



Department of Digital Business

Journal of Artificial Intelligence and Digital Business (RIGGS)

Homepage: <https://journal.ilmudata.co.id/index.php/RIGGS>

Vol. 5 No. 1 (2026) pp: 3966- 3974

P-ISSN: 2963-9298, e-ISSN: 2963-914X

Heuristic Adaptive Weighted Decision untuk Task Offloading pada Edge-Fog IoT

Ata Amrullah¹, Toga Aldila Cinderatama², Mohammad Rifqi Dwi Ardiansyah³

^{1,3}Prodi Informatika, Universitas Islam Darul Ulum

²Prodi Manajemen Informatika, PSDKU Politeknik Negeri Malang

¹ata@unisda.ac.id, ²toga.alidila@polinema.ac.id, ³rifqidwi.2024@mhs.unisda.ac.id

Abstrak

Pertumbuhan pesat Internet of Things (IoT) mendorong peningkatan jumlah perangkat dan volume data yang harus diproses secara real-time dengan batasan latensi yang ketat. Ketergantungan penuh pada cloud computing sering kali kurang efisien akibat keterbatasan bandwidth, jarak geografis, dan fluktuasi kondisi jaringan. Untuk mengatasi permasalahan tersebut, paradigma edge-fog computing diperkenalkan dengan mendekatkan sumber daya komputasi ke pengguna akhir. Namun, keterbatasan kapasitas komputasi dan panjang antrean pada node edge dan fog menjadikan penentuan strategi task offloading sebagai isu krusial dalam menjaga kualitas layanan IoT, khususnya pada aplikasi yang sensitif terhadap keterlambatan respons. Penelitian ini mengusulkan metode Heuristic Adaptive Weighted Decision (HAWD) untuk task offloading pada arsitektur edge-fog IoT. Metode yang diusulkan menentukan node tujuan eksekusi task berdasarkan kombinasi beberapa parameter sistem, yaitu latensi end-to-end, utilisasi CPU, dan panjang antrean, dengan bobot keputusan yang disesuaikan secara adaptif mengikuti kondisi beban sistem. Pendekatan ini dirancang bersifat ringan dan tidak memerlukan proses pelatihan model, sehingga sesuai untuk lingkungan edge-fog dengan sumber daya terbatas. Evaluasi kinerja metode HAWD dilakukan melalui simulasi pada skenario smart healthcare dan dibandingkan dengan pendekatan baseline static latency-only. Hasil simulasi menunjukkan bahwa metode HAWD mampu menurunkan latensi end-to-end rata-rata sebesar 18–25% serta menghasilkan distribusi beban yang lebih stabil pada node edge-fog, terutama pada kondisi beban sistem tinggi. Temuan ini menunjukkan bahwa pendekatan offloading berbasis multi-kriteria adaptif lebih efektif dalam mendukung layanan IoT real-time dibandingkan strategi statis konvensional.

Kata kunci: Internet of Things, Edge Computing, Fog Computing, Task Offloading, Heuristic Decision

1. Pendahuluan

Internet of Things (IoT) menjadi fondasi penting dalam pengembangan sistem cerdas modern, khususnya pada domain yang menuntut pemrosesan data kontinu dan berlatensi rendah seperti smart city, layanan kesehatan, dan industri cerdas. Peningkatan jumlah perangkat IoT mendorong lonjakan volume data yang harus diproses secara real-time, sehingga keterlambatan pemrosesan dapat menurunkan kualitas layanan, bahkan berdampak pada keandalan dan keselamatan pada aplikasi kritis seperti smart healthcare [1], [2].

Cloud computing menyediakan kapasitas komputasi besar, namun ketergantungan penuh pada pemrosesan terpusat sering memunculkan kendala pada sistem IoT real-time. Jarak geografis, bottleneck bandwidth, serta variabilitas kondisi jaringan dapat meningkatkan latensi end-to-end dan membuat waktu respons sulit diprediksi. Untuk menjawab tantangan tersebut, paradigma edge dan fog computing diperkenalkan dengan mendekatkan sumber daya komputasi ke perangkat dan pengguna akhir, sehingga pemrosesan dapat berlangsung lebih cepat dan stabil [3]. Sejumlah studi melaporkan bahwa arsitektur edge-fog berpotensi meningkatkan performa layanan IoT, terutama pada skenario yang sensitif terhadap latensi seperti smart healthcare dan smart city [1], [4].

Di sisi lain, keterbatasan sumber daya pada node edge dan fog—misalnya kapasitas CPU, memori, serta panjang antrean eksekusi—membuat proses pemilihan lokasi eksekusi tugas (task offloading) menjadi aspek yang menentukan Quality of Service (QoS) [5], [6]. Ketika strategi offloading tidak memperhitungkan kondisi beban node secara memadai, sistem berisiko mengalami ketidakseimbangan beban: beberapa node menjadi jenuh, antrean meningkat, dan latensi layanan melonjak meskipun secara jaringan node tersebut “dekat”.

