BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Tabel dibawah akan menjelaskan tinjauan pustaka yang digunakan pada penelitian ini untuk mengetahui perbedaan antara tinjauan pustaka dengan aplikasi yang akan dibuat :

Tabel 2.1 Perbandingan Tinjauan Pustaka

No.	Penulis	Obyek Penelitian	Teknologi	Hasil
1.	Muhammad Syaiful Adnan (2017)	Analisis Kinerja *Web Server* Dengan Metode *Load Balancing* Pada HAProxy	HAProxy, Apache Benchmark, *Round robin*	*Multi Node* lebih baik dibandingkan *Single Node*
2.	Syaqia Azizah, Asmunin (2017)	Implementasi *Load Balancing Web Server* Menggunakan HAProxy	HAProxy, Apache Benchmark, *Round Robin*	Performa *Web Server* untuk *request* dan *throughput* meningkat menggunakan HAProxy
3.	Dany Rahmana, Rakhmadhany Primananda, Widhi Yahya (2018)	Analisis *Load Balancing* Pada *Web Server* Menggunakan *Algoritme Weighted Least Connection*	IPVSADM, httperf, Weighted Least Connection	Server menerima bobot yang berbeda sesuai kemampuannya dan *CPU Usage* menurun dibandingkan *least connection*
4.	Gurasis Singh, Kamalpreet Kaur (2018)	*An Improved Weighted Least Connection Scheduling Algorithm for Load Balancing in Web Cluster Systems*	HAProxy, *Improved Weighted Least Connection*, Nginx *web server*	*Improved Weighted Least Connection* merupakan pengembangan dari *Weighted Least Connection* dan kinerjanya lebih baik dalam menangani *overload* server

5.	Fajar Zuhroni, Adian Fatchur Rochim, Eko Didik Widianto (2015)	*Analisis Performansi Layanan Kluster Server Menggunakan Penyeimbang Beban Dan Virtualbox*	HAProxy, httperf, Pound	HAProxy lebih unggul dibandingkan Pound, *response time* 70% lebih cepat dan *throughput* 20% lebih besar dan akses halaman per menit 60% lebih banyak.
6.	Yang diusulkan : Herbowo Prasetyo (2019)	Implementasi *Load Balancing* Pada *Web Server* Menggunakan *Algoritma Improved Weighted Least Connection*	HAProxy, *Improved Weighted Least Connection*, Apache Benchmark, LAMP Stack	*Multi Node* lebih baik dibanginkan *Single Node* dan tanpa load balancing. Kinerja CPU dan Memory lebih ringan juga *response time* lebih cepat, *throughput* lebi besar dan request per detik lebih banyak dibandingkan *Single Node* dan tanpa load balancing.

Penelitian yang dilakukan Muhammad Syaiful Adnan (2017) melakukan penelitian tentang kinerja *Web Server Single Node* dan *Multi Node* dengan metode *load balancing* pada HAProxy menggunakan sistem penjadwalan *Round Robin*. Hasil dari penelitian tersebut adalah kinerja *Multi Node* lebih baik dibandingkan *Single Node* untuk parameter yang diuji yaitu *throughput, request, response time, processor* dan *memory*.

Penelitian yang dilakukan Syaqia Azizah dan Asmunin (2017) melakukan penelitian tentang *Load Balancing Web Server* menggunakan HAProxy. Yang diuji adalah server *Single Server* dan HAProxy. Menggunakan *stress tool* Apache Benchmark, mensimulasi server dengan jumlah beban 250 *user*, 500 *user*, 750 *user*, 1000 *user* dan 1250 *user* menggunakan web statis. Hasil yang didapat yaitu menggunakan HAProxy untuk *Web Server* dapat meningkatkan performa *request* dan *throughput* dari web server dibandingkan *Single Server*.

Penelitian yang dilakukan Dany Rahmana, Rakhmadhany Primananda, dan Widhi Yahya (2018) melakukan penelitian tentang *Load Balancing* Pada *Web Server* menggunakan *Algoritme Weighted Least Connection*. Algoritma *Weighted Least Connection* membagi beban server berdasarkan *weight (*bobot). *Load Balancing* menggunakan *tool* IPVSADM yang terdapat *interface* Piranha dan pengujian menggunakan *httperf*. Pada pengujian nilai *throughput* akan meningkat jika diberi koneksi yang banyak. Berbanding terbalik dengan nilai *response time*, jika diberikan koneksi tinggi maka kinerjanya akan menurun. Server dengan bobot tertinggi mampu menangani 4000 *request* dan yang terendah mampu menangani 1000 *request*. Hasilnya untuk *CPU Usage*, menggunakan *algoritme Weighted Least Connection* lebih baik dari *least connection* jika diterapkan pada *server heterogen*.

