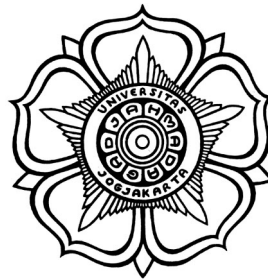


SKRIPSI

**OPTIMASI MULTIOBJEKTIF PENEMPATAN MESIN VIRTUAL DAN
PENENTUAN RUTE JARINGAN PADA *CLOUD DATA CENTER* BERBASIS
ALGORITMA GENETIKA *NONDOMINATED SORTING***

***MULTI-OBJECTIVE OPTIMIZATION OF VIRTUAL MACHINE
PLACEMENT AND NETWORK ROUTING IN CLOUD DATA CENTERS
BASED ON NONDOMINATED SORTING GENETIC ALGORITHM***



**GUSTI AGUNG RAMA AYUDHYA
20/459266/PA/19927**

**PROGRAM STUDI ILMU KOMPUTER
DEPARTEMEN ILMU KOMPUTER DAN ELEKTRONIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS GADJAH MADA
YOGYAKARTA**

2025

SKRIPSI

**OPTIMASI MULTIOBJEKTIF PENEMPATAN MESIN VIRTUAL DAN
PENENTUAN RUTE JARINGAN PADA *CLOUD DATA CENTER* BERBASIS
ALGORITMA GENETIKA *NONDOMINATED SORTING***

***MULTI-OBJECTIVE OPTIMIZATION OF VIRTUAL MACHINE
PLACEMENT AND NETWORK ROUTING IN CLOUD DATA CENTERS
BASED ON NONDOMINATED SORTING GENETIC ALGORITHM***

Usulan Penelitian



GUSTI AGUNG RAMA AYUDHYA
20/459266/PA/19927

**PROGRAM STUDI ILMU KOMPUTER
DEPERTEMEN ILMU KOMPUTER DAN ELEKTRONIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS GADJAH MADA
YOGYAKARTA**

2025

HALAMAN PENGESAHAN

SKRIPSI

OPTIMASI MULTIOBJEKTIF PENEMPATAN MESIN VIRTUAL DAN PENENTUAN RUTE JARINGAN PADA *CLOUD DATA CENTER* BERBASIS ALGORITMA GENETIKA *NONDOMINATED SORTING*

Telah dipersiapkan dan disusun oleh

GUSTI AGUNG RAMA AYUDHYA
20/459266/PA/19927

Telah dipertahankan di depan Tim Penguji
pada tanggal Mei 2025

Susunan Tim Penguji

Drs. Medi, S.Kom., M.Cs
Pembimbing

-
Ketua Penguji

-
Anggota Penguji

PERNYATAAN

Dengan ini saya menyatakan bahwa dalam Skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, Mei 2025

Gusti Agung Rama Ayudhya

Karya ini ku persembahkan kepada
Ibu, Bapak, dan adik-adikku tercinta serta teman-teman
seperjuangan di Ilmu Komputer Universitas Gadjah Mada

PRAKATA

Puji syukur penulis panjatkan ke hadapan Ida Hyang Widhi Wasa, Tuhan Yang Maha Esa, karena atas *asung kerta wara nugraha*-Nya, penulis dapat menyelesaikan skripsi ini dengan judul "Optimasi Multiobjektif Penempatan Mesin Virtual dan Penentuan Rute Jaringan pada *Cloud Data Center* Berbasis Algoritma Genetika *Nondominated Sorting*". Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Ilmu Komputer pada Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Gadjah Mada.

Penulis menyadari bahwa penyusunan skripsi ini tidak terlepas dari dukungan, bimbingan, dan bantuan dari berbagai pihak. Oleh karena itu, pada kesempatan ini, penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Ayah dan Ibu, yang telah membimbing, mendukung, mendoakan, serta membiayai penulis hingga dapat menempuh pendidikan di Universitas Gadjah Mada.
2. Gusti Ayu Bulan Adhistanaya dan Gusti Agung Deva Maheswara, kedua adik penulis, serta seluruh keluarga besar yang senantiasa memberikan semangat dan dukungan selama proses penyusunan skripsi ini.
3. Bapak Drs. Medi, M.Kom., selaku Dosen Pembimbing, yang telah dengan sabar membimbing, memberikan ilmu, arahan, masukan, serta koreksi selama proses penulisan skripsi ini.
4. Almarhumah Ibu Anny Kartika Sari, S.Si., M.Sc. dan Bapak Lukman Heryawan, S.T., M.T., selaku Dosen Pembimbing Akademik, atas segala bimbingan, nasihat, dan bantuan selama penulis menempuh studi di Program Studi Ilmu Komputer.
5. Tim Penguji, yang telah memberikan kritik, saran, serta masukan berharga untuk penyempurnaan skripsi ini.
6. Seluruh dosen dan staf Fakultas MIPA UGM, khususnya di Program Studi Ilmu Komputer, yang telah memberikan ilmu dan dukungan selama proses studi.
7. Teman-teman Ilmu Komputer Angkatan 2020 dan OmahTI, yang telah menjadi rekan seperjuangan dan selalu siap membantu, baik dalam perkuliahan maupun dalam penyusunan skripsi.

8. Rekan-rekan Tim KKN Punung Periode 4 Tahun 2023 Unit JI-081 (Senandung Punung), dan warga Desa Wareng, Kabupaten Pacitan, Jawa Timur, atas kebersamaan, pengalaman, dan kisah tak terlupakan selama lima puluh hari pengabdian.
9. Seluruh staf Perpustakaan dan Arsip Universitas Gadjah Mada, yang telah menyediakan fasilitas dan referensi penting dalam mendukung penyusunan skripsi ini.
10. Seluruh pihak yang tidak dapat disebutkan satu per satu, yang turut membantu dan mendukung penulis selama proses penulisan skripsi ini.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Oleh karena itu, penulis terbuka atas segala kritik dan saran yang membangun demi perbaikan ke depan. Penulis berharap skripsi ini dapat memberikan manfaat, baik dalam pengembangan ilmu komputer maupun sebagai referensi bagi penelitian selanjutnya.