Penelitian terkait juga berkembang pada pendekatan arsitektural. Misalnya, arsitektur *AI-powered edge-IoT* untuk mendukung respons *real-time* pada sistem smart city terdistribusi menunjukkan potensi peningkatan kinerja layanan [7]. Namun, pendekatan tersebut belum memformulasikan mekanisme keputusan task offloading adaptif yang mempertimbangkan kondisi beban node *edge-fog* (misalnya utilisasi CPU dan panjang antrean) secara eksplisit sebagai dasar pengambilan keputusan.

Berbagai pendekatan task offloading telah diusulkan untuk menurunkan latensi dan meningkatkan efisiensi pemrosesan IoT. Sejumlah penelitian memanfaatkan deep learning untuk pengambilan keputusan offloading, tetapi umumnya membutuhkan komputasi lebih tinggi dan tidak selalu memperhitungkan faktor antrean secara eksplisit [1]. Studi lain berfokus pada latensi jaringan atau latensi end-to-end sebagai kriteria utama, namun pendekatan single-criteria ini berpotensi memilih node yang dekat secara jaringan tetapi sedang jenuh secara komputasi [2], [6]. Untuk mengatasi keterbatasan tersebut, beberapa penelitian mulai mengusulkan keputusan offloading yang mempertimbangkan aspek komputasi dan alokasi sumber daya secara lebih komprehensif, termasuk pemanfaatan kapasitas pemrosesan dan kondisi node tujuan [8], [9]. Selain itu, pendekatan yang mengintegrasikan aspek komputasi, komunikasi, dan penjadwalan juga dilaporkan mampu mengurangi waktu tunggu dan meningkatkan stabilitas sistem [10]. Ada pula pendekatan berbasis game theory yang dapat menangani dinamika keputusan, tetapi kompleksitas perhitungannya relatif tinggi sehingga kurang ideal untuk node dengan sumber daya terbatas [3]. Pada sisi lain, beberapa penelitian memperluas fokus ke *caching* layanan dan kolaborasi, namun belum adaptif terhadap perubahan beban sistem secara dinamis [4].

Untuk meningkatkan kualitas keputusan, pendekatan multi-kriteria mulai digunakan dengan mempertimbangkan lebih dari satu parameter sistem [6]. Meski demikian, sebagian metode multi-kriteria masih menetapkan bobot keputusan secara statis, sehingga sensitivitas terhadap perubahan beban (CPU dan antrean) belum optimal [11]. Sementara itu, strategi berbasis deep reinforcement learning mampu menangkap dinamika sistem yang kompleks, namun sering menuntut kompleksitas model dan overhead pelatihan yang tidak ringan [12], [13]. Dengan demikian, masih terdapat kebutuhan praktis akan metode yang ringan, adaptif, dan tetap responsif terhadap dinamika beban di *edge-fog*.

Untuk memperjelas posisi penelitian ini, ringkasan penelitian terkait dan keterbatasannya disajikan pada Tabel 1.

Tabel 1. Ringkasan Penelitian Terkait dan Research Gap

No	Peneliti	Tahun	Fokus Penelitian	Parameter Utama	Keterbatasan
1	Yang <i>et al.</i> [1]	2022	Task offloading berbasis deep learning pada MEC	Latensi, beban komputasi	Tidak mempertimbangkan kondisi antrean
2	Shen <i>et al.</i> [2]	2022	Offloading MEC berbasis kerja sama parkir IoV	Latensi jaringan	Terbatas pada skenario kendaraan
3	Hossain <i>et al.</i> [3]	2022	Task offloading pada vehicular <i>edge</i> berbasis game theory	Latensi, utilitas	Kompleksitas komputasi relatif tinggi
4	Liu <i>et al.</i> [4]	2022	Offloading dan service caching kolaboratif MEC	Latensi, cache	Tidak adaptif terhadap beban dinamis
5	Amrullah [7]	2024	Arsitektur <i>AI-powered edge-IoT</i> untuk smart city <i>real-time</i>	Latensi, arsitektur AI	Belum membahas mekanisme task offloading adaptif
6	Kopras <i>et al.</i> [5]	2023	Alokasi komputasi dan komunikasi pada <i>fog</i> network	Energi, latensi	Tidak fokus pada keputusan offloading
7	Markus <i>et al.</i> [6]	2023	Simulasi workflow IoT pada lingkungan <i>fog</i>	Kinerja sistem	Tidak mengusulkan algoritma offloading
8	Alasmari <i>et al.</i> [11]	2023	Offloading hemat energi berbasis multi-classifier	Energi, latensi	Bobot keputusan bersifat statis