Penelitian yang dilakukan Gurasis Singh dan Kamalpreet Kaur (2018) melakukan penelitian tentang *An Improved Weighted Least Connection Scheduling Algorithm for Load Balancing in Web Cluster Systems*. Pengujian menggunakan Nginx *Web Server* dan HAProxy untuk membagi beban menggunakan algoritma *Improved Weighted Least Connection* dimana algoritma ini pengembangan dari *least connection* dengan menentukan *request* pada web server yang koneksinya lebih sedikit dan *weight* yang telah ditentukan pada masing – masing server. Cara ini dilakukan untuk menghindari *overload* pada web server. Hasilnya menggunakan algoritma *Improved Weighted Least Connection* lebih baik dibandingkan algoritma *least connection*.

Penelitian yang dilakukan Fajar Zuhroni, Adian Fatchur Rochim, dan Eko Didik Widianto (2015) tentang *Analisis Performansi Layanan Kluster Server*

Menggunakan Penyeimbang Beban Dan Virtualbox. Pengujian membandingkan performa *load balancer* HAProxy dan Pound dimana parameternya *response time*, *throughput* dan *request*. Hasil yang didapat yaitu HAProxy lebih unggul dibandingkan Pound, dilihat dari parameter *response time* 70% lebih cepat dan *throughput* 20% lebih besar dan akses halaman per menit 60% lebih banyak.

Penelitian yang diusulkan oleh Herbowo Prasetyo (2019) memiliki perbedaan yaitu berfokus pada implementasi dari *load balancing* menggunakan HAProxy sebagai pembagi beban, *Apache Benchmark* sebagai *stress tool* dan algoritma *Weighted Least Connection* untuk *Single Node* dan *Multi Node* juga tanpa Load Balancing. Membandingkan performa dari setiap skema, juga dapat dilihat hasilnya skema terbaik dari ketiganya.

2.2 Dasar Teori

2.2.1 Load Balancing

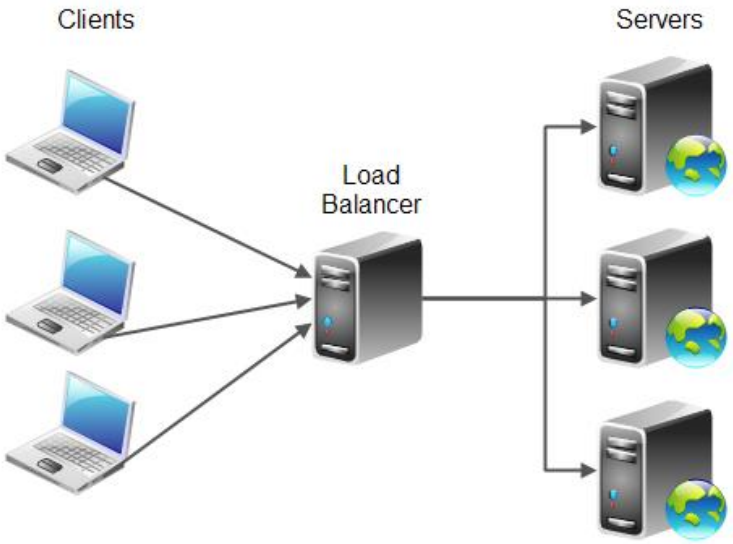
Load balancing adalah teknik membagi jumlah pekerjaan yang harus dilakukan komputer antara dua atau lebih komputer sehingga setiap komputer melakukan pekerjaan dalam jumlah dan waktu yang sama serta secara umum semua pengguna mendapatkan akses cepat. *Load balancing* bertujuan untuk mengoptimalkan sumber daya, memaksimalkan *throughput*, meminimalkan waktu respon dan menghindari kelebihan beban dari sumber daya tunggal (Lakhe, Shinde, Sukhthankar, & Reddy, 2016)

Load balancer adalah perangkat lunak yang berguna untuk membagi beban agar menjadi seimbang dalam melayani suatu request dari para user. Pembagian beban permintaan atau akses dari *user* akan merata ke semua server yang ada, kinerja server juga menjadi lebih baik.

