

OPTIMASI STRATEGI COMPUTATION OFFLOADING BERBASIS SWARM INTELLIGENCE DALAM LINGKUNGAN IOT- EDGE COMPUTING

AGUS MULYANA



**PROGRAM STUDI DOKTOR ILMU KOMPUTER
SEKOLAH SAINS DATA, MATEMATIKA, DAN INFORMATIKA
INSTITUT PERTANIAN BOGOR
BOGOR
2025**



PERNYATAAN MENGENAI DISERTASI DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA

Dengan ini saya menyatakan bahwa disertasi dengan judul “Optimasi Strategi *Computation Offloading* Berbasis *Swarm Intelligence* dalam Lingkungan *IoT- Edge Computing*” adalah karya saya dengan arahan dari dosen pembimbing dan belum diajukan dalam bentuk apa pun kepada perguruan tinggi mana pun. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka di bagian akhir disertasi ini.

Dengan ini saya melimpahkan hak cipta dari karya tulis saya kepada Institut Pertanian Bogor.

Bogor, Agustus 2025

Agus Mulyana
G6601211014



RINGKASAN

AGUS MULYANA. Optimasi Strategi *Computation Offloading* Berbasis *Swarm Intelligence* dalam Lingkungan IoT- *Edge Computing*. Dibimbing oleh TAUFIK DJATNA, SRI WAHJUNI, dan HERU SUKOCO.

Computation offloading merupakan proses pemindahan sebagian atau seluruh beban komputasi (*task*) dari sebuah perangkat dengan sumber daya terbatas, seperti perangkat *Internet of Things* (IoT) ke unit komputasi eksternal yang memiliki kapabilitas lebih tinggi, seperti *edge server*, *cloud server*, atau perangkat lain dalam jaringan. Tujuan utama dari teknik ini adalah untuk meningkatkan efisiensi energi, mengurangi waktu eksekusi, serta mengoptimalkan performa sistem secara keseluruhan. Beberapa strategi utama yang digunakan dalam *computation offloading* meliputi: 1) Identifikasi jenis *task* yang bertujuan untuk menentukan apakah suatu *task* layak untuk di-*offload* atau lebih efisien dijalankan secara lokal. 2) Penentuan target *offloading* yang bertujuan untuk memilih node tujuan yang optimal berdasarkan kapabilitas sumber daya, latensi, atau jarak. 3) Optimasi jalur komunikasi untuk menentukan rute transmisi data yang paling efisien, memperhitungkan faktor seperti *bandwidth*, interferensi jaringan, dan topologi dinamis. Lingkungan komputasi yang dinamis seperti dalam sistem IoT atau jaringan bergerak memunculkan berbagai tantangan, antara lain: 1) Variabilitas ketersediaan energi pada perangkat sumber, misalnya baterai yang relatif terbatas. 2) Fluktuasi *bandwidth* jaringan yang memengaruhi kecepatan dan keandalan transmisi data. 3) Kualitas dan kekuatan sinyal komunikasi nirkabel, yang dapat berubah tergantung pada lokasi dan interferensi. 4) Mobilitas *node*, terutama dalam sistem berbasis kendaraan (*vehicular networks*) atau *drone*. Banyak pendekatan dalam penelitian *computation offloading* masih mengandalkan strategi statis, yaitu pengambilan keputusan *offloading* yang tidak mempertimbangkan perubahan kondisi lingkungan secara *real-time*. Strategi seperti ini rentan menyebabkan *overload* pada *server* tujuan, peningkatan latensi, konsumsi energi yang tidak efisien, dan penurunan kualitas layanan.