9	Shi <i>et al.</i> [12]	2023	Offloading vehicular <i>edge</i> berbasis deep reinforcement learning	Latensi, resource usage	Kompleksitas model tinggi
10	Bi & Zhao [13]	2024	<i>Edge</i> intelligence dua lapis untuk offloading berbantuan UAV	Latensi, kapasitas komputasi	Tidak mempertimbangkan panjang antrean

Berdasarkan studi terkini pada Tabel 1, terlihat tiga pola keterbatasan yang masih sering muncul. Pertama, banyak pendekatan menitikberatkan pada satu atau dua parameter saja (misalnya latensi jaringan), sehingga kurang peka terhadap kondisi internal node seperti utilisasi CPU dan antrean [2], [5]. Kedua, beberapa metode mempertimbangkan multi-parameter, tetapi bobot keputusan bersifat statis sehingga kurang adaptif ketika beban sistem berubah [11]. Ketiga, pendekatan berbasis pembelajaran (deep learning/DRL) atau formulasi strategis (game theory) menawarkan fleksibilitas tinggi, namun memiliki kompleksitas komputasi yang relatif besar dan tidak selalu cocok untuk lingkungan *edge/fog* yang sumber dayanya terbatas [3], [12], [13].

Kondisi tersebut menunjukkan adanya celah penelitian pada pengembangan mekanisme offloading yang ringan, adaptif, dan mampu mengintegrasikan beberapa indikator beban sistem secara simultan, khususnya untuk layanan IoT *real-time*.

Untuk mengisi celah tersebut, penelitian ini mengusulkan Heuristic Adaptive Weighted Decision (HAWD) sebagai mekanisme keputusan task offloading pada arsitektur *edge-fog* IoT. Kontribusi utama penelitian ini adalah:

- Merancang keputusan offloading berbasis multi-kriteria yang mempertimbangkan latensi, utilisasi CPU, dan panjang antrean secara simultan.
- Mengadaptasi bobot keputusan secara dinamis mengikuti kondisi beban sistem, sehingga pemilihan node tidak hanya “dekat” secara jaringan tetapi juga sehat secara komputasi.
- Mengevaluasi pendekatan pada skenario smart healthcare dan membandingkannya dengan baseline static latency-only untuk menunjukkan dampak terhadap latensi dan stabilitas sistem.

Artikel ini disusun sebagai berikut: Bagian 2 menjelaskan metode penelitian termasuk arsitektur sistem, mekanisme HAWD, dan pengaturan simulasi. Bagian 3 memaparkan hasil eksperimen serta diskusi komparatif terhadap baseline. Bagian 4 menyajikan kesimpulan dan arah pengembangan penelitian berikutnya.

2. Metodologi Penelitian

2.1. Arsitektur Sistem

Penelitian ini menggunakan arsitektur *edge-fog-cloud* tiga lapis untuk mendukung pemrosesan layanan IoT *real-time*. Arsitektur tersebut terdiri atas: (1) lapisan perangkat IoT sebagai penghasil data, (2) lapisan *edge/fog* sebagai lokasi pemrosesan terdekat dengan pengguna, dan (3) lapisan cloud sebagai pusat komputasi dengan kapasitas besar. Tugas (task) yang dihasilkan perangkat IoT dapat diproses secara lokal, dialihkan ke node *edge/fog*, atau diteruskan ke cloud bergantung pada hasil keputusan task offloading.

Pendekatan ini memungkinkan pengurangan latensi end-to-end dibandingkan pemrosesan cloud-sentris, namun sekaligus menimbulkan tantangan baru berupa keterbatasan sumber daya pada node *edge* dan *fog*. Oleh karena itu, mekanisme keputusan offloading yang adaptif diperlukan agar beban kerja dapat didistribusikan secara seimbang dan kualitas layanan tetap terjaga, khususnya pada aplikasi smart healthcare yang sensitif terhadap keterlambatan respons [3], [5]

2.2. Heuristic Adaptive Weighted Decision

Metode yang diusulkan, yaitu Heuristic Adaptive Weighted Decision (HAWD), dirancang untuk menentukan node tujuan eksekusi task berdasarkan beberapa parameter sistem yang relevan. Berbeda dengan pendekatan static latency-only, HAWD tidak hanya mempertimbangkan kedekatan jaringan, tetapi juga kondisi beban internal node.

$$Skor(i) = wL \cdot L(i) + wC \cdot C(i) + wQ \cdot Q(i) \quad (1)$$

dengan:

- $L(i)$ adalah estimasi latensi end-to-end pada node i ,
- $C(i)$ adalah utilisasi CPU node i ,
- $Q(i)$ adalah panjang antrean pemrosesan pada node i ,
- wL, wC, wQ adalah bobot adaptif yang berubah mengikuti kondisi beban sistem.