Yogyakarta, Mei 2025

Penulis

DAFTAR ISI

Halaman Judul	ii
Halaman Pengesahan	iii
Halaman Pernyataan	iv
Halaman Persembahan	v
Halaman Motto	vi
PRAKATA	vii
DAFTAR ISI	ix
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
INTISARI	xiii
ABSTRACT	xv
I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	5
1.6 Metodologi Penelitian	5
1.7 Sistematika Penulisan	7
II TINJAUAN PUSTAKA	9
III DASAR TEORI	10
3.1 Representational State Transfer	10
3.2 JavaScript Object Notation	11

3.2.1	Definisi	11
3.2.2	Contoh	11
IV	ANALISIS DAN PERANCANGAN SISTEM	12
4.1	Deskripsi Umum Sistem	12
4.2	Analisis Kebutuhan Sistem	12
4.3	Pembuatan Sistem	12
4.3.1	Pembuatan Sistem Pengenalan Entitas Bernama	12
4.3.2	Pembuatan Sistem Ekstraksi Kalimat Pernyataan	12
4.4	Rancangan Antarmuka	12
4.4.1	Deskripsi	12
4.4.2	<i>Wireframe</i>	12
V	IMPLEMENTASI SISTEM	14
5.1	Spesifikasi	14
5.2	Implementasi Sistem Pengenalan Entitas Bernama	14
5.3	Implementasi Sistem Ekstraksi Kalimat Pernyataan	14
VI	PENGUJIAN DAN PEMBAHASAN SISTEM	15
6.1	Pengujian Sistem Pengenalan Entitas Bernama	15
6.2	Pengujian Sistem Ekstraksi Kalimat Pernyataan	15
VII	PENUTUP	16
7.1	Kesimpulan	16
7.2	Saran	16
DAFTAR PUSTAKA		17
A	BERKAS JSON UNTUK MODEL SISTEM PENGENALAN ENTITAS BERNAMA	18

DAFTAR TABEL

2.1	My caption	9
-----	------------	---

DAFTAR GAMBAR

INTISARI

Optimasi Multiobjektif Penempatan Mesin Virtual dan Penentuan Rute Jaringan pada *Cloud Data Center* Berbasis Algoritma Genetika *Nondominated Sorting*

Oleh

Gusti Agung Rama Ayudhya

20/459266/PA/19927

Komputasi awan telah menjadi infrastruktur utama dalam teknologi informasi karena kemampuannya menyediakan sumber daya secara *scalable* dan elastis sesuai permintaan pengguna. Kemampuan ini didukung oleh virtualisasi, yang memungkinkan penyewaan sumber daya tanpa pengelolaan langsung oleh pengguna, sekaligus meningkatkan efisiensi dan keberlanjutan *data center*. Salah satu tantangan utama dalam komputasi awan adalah penempatan VM (*virtual machine* atau mesin virtual) pada PM (*physical machine* atau mesin fisik) dan rekayasa lalu lintas (*traffic engineering*) jaringan *data center*, yang harus mempertimbangkan berbagai efisiensi operasional, seperti konsumsi energi dan efisiensi sumber daya, serta *Quality of Service* (QoS), seperti alokasi *bandwidth* dan *latency* komunikasi antar-VM.

Optimasi penempatan VM dan penentuan rute menjadi semakin kompleks karena adanya kendala yang sering kali saling bertentangan, sehingga lebih cocok diselesaikan menggunakan metode optimasi multiobjektif. Masalah ini bersifat *NP-complete*, sehingga lebih cocok diselesaikan menggunakan algoritma heuristik dibandingkan algoritma eksak. Penelitian ini menggunakan NSGA-III (*Nondominated Sorting Genetic Algorithm III*), sebuah algoritma genetika untuk menyelesaikan masalah multiobjektif, dengan tujuan mengoptimalkan konsumsi energi, efisiensi sumber daya dalam penempatan VM, alokasi *bandwidth*, dan *latency* komunikasi dalam penentuan rute jaringan secara bersamaan.

Evaluasi performa NSGA-III dilakukan melalui simulasi menggunakan CloudSim Plus, dengan mempertimbangkan berbagai skenario seperti topologi jaringan, kebutuhan sumber daya VM, kapasitas PM, serta parameter NSGA-III. Performa NSGA-III akan diukur menggunakan beberapa metrik, termasuk *hypervolume* dan rata-rata jarak antargenerasi. Selain itu, solusi yang diperoleh NSGA-III akan dibandingkan dengan solusi eksak yang diperoleh *LP solver* (pemecah pemrograman linier) serta beberapa kombinasi algoritma heuristik untuk penempatan VM dan penentuan rute.

Kata Kunci: Komputasi awan, *data center*, penempatan mesin virtual,

konsumsi energi, efisiensi sumber daya, masalah *bin packing*, perutean jaringan, alokasi *bandwidth*, *latency*, masalah *multicommodity flow*, metode optimasi multiobjektif, *Nondominated Sorting Genetic Algorithm III* (NSGA-III)

ABSTRACT

Multi-Objective Optimization of Virtual Machine Placement and Network Routing in Cloud Data Centers Based on Nondominated Sorting Genetic Algorithm

By

Gusti Agung Rama Ayudhya

20/459266/PA/19927

Cloud computing has become a key infrastructure in information technology due to its ability to provide scalable and elastic resources on demand. This capability is supported by virtualization, which enables resource rental without direct management by users while also improving efficiency and sustainability of data centers. One of the main challenges in cloud computing is the placement of virtual machines (VMs) on physical machines (PMs) and traffic engineering in data center networks, which must consider various operational efficiency factors, such as energy consumption and resource utilization, as well as Quality of Service (QoS) aspects, such as bandwidth allocation and communication latency between VMs.

VM placement and network routing optimization becomes increasingly complex due to conflicting constraints, making multiobjective optimization methods suitable for solving this problem. On the top of that, this problem is NP-complete, making heuristic algorithms more suitable than exact algorithms. This study employs NSGA-III (Nondominated Sorting Genetic Algorithm III), a genetic algorithm for solving multi-objective problems, aiming to optimize energy consumption, resource efficiency in VM placement, bandwidth allocation, and communication latency in network routing simultaneously.

The performance of NSGA-III is evaluated through simulations using CloudSim Plus, considering various scenarios such as network topology, VM resource demands, PM capacity, and NSGA-III parameters. The performance of NSGA-III is measured using several metrics, including hypervolume and the average intergenerational distance. Additionally, the solutions obtained by NSGA-III are compared with exact solutions obtained from linear programming (LP) solvers and several heuristic algorithm combinations for VM placement and routing.

