BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Tinjauan pustaka yang digunakan pada penelitian ini meliputi beberapa penelitian yang sudah dilakukan sebelumnya, sebagai berikut. Berbagai penelitian tentang *load balancing* terus dilakukan dan metode *load balancing* terus dikembangkan. Penelitian yang dilakukan oleh Gao, dkk (2008) tentang arsitektur baru dan algoritma untuk webserver cluster berbasis *linux virtual server* (LVS). Pada penelitian tersebut mereka merancang *web server* cluster dengan menggunakan LVS untuk membangun *load balancing* dalam menangani kesalahan pada server tunggal akibat *bottleneck*. Rancangan tersebut diujikan terhadap situs kecil dan menengah dengan biaya yang murah. Parameter sebagai acuannya adalah *throughput*. Hasil yang diperoleh dari penelitian tersebut adalah dengan *load balancing* dapat meningkatkan *throughput* sampai 1,5 kali. Penelitian yang dilakukan oleh Rabu, dkk (2012) tentang implementasi *web server* menggunakan metode LVS-NAT. Pada penelitian tersebut dilakukan untuk mengatasi masalah performa *web server* terhadap jumlah *request* yang meningkat. Berbagai cara mereka telah lakukan yaitu dengan pemutakhiran perangkat keras *web server* namun hanya bersifat sementara. Teknik yang meraka anjurkan

adalah implementasi *load balancing* dimana beban kerja *single server* dibagi ke dalam beberapa server yang ada. Hasil yang diperoleh adalah *load balancing* LVS-NAT mampu meningkatkan nilai *trhoughput response time* dan mampu mengoptimalkan utilisasi.

(Komarudin, 2012) membahas mengenai implementasi clustering pada jaringan komputer diskless menggunakan Red Hat Enterprise Linux 5. Pada penelitian ini, cluster dibangun dengan hardware berupa PC (personal computer) dengan sistem diskless, kemudian booting menggunakan PXE (Pre Execution Environment). Pengujian dilakukan dengan mengeksekusi program hello, kemudian dilakukan analisis perbandingan kinerja menggunakan satu node dan dua node.

2.2 Dasar Teori

2.2.1 Analisis Kinerja

Dalam kamus besar Bahsa Indonesia Departemen Pendidikan Nasional (2005) menjelaskan bahwa analisis adalah penyelidikan terhadap suatu peristiwa untuk mengetahui keadaan yang sebenarnya. Kinerja adalah kemampuan kerja, sesuatu yang dicapai, prestasi yang diperlihatkan.

Dapat disimpulkan bahwa pengertian analisis kinerja adalah penyelidikan terhadap suatu metode untuk mengetahui kerja metode tersebut melalui beberapa percobaan atau pengujian.

2.2.2 Web Server

Nancy J. Yeagar dan Robert E. McGrath dalam buku *web Server*

Teckhnology (1996:20) menjelaskan bahwa *web server* merupakan sebuah komputer dengan koneksi internet, mempunyai software untuk menjalankan komputer dan terkoneksi dengan sistem lain di internet. Komputer ini mempunyai komponen yang penting yaitu perangkat lunak untuk server web. Lebih jauh lagi server web harus mempunyai layanan. Layanan inilah yang menjadi alasan kenapa web server harus ada.

Leon Shklar dan Richard Rosen (2003:65) menjelaskan bahwa web server memungkinkan kita untuk melakukan akses HTTP (*Hyper Text Transfer Protocol*) ke sebuah situs yang sederhana berisi dokumen-dokumen dan informasi lain yang diatur menjadi sebuah struktur *tree,* yang sebenarnya lebih mirip dengan *filesystem* pada komputer. Server web yang modern mengimplementasikan berbagai *protocol* untuk melewatkan *request* ke perangkat lunak tambahan yang menyediakan konten dinamis. Seperti kita ketahui bersama bahwa perangkat lunak dinamis ini seperti PHP, Java, dan lain-lain.

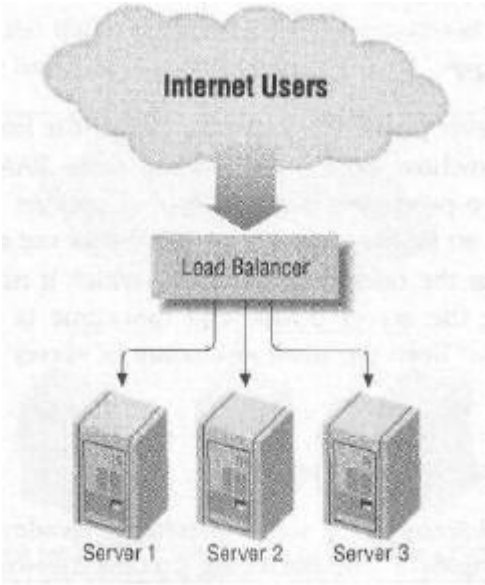
2.2.3 Load Balancing

Sebuah konsep yang gunanya untuk menyeimbangkan beban atau muatan pada infrastruktur Teknologi Informasi sebuah perusahaan atau instansi. Agar seluruh departemen atau bagian dapat memanfaatkan secara maksimal dan optimal yang berfungsi menggabungkan beberapa *line Internet Service Provider*. Load balancing Jaringan sangat penting bila skala dalam jaringan komputer makin besar, demikian juga traffic data yang ada dalam jaringan komputer makin lama makin tinggi.