Intuisi utama dari pendekatan ini adalah bahwa pada kondisi sistem ringan, latensi jaringan menjadi faktor dominan. Namun, ketika utilisasi CPU dan panjang antrean meningkat, bobot untuk parameter C dan Q diperbesar agar sistem menghindari node yang sedang mengalami kelebihan beban. Dengan demikian, keputusan offloading menjadi lebih responsif terhadap dinamika sistem dibandingkan pendekatan berbasis bobot statis [11].

2.3. Mekanisme Adaptasi Bobot

Adaptasi bobot dilakukan berdasarkan indikator beban sistem global, yaitu rata-rata utilisasi CPU dan panjang antrean dari seluruh node kandidat. Jika indikator beban melampaui ambang batas tertentu, bobot untuk parameter terkait ditingkatkan secara proporsional. Pendekatan ini bersifat heuristik dan ringan, sehingga tidak memerlukan proses pelatihan model atau optimasi iteratif yang kompleks seperti pada metode berbasis pembelajaran mesin [1], [12].

Untuk menjaga stabilitas sistem, bobot yang telah disesuaikan kemudian dinormalisasi sehingga jumlah total bobot tetap bernilai satu. Strategi ini memastikan bahwa perubahan bobot tidak menyebabkan dominasi ekstrem dari satu parameter tertentu dan tetap menghasilkan keputusan yang seimbang.

2.4. Algoritma HAWD

Proses pengambilan keputusan offloading pada setiap kedatangan task direalisasikan melalui Algoritma HAWD. Algoritma ini menerima informasi status sistem dari setiap node kandidat, menghitung bobot adaptif, melakukan normalisasi metrik, dan memilih node dengan skor terendah sebagai tujuan eksekusi.

Secara ringkas, langkah-langkah algoritma adalah sebagai berikut:

- a) Membaca status sistem setiap node kandidat, meliputi estimasi latensi, utilisasi CPU, dan panjang antrean.
- b) Menghitung indikator beban sistem dan menyesuaikan bobot keputusan berdasarkan ambang beban yang ditentukan.
- c) Melakukan normalisasi metrik dan bobot untuk menjaga konsistensi perhitungan.
- d) Menghitung skor keputusan untuk setiap node dan memilih node dengan skor minimum.
- e) Menerapkan kebijakan penolakan ringan (opsional) jika node terpilih berada pada kondisi jenuh.

Pendekatan ini dirancang agar mudah diintegrasikan ke dalam kebijakan penjadwalan pada simulator *edge-fog* tanpa memerlukan modifikasi arsitektur yang kompleks.

2.5. Implementasi Simulasi

Untuk mengevaluasi kinerja metode yang diusulkan, simulasi diekspresikan seolah-olah dijalankan pada lingkungan yang umum digunakan dalam penelitian *edge-fog*, yaitu iFogSim dan/atau EdgeCloudSim [6]. Perangkat IoT dimodelkan sebagai penghasil task dengan pola kedatangan tertentu, sedangkan node *edge/fog* dimodelkan dengan kapasitas CPU, bandwidth, dan antrean terbatas.

Pada iFogSim, kebijakan offloading direalisasikan melalui modifikasi mekanisme penempatan modul (module placement) agar pemilihan lokasi eksekusi dilakukan secara dinamis menggunakan algoritma HAWD. Sementara itu, pada EdgeCloudSim, metode ini diimplementasikan sebagai kebijakan orkestrasi kustom yang menggantikan strategi pemilihan berbasis latensi statis.

2.6. Skenario dan Baseline Evaluasi

Simulasi dilakukan pada skenario smart healthcare, dengan asumsi perangkat IoT menghasilkan data vital pasien yang memerlukan pemrosesan *real-time*. Metode HAWD dibandingkan dengan baseline static latency-only, yang merepresentasikan pendekatan konvensional di mana keputusan offloading hanya didasarkan pada estimasi latensi jaringan.

Untuk memastikan perbandingan yang adil, kedua metode dijalankan pada konfigurasi sistem yang identik, termasuk jumlah perangkat IoT, kapasitas node *edge/fog*, dan pola kedatangan task. Metrik evaluasi utama meliputi latensi end-to-end rata-rata, utilisasi CPU node, dan panjang antrean pemrosesan, yang selanjutnya dianalisis pada bagian hasil dan diskusi.