Keywords: cloud computing, data center, virtual machine placement, energy consumption, resource wastage, bin packing problem, network routing, bandwidth allocation, latency, multicommodity flow problem, multiobjective optimization method, Nondominated Sorting Genetic Algorithm

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seiring perkembangan teknologi informasi, *cloud computing* (komputasi awan) telah menjadi infrastruktur utama bagi penyedia layanan komputasi. Komputasi awan memungkinkan penggunaan sumber daya—seperti penyimpanan, jaringan, daya komputasi, basis data, *platform*, dan layanan aplikasi—secara efisien melalui Internet. Model komputasi ini semakin diminati karena mampu menyediakan sumber daya yang bersifat *scalable* dan elastis sesuai permintaan pengguna (*on-demand*).

Kemampuan ini dimungkinkan oleh teknologi virtualisasi. Virtualisasi memungkinkan pengguna menyewa dan mengakses sumber daya yang sudah diabstraksi tanpa perlu memiliki, mengelola, atau merawatnya secara langsung. Virtualisasi juga mendukung pendekatan *multi-tenancy*, yaitu penggunaan infrastruktur yang sama oleh banyak pengguna, sehingga penyedia layanan dapat menghemat biaya operasional (Cloudflare, t.t.). Dengan adanya virtualisasi, VM (*virtual machine* atau mesin virtual) dapat dibuat sebagai emulasi perangkat keras fisik yang mampu menjalankan sistem operasi dan aplikasi layaknya komputer fisik. VM ini dapat dengan mudah dijalankan, diskalakan, dimigrasikan, direplikasikan, atau dihancurkan, sehingga meningkatkan skalabilitas, elastisitas, fault-tolerance, ketersediaan, penghindaran bencana (*disaster avoidance*), dan pemulihan bencana (*disaster recovery*) (Hill dkk., 2013).

Namun, supaya dapat digunakan, VM harus ditempatkan pada PM (*physical machine* atau mesin fisik). Penempatan VM menjadi tantangan utama bagi penyedia layanan komputasi awan karena harus memperhatikan kebutuhan sumber daya VM serta kapasitas PM. Penempatan yang optimal dapat meningkatkan efisiensi data center dan metrik performa, seperti minimalisasi konsumsi energi, efisiensi penggunaan sumber daya, serta minimalisasi biaya komunikasi antar-VM.

Komunikasi antar-VM dalam *cloud data center* juga memainkan peran penting dalam kinerja aplikasi. *Cloud data center* skala besar, yang biasanya memiliki ribuan PM terhubung oleh perangkat jaringan, sering kali digunakan untuk mendukung aplikasi yang terdiri atas beberapa komponen saling bergantung.

Komponen yang saling berkomunikasi melalui jaringan dapat menyebabkan *delay*, yang berdampak pada jumlah tugas (*task*) yang dapat diproses oleh VM per satuan waktu. Penempatan VM dan pemilihan jalur komunikasi yang ideal mampu mengurangi *delay* komunikasi dengan menempatkan VM yang saling berkomunikasi intensif pada PM yang sama atau PM yang terhubung oleh jalur jaringan sesingkat mungkin.

Selain itu, manajemen energi di data center menjadi isu krusial dalam komputasi awan. Konsumsi energi tidak hanya memengaruhi biaya operasional tetapi juga berdampak pada lingkungan, seperti emisi karbon yang dihasilkan oleh perangkat fisik, terutama server. Menurut survei yang dilakukan oleh Lawrence Berkeley National Laboratory (2024), sekitar 60% total konsumsi energi di data center berasal dari mesin fisik. Mesin fisik *idle* (menyala tetapi tidak menjalankan *task* apapun) rata-rata mengonsumsi energi sebesar 70% dari energi yang digunakan oleh mesin fisik dengan utilisasi CPU maksimal (Beloglazov, Abawajy, dan Buyya, 2012). Dengan penempatan VM yang ideal, lebih banyak VM dapat ditempatkan pada PM dengan tingkat utilisasi tinggi. Mesin fisik yang *idle* dapat dimatikan, sehingga jumlah perangkat aktif berkurang dan konsumsi energi *data center* dapat ditekan. Pendekatan ini tidak hanya meningkatkan efisiensi energi tetapi juga mendukung keberlanjutan lingkungan.

Kebutuhan mengoptimalkan beberapa metrik performa sekaligus: konsumsi energi *data center*, efisiensi penggunaan sumber daya komputasi, alokasi *bandwidth*, dan *latency* komunikasi antar-VM, membuat penempatan VM semakin kompleks. Pengoptimalan metrik-metrik tersebut sering kali saling bertentangan, di mana pengoptimalan salah satu metrik dapat memperburuk metrik lainnya. Selain itu, terdapat kendala yang harus dipenuhi, seperti kebutuhan sumber daya dan kapasitas PM, serta pemilihan jalur komunikasi antar-VM, yang membatasi ruang solusi. Untuk itu, setiap metrik yang dipertimbangkan harus dioptimalkan secara setara tanpa memberikan prioritas pada salah satu metrik. Masalah ini sangat cocok untuk diselesaikan menggunakan metode optimasi multiobjektif.

Masalah penempatan VM merupakan salah satu bentuk masalah *bin packing* (Fatima dkk., 2018), sedangkan masalah pemilihan jalur komunikasi antar-VM merupakan salah satu bentuk masalah *multicommodity flow* (Fortz, Gouveia Joyce-Moniz, 2017). Keduanya dikenal memiliki kompleksitas *NP-complete*, sehingga secara keseluruhan, masalah ini juga berkompleksitas *NP-complete*. Oleh karena itu, masalah ini kurang cocok diselesaikan dengan algoritma eksak. Sebagai gantinya,

berbagai algoritma heuristik dan metaheuristik telah dikembangkan untuk mencari solusi yang cukup optimal dengan pendekatan yang lebih efisien,

Dalam penelitian ini, penulis menggunakan algoritma NSGA-III (*Nondominated Sorting Genetic Algorithm III*), salah satu varian algoritma genetika yang dirancang untuk menyelesaikan masalah multiobjektif. Algoritma ini akan digunakan untuk menempatkan VM pada PM serta memilih rute komunikasi antar-VM, dengan tujuan mengoptimalkan empat metrik performa sekaligus: konsumsi energi *cloud data center*, efisiensi penggunaan sumber daya komputasi, alokasi *bandwidth*, dan *latency* komunikasi antar-VM.