Load Balancer menggunakan beberapa peralatan yang sama untuk menjalankan tugas yang sama. Hal ini memungkinkan pekerjaan dilakukan dengan lebih cepat dibandingkan apabila dikerjakan oleh hanya 1 peralatan saja dan dapat meringankan beban kerja peralatan, serta mempercepat waktu respons.

Load Balancer bertindak sebagai penengah diantara layanan utama dan pengguna, dimana layanan utama merupakan sekumpulan server/mesin yang siap melayani banyak pengguna. Disaat *Load Balancer* menerima permintaan layanan dari *user*, maka permintaan tersebut akan diteruskan ke server utama. *Load Balancer* dengan pintar dapat menentukan server mana yang memiliki *load* yang lebih rendah dan respons yang lebih cepat. Bahkan bisa menghentikan akses ke

server yang sedang mengalami masalah dan hanya meneruskannya ke server yang dapat memberikan layanan. Hal ini salah satu kelebihan yang umumnya dimiliki *load balancer*, sehingga layanan seolah olah tidak ada gangguan di mata pengguna.



Gambar 2.1 Load Balancing

2.2.2 Web Server

Web server adalah sebuah komputer yang mana kandungan web disimpan. Pada dasarnya *web server* yang digunakan untuk meng-host situs web tetapi ada server web lain juga seperti game, penyimpanan, FTP, e-mail dll. Situs web adalah koleksi halaman web whileweb server adalah sebuah *software* yang menanggapi permintaan sumber daya web (Tutorials Point Originated, 2017).

Web server adalah perangkat lunak yang berfungsi sebagai penerima permintaan yang dikirimkan melalui *browser* kemudian memberikan tanggapan permintaan dalam bentuk halaman situs web atau lebih umumnya dalam dokumen HTML.

Fungsi utama *Web server* adalah untuk melakukan atau akan mentransfer berkas permintaan pengguna melalui protokol komunikasi yang telah ditentukan sedemikian rupa. halaman web yang diminta terdiri dari berkas teks, video, gambar, file dan banyak lagi. pemanfaatan web server berfungsi untuk mentransfer seluruh aspek pemberkasan dalam sebuah halaman web termasuk yang di dalam berupa teks, video, gambar dan banyak lagi.

2.2.3 Improved Weighted Least Connection

Improved Weighted Least Connection merupakan pengembangan dari *Least Connection* dimana perbedaannya terletak pada penentuan berat pada masing – masing server dan mengarahkan *request* ke server yang paling sedikit mendapatkan koneksi.

Menggunakan algoritma *Improved Weighted Least Connection*, memungkinkan server baru tidak akan *overload* pada saat *request* sedang tinggi dan performa server menjadi lebih optimal. Pada file haproxy.cfg, atur pada bagian backend untuk balance ditulis *leastconn* dan menambahkan *weight* pada list server seperti dibawah.

```
backend web_server
        mode tcp
        mode http
        stats enable
        stats hide-version
        stats refresh 20s
        stats show-node
        stats uri /haproxy?stats
        balance leastconn
        option allbackups
        option forwardfor
        option httpclose
        server web1.local 192.168.43.101:80 weight 50 check
```

```
        server  web2.local  192.168.43.102:80  weight  100
check backup
```

2.2.4 Apache Benchmark

ApacheBench (ab) adalah program komputer *single-threaded command line* untuk mengukur kinerja server web HTTP. Awalnya dirancang untuk menguji Server HTTP Apache, cukup umum untuk menguji server web apa pun.

Tools ab dilengkapi dengan sumber distribusi dari standar Apache, dan seperti server web Apache itu sendiri, tools Apache Benchmark bersifat gratis, perangkat lunak *opensource* dan didistribusikan di bawah ketentuan Lisensi Apache.

Menggunakan Apache Benchmark dapat mengetahui seperti *Hostname Server, Request Per Second, Time Per Request, Transfer Rate, Keep-Alive Requests, Concurrency Level* dan banyak lagi.

2.2.5 VirtualBox

Oracle VM VirtualBox adalah perangkat lunak virtualisasi, yang dapat digunakan untuk mengeksekusi sistem operasi "tambahan" di dalam sistem operasi "utama". Sebagai contoh, jika seseorang mempunyai sistem operasi MS Windows yang terpasang di komputernya, maka seseorang tersebut dapat pula menjalankan sistem operasi lain yang diinginkan di dalam sistem operasi MS Windows.

