

BAB I

PENDAHULUAN

1.1 Latar Belakang

Load balancer merupakan komponen dalam infrastruktur jaringan modern yang berfungsi mendistribusikan beban trafik secara dinamis ke sekelompok *server*. Teknologi ini lahir dari kebutuhan untuk mengatasi keterbatasan *server* tunggal dalam menangani permintaan yang bersifat fluktuatif, terutama di era komputasi awan dan aplikasi berbasis *microservices*. Pada dasarnya, *load balancer* tidak hanya meningkatkan kinerja sistem tetapi juga berfungsi dalam menjamin kelangsungan layanan [1]. *Load balancer* mampu meningkatkan ketersediaan layanan dengan mekanisme *failover* otomatis ketika *server* mengalami gangguan, skalabilitas horizontal melalui penambahan *server backend* tanpa mengganggu operasional, serta optimalisasi kinerja dengan memastikan tidak ada *server* yang kelebihan beban sementara yang lain menganggur. [2]

Load balancer dapat diklasifikasikan berdasarkan dua pendekatan yaitu metode distribusi dan *OSI layer*. Berdasarkan metode distribusi, algoritma seperti *Round Robin*, *Weighted Round Robin*, *Least Connection*, *Weighted Least Connection*, *IP Hash*, *Least Response Time*, dan *Resource Based Method*. Sementara dari sisi *OSI Layer*, terdapat *Layer 4 (Transport Layer)* yang bekerja pada protokol TCP/UDP, dan *Layer 7 (Application Layer)* yang mampu memproses konten HTTP/HTTPS [3]. Implementasinya dapat berbentuk perangkat keras seperti F5 BIG-IP, atau perangkat lunak seperti Traefik, NGINX, Seesaw, dan HAProxy.

HAProxy (*High Availability Proxy*) menjadi salah satu *software open-source* untuk *load balancing* dan *failover* untuk aplikasi berbasis FTP (File Transfer Protocol) dan HTTP (Hyper Text Transfer Protocol) yang mampu menghubungkan client dengan *server* juga mendistribusikan trafik. [4] Dalam mendistribusikan beban trafik, HAProxy menyediakan berbagai algoritma, seperti *Round Robin*, *Least Connection*, *Source IP Hash*, *Weighted Round Robin*, *Weighted Least Connection*. Pada penelitian ini, Penulisan akan berfokus pada Algoritma *Weighted Round Robin* yang bekerja dengan mendistribusikan trafik secara bergiliran kepada setiap *server* dalam *pool*, namun dengan mempertimbangkan bobot yang telah ditetapkan untuk setiap *server* dan algoritma *Weighted Least Connection* yang bekerja dengan cara mempertimbangkan jumlah koneksi aktif pada setiap *server* dan mengarahkan trafik baru ke *server* dengan beban koneksi paling ringan. Pendekatan ini dinilai lebih dinamis karena mampu merespons perubahan beban secara *real-time*, sehingga mengurangi risiko *overload* pada *server* tertentu. [5] [6]

Untuk menjalankan *server-server* tersebut dibutuhkan sebuah *Virtual Environment* yang memungkinkan efisiensi sumber daya melalui virtualisasi perangkat keras fisik menjadi beberapa *Virtual Machine* yang berjalan secara independen. Teknologi ini memungkinkan organisasi untuk mengonsolidasikan *server*, mengurangi biaya operasional, dan meningkatkan fleksibilitas dalam manajemen infrastruktur. Dengan *Virtual Environment*, sumber daya seperti CPU, RAM, dan penyimpanan dapat dialokasikan secara dinamis sesuai kebutuhan, meminimalkan pemborosan kapasitas sekaligus memudahkan skalabilitas [7]. Selain itu, *Virtual Environment* juga mendukung isolasi sistem, memungkinkan

pengujian aplikasi atau konfigurasi baru tanpa mengganggu produksi, serta memfasilitasi *disaster recovery* dengan fitur *snapshot* dan migrasi *live*.

Ada beberapa jenis *Virtual Environment* yang umum digunakan, mulai dari *hypervisor* tipe-1 seperti VMware ESXi, Microsoft Hyper-V, dan Proxmox *Virtual Environment* yang berjalan langsung di atas perangkat keras fisik, hingga *hypervisor* tipe-2 seperti Oracle VirtualBox dan VMware Workstation yang berjalan di atas sistem operasi utama. Solusi berbasis container seperti *Docker* dan *Linux Container* (LXC) juga termasuk dalam kategori *Virtual Environment* ringan yang mengvirtualisasi hanya pada level aplikasi. Dalam konteks *load balancing*, *Virtual Environment* seperti Proxmox *Virtual Environment* memungkinkan administrator sistem untuk mengelola infrastruktur sistem dengan fleksibilitas tinggi. [8].