Kinerja NSGA-III akan dievaluasi pada berbagai skenario, mencakup parameter NSGA-III itu sendiri, variasi topologi jaringan, kebutuhan sumber daya setiap VM, kapasitas sumber daya tiap PM, *bandwidth* yang diperlukan oleh pasangan VM yang saling berkomunikasi, serta kapasitas dan *latency* setiap *link* dalam jaringan. Penelitian ini menggunakan CloudSim Plus untuk memodelkan dan mensimulasikan lingkungan komputasi awan beserta algoritma penempatan VM dan pemilihan jalur komunikasinya.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, terdapat beberapa masalah yang hendak dipecahkan melalui penelitian ini:

1. Bagaimana mengembangkan algoritma penempatan VM dan penentuan rute komunikasi antar-VM yang mampu menangani *trade-off* antara konsumsi energi, efisiensi penggunaan sumber daya, alokasi *bandwidth*, dan *latency* komunikasi antar-VM secara bersamaan?
2. Bagaimana memastikan algoritma yang dikembangkan dapat menghasilkan solusi optimal Pareto yang menggambarkan berbagai skenario kompromi antar metrik performa, sehingga memungkinkan pembuat keputusan memilih solusi yang paling sesuai?
3. Bagaimana kinerja algoritma NSGA-III dalam menyelesaikan masalah penempatan VM dan penentuan jalur komunikasi antar-VM, seperti kualitas *Pareto front*, efisiensi komputasi, dan stabilitas solusi dalam berbagai skenario lingkungan *data center*?

1.3 Batasan Masalah

Supaya penelitian lebih terfokus, terdapat beberapa batasan yang diterapkan:

1. Evaluasi melalui Simulasi

Kinerja algoritma dievaluasi melalui simulasi menggunakan CloudSim Plus. Oleh karena itu, performa algoritma dalam lingkungan nyata yang lebih dinamis dapat berbeda dengan hasil simulasi.

2. Asumsi Lingkungan Statis

Penelitian ini mengasumsikan lingkungan *cloud* bersifat statis, di mana semua parameter permasalahan, seperti jumlah VM dan PM, kebutuhan sumber daya dari VM, kapasitas PM, serta karakteristik jaringan, tidak mengalami perubahan selama simulasi berlangsung. Dengan demikian, algoritma yang dikembangkan bersifat *offline* dan tidak menangani perubahan dinamis, seperti kedatangan atau penghentian VM secara *real-time*.

3. Komunikasi Antar-VM dalam PM yang Sama

Jika dua VM ditempatkan pada PM yang sama, komunikasi antar-VM dilakukan melalui *memory sharing* tanpa memanfaatkan jaringan tambahan. Kebutuhan memori yang dialokasikan untuk setiap VM diasumsikan sudah mencakup kebutuhan tambahan untuk *memory sharing*.

4. Penggunaan *Software Defined Network* (SDN) pada *Data Center*

Penelitian ini mengasumsikan bahwa informasi mengenai topologi jaringan *data center* tersedia bagi pembuat keputusan, termasuk struktur topologi, kapasitas setiap *link*, dan kecepatan transmisinya. Dengan adanya SDN, jalur komunikasi antar-VM dapat ditentukan berdasarkan informasi tersebut.

5. Penggunaan Banyak Jalur untuk Setiap Komunikasi

Setiap komunikasi antar-VM dapat menggunakan lebih dari satu jalur secara bersamaan (*multipath routing*). Hal ini didukung oleh protokol seperti *Multipath Transmission Control Protocol* (MP-TCP) yang memungkinkan distribusi lalu lintas melalui beberapa jalur dalam jaringan.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk menyelesaikan masalah yang telah dirumuskan, antara lain:

1. Mengembangkan algoritma penempatan VM dan penentuan rute komunikasi berbasis NSGA-III yang mampu menangani *trade-off* antara empat metrik: konsumsi energi, penggunaan sumber daya, alokasi *bandwidth*, dan *latency* komunikasi antar-VM.
2. Merancang algoritma yang menghasilkan himpunan solusi optimal Pareto, sehingga pembuat keputusan dapat memilih solusi yang sesuai dengan kebutuhan berdasarkan kompromi antar metrik.
3. Mengevaluasi kinerja algoritma NSGA-III melalui simulasi menggunakan CloudSim Plus dengan mengukur kualitas *Pareto front*, efisiensi komputasi, dan stabilitas solusi pada berbagai skenario lingkungan *data center*.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Kontribusi terhadap pengembangan algoritma penempatan VM dan pemilihan rute komunikasi antar-VM melalui pendekatan optimasi multiobjektif berbasis NSGA-III. Hasil penelitian ini dapat menjadi acuan bagi penelitian selanjutnya dalam bidang penjadwalan dan pengelolaan sumber daya di *cloud computing*.
2. Potensi penerapan dalam lingkungan komputasi awan nyata, khususnya dalam meningkatkan efisiensi penggunaan sumber daya, mengoptimalkan konsumsi energi, dan mengurangi *latency* komunikasi antar-VM.
3. Menyediakan referensi bagi penyedia layanan komputasi awan dalam merancang strategi penempatan VM yang mempertimbangkan kompromi antara berbagai metrik performa, sehingga dapat meningkatkan QoS dan efisiensi operasional *data center*.

1.6 Metodologi Penelitian

Penelitian ini dilakukan melalui beberapa tahapan sebagai berikut:

1. Studi Literatur

Tahap ini mencakup kajian terhadap penelitian terdahulu yang berkaitan dengan penempatan VM, pemilihan rute komunikasi antar-VM, serta metode optimasi multiobjektif berbasis algoritma evolusioner. Literatur yang dikaji mencakup

buku, jurnal ilmiah, serta publikasi konferensi yang relevan. Selain itu, studi juga dilakukan terhadap teknologi yang digunakan, seperti NSGA-III sebagai algoritma optimasi, CloudSim Plus untuk simulasi, dan DOCPLEX sebagai *LP solver*.