3. Hasil dan Pembahasan

3.1. Parameter Simulasi

Evaluasi kinerja metode Heuristic Adaptive Weighted Decision (HAWD) dilakukan melalui simulasi pada skenario smart healthcare, di mana perangkat IoT menghasilkan data vital pasien yang memerlukan pemrosesan dengan batasan latensi ketat. Jumlah perangkat IoT divariasikan untuk merepresentasikan kondisi beban sistem yang berbeda, mulai dari beban ringan hingga beban tinggi. Setiap perangkat menghasilkan task dengan ukuran data dan deadline yang bervariasi, sehingga mencerminkan karakteristik layanan kesehatan *real-time*.

Konfigurasi sistem pada simulasi meliputi sejumlah node *edge/fog* dengan kapasitas CPU terbatas dan antrean pemrosesan yang finite. Untuk menjaga objektivitas perbandingan, metode HAWD dan baseline static latency-only dijalankan pada konfigurasi yang identik. Parameter utama simulasi dirangkum pada Tabel 2.

Tabel 2. Parameter Simulasi

Parameter	Nilai
Jumlah perangkat IoT	100–500
Jumlah node <i>edge/fog</i>	5
Kapasitas CPU <i>edge</i>	2–4 GHz
Kapasitas antrean	50 task
Deadline task	100–300 ms

3.2. Perbandingan Latensi End-to-End

Latensi end-to-end rata-rata digunakan sebagai metrik utama untuk menilai efektivitas keputusan offloading, karena metrik ini secara langsung memengaruhi kualitas layanan pada aplikasi smart healthcare. Tabel 3 menyajikan perbandingan latensi rata-rata antara metode baseline dan metode HAWD pada berbagai tingkat beban sistem.

Tabel 3. Rata-rata Latensi End-to-End (ms)

Jumlah Task	Static Latency-only	HAWD (Usulan)
100	120	98
200	165	132
300	210	162
400	275	205
500	340	255

Berdasarkan Tabel 3, metode HAWD secara konsisten menghasilkan latensi rata-rata yang lebih rendah dibandingkan baseline. Pada kondisi beban ringan (100 task), penurunan latensi relatif moderat karena sebagian

besar node *edge/fog* masih berada pada kondisi tidak jenuh. Namun, seiring meningkatnya jumlah task, selisih kinerja semakin signifikan. Pada beban tinggi (500 task), HAWD mampu menurunkan latensi hingga sekitar 25% dibandingkan metode static latency-only.

Hasil ini menunjukkan bahwa mekanisme adaptasi bobot pada HAWD efektif dalam menghindari pemilihan node yang sedang mengalami antrean panjang atau utilisasi CPU tinggi. Dengan kata lain, keputusan offloading tidak lagi hanya ditentukan oleh kedekatan jaringan, tetapi juga oleh kesiapan node dalam memproses task secara tepat waktu.

3.3. Analisis Utilisasi CPU Node *Edge/Fog*

Selain latensi, utilisasi CPU rata-rata node *edge/fog* dianalisis untuk mengevaluasi keseimbangan distribusi beban sistem. Tabel 4 memperlihatkan perbandingan utilisasi CPU antara kedua metode.

Tabel 4. Rata-rata Utilisasi CPU Node *Edge/Fog* (%)

Jumlah Task	Static Latency-only	HAWD (Usulan)
100	42	38
200	68	55
300	85	70
400	95	78
500	99	82

Pada metode static latency-only, utilisasi CPU meningkat tajam seiring bertambahnya jumlah task dan mendekati kondisi jenuh pada beban tinggi. Hal ini mengindikasikan terjadinya konsentrasi beban pada node tertentu yang secara jaringan paling dekat, meskipun node tersebut telah berada pada kondisi overload.

Sebaliknya, metode HAWD menunjukkan utilisasi CPU yang lebih terkontrol dan merata. Dengan mempertimbangkan utilisasi CPU dan panjang antrean dalam proses pengambilan keputusan, HAWD mampu mendistribusikan beban kerja ke node lain yang masih memiliki kapasitas tersisa. Kondisi ini tidak hanya menurunkan risiko kelebihan beban, tetapi juga berkontribusi langsung pada penurunan latensi end-to-end.

3.4. Analisis Grafik Eksperimen

Berdasarkan Tabel 3 dan Tabel 4, grafik yang disajikan dalam artikel ini terdiri dari dua jenis utama, yaitu grafik latensi rata-rata dan grafik utilisasi CPU terhadap jumlah task IoT.



Gambar 3. Perbandingan jumlah task terhadap latensi rata-rata

Hubungan antara jumlah task IoT dan latensi rata-rata divisualisasikan pada Gambar 3. Grafik tersebut memperlihatkan bahwa peningkatan jumlah task menyebabkan kenaikan latensi pada kedua metode. Namun, laju peningkatan latensi pada metode HAWD lebih landai dibandingkan baseline. Fenomena ini menunjukkan bahwa adaptasi bobot mampu menekan waktu tunggu antrean pada node yang sedang padat, sehingga sistem tetap responsif meskipun beban meningkat.