Penelitian ini bertujuan untuk mengoptimalkan strategi *computation offloading* agar adaptif terhadap lingkungan komputasi yang berkaitan dengan proses identifikasi jenis *task*, waktu, tujuan, dan penentuan jalur *computation offloading* yang tepat. Metode yang digunakan adalah mengembangkan kerangka kerja *computation offloading* dengan mengintegrasikan algoritma *Artificial Bee Colony* (ABC) dengan *Markov Decision Process* (MDP) selanjutnya disebut ABC-MDP. Tujuan integrasi algoritma ABC dengan MDP adalah untuk menghasilkan algoritma yang memiliki *fitness value* dan konvergensi yang adaptif sesuai fungsi tujuan dalam setiap iterasi komputasi. Algoritma ABC-MDP terdiri dari algoritma ABC berfokus pada pencarian sumber daya komputasi dengan cara mengeksplorasi atau mengeksploitasi, sedangkan MDP digunakan untuk menghitung *reward* dari setiap *action* keputusan berdasarkan *state* komputasi untuk setiap siklusnya. Berdasarkan nilai *reward* tersebut maka langkah komputasi pada siklus berikutnya akan lebih pasti, apakah melanjutkan eksploitasi pada suatu sumber daya komputasi atau berpindah untuk eksplorasi sumber daya komputasi yang baru. Luaran dari penelitian ini berupa kerangka kerja *computation offloading* adaptif yang terdiri dari arsitektur sistem, metode, formula dan algoritma. Evaluasi kinerja sistem dilakukan dalam tiga mode yaitu: 1) evaluasi numerik dalam lingkungan Python, 2) simulasi *computation offloading* berbasis *testbed* dan 3) implementasi di lingkungan nyata dalam studi kasus optimasi pemetikan teh di PPTK Gambung berbasis *internet of Things* dan *edge computing*.

@Hak Cipta Dilindungi Undang-undang

IPB University

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah
b. Pengutipan tidak merugikan kepentingan yang wajar IPB University.

2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB University.



Hak Cipta Dilindungi Undang-undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah
b. Pengutipan tidak merugikan kepentingan yang wajar IPB University.
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB University.

Mode evaluasi numerik akan membandingkan kinerja algoritma *baseline* GWO-WOA dengan ABC-MDP untuk menyelesaikan persamaan *Sphere*, *Rosenbrock* dan *Rastrigin* yang bertujuan untuk menghitung *fitness value*, konvergensi, kompleksitas ruang, dan kompleksitas waktu. Berdasarkan evaluasi numerik, algoritma ABC-MDP memiliki konsumsi memori lebih besar dari *baseline* algoritma GWO-WOA, hal ini karena algoritma ABC-MDP harus menyimpan *state* dari setiap *action*, sedangkan pada GWO-WOA tidak ada penyimpanan *state*. Pada evaluasi kompleksitas waktu, ABC-MDP memiliki waktu komputasi relatif lebih lambat dibandingkan GWO-WOA pada saat awal iterasi, kondisi ini karena ABC-MDP memiliki tahapan komputasi selain menghitung nilai *fitness* juga menghitung *reward* dari setiap *action* yang dilakukan. Ketika sebuah nilai *fitness* dan *reward* sudah didapatkan algoritma ABC-MDP dapat mencapai nilai optimal dan konvergen lebih cepat antara 15% sampai dengan 20% dibandingkan dengan GWO-WOA.

Mode simulasi berbasis *testbed* terdiri dari aplikasi pada perangkat *smartphone* sebagai *task generator* dengan dukungan fitur *interoperability* untuk pemilihan jalur komunikasi, baik GSM, Wi-Fi maupun LoRa. Pada bagian *edge server* dilengkapi dengan aplikasi untuk menerima, mengidentifikasi, mengeksekusi *task* dan memberikan hasil eksekusi kepada unit pengirim. Simulasi ini menggunakan enam skenario yang bertujuan untuk membuat variasi ketersediaan jumlah IoT, *edge computing server*, dan nilai iterasi komputasi. Tujuan pengaturan parameter ini adalah untuk mengukur adaptabilitas dan keandalan kerangka kerja *computation offloading* berbasis ABC-MDP dibandingkan dengan GWO-WOA. Indikator kinerja diukur dengan lima parameter yaitu: 1) *fitness value*, 2) latensi, 3) energi, 4) *bandwidth*, dan 5) *throughput*. ABC-MDP mendapatkan nilai kinerja lebih baik dari GWO-WOA pada konvergensi *fitness value* 5,53% lebih optimal, latensi lebih rendah 3,53%, energi lebih hemat 1,97% dan *throughput* lebih besar 0,43 % dibandingkan GWO-WOA, sedangkan untuk konsumsi *bandwidth* masih lebih besar 2,22% dari GWO-WOA.