2. Perancangan Algoritma Penempatan VM dan Lingkungan Simulasi

Pada tahap ini, dilakukan perancangan model optimasi multiobjektif yang mencakup:

- (a) Formulasi masalah optimasi dalam bentuk fungsi objektif dan kendala.
- (b) Pemilihan metode representasi kromosom untuk solusi penempatan VM dan pemilihan rute komunikasi.
- (c) Perancangan operator algoritma genetika untuk NSGA-III, termasuk crossover, mutasi, dan mekanisme perbaikan.
- (d) Penentuan skenario simulasi, seperti topologi jaringan, konfigurasi VM dan PM, serta parameter simulasi lainnya.

3. Implementasi Algoritma

Tahap ini mencakup implementasi algoritma NSGAIII dalam lingkungan simulasi CloudSim Plus. Implementasi meliputi:

- (a) Implementasi algoritma NSGA-III ke dalam program Java.
- (b) Integrasi algoritma NSGA-III dengan CloudSim Plus untuk menilai performa solusi yang dihasilkan.
- (c) Pengujian awal untuk memastikan algoritma berjalan sesuai dengan perancangan.

4. Pengujian Algoritma pada Lingkungan Simulasi dan Evaluasi Kinerja

Algoritma yang telah diimplementasikan diuji pada berbagai skenario simulasi dengan variasi parameter, seperti:

- (a) Topologi jaringan yang berbeda (misalnya, FatTree, BCube, DCell).
- (b) Jumlah VM dan PM yang bervariasi.
- (c) Kebutuhan sumber daya setiap VM dan kapasitas sumber daya di setiap PM yang bervariasi.

Evaluasi kinerja dilakukan dengan membandingkan kualitas solusi Pareto yang dihasilkan, efisiensi komputasi algoritma, serta stabilitas hasil dalam berbagai skenario.

5. Penulisan Laporan

Hasil penelitian yang mencakup formulasi masalah, perancangan algoritma, hasil eksperimen, dan analisis kinerja algoritma didokumentasikan dalam bentuk skripsi.

1.7 Sistematika Penulisan

Skripsi ini ditulis dalam tujuh bab dengan struktur sebagai berikut:

Bab I: Pendahuluan

Bab ini membahas latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan skripsi ini.

Bab II: Tinjauan Pustaka

Bab ini membahas penelitian terdahulu yang berkaitan dengan metode penempatan VM, optimasi multiobjektif, serta pemodelan *cloud data center*. Selain itu, tinjauan terhadap algoritma evolusioner seperti NSGA-III juga disertakan untuk memberikan dasar pengembangan metode dalam penelitian ini.

Bab III: Landasan Teori

Bab ini membahas teori dan konsep yang mendukung penelitian, termasuk komputasi awan, metode optimasi multiobjektif, algoritma genetika, dan NSGA-III. Selain itu, bab ini menguraikan cara kerja NSGA-III serta formulasi masalah penempatan VM dan pemilihan jalur komunikasi antar-VM sebagai masalah optimasi multiobjektif.

Bab IV: Analisis dan Rancangan Sistem

Bab ini membahas analisis masalah, perancangan algoritma, perancangan lingkungan simulasi, serta rancangan evaluasi kinerja algoritma.

Bab V: Implementasi

Bab ini membahas implementasi algoritma NSGA-III, integrasi NSGA-III dengan lingkungan simulasi CloudSim Plus, pengaturan parameter eksperimen, serta mengevaluasi kinerja algoritma berdasarkan skenario yang dipertimbangkan dalam simulasi.

Bab VI: Hasil dan Pembahasan

Bab ini membahas hasil evaluasi kinerja algoritma berdasarkan simulasi yang

telah dilakukan. Analisis dilakukan terhadap kualitas solusi Pareto, efisiensi komputasi, serta stabilitas algoritma dalam berbagai skenario simulasi.

Bab VII: Kesimpulan

Bab ini berisi kesimpulan penelitian serta saran untuk pengembangan lebih lanjut.

BAB II

TINJAUAN PUSTAKA

Seperti yang telah dibahas pada bab sebelumnya, masalah penempatan mesin virtual memiliki kompleksitas *NP-complete*. Oleh karena itu, berbagai algoritma heuristik dan metaheuristik telah dikembangkan untuk menangani masalah ini secara lebih efisien.

Metode-metode tersebut dirancang untuk mengoptimalkan berbagai metrik performa, terutama konsumsi daya server dan pemborosan sumber daya. Meskipun berbagai model konsumsi daya server (Ahmed, Bollen Alvarez, 2021) dan model penggunaan sumber daya telah dikembangkan, pada sebagian besar penelitian, konsumsi daya server diukur berdasarkan model yang dikembangkan oleh Beloglazov, Abawajy, dan Buyya (2021), sementara pemborosan sumber daya dihitung menggunakan model dari Gao dkk. (2013). Sejumlah penelitian menyederhanakan pengukuran konsumsi energi *data center* dengan cara menghitung banyaknya server yang aktif (sedang menjalankan mesin virtual).

2.1 Algoritma Penempatan Mesin Virtual

2.1.1 Metode Heuristik

Beberapa adaptasi algoritma klasik untuk masalah *bin packing* seperti *First Fit* (FF), *First Fit Decreasing* (FFD), *Random Fit* (RF), *Best Fit* (BF), dan *Best Fit Decreasing* (BFD), dapat digunakan untuk menentukan penempatan VM yang optimal (Alharabe, Rakrouki Aljohani, 2022). Namun, solusi yang didapat belum cukup optimal. Selain itu, algoritma ini lebih cocok digunakan untuk mengoptimalkan satu objektif saja, seperti konsumsi energi. Untuk memperoleh solusi yang lebih optimal, metode seperti MinPR (Azizi, Zandsalimi Li, 2020), GRVMP (*Greedy Randomized Virtual Machine Placement*) (Azizi dkk., 2021), dan CRBFF (*Combinated Random Best First Fit*) (Yousefi Babamir, 2024) dikembangkan untuk meminimalkan konsumsi energi sekaligus mengurangi pemborosan sumber daya. Selain itu, algoritma non-greedy seperti WPRVMP (*Weighted PageRank-based Virtual Machine Placement*) memanfaatkan algoritma *weighted PageRank* untuk mengurangi jumlah server aktif sambil memaksimalkan pemanfaatan sumber daya server tersebut.

