BAB III

METODOLOGI

3.1. Pendahuluan

Pada bagian bab 3 ini akan menjelasakan tentang metodologi yang digunakan dalam Tugas Akhir yang berjudul Monitoring *Load Processor* dan Jaringan Pada *Cloud Computing* dengan *Fuzzy Logic*. Dan akan berfokus mengenai tahapan untuk merancang sistem monitoring secara sistematis yang meliputi beberapa tahapan dan direpresentasikan pada suatu kerangka kerja.

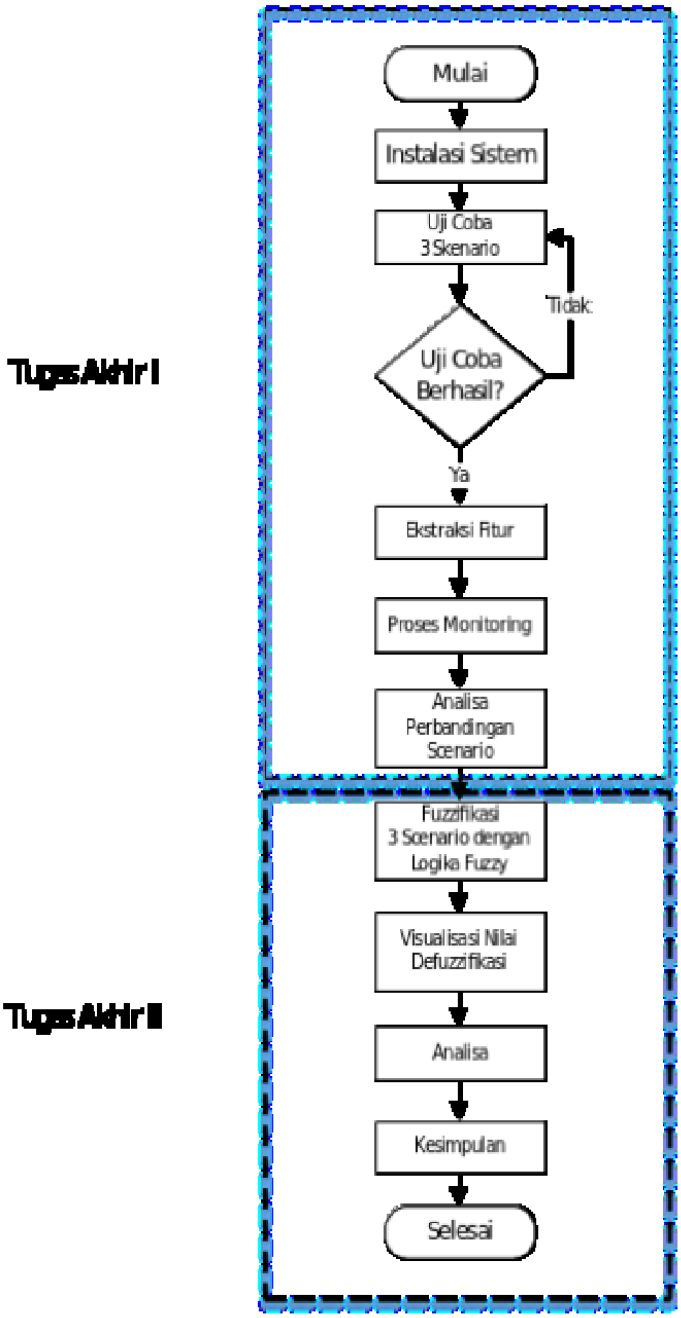
3.2. Kerangka Penelitian

Tahapan pertama dalam kerangka kerja ini adalah menghitung bebarapa besar energi atau daya yang digunakan *Private Cloud Computing* untuk mengetahui kecepataan dan daya yang digunakan pada *Cloud Computing* dengan menggunakan metode *Fuzzy Logic*. Sehingga tahapan kerja dilakukan secara terstruktur dan mengikuti alur yang telah di tentukan.