Pada mode implementasi di area PPTK Gambung kerangka kerja *computation offloading* diterapkan dalam perangkat *Internet of Things*, *multi-access edge computing* dan *edge computing server* untuk melakukan *computation offloading*. Jenis *task* yang digunakan adalah *real-time task* dengan 4 tipe data yaitu data dokumen, data gambar, data video, dan data alfanumerik. Setiap *task* akan diidentifikasi berdasarkan jenis *task*, batas waktu, prioritas, dan tipe data. Pada mode implementasi ini melibatkan perangkat IoT dan *Multi Access Edge Computing* bergerak berbasis GPS untuk melakukan *computation offloading* ke *edge computing server*. Perangkat IoT dan *Multi Access Edge Computing* dapat memilih jalur *computation offloading* apakah menggunakan jalur komunikasi GSM, Wi-Fi atau LoRa secara adaptif berdasarkan nilai RSSI. Untuk bertukar pesan antara IoT, *Multi Access Edge Computing* dan *edge computing server* menggunakan protokol pesan HTTP atau MQTT. Berdasarkan pengujian dengan dengan 130 *task* yang memiliki data beragam dan batas waktu, maka terdapat 73 *task* atau 56,15% berhasil dioptimasi dan terdapat 47 *task* atau 43,84 % tidak berhasil dioptimasi. Ketidakterhasilan optimasi disebabkan oleh batasan waktu *real-time task* yang tidak dapat terpenuhi karena latensi yang dihasilkan lebih besar dari batas waktu komputasi untuk mengeksekusi *task*.

Kata Kunci: *Artificial Bee Colony*, *Computation Offloading*, *Edge Computing*, *Internet of Things*, *Markov Decision Process*.



SUMMARY

Computation Offloading Strategy Based on Swarm Intelligence in IoT Environment - Edge Computing. Supervised by TAUFIK DJATNA, SRI WAHJUNI and HERU SUKOCO.

Computation offloading is the process of transferring part or all of the computational load (task) from a device with limited resources, such as an Internet of Things (IoT) device to an external computing unit that has higher capabilities, such as an edge server, cloud server, or other device in the network. The main goal of this technique is to improve energy efficiency, reduce execution time, and optimize overall system performance. Some of the main strategies used in computation offloading include: 1) Identification of the type of task which aims to determine whether a task is worthy of being offloaded or more efficiently run locally. 2) Determination of the offloading target which aims to select the optimal destination node based on resource capabilities, latency, or distance. 3) Optimization of communication paths to determine the most efficient data transmission route, taking into account factors such as bandwidth, network interference, and dynamic topology. Dynamic computing environments such as in IoT systems or mobile networks pose various challenges, including: 1) Variability in energy availability on source devices, such as relatively limited batteries. 2) Fluctuations in network bandwidth that affect the speed and reliability of data transmission. 3) The quality and strength of wireless communication signals, which can change depending on location and interference. 4) Node mobility, especially in vehicular networks or drones. Many approaches in computation offloading research still rely on static strategies, namely offloading decision-making that does not consider changes in environmental conditions in real time. Such strategies are prone to overloading the destination server, increasing latency, inefficient energy consumption and decreasing service quality.

This study aims to optimize the computation offloading strategy to be adaptive to the computing environment related to the process of identifying the type of task, time, purpose and determining the right computation offloading path. The method used is to develop a computation offloading framework by integrating the Artificial Bee Colony (ABC) algorithm with the Markov Decision Process (MDP) hereinafter referred to as ABC-MDP. The purpose of integrating the ABC algorithm with MDP is to produce an algorithm that has a fitness value and adaptive convergence according to the objective function in each computation iteration. ABC-MDP algorithm consists of ABC algorithm focuses on finding computational resources by exploring or exploiting, while MDP is used to calculate the reward of each decision action based on the computational state for each cycle. Based on the reward value, the computational steps in the next cycle will be more certain, whether to continue exploitation on a computational resource or move to explore new computational resources. The output of this study is an adaptive computation offloading framework consisting of system architecture, methods, formulas and algorithms. The system performance evaluation was conducted in three modes, namely: 1) numerical evaluation in Python environment, 2) testbed-based computation offloading simulation and 3) implementation in real environment in a case study of tea picking optimization at PPTK Gambung based on the Internet of Things and edge computing.

The numerical evaluation mode will compare the performance of the baseline GWO-WOA algorithm with ABC-MDP to solve the Sphere, Rosenbrock, and Rastrigin equations which aim to calculate the fitness value, convergence, space complexity and

@Hak Cipta IPB University

IPB University

Hak Cipta Dilindungi Undang-undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah
b. Pengutipan tidak merugikan kepentingan yang wajar IPB University.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB University.



time complexity. Based on the numerical evaluation, the ABC-MDP algorithm has a larger memory consumption than the baseline GWO-WOA algorithm, this is because the ABC-MDP algorithm must store the state of each action, while in GWO-WOA there is no state storage. In the time complexity evaluation, ABC-MDP has a relatively slower computing time than GWO-WOA at the beginning of the iteration, this condition is because ABC-MDP has a computing stage in addition to calculating the fitness value also calculating the reward of each action taken. When a fitness value and reward have been obtained, the ABC-MDP algorithm can achieve optimal values and converge faster between 15% and 20% compared to GWO-WOA.