2.1.2 Metode Metaheuristik

Pendekatan metaheuristik, khususnya algoritma evolusioner, banyak digunakan dalam optimasi penempatan mesin virtual. Gao dkk. (2013) mengembangkan VMPACS (*Virtual Machine Placement with Ant Colony System*) berbasis ACO (*Ant Colony Optimization*) untuk meminimalkan konsumsi daya server dan pemborosan sumber daya. Alharabe, Rakrouki, dan Aljohani (2022) memperkenalkan HACOS, yang mengintegrasikan ACO dengan *simulated annealing* untuk mengoptimalkan *network traffic* dan tingkat penggunaan maksimum pada *link* jaringan. Liu dkk. (2018) menciptakan OEMACS untuk mengurangi jumlah server aktif dalam *data center*. Wei dkk. (2019) mengembangkan AP-ACO (*Adaptive Parameter Ant Colony Optimization*), yang parameternya dapat beradaptasi, untuk meminimalkan konsumsi daya dan biaya komunikasi antar-VM.

ACO (*Ant Colony Optimization*) Pendekatan metaheuristik, khususnya algoritma evolusioner, banyak digunakan dalam optimasi penempatan mesin virtual. Gao dkk. (2013) mengembangkan VMPACS (*Virtual Machine Placement with Ant Colony System*) berbasis ACO (*Ant Colony Optimization*) untuk meminimalkan konsumsi daya server dan pemborosan sumber daya. Alharabe, Rakrouki, dan Aljohani (2022) memperkenalkan HACOS, yang mengintegrasikan ACO dengan *simulated annealing* untuk mengoptimalkan *network traffic* dan tingkat penggunaan maksimum pada *link* jaringan. Liu dkk. (2018) menciptakan OEMACS untuk mengurangi jumlah server aktif dalam *data center*. Wei dkk. (2019) mengembangkan AP-ACO (*Adaptive Parameter Ant Colony Optimization*), yang parameternya dapat beradaptasi, untuk meminimalkan konsumsi daya dan biaya komunikasi antar-VM.

Algoritma Genetika Algoritma genetika banyak digunakan dalam optimasi penempatan VM, termasuk dengan metode pengkodean yang terinspirasi dari masalah *bin packing*. Metode pengkodean standar merepresentasikan solusi sebagai *array* yang menunjukkan indeks kotak tempat setiap item diletakkan. Namun, Falkenauer (1992) mengusulkan *Grouping Genetic Algorithm* (GGA) untuk mengatasi kelemahan pengkodean standar dengan menambahkan daftar label yang diperbolehkan, sehingga *crossover* dan mutasi tetap mempertahankan struktur partisi.

Wu (2021) menerapkan GGA untuk menempatkan VM pada PM identik guna meminimalkan konsumsi energi. Xu Fortes (2010) mengombinasikan GGA dengan logika *fuzzy* untuk meminimalkan pemborosan sumber daya, konsumsi daya, dan

suhu tertinggi PM, dengan menggunakan *hash table* sebagai representasi kromosom. Liu dkk. (2014) mengadaptasi GGA dalam *Nondominated Sorting Genetic Algorithm* untuk mengoptimalkan jumlah PM aktif, lalu lintas jaringan, dan keseimbangan penggunaan sumber daya, meskipun tanpa mempertimbangkan topologi jaringan secara eksplisit. Sonklin Sonklin (2023) menerapkan GGA untuk penempatan VM berdasarkan tipe yang telah ditentukan penyedia layanan *cloud*.

Tang Pan (2014) mengasumsikan topologi jaringan *data center* berbasis hierarki tiga tingkat: *core*, *aggregation*, dan *edge*. Mereka mengklasifikasikan komunikasi antar-VM ke dalam empat kategori berdasarkan lokasi VM, dengan tujuan meminimalkan konsumsi energi jaringan dan PM. Meskipun menggunakan algoritma genetika standar, mereka menerapkan algoritma khusus untuk memperbaiki kromosom yang rusak dan prosedur optimasi lokal guna mengurangi jumlah PM aktif.

Metode Metaheuristik Lainnya Selain ACO dan algoritma genetika, algoritma evolusioner lainnya juga diterapkan. Balaji, Kiran, dan Kumar (2023) menggunakan *firefly algorithm* untuk meminimalkan konsumsi daya, sementara Ghetas (2021) menerapkan *monarch butterfly optimization* dalam MBO-VM untuk mengoptimalkan konsumsi daya dan pemborosan sumber daya. Tripathi, Pathak, dan Vidyarthi (2020) memodifikasi BDA (*Binary Dragonfly Algorithm*) menjadi VMPDA (*Virtual Machine Placement using Dragonfly Algorithm*) untuk mengurangi pemborosan sumber daya. Zhao, Zhou, dan Li (2019) mengembangkan GATA, algoritma hibrida berbasis algoritma genetika dan *tabu search*, untuk meminimalkan konsumsi daya dan meningkatkan *load balance*.

2.1.3 Metode *Machine Learning*

Metode *machine learning*, khususnya *reinforcement learning* (RL), juga banyak digunakan. Caviglione (2021) memanfaatkan *deep reinforcement learning* untuk meminimalkan konsumsi daya server, risiko gangguan perangkat keras, dan interferensi antar-VM. Ghasemi, Haghighat, dan Keshavarzi (2024) mengembangkan dua algoritma untuk menentukan penempatan VM yang bertujuan meminimalkan penggunaan energi, mengurangi pemborosan sumber daya, dan memaksimalkan *load balance*: VMPMFuzzyORL dan MRRL. VMPMFuzzyORL mengintegrasikan *reinforcement learning* (RL) dengan sistem *fuzzy*, sementara MRRL menggabungkan RL dengan algoritma *k-means*. Pada MRRL, algoritma *k-means* digunakan untuk membentuk kluster-kluster VM, sedangkan RL memetakan setiap kluster ke server

tertentu. Sebaliknya, pada VMPMFuzzyORL, RL langsung digunakan untuk memetakan masing-masing VM ke server tertentu, dengan *reward* dari setiap aksi ditentukan oleh sistem *fuzzy* yang mengevaluasi ketiga metrik performa tersebut. Qin dkk. (2020) memperkenalkan VMPMORL untuk meminimalkan konsumsi energi dan pemborosan sumber daya. Pada VMPMORL, MDP (*Markov Decision Process*) dimodifikasi menjadi MDP multi-objektif di mana *reward signal* dan \hat{Q} -value untuk setiap objektif direpresentasikan sebagai vektor *reward* dan vektor \hat{Q} -value. Jarak setiap vektor \hat{Q} -value terhadap titik utopia, titik dengan koordinat ke- i yang berupa nilai terbaik untuk fungsi objektif ke- i , dihitung menggunakan metrik Chebyshev. Aksi dengan vektor \hat{Q} -value yang memiliki jarak terkecil dari titik utopia dicari menggunakan algoritma ϵ -greedy.