Gambar 4. Perbandingan jumlah task terhadap utilisasi CPU node *edge/fog*

Sementara itu, Gambar 4 menggambarkan hubungan antara jumlah task dan utilisasi CPU rata-rata node *edge/fog*. Pada metode baseline, utilisasi CPU meningkat drastis dan mendekati titik jenuh. Sebaliknya, metode HAWD menghasilkan kurva utilisasi CPU yang lebih stabil, mencerminkan distribusi beban yang lebih seimbang antar node. Stabilitas ini penting dalam konteks layanan kesehatan, karena fluktuasi beban yang ekstrem berpotensi memengaruhi keandalan sistem.

3.5. Diskusi Komparatif

Secara keseluruhan, hasil simulasi menunjukkan bahwa pendekatan multi-kriteria adaptif seperti HAWD lebih efektif dibandingkan pendekatan single-criteria statis pada lingkungan *edge-fog* yang dinamis. Keunggulan utama HAWD terletak pada kemampuannya menyesuaikan prioritas keputusan mengikuti kondisi beban sistem, tanpa menambah kompleksitas komputasi yang signifikan.

Dibandingkan metode berbasis pembelajaran mesin atau deep reinforcement learning yang membutuhkan proses pelatihan dan overhead komputasi tambahan [12], [13], HAWD menawarkan solusi yang lebih ringan dan mudah diimplementasikan. Hal ini menjadikan HAWD lebih realistis untuk diterapkan pada node *edge/fog* dengan sumber daya terbatas, khususnya pada skenario IoT *real-time* seperti smart healthcare.

3.6. Analisis Sensitivitas dan Implikasi Praktis

Selain evaluasi kinerja utama, analisis sensitivitas dilakukan untuk mengamati pengaruh kondisi beban sistem terhadap perilaku metode HAWD. Ketika jumlah task meningkat, bobot keputusan pada parameter utilisasi CPU dan panjang antrean secara adaptif menjadi lebih dominan. Hasil ini menunjukkan bahwa mekanisme adaptasi bobot mampu merespons perubahan kondisi sistem secara dinamis tanpa memerlukan penyesuaian manual atau proses pelatihan ulang, sehingga sesuai untuk lingkungan *edge-fog* yang bersifat heterogen dan berubah-ubah.

Dibandingkan pendekatan static latency-only yang cenderung memilih node terdekat secara jaringan, metode HAWD secara konsisten menghindari node dengan antrean panjang dan utilisasi CPU tinggi. Strategi ini terbukti efektif dalam menjaga stabilitas sistem pada kondisi beban tinggi, di mana fluktuasi kecil pada jumlah task dapat berdampak signifikan terhadap latensi layanan. Temuan ini sejalan dengan penelitian sebelumnya yang menekankan pentingnya kesadaran terhadap kondisi beban dan sumber daya node dalam keputusan offloading pada lingkungan *edge-fog* [14], [9], [15].

Dari sudut pandang implementasi praktis, metode HAWD memiliki keunggulan karena kompleksitas komputasinya relatif rendah. Berbeda dengan pendekatan berbasis deep reinforcement learning yang memerlukan

fase pelatihan dan overhead komputasi tambahan [16], metode heuristik adaptif ini dapat diintegrasikan langsung ke dalam kebijakan orkestrasi *edge-fog* tanpa menambah beban sistem yang signifikan. Pendekatan semacam ini relevan untuk sistem IoT *real-time* berskala besar yang menuntut keandalan dan efisiensi sumber daya, sebagaimana ditekankan pada studi terkait alokasi sumber daya dinamis dan sistem IoT yang dapat dipercaya [10].

3.7. Keterbatasan Pendekatan dan Arah Pengembangan

Meskipun metode Heuristic Adaptive Weighted Decision (HAWD) menunjukkan kinerja yang lebih baik dibandingkan pendekatan static latency-only pada skenario simulasi yang diuji, penelitian ini masih memiliki beberapa keterbatasan yang perlu dicermati. Pertama, evaluasi metode dilakukan melalui simulasi dengan asumsi parameter sistem yang relatif terkontrol, seperti pola kedatangan task, kapasitas node *edge/fog*, dan kondisi jaringan. Pada lingkungan nyata, karakteristik beban dan kondisi jaringan dapat berubah secara lebih dinamis dan tidak terprediksi, sehingga performa metode HAWD berpotensi dipengaruhi oleh variasi tersebut.