Tahapan selanjutnya adalah menentukan nilai variabel dengan menggunakan metode yang telah digunakan yaitu *fuzzy logic* sebagai unit interferensinya. Sehigga setelah penulis melakukan tahapan tersebut maka penulis dapat dengan mudah merancang software monitoring secara interaface sehingga akan menghasilkan variabel yaitu *Input/Oput* yang juga akan diterapakan dengan memberikan notifikasi secara otomtis. Sehingga akan didapatkan nilai yang tepat sehingga dapat dilakukan untuk mencari kronologi menotoring untuk mendapatkan skenario hasil pengujian sistem monitoring yang berupa nilai masing otput/input. Berikut merupakan kerangka peneletian pada tugas akhir ini.

Langkah awal dalam melakukan proses monitoring adalah dengan melakukan monitoring 2 komputer *virtual client*. Sehingga komputer

client tersebut akan menghasilkan data yang mana data tersebut akan dilakukan perhitungan secara keseluruhan pada saat *client* akan mengakses *server*. Sehingga dari data yang didapatkan dapat dilakukan validasi yang didasarkan dengan penggunaan CPU dari *server cloud computing* yang digunakan.



Gambar 3.1 Tahapan Keranngka Kerja

3.3. Kebutuhan Perangkat Lunak

Dalam proses penelitian ini dalam melakukan perancangan systemnya maka yang diperlukan yaitu kebutuhan perangkat keras dan perangkat lunak. Adapun penjabaran dari masing-masing kebutuhan perangkat adalah sebagai berikut:

3.3.1. Perangakat Keras

Perangkat keras yang digunakan dalam penelitian adalah 1 PC sebagai *server*, 1 PC sebagai penyerang, 1 *router* untuk membangun jaringan *cloud*. Dengan spec user Spesifikasi kebutuhan perangkat keras sebagai berikut:

Tabel 1 Perangkat Keras Penelitian

Sistem	***Tools***	**Keterangan**
Server	PC	Ubuntu 16.04
Penyerang	PC	Kali Linux v.4.13
Laptop	15 85th Generation	RAM 8GB Nvidia Mx150
Jaringan	Router	Huawei HG8245A

3.3.2. Perangkat Lunak

Perangkat lunak yang digunakan dalam penelitian merupakan *tools* owncloud, wireshark dan pyhton. Spesifikasi kebutuhan perangkat lunak dijelaskan pada tabel dibawah:

Tabel 2 Perangkat Lunak Penelitian

Sistem	**Tools**	**Keterangan**
Netdata	Linux	Speedtest
Cloud	Owncloud	Owncloud X

Capture	Wireshark	Wireshark 2.44
DSTAT	Linux	Dstat on Linux Mint

3.4. Parameter yang digunakan

Dalam penelitian tugas akhir ini tujuannya adalah menganalisa sebeberapa besar penggunaan daya dan konsumsi energy pada Utilitas CPU di *Private cloud Computing* pada saat keadaan sibuk. Seperti yang sudah dijelaskan sebelumnya pengujian di dalam scenario dilakukan dengan mengakses server *Cloud Computing* yang telah dibangun. Adapun parameter yang digunakan adalah sebagai berikut:

3.4.1. Penggunaan Daya

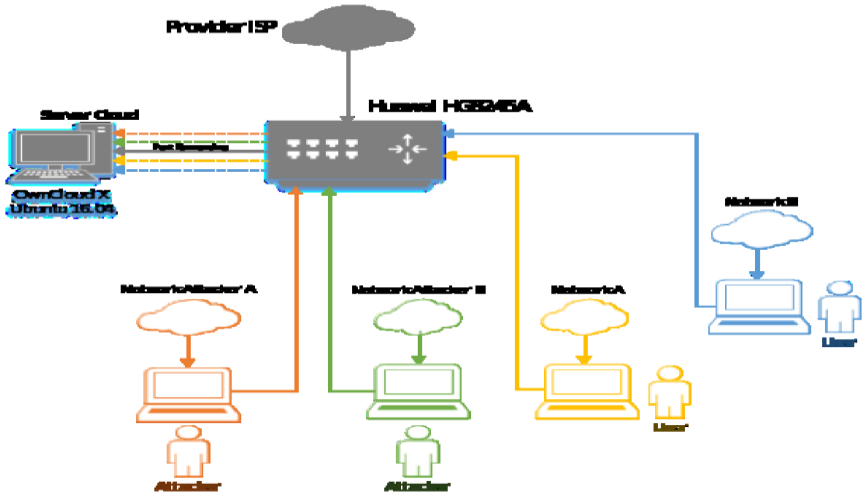
Penggunaan daya sebenarnya sangat dipengaruih dengan adanya sumber utama yang berupa energy yang biasanya disebut dengan daya supply. Yang mana daya yang di gunakan dapat dilihat pada saat server dijalankan dengan melihat perkembanganya dari utilitas CPU.

3.4.2. Konsumsi Energy

Konsumsi energy dari cloud computing adalah dapat didefinisikan sebagai total energy yang dipakai ketika server digunakan sehingga nilai dari konsumsi daya pada saat idle dan maksimum pada utilitas CPU.

3.4.3. Perancangan Topologi

Pada penelitian tugas akhir, akan dirancang topologi untuk membangun *cloud.* Topologi menggunakan 1 *router* kemudian di-*port forwarding* ke IP *server cloud* sehingga *cloud* dapat diakses dari luar jaringan (bersifat publik). Berikut topologi penelitian tugas akhir.



Gambar 3.2 Topologi Arsitektur Penelitian

3.4.4. OwnCloud sebagai Cloud Computing

OwnCloud adalah perangkat lunak yang digunakan membuat layanan file hosting. Aplikasi ini mirip dengan Dropbox, Google Drive, dan penyimpanan lainnya. Cocok untuk mencoba membuat server file sendiri dan mengkonfigurasinya sesuai dengan kebutuhan. OwnCloud menyediakan layanan pengelolaan file yang baik, mengambil dan mengakses file dengan lancar, serta terhubung langsung dengan teknologi berbasis web.

Gambar 3.3 OwnCloud X sebagai Cloud

3.4.5. DSTAT

Dstat adalah pengganti serbaguna untuk vmstat, iostat, netstat dan ifstat. Dstat mengatasi beberapa keterbatasan dan menambahkan beberapa fitur tambahan, lebih banyak counter dan fleksibilitas. Dstat berguna untuk memonitor sistem selama pengujian, benchmark, atau pemecahan masalah kinerja. Dstat memungkinkan untuk melihat semua sumber daya sistem secara real-time, bisa dengan mudah dalam pemantauan.

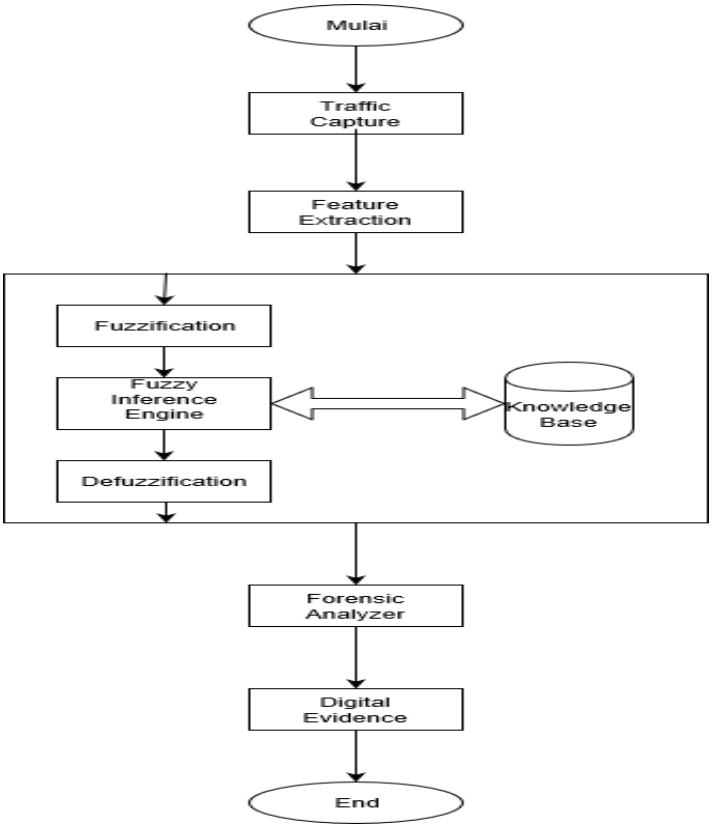
Dstat memberi informasi selektif terperinci dalam kolom dan dengan jelas menunjukkan berapa besar dan satuan output yang ditampilkan. Lebih sedikit ketidakpastian, lebih sedikit kesalahan. Dan yang paling penting, sangat mudah untuk menulis plugin untuk mengumpulkan counter sendiri. Output Dstat secara default dirancang untuk ditafsirkan secara real-time, namun dapat mengekspor detail ke output CSV ke file yang akan diimpor kemudian ke Gnumeric atau Excel untuk menghasilkan grafik.