The testbed-based simulation mode consists of an application on a smartphone device as a task generator with interoperability feature support for selecting communication paths, either GSM, Wi-Fi or LoRa. The edge server section is equipped with an application to receive, identify, execute tasks and provide execution results to the sending unit. This simulation uses six scenarios that aim to vary the availability of the number of IoT, edge computing servers and computation iteration values. The purpose of setting these parameters is to measure the adaptability and reliability of the ABC-MDP-based computation offloading framework compared to GWO-WOA. Performance indicators are measured by five parameters, namely: 1) fitness value, 2) latency, 3) energy, 4) bandwidth, and 5) throughput. ABC-MDP gets a better performance value than GWO-WOA at 5.53% more optimal fitness value convergence, 3.53% lower latency, 1.97% more energy efficient and 0.43% greater throughput compared to GWO-WOA, while for bandwidth consumption it is still 2.22% greater than GWO-WOA.

In the implementation mode in the PPTK Gambung area, the computation offloading framework is applied in Internet of Things devices, multi-access edge computing and edge computing servers to perform computation offloading. The type of task used is a real-time task with 4 types of data, namely document data, image data, video data and alphanumeric data. Each task will be identified based on the type of task, time limit, priority and data type. In this implementation mode, IoT devices and GPS-based mobile Multi Access Edge Computing are involved to perform computation offloading to the edge computing server. IoT devices and Multi Access Edge Computing can choose the computation offloading path whether to use GSM, WiFi or LoRa communication paths adaptively based on RSSI values. To exchange messages between IoT, Multi Access Edge Computing and edge computing servers, the HTTP or MQTT message protocol is used. Based on testing with 130 tasks that have various data and time limits, 73 tasks or 56.15% were successfully optimized and 47 tasks or 43.84% were not successfully optimized. The failure of optimization is caused by the real-time task time constraints not being met because the resulting latency is greater than the computation time limit for executing the task.

Keywords: Artificial Bee Colony, Computation Offloading, Edge Computing, Internet of Things, Markov Decision Process.



@Hak cipta milik IPB University

Hak Cipta Dilindungi Undang-undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah
 - b. Pengutipan tidak merugikan kepentingan yang wajar IPB University.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB University.

© Hak Cipta milik IPB, tahun 2025
Hak Cipta dilindungi Undang-Undang

Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan atau menyebutkan sumbernya. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik, atau tinjauan suatu masalah, dan pengutipan tersebut tidak merugikan kepentingan IPB.

Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apa pun tanpa izin IPB.

OPTIMASI STRATEGI COMPUTATION OFFLOADING BERBASIS SWARM INTELLIGENCE DALAM LINGKUNGAN IOT- EDGE COMPUTING

AGUS MULYANA

Disertasi
sebagai salah satu syarat untuk memperoleh gelar
Doktor pada
Program Studi Doktor Ilmu Komputer

**PROGRAM STUDI DOKTOR ILMU KOMPUTER
SEKOLAH SAINS DATA, MATEMATIKA, DAN INFORMATIKA
INSTITUT PERTANIAN BOGOR
BOGOR
2025**



@Hak cipta milik IPB University

Hak Cipta Dilindungi Undang-undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah
 - b. Pengutipan tidak merugikan kepentingan yang wajar IPB University.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB University.

Penguji Luar Komisi Pembimbing pada Ujian Tertutup Disertasi:

- 1 Dr. Hendra Rahmanwan, S.Kom, M.T.
- 2 Dr. Hanif Fakhurroza, S.Si, M.T.

Promotor Luar Komisi Pembimbing pada Sidang Promosi Terbuka Disertasi:

- 1 Dr. Hendra Rahmanwan, S.Kom, M.T.
- 2 Dr. Hanif Fakhurroza, S.Si, M.T.



Judul Disertasi : Optimasi Strategi *Computation Offloading* Berbasis *Swarm Intelligence* dalam Lingkungan IoT- *Edge Computing*

Nama : Agus Mulyana
NIM : G6601211014

Disetujui oleh

Pembimbing 1:

Prof. Dr. Eng. Ir. Taufik Djatna, M.Si.