2.2 Metode Optimasi Multiobjektif untuk Masalah Penempatan Mesin Virtual

Beberapa metode yang disebutkan sebelumnya, seperti VMPMORL (Qin dkk, 2020) dan algoritma buatan Caviglione (2021), merumuskan masalah penempatan mesin virtual sebagai optimasi multiobjektif, di mana solusi diperoleh dengan menyeimbangkan setiap objektif (metrik performa) secara bersamaan untuk menghasilkan sejumlah solusi Pareto. Akan tetapi, sebagian besar penelitian yang telah dibahas belum menerapkan pendekatan ini. Hal tersebut dapat ditunjukkan dari penyerdehanaan yang dilakukan oleh metode-metode ini menjadi objektif tunggal melalui skalarisasi fungsi. Misalnya, untuk n fungsi objektif f_1, f_2, \dots, f_n , fungsi yang dihasilkan adalah $f = a_1 f_1 + a_2 f_2 + \dots + a_n f_n$.

Agar dapat mengeksplorasi solusi Pareto secara efisien, algoritma MOEA (*Multi-Objective Evolutionary Algorithm*) sering digunakan dalam pendekatan ini. MOEA/D (*Multi-Objective Evolutionary Algorithm based on Decomposition*) digunakan untuk mengoptimalkan konsumsi daya server, pemborosan CPU, dan waktu propagasi (Gopu Venkataraman, 2019), sementara NSGA-III (*Non-dominated Sorting Genetic Algorithm*) digunakan untuk meminimalkan konsumsi daya, pemborosan sumber daya, dan *network transmission delay* (Gopu dkk., 2023). Ye, Yin, dan Lin mengembangkan EEKnEA (*Energy-Efficient Knee Point-driven Evolutionary Algorithm*) untuk meminimalkan konsumsi daya, memaksimalkan *load balance*, memaksimalkan rata-rata pemanfaatan sumber daya, dan memaksimalkan rata-rata "robustness" server.

Tao dkk. (2016) mengembangkan BGM-BLA (*Binary Graph Matching-*

Based Bucket Code Learning Algorithm) untuk mengubah penempatan VM sehingga jumlah server aktif, komunikasi antar-VM, dan biaya migrasi VM menjadi seminimal mungkin. Sesuai dengan namanya, algoritma ini mengombinasikan algoritma *bucket-code learning* dan *binary graph matching*. BGM-BLA dibagi dalam dua tahap: pembentukan grup-grup VM dan menentukan server yang cocok sebagai tempat baru masing-masing grup tersebut. Algoritma *bucket-code learning* digunakan untuk mencari beberapa kandidat solusi optimal, sedangkan *binary graph matching* digunakan untuk mengevaluasi dan membandingkan kandidat-kandidat tersebut berdasarkan ketiga objektif tersebut. Kemudian, solusi tersebut dieksplorasi melalui tahap *learning* dan mutasi.

2.3 Menyelesaikan Masalah Penempatan Mesin Virtual sekaligus Masalah Penentuan Rute Jaringan

Sebagian besar metode sebelumnya juga belum memanfaatkan topologi jaringan *data center* sebagai informasi penting dalam menentukan penempatan VM, terutama untuk VM yang berkomunikasi dengan VM lain. Bahkan, metode yang mengoptimalkan metrik kinerja jaringan sering kali hanya memodelkan *data center* sebagai sekumpulan server yang dapat saling berkomunikasi dengan *bandwidth* tetap. Untuk mengatasi kekurangan ini, beberapa metode dikembangkan untuk menentukan tidak hanya penempatan VM tetapi juga rute komunikasi antar-VM. Algoritma HACOS merupakan salah satu contoh metode tersebut (Alharabe, Rakrouki Aljohani, 2022). Akan tetapi, HACOS mengasumsikan lingkungan *cloud* yang statis. Oleh karena itu, sejumlah algoritma dirancang untuk lingkungan *cloud* yang dinamis, di mana *data center* melayani banyak *tenant* (pengguna) dengan kebutuhan sumber daya yang beragam. Setiap *tenant* dapat masuk dan keluar dari sistem pada waktu yang berbeda, sehingga algoritma-algoritma tersebut bersifat adaptif dan dijalankan secara berkala selama *data center* aktif.

Jiang dkk. (2012) menggagas sebuah algoritma heuristik yang menggunakan teknik aproksimasi rantai Markov untuk menentukan penempatan VM serta memilih *link* komunikasi yang dapat mengurangi jumlah server aktif dan rata-rata tingkat penggunaan *link*. Tidak seperti algoritma sebelumnya di mana lingkungan *cloud* bersifat statis, algoritma ini dirancang untuk lingkungan *cloud* di mana jumlah *tenant* pada waktu tertentu dimodelkan menggunakan antrean $M/M/\infty$ dan algoritma ini dijalankan setiap kali ada *tenant* yang masuk atau keluar dari sistem. Fang dkk.

(2013) mengembangkan pendekatan heuristik untuk menentukan penempatan VM dan rute komunikasi antar-VM untuk meminimalkan konsumsi daya server, biaya migrasi VM, dan *delay* komunikasi dalam jaringan. Algoritma ini dirancang khusus untuk *data center* berbasis OpenFlow dengan topologi *fat tree*. Sementara itu, Luo dkk. (2014) mengusulkan algoritma yang meminimalkan biaya komunikasi jaringan dengan memanfaatkan *minimum tree level* antar-VM berdasarkan topologi jaringan dan penempatan VM di *data center*. Algoritma ini terdiri dari dua tahap: pertama, mengevaluasi apakah total *traffic* pada setiap *switch* melebihi ambang batas yang ditentukan; kedua, memigrasikan VM ke server lain jika ambang batas terlampaui dan menentukan ulang rute komunikasi antar-VM. Algoritma ini dijalankan secara berkala pada interval waktu tertentu.