Secara harafiah, *load balancing* artinya adalah pembagian beban (load)

menjadi seimbang (balance). Load balancing adalah suatu teknik untuk memanfaatkan sumber daya pengolahan yang tersedia secara lebih efektif dengan cara membagi pekerjaan berdasarkan metode pembagian tertentu (Balasubramanian, et al. 2004). Ketika sebuah server sedang diakses oleh para pengguna, maka server tesebut sebenarnya sedang dibebani karena harus melakukan proses terhadap permintaan para penggunanya. Jika penggunanya banyak, maka proses yang dilakukan juga menjadi banyak.

Tony Bourke (2001), *load balance* merupakan sebuah proses dan teknologi yang menyalurkan lalu lintas (*traffic*) diantara beberapa server menggunakan perangkat berbasis jaringan. Perangkat ini (*load balancing server*) menahan atau menangkap *traffic* yang bertujuan kepada sebuah alamat kemudian me-*redirect traffic* tersebut kepada banyak server. Proses *load balance* ini bersifat transparan terhadap pengguna yang melakukan *request* ke server *load balancer*. Ada puluhan bahkan ratusan server yang beroperasi dibalik sebuah alamat.



Gambar 2.1 Load Balancing

2.2.4 Cluster

Sekumpulan komputer (umumnya server) independen yang beroperasi dan terlihat oleh client seolah-olah komputer-komputer tersebut adalah satu buah unit komputer. Proses menghubungkan beberapa komputer agar dapat bekerja seperti itu dinamakan dengan *clustering*.

Thomas Sterling (2001:14) dalam bukunya *Beowulf Cluster Computer with Linux* menjelaskan bahwa *clustering* adalah konsep dan teknik yang *powerful* untuk mendapatkan sebuah kapabilitas yang sangat besar dari sekumpulan (*class*) komponen atau benda. *Cluster* merupakan mekanisme yang mendasar untuk membuat komplekitas dan keberagaman melalui pengumpulan dan penggabungan dari elemen-elemen dasar yang sederhana.

Jadi, *cluster* dalam konteks komputer adalah sekumpulan komputer yang identik dan bekerja bersama unutk mendapatkan kapabilitas yang besar dengan cara menggabungkan atau mengkombinasikan komputer-komputer yang sederhana.

Dalam bidang sistem komputasi, *clustering* digunakan untuk membuat struktur sistem baru dari elemen komputasi yang ada untuk menghasilkan kapabilitas komputasi yang 10 kali lipat lebih murah dari pendekatan lain (Thomas Sterling, 2001:14). Kemudian Thomas juga menyebutkan bahwa beberapa kompuer terbesar di dunia adalah sistem *cluster*. Sistem *cluster* dapat melakukan pekerjaan dengan skala medium, namun dengan biaya yang rendah karena menggunakan teknologi yang berbasis PC.

Sistem ini menjadi populer karena menghasilkan harga yang murah dan

kinerja yang tinggi, fleksibilitas konfigurasi dan *upgrade*, kemampuan untuk mengembangkan tool-tool yang baik, serta membuka kesempatan yang besar untuk pengembangan aplikasi komputasi baru.

2.2.5 VirtualBox

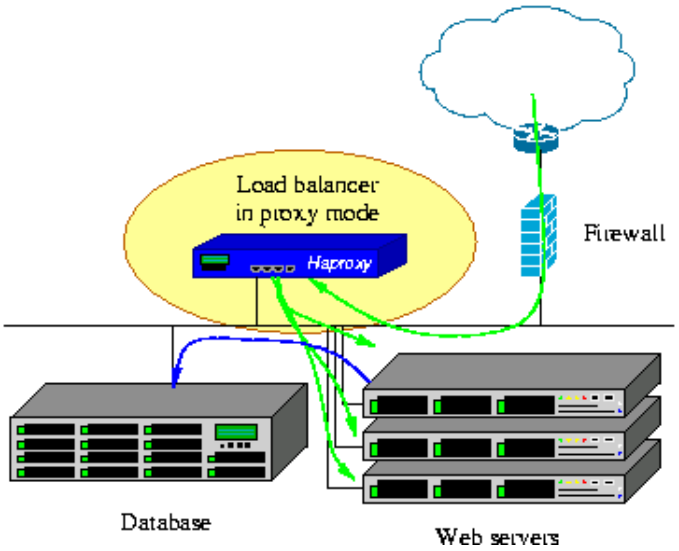
Perangkat lunak virtualisasi, yang dapat digunakan untuk mengeksekusi sistem operasi "tambahan" di dalam sistem operasi "utama". Sebagai contoh, jika seseorang mempunyai sistem operasi MS. Windows yang terpasang di komputernya, maka seseorang tersebut dapat pula menjalankan sistem operasi lain yang diinginkan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