Fungsi ini sangat penting jika seseorang ingin melakukan ujicoba dan simulasi instalasi suatu sistem tanpa harus kehilangan sistem yang ada. Aplikasi dengan fungsi sejenis VirtualBox lainnya adalah VMware dan Microsoft Virtual PC.

2.2.6 HAProxy

HAProxy (*High Availability Proxy*) adalah sebuah aplikasi *opensource* berbasis Linux yang biasa digunakan sebagai *load balancing trafic* jaringan. Karena TCP, HAProxy dapat digunakan tidak hanya untuk HTTP namun juga protokol lainnya. Selain untuk load balancer HAProxy juga bermanfaat sebagai *proxy*.

Load balancing umumnya dikelompokkan dalam dua kategori : *Layer* 4 dan *Layer* 7, *Layer* 4 *load balance* bertindak pada data di *network* TCP (IP, TCP, FTP,UDP). *Layer* 7 *load balance* mendistribusikan permintaan dari *client* berdasarkan data yang ditemukan pada *layer Application* seperti HTTP.

2.2.7 LAMP

LAMP (Linux, Apache, MariaDB, PHP) merupakan sebuah paket perangkat lunak bebas terdiri dari Web Server Apache, Database MariaDB dan Bahasa yang digunakan yaitu PHP. Bebas dalam arti menggunakan, mempelajari, mengubah, menyalin atau menjual sebuah perangkat lunak, pengguna tidak perlu meminta izin dari siapapun.

2.2.8 SNMP

SNMP (*Simple Network Protocol Management*) adalah sebuah protokol apikasi pada jaringan TCP/IP yang menangani manajemen jaringan. Protokol ini didesain sehingga pengguna dapat dengan mudah memantau kondisi jaringan komputer. Pemantauan kondisi jaringan dapat dilakukan dengan cara pengumpulan nilai-nilai informasi dari kondisi jaringan secara jarak jauh atau menggunkan satu

pusat pengamatan. SNMP menjadi protokol yang terus dikembangkan karena banyak perangkat jaringan yang mendukung dan tersedia layanan SNMP seperti router, switch, server, workstation, dan printer.

Protokol SNMP pada jaringan TCP/IP menggunakan transport UDP oleh karena itu dalam penggunaannya tidak akan membebani trafik jaringan. Pada sistem pemantauan jaringan dengan menggunakan layanan SNMP, terdapat tiga komponen dasar antara lain :

1. Manajer SNMP

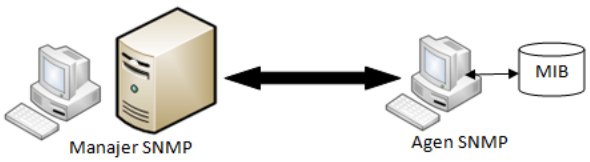
   Manajer adalah perangkat yang menjalankan dan dapat menangani tugas-tugas manajemen jaringan.

2. Agen SNMP

   Agen SNMP adalah perangkat pada jaringan yang akan diamati dan dikelola. Setiap agen akan merespon dan menjawab permintaan manajer SNMP.

3. Management Information Base (MIB)

   MIB pada SNMP dapat dikatakan sebagai tempat penyimpanan informasi yang dimiliki agen. MIB yang terdapat pada SNMP didefinisikan secara hirarki dan setiap bagian mempunyai identifikasi objek (OID).



Gambar 2.2 Manajer, Agen, dan MIB SNMP

2.2.9 Netdata

Netdata adalah sebuah aplikasi monitoring performa dan keadaan sistem yang terdistribusi, dan *real-time*. Merupakan agen monitoring yang sangat optimal pada semua sistem dan kontainer.

Netdata memberikan wawasan yang tak tertandingi, secara real-time, dari segala sesuatu yang terjadi pada sistem yang dijalankannya (termasuk server web, database, aplikasi), menggunakan dasbor web yang sangat interaktif. Netdata dapat berjalan secara mandiri, tanpa komponen pihak ketiga, atau dapat diintegrasikan ke alat pemantauan yang ada (Prometheus, Graphite, OpenTSDB, Kafka, Grafana, dll).

Netdata sangat cepat dan efisien, dirancang untuk berjalan secara permanen di semua sistem (server fisik & virtual, kontainer, perangkat IoT), tanpa mengganggu fungsi intinya.

Netdata adalah perangkat lunak open-source gratis dan saat ini berjalan di Linux, FreeBSD, dan MacOS.