3.5 Feature Extraction

Setelah pengujian serangan *brute force* yang direkam menggunakan wireshark dalam jenis file *packet capture* (pcap), penelitian dilanjutkan dengan melakukan *feature extraction*. *Feature extraction* berfungsi untuk membuat file *Comma Separated Value (CSV)* dari *raw* data hasil *capture* wireshark (pcap) yang bertujuan untuk mengenali pola serangan dari dataset yang telah dilakukan. *Feature extraction* dilakukan menggunakan bahasa pemrograman python.

3.6 Perancangan Logika Fuzzy untuk Forensik Jaringan

Logika fuzzy untuk forensik jaringan dibagi menjadi tujuh tahapan yaitu : *the traffic capture, the feature extraction, the fuzzification, the fuzzy inference and knowledge base, the defuzzification, forensic analyzer,* dan *digital evidence*.



Gambar 3.5 *Flowchart* Perancangan Logika Fuzzy

3.6.1 *Traffic Capture*

Pada tahap ini digunakan wireshark untuk menangkap data trafik jaringan dari skenario pengujian.

3.6.2 Defuzzyfikasi

Defuzzyfikasi merupakan langkah proses terakhir dalam system logika fuzzy yang merupakan proses pemetaan besaran dari himpunana fuzzy kedalam bentuk nilai crisp. Metode yang digunakan adalah metode keanggotaan maximum. Defuzzyfikasi juga dapat digunakan sebagai proses pengubahan data-data fuzzy menjadi data-data yang dapat dikirimkan ke peralatan pengendali. Defuzzifikasi didalam prosesnya terdapat fungsi grafik keanggotaan yang difungsikan untuk menentukan batasan output fuzzy yang diinginkan. Defuzzifikasi bekerja berdasarkan pengetahuan yang disebut dengan sistem knowledge base. Dalam sistem knowledge base terbagi menjadi dua yaitu data base dan rule base.

Didalam database berisi definisi-definisi penting yang membahas mengenai parameter fuzzy seperti himpunan fuzzy yang berfungsi sebagai keanggotaan yang sudah didefinisikan sebagai variable linguistik. Input dari proses defuzzifikasi berupa himpunan fuzzy yang didapatkan dari komposisi aturan-aturan fuzzy sedangkan outputnya dihasilkan dari bilangan domain himpunan fuzzy. Sehingga jika diberikan himpunan fuzzy dalam range maka terdapat nilai crisp tertentu yang dapat dijadikan sebagai output.

$$\text{Def} = \sum \frac{[\mu(Normal).\widehat{normal} + \mu(Serangan).\widehat{Serangan}]}{[\mu(normal) + \mu(serangan)]} \ldots\ldots\ldots(7)$$

Keterangan:

Def : nilai akhir defuzifikasi

μ (x) : nilai derajat keanggotaan *output*

$\widehat{x}$: nilai singleton variabel *output*.