2.2.6 HAproxy

Haproxy adalah produk *opensource* yang mendukung keperluan penyeimbang beban dan *failover web server*, banyak digunakan untuk keperluan *reverse proxy* di site-site yang trafik hariannya tinggi. Pada Haproxy memiliki beberapa parameter didalamnya yaitu sebagai berikut.

1. Global parameter : berisi parameter manajemen proses dan keamanan, *tuning kinerja, debugging,* dan *userlist.*
2. *Proxy.*
3. Konfigurasi server.
4. Manipulasi HTTP.
5. *Accesslist.*
6. *Loogging.*

7. Statistik dan monitoring.



Gambar 2.2 Skema HAProxy

Haproxy adalah penyeimbang beban yang dapat memuat dan menyeimbangkan layanan TCP. Khususnya cocok untuk load balancing HTTP HAProxy sama seperti dengan Heartbeat hanya saja pada heartbeat tidak ada fasilitas berupa fitur statistik menggunakan webbased html.

Pada haproxy ini juga sudah tersedia fitur algoritma *Round Robin* jadi untuk menggunakannya seorang teknisi tinggal mensetting pada file configurasinya saja.

2.2.7 NodeJS

Nodejs merupakan salah satu peranti pengembang yang dapat digunakan untuk membuat aplikasi berbasis *Cloud*. Node.js dikembangkan dari *engine JavaScript*yang dibuat oleh *Google* untuk *browser Chrome* ditambah dengan *libuv* serta beberapa pustaka lainnya.Node.js menggunakan *JavaScript* sebagai bahasan pemrograman dan *event-driven, non-blocking I/O (asynchronous)* model yang

membuatnya ringan dan efisien.Node.js memiliki fitur *built-in* HTTP *server library* yang menjadikannya mampu menjadi sebuah *web server* tanpa bantuan *software* lainnya seperti *Apache* dan *Nginx*.

Pada dasarnya, Node.js adalah sebuah *runtime environment* dan *script library*. Sebuah *runtime environment* adalah sebuah software yang berfungsi untuk mengeksekusi, menjalankan dan mengimplementasikan fungsi-fungsi serta cara kerja inti dari suatu bahasa pemprograman.Sedangkan *script library* adalah kumpulan, kompilasi atau bank data berisi skrip/kode-kode pemrograman.

2.2.8 MongoDB

MongoDB adalah salah satu *softwareNoSQL* yang termasuk dalam kategori *Document Store/ Document-Oriented Database*, yaitu data disimpan dalam bentuk dokumen. Suatu dokumen bisa di ibaratkan seperti suatu *record* dalam basis data relasional dan isi dari masing-masing dokumen tersebut bisa berbeda-beda dan ada pula yang sama. Hal ini berbeda dengan basis data relasional yang menetapkan keseragaman kolom serta tipe data dengan data yang *NULL* jika tidak terdapat data. MongoDB menyimpan data dalam bentuk dokumen dengan menggunakan format *JSON*. Konsep dasar yang harus dipahami dalam MongoDB sebagai *document-oriented database* adalah *documents* dan *collections*. Sama halnya dengan basis data relasional, MongoDB menyimpan data dalam suatu basis data. Di dalam basis data tersebut terdapat *collections* yang bisa diibaratkan seperti tabel dalam basis data relasional. *Collections* digunakan untuk menyimpan dokumen (*documents*). Dalam istilah basis data relasional, *documents* adalah *records*.

2.2.9 Performansi Jaringan dan Server

a. Throughput

Throughput adalah jumlah bit yang diterima dengan sukses perdetik melalui sebuah sistem atau media komunikasi dalam selang waktu pengamatan tertentu. Umumnya *throughput* direpresentasikann dalam satuan bit per second (bps). Aspek utama *throughput* yaitu berkisar pada ketersediaan *bandwitch* yang cukup untuk menjalankan aplikasi. Hal ini menentukan besarnya trafik yang dapat diperoleh suatu aplikasi saat melewati jaringan.

b. Response Time

Response time adalah selang waktu antara user memasukkan suatu perintah ke dalam sistem hingga mendapat balasan selengkapnya dari sistem.