Proxmox *Virtual Environment* memiliki kemampuan untuk menjalankan banyak *Virtual Machine* secara bersamaan. Ketika dikombinasikan dengan HAProxy, sistem ini memungkinkan pengelolaan beban trafik yang lebih dinamis dan responsif. Dalam lingkungan virtualisasi, sumber daya seperti CPU, memori, dan *bandwidth* dapat dialokasikan ulang secara *real-time* sesuai kebutuhan. HAProxy dapat menyesuaikan distribusi trafik berdasarkan perubahan alokasi sumber daya ini, sehingga mencegah kelebihan beban maupun pengangguran pada *Virtual Machine*. Pendekatan ini penting untuk menjaga ketersediaan layanan dan memastikan pengalaman pengguna tetap optimal, khususnya pada sistem yang memerlukan tingkat responsivitas tinggi.

Penelitian terkait *Load balancing* telah mengalami perkembangan yang signifikan. Penelitian yang dilakukan oleh [9] berfokus pada permasalahan

peningkatan beban *server* web akibat tingginya permintaan pengguna, serta bagaimana algoritma penjadwalan seperti *Weighted Round Robin* dan *Weighted Least Connection* dapat mengoptimalkan distribusi beban pada *load balancer* berbasis NGINX. Penelitian tersebut menggunakan tiga *server* web dengan bobot 10:15:20 dan melakukan pengujian menggunakan *Apache Benchmark* terhadap 1000, 2000, dan 5000 request untuk mengevaluasi parameter *throughput*, *Response Time*, dan *Packet Loss*. Hasil penelitian tersebut menunjukkan bahwa algoritma *Weighted Least Connection* memiliki kinerja yang lebih baik dibanding *Weighted Round Robin* pada aspek *throughput* dan *Response Time*, sedangkan *Weighted Round Robin* unggul pada parameter *Packet Loss*.

Penelitian Selanjutnya yang diteliti oleh [10] membandingkan berbagai skenario pembagian bobot menggunakan algoritma *Weighted Round Robin* saja, seperti 2:1, 3:1, 4:1, dan 5:1, pada dua buah *server* dengan spesifikasi berbeda. Permasalahan penelitian tersebut adalah bagaimana cara mendistribusikan beban secara merata pada lingkungan virtualisasi Proxmox *Virtual Environment* agar tidak terjadi penumpukan trafik pada satu *server* saja. Metode yang digunakan yaitu dengan menerapkan *load balancing* menggunakan HAProxy dan algoritma *Weighted Round Robin* pada dua *virtual server* yang memiliki perbedaan jumlah core CPU dan kapasitas RAM. Pengujian dilakukan berdasarkan parameter *Quality Of Services* dengan mengukur *throughput*, *Delay*, *Packet Loss*, serta penggunaan CPU melalui skenario pembagian bobot yang berbeda. Hasilnya menunjukkan bahwa pembagian beban dengan rasio 2:1 memberikan performa terbaik, dengan nilai *throughput* tertinggi sebesar 6.717 Mbit/s, *Delay* terendah sebesar 0.505 ms,

serta tidak adanya *Packet Loss*. Selain itu, skenario ini juga menghasilkan penggunaan CPU yang paling seimbang di antara kedua *server*, sehingga dinilai sebagai konfigurasi yang paling optimal dalam konteks penelitian tersebut.

Setelah itu terdapat penelitian yang dilakukan oleh [11] bertujuan untuk membandingkan kinerja algoritma *Weighted Round Robin* dan *Weighted Least Connection* pada skenario *load balancing* menggunakan *web server* NGINX di lingkungan *Virtual Private Server* komersial. Permasalahan yang diangkat berfokus pada efektivitas pembagian beban trafik dalam menangani permintaan akses terhadap *file* statis seperti gambar, video, dan dokumen PDF, yang disajikan dalam bentuk layanan *cloud storage*. Metode penelitian bersifat eksperimental dengan skenario pengujian menggunakan *Apache JMeter* untuk mensimulasikan trafik yang dibebankan pada tiga node *server* dengan bobot konfigurasi 2:3:5. Hasil penelitian menunjukkan bahwa algoritma *Weighted Least Connection* secara konsisten unggul dibandingkan *Weighted Round Robin* dalam empat parameter pengujian, yaitu *throughput*, *Response Time*, *error request*, dan penggunaan CPU. Secara spesifik, *throughput Weighted Least Connection* tercatat 38.31 thread/sec lebih tinggi dari *Weighted Round Robin*, *Response Time* rata-rata lebih cepat 31.77 detik, dan *error request* 39.66% lebih rendah. Hasil Penelitian tersebut menegaskan bahwa *Weighted Least Connection* lebih adaptif terhadap kondisi beban dinamis karena mempertimbangkan jumlah koneksi aktif saat mendistribusikan permintaan, berbeda dengan *Weighted Round Robin* yang hanya mengandalkan bobot statis.