3.6.3. Fuzzy Inference Engine and knowledge base

Merupakan fungsi dari keanggotaan yang menghasilkan data variable berupa input yang telah dibentuk. Logika fuzzy digunakan untuk forensik jaringan yang merupakan aturan untuk kegunaan IF-THEN pada output yang akan dihasilkan yaitu berupa trafik normal dan trafik serangan. Dalam fuzy inference engine merupakan type rules yang mengkonversi fuzzy input ke fuzzy output sedangakan fuzy output dari inference engine dari ineference engine menjadi crisp menggunakan fungsi keanggotaan, *reverse* dari fuzzifier. Knowledge base mempunyai fungsi penting dalam pengendalian dengan logika fuzzy karena semua proses sudah difuzzifikasi, inferensi dan defuzzifikasi bekerja berdsarkan pengetahuan yang ada pada knowledge base. Knowledge base terbagi menjadi dua yaitu database dan rulebase. Data base berisi definisi-definisi penting mengenai parameter fuzzy seperti himpunan fuzzy dengan fungsi keanggotaan yang telah di definisikan untuk setiap

variablenya pada linguistik yang telah ada. Berikut merupakan table variable lingustik (*rules based*) pada processor yang akan difuzzykan.

Tabel 4 Variable Lingustik (*Rules Based*) pada *Fuzzy Logic*

No	RAM	CPU	HARDISK	PERFORMA
1	Ringan	Ringan	Ringan	Ringan
2	Ringan	Normal	Ringan	Ringan
3	Ringan	Berat	Ringan	Ringan
4	Ringan	Ringan	Normal	Ringan
5	Ringan	Normal	Normal	Normal
6	Ringan	Berat	Normal	Normal
7	Ringan	Ringan	Berat	Ringan
8	Ringan	Normal	Berat	Normal
9	Ringan	Berat	Berat	Berat
10	Normal	Ringan	Ringan	Ringan
11	Normal	Normal	Ringan	Normal
12	Normal	Berat	Ringan	Normal
13	Normal	Ringan	Normal	Normal
14	Normal	Normal	Normal	Normal
15	Normal	Berat	Normal	Normal
16	Normal	Ringan	Berat	Normal
17	Normal	Normal	Berat	Normal
18	Normal	Berat	Berat	Berat
19	Berat	Ringan	Ringan	Ringan
20	Berat	Normal	Ringan	Normal
21	Berat	Berat	Ringan	Berat
22	Berat	Ringan	Normal	Normal
23	Berat	Normal	Normal	Normal
24	Berat	Berat	Normal	Berat
25	Berat	Ringan	Berat	Berat
26	Berat	Normal	Berat	Berat

27	Berat	Berat	Berat	Berat

$$\mu_{sf}[x] = \min(\mu kf1[x], \mu kf2[x], \mu kf3[x], \mu kf4[x], \mu kf5[x], ...) \quad \text{……(8)}$$

Keterangan:

$\mu_{sf}[x]$: nilai keanggotaan solusi fuzzy sampai aturan ke-i;

$\mu_{kf}\, i[x]$: nilai keanggotaan konsekuen fuzzy setiap aturan ke-i, i = 1,2,3,.....

$$\mu_{sf}[x] = \max(\mu kf1[x], \mu kf2[x], \mu kf3[x], \mu kf4[x], \mu kf5[x], ...) \quad \text{……...(9)}$$

Keterangan:

$\mu_{sf}[x]$: nilai keanggotaan solusi fuzzy sampai aturan ke-i;

$\mu_{kf}\, i[x]$: nilai keanggotaan konsekuen fuzzy setiap aturan ke-i, i = 1,2,3,.....