Pembimbing 2:

Dr. Ir. Sri Wahjuni, M.T.



Pembimbing 3:

Dr. Eng. Heru Sukoco, M.T.



Diketahui oleh

Ketua Program Studi:

Prof. Dr. Imas Sukaesih Sitanggang, S.Si., M.Kom.
NIP 197501301998022001



Dekan Sekolah Sains Data Matematika dan Informatika

Prof. Dr. Ir. Agus Bueno, M.Si, M.Kom
NIP 196607021993011001



Tanggal Ujian:
06 Mei 2025

Tanggal Lulus:

PRAKATA

Puji dan syukur penulis panjatkan kepada Allah subhanahu wa ta'ala atas segala karunia-Nya sehingga karya ilmiah ini berhasil diselesaikan. Tema yang dipilih dalam penelitian yang dilaksanakan sejak bulan Juni 2023 sampai bulan Desember 2024 ini adalah *computation offloading*, dengan judul “Optimasi Strategi *Computation Offloading* berbasis *Swarm Intelligence* dalam Lingkungan *IoT- Edge Computing*”.

Ucapan terima kasih penulis sampaikan, dan penghargaan yang tinggi diberikan kepada:

1. Para dosen pembimbing yang terdiri atas Prof. Dr. Eng. Ir Taufik Djatna, M.Si, Dr. Ir Sri Wahjuni, M.T. dan Dr. Eng Heru Sukoco, S.Si, M.T. yang telah membimbing, memberi saran, arahan dan motivasi kepada penulis selama penelitian disertasi ini. Semoga Allah SWT membalas kebaikan Bapak/Ibu semua, Aamiin.
2. Penguji luar komisi pembimbing, Dr. Hendra Rahmawan, S.Kom., M.T. dan Dr. Hanif Fakhurroja, S.Si., M.T., atas saran, koreksi dan diskusi dalam evaluasi disertasi ini.
3. Sekolah Pascasarjana IPB University, Ketua Departemen Ilmu Komputer, Kepada Dr. Sony Hartono Wijaya, S.Kom., M.Kom., Ketua Program Studi Doktor Ilmu Komputer, Prof. Dr. Imas Sukaesih Sitanggang, S.Si., M.Kom., dan semua dosen di Departemen Ilmu Komputer IPB University.
4. Khusus kepada kedua orang tua, Bapak Ta'ip dan Ibu Uka Suwarsih yang telah mendidik penulis sejak kecil hingga saat ini. Istri tercinta Erna Hernawati, S.IP, buat Ananda Azkiya Izzatunnisa dan Ananda Muhammad Syauqi sebagai penyemangat penulis, serta seluruh keluarga yang telah memberikan dukungan, do'a, dan kasih sayangnya selama penulis menyelesaikan studi S3 di IPB. Semoga kasih sayang, dukungan dan kesabaran selama ini mendapat keberkahan dari Allah SWT.
5. Rektor Universitas Komputer Indonesia Prof. Dr. Ir. Eddy Soeryanto Soegoto, M.T., Dekan FTIK-UNIKOM dan seluruh kolega di Program Studi Sistem dan Teknik Komputer-FTIK-UNIKOM.
6. Semua sahabat penelitian di PPTK Gambung dan PT. Makerindo Prima Solusi yang telah banyak memberikan masukan, tanggapan serta dukungan selama penelitian.
7. Segenap sahabat program Doktor Ilmu Komputer, khususnya Angkatan 2021 serta semua kawan kawan lainnya yang banyak membantu selama studi.

Semoga karya ilmiah ini bermanfaat bagi pihak yang membutuhkan dan bagi kemajuan ilmu pengetahuan.