Berdasarkan 3 penelitian sebelumnya dapat disimpulkan penerapan algoritma *Weighted Round Robin* dan *Weighted load balancing* untuk *load balancer* efektif

untuk diterapkan. Oleh karena itu penulis mengangkat sebagai penelitian, perbedaan penelitian ini dengan penelitian sebelumnya adalah penelitian ini secara khusus mengimplementasikan algoritma *Weighted Round Robin* dan *Weighted Least Connection* pada HAProxy dalam lingkungan virtualisasi berbasis bare-metal menggunakan Proxmox *Virtual Environment*. Penelitian ini melakukan pembagian bobot statis 3:2:1 pada tiga node *backend* untuk menguji efektivitas algoritma dalam skenario nyata. Selain itu, penggunaan satu node terdedikasi sebagai *load balancer* memungkinkan simulasi trafik lebih terkonsentrasi. Penilaian kinerja dilakukan secara menyeluruh dengan mengukur parameter *Quality Of Services* yang meliputi *latency*, *throughput*, *Packet Loss*, *jitter*, dan *Response Time* sehingga memberikan analisis yang lebih mendalam dan komprehensif terhadap performa masing-masing algoritma dalam kondisi virtualisasi yang menyerupai lingkungan produksi.

1.2 Identifikasi Masalah

Identifikasi masalah dalam penelitian ini adalah bagaimana kinerja dan keandalan algoritma *Weighted Round Robin* pada *load balancer* HAProxy dalam penerapannya di Proxmox *Virtual Environment*. Dalam konteks ini, beberapa masalah yang dihadapi meliputi:

1. Perlunya dilakukan analisis *Quality Of Services* menggunakan parameter *latency*, *throughput*, *Packet Loss*, *jitter*, dan *Response Time* terhadap kinerja algoritma *Weighted Round Robin* dalam penerapan *load balancing*

menggunakan HAProxy pada lingkungan virtualisasi Proxmox *Virtual Environment*.

2. Perlunya dilakukan analisis *Quality Of Services* menggunakan parameter *latency, throughput, Packet Loss, jitter, dan Response Time* terhadap kinerja algoritma *Weighted Least Connection* dalam penerapan *load balancing* menggunakan HAProxy pada lingkungan virtualisasi Proxmox *Virtual Environment*.
3. Perlunya dilakukan perbandingan analisis *Quality Of Services* menggunakan parameter *latency, throughput, Packet Loss, jitter, dan Response Time* terhadap kinerja algoritma *Weighted Round Robin* dan *Weighted Least Connection* dalam konteks penerapan *load balancing* menggunakan HAProxy di lingkungan Proxmox *Virtual Environment*.

1.3 Rumusan Masalah

Berdasarkan Identifikasi masalah dan latar belakang penelitian, terdapat beberapa rumusan masalah dalam penelitian ini sebagai berikut :

1. Bagaimana kinerja algoritma *Weighted Round Robin* dalam mendistribusikan beban menggunakan HAProxy pada lingkungan virtualisasi Proxmox *Virtual Environment* berdasarkan analisis *Quality Of Services* seperti *latency, throughput, Packet Loss, jitter, dan Response Time*?
2. Bagaimana kinerja algoritma *Weighted Least Connection* dalam mendistribusikan beban menggunakan HAProxy pada lingkungan virtualisasi Proxmox *Virtual Environment* berdasarkan analisis *Quality Of Services* seperti *latency, throughput, Packet Loss, jitter, dan Response Time*?

3. Bagaimana perbandingan kinerja antara algoritma *Weighted Round Robin* dan *Weighted Least Connection* dalam menerapkan *load balancing* menggunakan HAProxy pada Proxmox *Virtual Environment* ditinjau dari parameter *Quality Of Services* yaitu *latency, throughput, Packet Loss, jitter, dan Response Time*?