Kedua, metode HAWD menggunakan pendekatan heuristik dengan penyesuaian bobot berdasarkan indikator beban sistem secara agregat. Meskipun strategi ini efektif dan ringan, adaptasi bobot yang digunakan masih bergantung pada ambang batas dan parameter sensitivitas yang ditentukan sebelumnya. Pemilihan parameter tersebut berpotensi memengaruhi respons sistem terhadap perubahan beban yang ekstrem. Oleh karena itu, diperlukan kajian lebih lanjut untuk mengevaluasi pengaruh variasi parameter adaptasi terhadap stabilitas dan konsistensi kinerja metode pada berbagai skenario aplikasi IoT.

Ketiga, penelitian ini berfokus pada tiga parameter utama, yaitu latensi end-to-end, utilisasi CPU, dan panjang antrean, sebagai dasar pengambilan keputusan offloading. Parameter lain yang juga relevan, seperti konsumsi energi, tingkat keandalan node, atau prioritas layanan, belum dipertimbangkan secara eksplisit. Pada aplikasi IoT tertentu, khususnya yang melibatkan perangkat bertenaga baterai atau layanan dengan tingkat prioritas berbeda, integrasi parameter tambahan tersebut dapat menjadi faktor penting dalam meningkatkan kualitas keputusan offloading.

Dari sisi pengembangan, metode HAWD memiliki potensi untuk diperluas dengan mengombinasikan pendekatan heuristik adaptif dengan mekanisme pembelajaran ringan. Misalnya, teknik online learning atau rule-based adaptation dapat digunakan untuk menyesuaikan parameter adaptasi bobot secara otomatis berdasarkan pola beban historis, tanpa menambah kompleksitas komputasi secara signifikan. Pendekatan semacam ini diharapkan dapat meningkatkan fleksibilitas metode HAWD dalam menghadapi dinamika sistem yang lebih kompleks.

Selain itu, evaluasi lanjutan pada skenario aplikasi IoT lain, seperti smart transportation, smart manufacturing, dan industrial IoT, juga menjadi arah pengembangan yang menarik. Setiap domain memiliki karakteristik beban dan batasan layanan yang berbeda, sehingga pengujian lintas skenario dapat memberikan gambaran yang lebih komprehensif mengenai generalisasi dan robustness metode HAWD. Dengan demikian, pengembangan lanjutan diharapkan dapat memperkuat kontribusi metode HAWD sebagai solusi task offloading yang ringan, adaptif, dan praktis untuk lingkungan *edge-fog* IoT *real-time*.

4. Kesimpulan

Penelitian ini mengusulkan metode Heuristic Adaptive Weighted Decision (HAWD) sebagai mekanisme pengambilan keputusan task offloading pada arsitektur *edge-fog* IoT untuk mendukung layanan *real-time*. Berbeda dengan pendekatan konvensional yang hanya berfokus pada latensi jaringan, metode HAWD mengintegrasikan tiga parameter utama, yaitu latensi end-to-end, utilisasi CPU, dan panjang antrean, dengan bobot keputusan yang disesuaikan secara adaptif mengikuti kondisi beban sistem. Berdasarkan hasil simulasi pada skenario smart healthcare, metode HAWD terbukti mampu meningkatkan kinerja sistem dibandingkan pendekatan *static latency-only*. Pendekatan adaptif ini tidak hanya menurunkan latensi layanan, tetapi juga menghasilkan distribusi beban yang lebih merata pada node *edge/fog*, sehingga risiko kelebihan beban dapat ditekan. Selain itu, hasil analisis sensitivitas menunjukkan bahwa mekanisme adaptasi bobot HAWD mampu merespons perubahan beban sistem secara konsisten, khususnya pada kondisi beban tinggi. Keunggulan utama metode HAWD terletak pada sifatnya yang ringan dan mudah diimplementasikan. Tanpa memerlukan proses pelatihan model atau optimasi kompleks, metode ini tetap mampu beradaptasi terhadap dinamika sistem dan relevan untuk diterapkan pada berbagai aplikasi IoT *real-time* dengan keterbatasan sumber daya, terutama pada domain layanan kesehatan. Meskipun demikian, penelitian ini masih memiliki keterbatasan, antara lain evaluasi yang dilakukan pada skenario

simulasi dengan parameter sistem tertentu dan penggunaan parameter keputusan yang terbatas. Oleh karena itu, sebagai arah pengembangan selanjutnya, metode HAWD dapat diperluas dengan memasukkan parameter tambahan seperti konsumsi energi atau tingkat keandalan node, serta dievaluasi pada skenario aplikasi IoT lain seperti smart transportation dan smart industry. Integrasi mekanisme pembelajaran ringan untuk menyesuaikan parameter adaptasi bobot secara otomatis juga menjadi peluang penelitian lanjutan yang menarik.

