1. Pendahuluan

a. Latar Belakang

Mengelola infrastruktur TI yang kuat dan andal merupakan tantangan bagi perusahaan yang menangani ratusan layanan setiap hari. *Server* adalah komputer yang menyediakan layanan melalui jaringan komputer, dan platform *web server* digunakan untuk menjalankan aplikasi [1]. Namun, peningkatan permintaan layanan dapat menyebabkan gangguan jika infrastruktur *server* tidak memadai. Untuk meningkatkan kapasitas dan kinerja *server* tanpa menambah *server* fisik, teknologi *clustering* di lingkungan virtualisasi dapat digunakan. Virtualisasi memungkinkan satu komputer berfungsi sebagai beberapa komputer virtual dengan arsitektur yang sama dengan komputer fisik [2].

Penelitian ini memanfaatkan platform *open-source Haproxy* sebagai mesin web server *load balancer*. *Haproxy* dipilih karena murah, cepat, andal, dan menawarkan ketersediaan tinggi, *failover*, serta *load balancing*. Selain itu, *Haproxy* dapat digunakan sebagai *proxy* untuk aplikasi berbasis *TCP* dan *HTTP* [3]. Untuk mendukung *load balancer*, teknik virtualisasi menggunakan *Docker* diterapkan. *Docker* memungkinkan pembangunan, pengujian, dan peluncuran aplikasi dalam *container* [4]. Namun, penerapannya pada satu server dapat menimbulkan tantangan, terutama dengan peningkatan beban lalu lintas.

Docker Swarm, versi baru *Docker*, diciptakan untuk mengelola *cluster Docker* yang berjalan di beberapa node yang disebut *Swarm* [5]. Dalam konteks ini, penerapan algoritma penyeimbangan beban di *Docker Swarm* dapat meningkatkan kinerja *load balancer*. Algoritma penyeimbang beban yang umum digunakan adalah *Round Robin* dan *Least Connection*. Algoritma *Round Robin* mendistribusikan beban secara berurutan ke setiap node dalam interval waktu tertentu, sementara algoritma *Least Connection* membagi beban berdasarkan jumlah koneksi yang dibuat *node*. *Node* dengan koneksi paling sedikit akan menerima beban berikutnya [6].

Dalam lingkungan riil *web server*, dimana lingkungan riil *web server* merujuk pada situasi di mana *web server* digunakan dalam operasi yang sebenarnya, melayani permintaan dari pengguna nyata dengan berbagai tingkat beban dan kondisi jaringan yang berubah-ubah. Performa *load balancer* tidak hanya diukur berdasarkan satu parameter saja. Sebaliknya, berbagai parameter kinerja seperti penggunaan *CPU*, penggunaan memori, *throughput* jaringan, dan waktu respons harus diperhitungkan secara bersama-sama. Namun saat ini dalam evaluasi kinerja *load balancer* sering kali terbatas pada pengukuran parameter individual tanpa mempertimbangkan interaksi antara berbagai parameter tersebut. Hal ini dapat menyebabkan hasil yang kurang akurat dalam menggambarkan kinerja load balancer dalam kondisi dunia nyata.

Penelitian ini berfokus pada evaluasi kinerja dua metode *load balancer*, *Round Robin* dan *Least Connection*, dalam lingkungan *Docker Swarm* pada *virtual machine*. Pengujian membandingkan penggunaan sumber daya CPU dan RAM, *throughput*, dan *response time*, serta mempertimbangkan kondisi riil lingkungan *web server* dengan variasi beban kerja dan skenario lalu lintas, menggunakan pembobotan antara keempat parameter tersebut untuk menentukan algoritma terbaik yang paling efisien dan andal guna memberikan gambaran yang lebih jelas mengenai performa *load balancer* dalam situasi sebenarnya.

b. Topik dan Batasannya

Penelitian ini menggunakan beberapa batasan untuk melakukan *load testing*. *Apache Bench* digunakan untuk mensimulasikan dan mengukur beban pada *web server*, sedangkan *VirtualBox* pada *virtual machine* digunakan untuk sebagai lingkungan pengujian. *Web server* dan *load balancer* yang digunakan adalah *Haproxy*. Pengujian melibatkan empat node, terdiri dari satu *node manager* dan tiga *node worker*. Kinerja sistem dievaluasi berdasarkan metrik penggunaan CPU, penggunaan memori, *throughput*, dan waktu respons. Jumlah *request* yang diuji mencakup 10.000, 15.000, 20.000, dan 25.000 untuk menilai penanganan beban kerja yang berbeda. Selain itu, konfigurasi CPU dan RAM yang digunakan meliputi CPU dengan 1 *Core*, 2 *Core*, dan 3 *Core*, serta RAM sebesar 1 GB, 2 GB, 4 GB, dan 6 GB untuk melihat dampak berbagai sumber daya pada kinerja sistem. Penelitian ini juga menggunakan *Docker Compose* versi 3.8 untuk mengatur lingkungan kontainer. MySQL versi 8.0 digunakan sebagai basis data untuk WordPress versi 6.2.2. Selain itu, Haproxy versi 2.8.2 digunakan sebagai *load balancer* dalam pengujian ini.

c. Tujuan

Penelitian ini mensimulasikan penggunaan *Docker Swarm* dalam lingkungan virtualisasi desktop untuk menguji kinerja algoritma *load balancer*. Pengujian dilakukan dengan menggunakan dua algoritma *load balancing*, yaitu *Round Robin* dan *Least Connection*. Tujuan utama penelitian ini adalah untuk menentukan algoritma *load balancing* yang paling efektif berdasarkan beberapa parameter kinerja utama, seperti penggunaan *CPU*, penggunaan memori, *throughput* jaringan, dan waktu respons. Selain itu, penelitian ini juga mempertimbangkan kondisi riil lingkungan *web server* dengan menerapkan pembobotan antara keempat parameter tersebut, sehingga dapat memberikan gambaran yang lebih komprehensif tentang performa *load balancer* dalam situasi dunia nyata. Dengan mengumpulkan dan menganalisis data dari berbagai skenario pengujian, penelitian ini bertujuan untuk memberikan rekomendasi mengenai algoritma *load balancing* yang dapat memberikan kinerja optimal dalam pengelolaan beban kerja pada *Docker Swarm*. Hasil dari penelitian ini diharapkan dapat memberikan wawasan yang bermanfaat bagi pengembang dan administrator sistem dalam mengoptimalkan distribusi beban pada lingkungan virtualisasi.