Tabel 2.1: My caption

Nama	Kegiatan	Algoritma	Perbedaan dengan peneliti
Yusuf	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

BAB III

DASAR TEORI

3.1 Representational State Transfer

Komputasi awan atau *cloud computing* merupakan teknologi yang mampu menyajikan sumber daya teknologi informasi sebagai layanan web yang dapat diakses melalui Internet. Menurut *National Institute of Standards and Technology* (NIST), terdapat lima karakteristik utama yang menjadi pembeda *cloud computing* dari model komputasi lainnya.

1. ***On-Demand Self-Service (Layanan Mandiri Sesuai Permintaan)*** : Pengguna dapat mengakses dan mengatur sumber daya komputasi sesuai kebutuhan tanpa perlu interaksi dengan penyedia layanan *cloud*.
2. ***textitBroad Network Access (Akses Jaringan Luas)*** Layanan cloud dapat diakses dari mana saja melalui Internet.
3. ***textitResource Pooling (Penggabungan Sumber Daya)*** Sumber daya digabungkan dalam satu infrastruktur dan dibagikan kepada banyak pengguna melalui model multi-tenant, di mana sumber daya fisik maupun virtual secara dinamis diberikan sesuai dengan permintaan pengguna.
4. ***textitRapid Elasticity (Elastisitas Cepat)*** Kapabilitas komputasi dapat dengan cepat ditingkatkan atau dikurangi sesuai dengan kebutuhan pengguna.
5. ***textitMeasured Service (Layanan Terukur)*** Penggunaan sumber daya dipantau, diukur, dan ditagihkan sesuai dengan konsumsi sebenarnya, seperti model pay-as-you-go.

Kelima karakteristik tersebut memungkinkan cloud computing memberikan efisiensi, fleksibilitas, dan skalabilitas tinggi bagi pengguna.

Di dalam lingkungan *cloud*, satu atau lebih VM dengan sistem operasi, aplikasi berjalan, dan spesifikasi yang beragam seperti kebutuhan daya komputasi, memori, kapasitas penyimpanan, dan *bandwidth* jaringan minimal dapat berbagi server yang sama. Hal ini dimungkinkan berkat kemampuan penyedia layanan *cloud* dalam memvirtualisasikan sumber daya bagi dan membagikan sumber daya kepada berbagai VM. Kemampuan ini memungkinkan penggunaan server yang lebih sedikit dan lebih efektif.

3.2 JavaScript Object Notation

3.2.1 Definisi

3.2.2 Contoh

BAB IV

ANALISIS DAN PERANCANGAN SISTEM

4.1 Deskripsi Umum Sistem

4.2 Analisis Kebutuhan Sistem

4.3 Pembuatan Sistem

4.3.1 Pembuatan Sistem Pengenalan Entitas Bernama

4.3.2 Pembuatan Sistem Ekstraksi Kalimat Pernyataan

4.4 Rancangan Antarmuka

4.4.1 Deskripsi

4.4.2 Wireframe

Lorem ipsum odor amet, consectetur adipiscing elit. Cursus viverra fames inceptos neque imperdiet nostra duis. Dignissim arcu at tempor mattis curae sed nascetur aliquet luctus. Netus arcu venenatis semper suscipit consequat. Phasellus congue sodales blandit ultricies donec dignissim. Dapibus at odio penatibus mauris adipiscing fusce sodales. Quisque nullam massa ullamcorper curae neque vehicula ultricies. Primis bibendum etiam velit viverra arcu etiam sed malesuada ut.

Vulputate ad malesuada elementum et mollis parturient sodales. Netus lectus vitae sit risus netus ipsum congue diam. Faucibus nascetur malesuada risus luctus ridiculus. Suspendisse nec ridiculus accumsan justo parturient metus iaculis. Montes nulla ultricies fringilla nascetur nisi dignissim massa lectus sagittis. Mi tellus orci nullam etiam scelerisque pretium inceptos id feugiat. Lacus luctus natoque placerat cursus faucibus. Luctus porta eget orci nullam magna nostra viverra eget.

Aptent accumsan ac torquent nibh magna tincidunt facilisis facilisi. Libero quis dignissim rhoncus aptent sapien faucibus nostra. Hendrerit volutpat faucibus diam sollicitudin aliquet diam lacus. Hac sed est dictum felis lacus congue at potenti. Metus sollicitudin varius suspendisse consequat scelerisque curae. Luctus porttitor cursus vel neque ipsum egestas. At orci sagittis pulvinar curabitur; ipsum adipiscing nullam diam. Pulvinar euismod interdum aliquam commodo augue aliquam erat.

Facilisi dictum imperdiet elit arcu erat dignissim neque. Hac tristique potenti; curabitur fusce aenean leo.

Diam euismod facilisis libero in sem. Ad et justo morbi vel justo primis ipsum cras et? Fermentum lacinia faucibus tristique pharetra fringilla ad. Eu ut integer consequat odio molestie. Nisl lectus ornare erat primis amet laoreet ultricies ligula consequat. Nibh tristique integer iaculis eget phasellus est magna. Fames risus rhoncus turpis sem ad netus massa efficitur.

Fames litora imperdiet accumsan nascetur nam arcu cursus. Odio vel sed platea tempor aptent senectus, consectetur conubia. Leo aenean vitae ultrices quis proin sit. Litora dictum torquent interdum morbi velit adipiscing. Nostra pharetra facilisi iaculis bibendum taciti quisque erat. Justo phasellus sed massa convallis turpis magnis facilisis. Dignissim libero sapien phasellus hendrerit ultricies. Adipiscing faucibus sodales justo hendrerit sagittis imperdiet felis maximus.

BAB V

IMPLEMENTASI SISTEM

- 5.1 Spesifikasi**
- 5.2 Implementasi Sistem Pengenalan Entitas Bernama**
- 5.3 Implementasi Sistem Ekstraksi Kalimat Pernyataan**

BAB VI

PENGUJIAN DAN PEMBAHASAN SISTEM

- 6.1 Pengujian Sistem Pengenalan Entitas Bernama**
- 6.2 Pengujian Sistem Ekstraksi Kalimat Pernyataan**

BAB VII

PENUTUP

7.1 Kesimpulan

7.2 Saran

DAFTAR PUSTAKA

Crockford, Douglas., 2006, *The application/json media type for javascript object notation (json)*.

LAMPIRAN A
BERKAS JSON UNTUK MODEL SISTEM PENGENALAN
ENTITAS BERNAMA