BAB I
PENDAHULUAN

1.1 Latar Belakang

Peningkatan mutu pelayanan server dapat dilakukan dengan meningkatkan beberapa aspek, antara lain kinerja, keandalan, dan efisiensi kerja server guna memberikan pengalaman yang lebih baik kepada pengguna [1]. Upaya peningkatan kinerja server dengan permintaan tinggi sangat penting dilakukan agar dapat memberikan kecepatan akses yang memadai dalam skala lebih besar. Berdasarkan penelitian mengenai analisis kualitas server atau *Quality of Service* (QoS), ditemukan pengaruh signifikan antara peningkatan kinerja server dengan skalabilitas pelayanan aplikasi server [2].

Peningkatan kinerja server merupakan aspek penting dalam pengelolaan sistem terdistribusi untuk meningkatkan kinerja, ketersediaan, dan skalabilitas aplikasi [3]. Salah satu metode yang sering digunakan adalah *load balancing*. Dalam konteks ini, *load balancing* memainkan peran krusial dalam mendistribusikan lalu lintas aplikasi secara merata dan efisien di lingkungan terdistribusi. Salah satu teknologi yang dapat digunakan untuk meningkatkan kinerja server adalah Docker Swarm [3]. Docker adalah implementasi konsep *microservice container*, yaitu pembangunan infrastruktur server dalam bentuk *microservice* yang terisolasi dan berukuran kecil karena hanya memuat layanan (*service*) berskala mikro [4].

Keunggulan menggunakan *microservice* adalah penggunaan *resource* server yang lebih kecil dibandingkan dengan metode pengelolaan server konvensional. Secara konvensional, server disusun dalam satu sistem besar yang memuat banyak layanan atau dikenal sebagai *monolith*. Banyaknya layanan dalam sistem *monolith* menjadi beban bagi server dalam menyediakan layanan aplikasi. Sebaliknya, *microservice* memiliki beban sistem yang kecil, sehingga penggunaan *resource* server lebih efisien dibandingkan *monolith* [5].

Selain mengurangi penggunaan *resource* server, optimalisasi kinerja layanan juga dapat dicapai melalui sistem layanan terdistribusi, yakni dengan konsep *load balancing* di antara beberapa server penyedia aplikasi. Konsep ini dikenal sebagai *load balancing*. Keunggulan Docker terletak pada kemampuan *load balancing* secara *native* melalui fitur Docker Swarm [5]. Dengan fitur ini, beberapa server dalam klaster dapat berfungsi sebagai satu kesatuan server terdistribusi dan melakukan *load balancing*. Penggunaan Docker Swarm mampu memberikan dua optimalisasi sekaligus, yaitu optimalisasi penggunaan sumber daya server dan optimalisasi jaringan [6].

Secara umum, keunggulan Docker Swarm sebagai platform orkestrasi kontainer adalah efisiensi tinggi dalam pengelolaan klaster. Docker Swarm memungkinkan pengguna membuat dan mengelola klaster Docker yang terdiri dari beberapa mesin *host* dan secara otomatis mendistribusikan aplikasi. *Load balancing* membantu mendistribusikan lalu lintas aplikasi secara merata di seluruh kontainer dalam klaster, sehingga mencegah penumpukan lalu lintas di satu titik dan meningkatkan ketersediaan aplikasi [2]. Docker Swarm juga menawarkan metode *load balancing* untuk mengoptimalkan jaringan.

Universitas Riau (UNRI) telah memanfaatkan teknologi Docker Swarm sejak 2019. Layanan utama perpustakaan, yaitu "Digilib" dengan alamat https://digilib.unri.ac.id, dikelola dalam klaster server Docker Swarm di bawah tanggung jawab Sub Koordinator IT Perpustakaan. Klaster tersebut terdiri dari 20 *worker* Docker Swarm. Selama empat tahun penggunaan, server semakin stabil dan mampu menangani permintaan dalam skala besar. Sebagai alternatif dan pembanding, UNRI tetap menggunakan server cadangan berbasis teknologi konvensional (*monolith*).

Saat ini, tantangan besar bagi UNRI adalah ketika diminta melaporkan peningkatan kinerja server dengan *load balancing* Docker Swarm dibandingkan teknologi konvensional. Laporan ini akan menjadi dasar penganggaran sistem di masa mendatang. UNRI membutuhkan kajian ilmiah yang mendalam dan dapat

dipertanggungjawabkan mengenai peningkatan kinerja ini, sementara sumber daya yang dimiliki terbatas. Oleh karena itu, diperlukan riset khusus untuk mengukur peningkatan kinerja server dengan teknologi baru dibandingkan dengan teknologi konvensional.

Sebagai mahasiswa bidang teknologi informasi, peneliti merasa bertanggung jawab untuk melakukan pengkajian mendalam mengenai masalah tersebut. Peneliti melihat tantangan di UNRI sebagai ujian kompetensi dalam bidang yang relevan. Secara teoritis, *load balancing* menggunakan Docker Swarm hampir pasti lebih unggul dibandingkan teknologi konvensional. Namun, UNRI belum memiliki analisis yang tepat dengan metode dan data akurat. Tantangan inilah yang mendorong peneliti untuk mengambil bagian dalam menjawab masalah tersebut melalui penelitian ilmiah.

Untuk menjawab tantangan tersebut, diperlukan dasar teoritis yang komprehensif terkait optimisasi peningkatan kinerja, efisiensi, dan keandalan operasional [7]. Publikasi terkait memberikan konsep penting tentang optimalisasi server yang relevan bagi UNRI.