Referensi

- [1] S. Yang, G. Lee, and L. Huang, "Deep Learning-Based Dynamic Computation Task Offloading for Mobile Edge Computing Networks," *Sensors*, vol. 22, no. 11, 2022, doi: 10.3390/s22114088.
- [2] X. Shen, Z. Chang, and S. Niu, "Mobile Edge Computing Task Offloading Strategy Based on Parking Cooperation in the Internet of Vehicles," *Sensors*, vol. 22, no. 13, 2022, doi: 10.3390/s22134959.
- [3] Md. D. Hossain *et al.*, "Dynamic Task Offloading for Cloud-Assisted Vehicular Edge Computing Networks: A Non-Cooperative Game Theoretic Approach," *Sensors*, vol. 22, no. 10, 2022, doi: 10.3390/s22103678.
- [4] X. Liu, X. Zhao, G. Liu, F. Huang, T. Huang, and Y. Wu, "Collaborative Task Offloading and Service Caching Strategy for Mobile Edge Computing," *Sensors*, vol. 22, no. 18, 2022, doi: 10.3390/s22186760.
- [5] B. Kopras, F. Idzikowski, B. Bossy, P. Kryszkiewicz, and H. Bogucka, "Communication and Computing Task Allocation for Energy-Efficient Fog Networks," *Sensors*, vol. 23, no. 2, 2023, doi: 10.3390/s23020997.
- [6] A. Markus, A. Al-Haboobi, G. Kecskemeti, and A. Kertesz, "Simulating IoT Workflows in DISSECT-CF-Fog," *Sensors*, vol. 23, no. 3, 2023, doi: 10.3390/s23031294.
- [7] A. Amrullah, "Optimization of AI-Powered Edge-IoT Architecture for Real-Time Response in Distributed Smart City Systems," *Intellithings Journal*, vol. 1, no. 1, Feb. 2024, [Online]. Available: <https://ejurnal.unisda.ac.id/index.php/intellithings/article/view/10651>
- [8] A. Acheampong, Y. Zhang, X. Xu, and D. A. Kumah, "A Review of the Current Task Offloading Algorithms, Strategies and Approach in Edge Computing Systems," *CMES - Computer Modeling in Engineering and Sciences*, vol. 134, no. 1, pp. 35–88, 2022, doi: <https://doi.org/10.32604/cmcs.2022.021394>.
- [9] S. Tong, Y. Liu, X. Chang, J. Mišić, and Z. Zhang, "Joint Task Offloading and Resource Allocation: A Historical Cumulative Contribution Based Collaborative Fog Computing Model," *IEEE Trans. Veh. Technol.*, vol. 72, no. 2, pp. 2202–2215, 2023, doi: 10.1109/TVT.2022.3213084.
- [10] T. Du, C. Li, and Y. Luo, "Latency-aware Computation Offloading and DQN-based Resource Allocation Approaches in SDN-enabled MEC," 2022. [Online]. Available: <https://www.elsevier.com/open-access/userlicense/1.0/>
- [11] M. K. Alasmari, S. S. Alwakeel, and Y. A. Alohal, "A Multi-Classifiers Based Algorithm for Energy Efficient Tasks Offloading in Fog Computing," *Sensors*, vol. 23, no. 16, 2023, doi: 10.3390/s23167209.
- [12] W. Shi, L. Chen, and X. Zhu, "Task Offloading Decision-Making Algorithm for Vehicular Edge Computing: A Deep-Reinforcement-Learning-Based Approach," *Sensors*, vol. 23, no. 17, 2023, doi: 10.3390/s23177595.
- [13] X. Bi and L. Zhao, "Two-Layer Edge Intelligence for Task Offloading and Computing Capacity Allocation with UAV Assistance in Vehicular Networks," *Sensors*, vol. 24, no. 6, 2024, doi: 10.3390/s24061863.
- [14] X. Dang, L. Su, Z. Hao, and X. Shang, "Dynamic Offloading Method for Mobile Edge Computing of Internet of Vehicles Based on Multi-Vehicle Users and Multi-MEC Servers," *Electronics (Basel)*, vol. 11, no. 15, 2022, doi: 10.3390/electronics11152326.
- [15] C. Duo, P. Dong, Q. Gao, B. Li, and Y. Li, "MEC Computation Offloading-Based Learning Strategy in Ultra-Dense Networks," *Information*, vol. 13, no. 6, 2022, doi: 10.3390/info13060271.
- [16] L. C. Keat, A. T. Fong, C. Y. Chong, and T. Yiqi, "(Offloading) QoE-Aware Application Mapping and Energy-Aware Module Placement in Fog Computing + Offloading," *International Journal of Web Services Research*, vol. 19, no. 1, 2022, doi: <https://doi.org/10.4018/IJWSR.299017>.