3.6.4. Fuzzifikasi

Proses untuk mengubah variable non fuzzy atau biasanya dikenal dengan variable numerik menjadi variable fuzzy yaitu variable linguistik. Input dalam proses defuzzifikasi adalah himpunan fuzzy yang didapatkan dari komposisi aturan-aturan fuzzy. Sedangkan output yang dihasilkan adalah bilangan domain dari himpunan fuzzy tersebut. Jika diberikan satu himpunan fuzzy dalam range tersebut maka dapat diambil nilai crispnya sebagai keluaran. Berikut table yang menunjukkan rentang nilai Performa Fuzzy pada Processor

- Performa RAM

Tabel 5 Performa RAM

Variabel Linguistik	Rentang Nilai
Ringan	0 - 600 MB
Normal	0 -1200 MB
Berat	600 – 200 MB

- Performa CPU

Tabel 6 Performa CPU

Variabel Linguistik	Rentang Nilai
Ringan	0 – 30%
Normal	30 – 60%
Berat	60 – 100%

- Performa Hardisk

Tabel 7 Performa Hardisk

Variabel Linguistik	Rentang Nilai
Ringan	0 – 2500 MB/s
Normal	2500 – 5000 MB/s
Berat	5000 – 8000 MB/s

3.7 Parameter Pengujian QoS Berdasarkan Standarisasi ITU-T G.1010

Berikut tabel referensi standarisasi QoS ITU-T G.1010 sebagai berikut:

Standarisasi QoS ITU-T G.1010

Tabel 8 Standarisasi QoS

Parameter QoS	Data
Throughput	NA

Delay	Preffered < 15 s ; Acceptable < 60 s Nb : Amount of Data 10KB – 10 MB
Packet Loss	0 %

❖ Throughput

Throughput akan digunakan untuk mengukur jumlah paket atau bit yang berhasil melewati jaringan. Throughput juga termasuk bandwidth aktual yang dapat diukur dengan waktu sehingan kecepatannya dapat di transfer dan diukur dalam bit per second (bps). Sehingga dapat melihat jumlah total paket yang masuk dan yang berhasil diamati oleh *destination* selama interval waktu tertentu. Berikut adalah persamaan dari *throughput* :

$$\textit{Throughput} = \frac{\text{ukuran data yang diterima}}{\text{Waktu pengiriman data}}$$

❖ Delay

Merupakan waktu tunda paket yang digunakan untuk proses transmisi dari node sumber ke node yang dituju dan persamaan delay adalah sebagai berikut:

$$\text{Rata-rata delay} = \frac{\text{total waktu pengiriman paket}}{\text{Total paket}}$$

❖ Packet Loss

Packet loss ditugaskan untuk menunjukan total paket yang hilang sehingga dapat terjadi kesalahan pada jaringan nirkabel seperti link terputus atau buffer yang tidak cukup diakibatkan kemacetan jaringan saat link menjadi kelebihan beban. Yang harus diperhatikan adalah rasio packet loss harus di minimum agar QoS tetap terjaga dengan baik. ITU (International Telecommunication Union) menurut standarnya nilai packet loss harus dijaga pada tingkat minimum. Adapun persamaan untuk menghitung nilai packet loss sebagai berikut:

$$\text{Packet Loss} = \frac{\sum \text{packet delivered} - \text{packet received}}{\sum \text{packet delivered}} \times 100\%$$

Keterangan :

∑ Packet delivered : Jumlah packet yang dikirim

∑ packet received : jumlah paket yang diterima

3.7 Skenario Pengujian

Pada pengujian penelitian ini akan menggunakan beberapa skenario serangan maupun akses dari user menggunakan beberapa sistem operasi. Berikut adalah scenario pengujiannya:

Tabel 9 Skenario Pengujian

No	Skenario	Keterangan
1	Ketika cloud diserang pada *brute force* menggunakan Kali Linux v.4.13	Dengan wordlist 1000 dalam durasi 10 menit
2	Ketika cloud diserang DoS	Durasi 10 menit
3	Ketika user mengupload file	1 GB – 10 GB
4	Memonitoring CPU	Secara real time