1.4 Batasan Masalah

Agar penelitian tetap dalam jalur dan tidak tersebar fokusnya, penelitian ini dibatasi sebagai berikut:

1. *Load balancer* yang digunakan adalah HAProxy, tanpa membandingkan dengan *software load balancing* lainnya seperti NGINX atau Traefik.
2. Penelitian menggunakan dua algoritma distribusi *load balancing*, yaitu *Weighted Round Robin* dan *Weighted Least Connection*
3. Pengukuran penelitian digunakan menggunakan parameter pada *Quality Of Services*, yaitu *Delay (Latency), Throughput, Packet Loss, Jitter, dan Response Time*.
4. Penelitian akan diuji menggunakan *Apache Benchmark* dan *Wireshark*.
5. *Client* pengujian dilakukan di perangkat terpisah namun tetap tersambung via ethernet.
6. Penelitian ini tidak menerapkan *failover*.
7. Aspek keamanan, kestabilan jangka panjang, atau konsumsi energi dari masing-masing algoritma tidak termasuk dalam cakupan penelitian ini.

8. Penelitian ini tidak mencakup pengaruh faktor eksternal seperti gangguan jaringan, kegagalan perangkat keras, atau serangan siber terhadap performa *load balancing*.

1.5 Tujuan Penelitian

Sesuai dengan judul penelitian, maka tujuan dari penelitian ini diantara sebagai berikut :

1. Menganalisis kinerja algoritma *Weighted Round Robin* dalam mendistribusikan beban menggunakan HAProxy pada lingkungan virtualisasi Proxmox *Virtual Environment* berdasarkan analisis *Quality Of Services* seperti *latency*, *throughput*, *Packet Loss*, *jitter*, dan *Response Time*.
2. Menganalisis kinerja algoritma *Weighted Least Connection* dalam mendistribusikan beban menggunakan HAProxy pada lingkungan virtualisasi Proxmox *Virtual Environment* berdasarkan analisis *Quality Of Services* seperti seperti *latency*, *throughput*, *Packet Loss*, *jitter*, dan *Response Time*.
3. Menganalisis perbandingan kinerja antara algoritma *Weighted Round Robin* dan *Weighted Least Connection* dalam menerapkan *load balancing* menggunakan HAProxy pada Proxmox *Virtual Environment* ditinjau dari parameter *Quality Of Services* yaitu *latency*, *throughput*, *Packet Loss*, *jitter*, dan *Response Time*.

1.6 Manfaat Penelitian

Penelitian ini memiliki potensi untuk memberikan manfaat dalam berbagai aspek, yaitu:

1. Menambah wawasan mengenai efektivitas algoritma *Weighted Round Robin* dan *Weighted Least Connection* dalam konteks *load balancing* di lingkungan virtualisasi.
2. Memberikan referensi dalam memilih algoritma *load balancing* yang paling optimal untuk diterapkan pada infrastruktur berbasis Proxmox *Virtual Environment*.

1.7 Sistemika Tugas Akhir

Adapun sistematika penulisan tugas akhir ini terdiri dari tiga bab. Dalam penulisan tugas akhir ini dilakukan beberapa tahap yang berisi tahap-tahapan agar masalah dapat terpecahkan. Adapun rincian bab-bab nya adalah sebagai berikut,

BAB I PENDAHULUAN

Pada bab ini membahas latar belakang, identifikasi masalah, rumusan masalah, tujuan dan manfaat, batasan masalah dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Berisi penjelasan teori – teori dasar yang berhubungan dengan topik penelitian dan bersumber dari berbagai literatur sejenis yang berhubungan dengan Proxmox dan algoritma *Weighted Round Robin* dan *Weighted Least Connection*

BAB III METODOLOGI PENELITIAN

Pada bab ini membahas tentang gambaran umum objek penelitian, penguraian kerangka penelitian, penguraian cara kerja penelitian, uraian metode pengumpulan data, uraian model penelitian dan uraian hasil analisis sistem berjalan, analisis sistem usulan serta analisis permasalahan.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini berisi penjelasan tentang hasil implementasi dari cara kerja penelitian serta pembahasan dan penghitungan dari hasil pengujian.

BAB V PENUTUP

Bab ini menjelaskan kesimpulan, keterbatasan, serta saran terhadap penyusunan skripsi serta evaluasi dari penelitian yang diuji di skripsi ini.