Penelitian Caen (2015) dalam *Docker High Performance* menggambarkan optimalisasi performa server di lingkungan klaster [8]. Penelitian tersebut merumuskan analisis kinerja dan optimisasi komprehensif dalam pengelolaan serta perencanaan sistem awan [9] dengan tujuan meningkatkan kinerja, efisiensi, dan keandalan di lingkungan awan. Konsep optimalisasi yang disampaikan dalam penelitian tersebut mencakup kinerja, efisiensi, dan keandalan [7], yang dapat diimplementasikan di UNRI.

Analisis kinerja server mencakup pemantauan sumber daya fisik, virtualisasi, jaringan, dan aplikasi yang berjalan di atas infrastruktur *Infrastructure as a Service* (IaaS) [7]. Dasar teoritis tersebut sejalan dengan konsep pemantauan infrastruktur server [11].

Pemantauan kinerja server yang optimal membuka peluang besar untuk meningkatkan kinerja server secara keseluruhan [5]. Melalui metode pemantauan

yang memadai, seperti pengukuran kualitas layanan, dapat ditemukan rumusan optimal untuk meningkatkan kinerja server [12]. Hal ini bertujuan memberikan layanan terbaik bagi pengguna, sehingga pengguna dapat merasakan dampak positif [13]. Secara teknis, pemantauan dilakukan terhadap beberapa aspek yang dijadikan parameter kinerja [19], seperti waktu respons, waktu muat halaman, keandalan, dan skalabilitas. Aspek-aspek tersebut merupakan bagian dari metrik QoS [14].

QoS adalah prinsip pengukuran kinerja yang mencakup beberapa parameter yang mempengaruhi performa server [3]. Penelitian QoS membutuhkan metode yang tepat dalam pengumpulan data, pemilihan alat, serta teknologi dan metode analisis untuk menilai kinerja server [14]. Dengan demikian, QoS dapat diterapkan secara komprehensif pada server lain. Kinerja pusat data, termasuk pengumpulan data QoS, sangat memengaruhi peningkatan kinerja server secara keseluruhan [11]. Keberhasilan penerapan QoS bergantung pada strategi pengumpulan data yang efektif, teknik pengukuran yang tepat, alat pemantauan yang relevan, dan metode analisis yang akurat [16].

Berdasarkan beberapa penelitian di atas, peneliti menemukan solusi yang layak ditawarkan, termasuk metode analisis, data, dan pengukuran yang sesuai dan standar untuk menjawab tantangan di UNRI. Hasil analisis ilmiah ini akan disusun dalam skripsi berjudul *"ANALISIS PERBANDINGAN KINERJA SERVER MENGGUNAKAN LOAD BALANCING DENGAN DOCKER SWARM (STUDI KASUS: APLIKASI DIGILIB PERPUSTAKAAN UNIVERSITAS RIAU) "*.

1.2 Rumusan Masalah

Berdasarkan penjelasan di atas, maka masalah dalam penelitian ini dapat dirumuskan "Bagaimana analisis Perbandingan kinerja *server* menggunakan *load balancing* dengan metode *docker swarm* pada aplikasi Digilib Perpustakaan Universitas Riau?

1.3 Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Analisis kinerja server dilakukan menggunakan Docker Swarm hanya pada **Docker version 20.10.21, build 20.10.21-0ubuntu1~22.04.3**.
2. Sistem operasi yang digunakan adalah **Linux Ubuntu Server versi 22.04**, rilis April 2022.
3. Data yang dikumpulkan berasal dari *resource QoS* berdasarkan studi kasus selama masa penelitian berlangsung, sesuai dengan jadwal penelitian.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah di atas, tujuan penelitian ini adalah sebagai berikut:

1. Menganalisis *QoS* pada aplikasi Digilib menggunakan metode konvensional.
2. Menganalisis *QoS* pada aplikasi Digilib dengan *load balancing* menggunakan metode Docker Swarm.
3. Membandingkan hasil analisis *QoS* antara penggunaan metode konvensional dan metode *load balancing* dengan Docker Swarm.

1.5 Manfaat Penelitian

Penelitian ini diharapkan memberikan manfaat sebagai berikut:

1. Menjadi bahan rekomendasi bagi pengembangan aplikasi Digilib Perpustakaan Universitas Riau di masa depan.
2. Menjadi referensi ilmiah bagi pengembangan ilmu pengetahuan, khususnya dalam optimalisasi jaringan menggunakan *load balancing* dengan metode Docker Swarm.

1.6 Sistematika Penulisan

Sistematika penulisan skripsi ini dibagi menjadi lima bab, yaitu:

- **BAB I PENDAHULUAN**

  Bab ini berisi latar belakang penelitian, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

- **BAB II LANDASAN TEORI**

  Bab ini membahas teori-teori dasar yang relevan dengan topik penelitian serta penelitian terdahulu yang berkaitan dengan topik yang diteliti. Selain itu, bab ini membandingkan perbedaan dan kesamaan di antara penelitian terdahulu serta menjelaskan teori yang digunakan sebagai landasan implementasi dalam penelitian ini.

- **BAB III METODE PENELITIAN**

  Bab ini menjelaskan metode yang digunakan dalam penelitian, termasuk lokasi dan waktu penelitian, alat dan bahan, serta teknik pengumpulan dan analisis data yang digunakan.

- **BAB IV HASIL DAN PEMBAHASAN**

  Bab ini memaparkan hasil penelitian atau uji coba yang telah dilakukan. Hasil tersebut kemudian dianalisis secara mendalam untuk memberikan informasi terkait QoS server berdasarkan hasil eksperimen.

- **BAB V PENUTUP**

  Bab ini berisi kesimpulan dan saran. Kesimpulan penelitian mencakup hasil analisis QoS, yaitu apakah layanan jaringan berada dalam kategori baik atau tidak. Selain itu, saran disusun untuk memberikan masukan bagi penelitian selanjutnya dan solusi perbaikan untuk pengembangan lebih lanjut.