Bogor, Agustus 2025

Agus Mulyana



DAFTAR ISI

DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
DAFTAR LAMPIRAN	xiv
DAFTAR ISTILAH	i
PENDAHULUAN	1
1.1 Latar Belakang	2
1.2 Rumusan Masalah	4
1.3 Tujuan	5
1.4 Manfaat	5
1.5 Ruang Lingkup	5
1.6 Kebaruan	6
II TINJAUAN PUSTAKA	7
2.1 <i>Edge Computing</i>	7
2.2 <i>Computation Offloading</i>	10
2.3 <i>Strategi Computation Offloading</i>	12
2.4 <i>Kerangka Kerja Computation Offloading</i>	13
2.4.1 <i>Arsitektur Sistem</i>	13
2.4.2 <i>Proses Computation Offloading</i>	14
2.4.3 <i>Metode Pengambilan Keputusan</i>	14
2.4.4 <i>Parameter yang dipertimbangkan</i>	14
2.4.5 <i>Taksonomi Computation Offloading</i>	14
2.5 <i>Swarm Intelligence</i>	15
2.6 <i>Optimasi</i>	18
2.6.1 <i>Artificial Bee Colony (ABC)</i>	19
2.6.2 <i>Grey Wolf Optimization – Whale Optimization Algorithm</i>	21
2.6.3 <i>Pergerakan Serigala Abu-Abu dalam GWO</i>	22
2.6.4 <i>Strategi Perburuan Paus dalam WOA</i>	22
2.7 <i>Markov Decision Process (MDP)</i>	25
2.7.1 <i>Kebijakan dan Fungsi Nilai</i>	27
2.8 <i>State of The Art Penelitian</i>	28
2.9 <i>Skenario Evaluasi Kinerja Sistem</i>	30
2.10 <i>Evaluasi Quality of Service (QoS)</i>	31
2.10.1 <i>Latensi ($T(x)$)</i>	31
2.10.2 <i>Konsumsi Energi ($E(x)$)</i>	32
2.10.3 <i>Throughput ($C(x)$)</i>	32
2.10.4 <i>Keandalan ($R(x)$)</i>	32
III METODE	33
3.1 Waktu dan Tempat Penelitian	33
3.2 Alat, Bahan, dan Data	33

3.3	Alur Kerja Penelitian	34
3.4	Tahapan Penelitian dan Indikator Keberhasilan	35
3.5	Model Sistem Lingkungan IoT-Edge Computing	36
3.5.1	Fungsi Tujuan	37
3.5.2	Perhitungan Parameter <i>Quality of Service</i> (QoS)	37
3.5.3	Batasan	39
3.6	Pengembangan Kerangka Kerja <i>Edge Computing</i>	39
3.6.1	Peran <i>Orchestrator</i> untuk <i>Computation Offloading</i>	39
3.6.2	Tahapan Pengembangan <i>Orchestrator Edge Computing</i>	41
3.7	Pengembangan Algoritma ABC-MDP	43
3.8	Desain Evaluasi Numerik ABC-MDP	47
3.8.1	Tujuan dan Desain Sistem Evaluasi Numerik	47
3.8.2	Evaluasi <i>Fitness Value</i> dan Konvergensi	48
3.8.3	Evaluasi Ruang dan Waktu	49
3.9	Desain Simulasi dan Evaluasi Kinerja ABC-MDP untuk <i>Computation Offloading</i> Berbasis <i>Testbed</i>	51
3.10	Desain Implementasi dan Evaluasi Kinerja ABC-MDP untuk <i>Computation Offloading</i>	52
IV	HASIL DAN PEMBAHASAN	57
4.1	Hasil Evaluasi Numerik	57
4.1.1	Evaluasi Konsumsi Memori dan Waktu Eksekusi	60
4.2	Hasil Simulasi <i>Computation Offloading</i> Berbasis <i>Testbed</i>	62
4.2.1	Simulasi <i>Computation Offloading</i> berbasis <i>Testbed</i> Skenario 1	62
4.2.2	Simulasi <i>Computation Offloading</i> berbasis <i>Testbed</i> Skenario 2	63
4.2.3	Simulasi <i>Computation Offloading</i> berbasis <i>Testbed</i> Skenario 3	63
4.2.4	Simulasi <i>Computation Offloading</i> berbasis <i>Testbed</i> Skenario 4	64
4.2.5	Simulasi <i>Computation Offloading</i> berbasis <i>Testbed</i> Skenario 5	65
4.2.6	Simulasi <i>Computation Offloading</i> berbasis <i>Testbed</i> Skenario 6	66
4.3	Skema Implementasi <i>Computation Offloading</i>	68
4.3.1	Evaluasi Kerangka Kerja <i>Computation offloading</i> pada Mode <i>Data Offloading</i>	69
4.3.2	Evaluasi Kerangka Kerja <i>Computation Offloading</i> Pada Perangkat Bergerak	77
4.4	Pembahasan Hasil Evaluasi	78
4.5	Kerbatasan Penelitian	79
4.6	Peneliti Selanjutnya	80
V	SIMPULAN DAN SARAN	81
5.1	Simpulan	81
5.2	Saran	81
	DAFTAR PUSTAKA	82
	LAMPIRAN	87
	RIWAYAT HIDUP	105