

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil pengujian dan analisis terhadap dua algoritma *load balancing*, yaitu *Weighted Round Robin* (WRR) dan *Weighted Least Connection* (WLC), dapat disimpulkan bahwa keduanya memiliki keunggulan masing-masing yang sesuai dengan karakteristik beban dan kebutuhan sistem. Penelitian dilakukan pada dua kondisi utama, yaitu kondisi *idle* dan *load testing*, dengan parameter evaluasi meliputi *Delay*, *throughput*, *Packet Loss*, *jitter*, dan *Response Time*. Untuk memberikan gambaran yang lebih jelas mengenai performa masing-masing algoritma, berikut ini disajikan uraian terperinci berdasarkan rumusan masalah penelitian.

1. Algoritma *Weighted Round Robin* (WRR) memiliki kemampuan yang baik dalam mendistribusikan beban secara proporsional berdasarkan bobot yang telah ditentukan. Keunggulan WRR terlihat pada parameter *throughput*, yang menunjukkan kemampuan tinggi dalam mentransfer data, terutama saat sistem berada dalam kondisi beban berat. Karakteristik ini menjadikan WRR cocok untuk sistem yang membutuhkan kapasitas transmisi data yang besar dan stabil. Namun, WRR cenderung menghasilkan *delay*, *jitter*, dan *response time* yang lebih tinggi, sehingga kurang optimal digunakan pada sistem yang menuntut kecepatan respons dan kestabilan trafik secara *real-time*.

2. Algoritma *Weighted Least Connection* (WLC) unggul dalam hal responsivitas dan adaptivitas terhadap trafik yang dinamis. Dengan memanfaatkan jumlah koneksi aktif sebagai acuan distribusi beban, WLC mampu menjaga nilai *delay*, *jitter*, dan *response time* tetap rendah dan stabil. Keunggulan ini membuat WLC lebih sesuai diterapkan dalam layanan yang menuntut kontinuitas dan waktu tanggap yang cepat. Meskipun *throughput* yang dihasilkan tidak setinggi WRR, algoritma ini tetap menjaga efisiensi jaringan dan distribusi trafik secara seimbang.
3. Kedua algoritma menunjukkan hasil yang setara pada parameter *packet loss*, dengan tingkat kehilangan paket yang rendah dan stabil dalam berbagai kondisi pengujian. Secara keseluruhan, WRR lebih ideal digunakan untuk sistem dengan kebutuhan transfer data besar dan cenderung stabil, sedangkan WLC lebih tepat untuk skenario yang menuntut respons cepat dan adaptasi terhadap perubahan beban secara real-time. Pemilihan algoritma load balancing perlu disesuaikan dengan karakteristik dan prioritas sistem agar kinerja dan efisiensi layanan dapat tercapai secara optimal.
4. Kedua algoritma menunjukkan kemampuan terbaiknya Ketika diberikan beban konkuren lebih dari 1000

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, maka penulis memberikan saran, yaitu:

1. Untuk pengujian lebih lanjut, disarankan agar pengujian juga dilakukan menggunakan pendekatan ISP-level *load balancing* guna melihat performa

distribusi trafik pada skala yang lebih besar dan kompleks. Pengujian dengan ISP load balancer dapat memberikan perspektif baru terhadap pengelolaan lalu lintas multi-jalur, terutama dalam hal optimisasi bandwidth eksternal, ketersediaan koneksi internet, dan efisiensi rute data.

2. Pengujian saat ini menggunakan tiga server dengan bobot tertentu. Sebaiknya pengujian ke depan mempertimbangkan variasi jumlah server (misalnya 5 atau lebih) dan mempertimbangkan spesifikasi *server*, serta distribusi beban yang lebih dinamis atau tidak seimbang untuk mengevaluasi adaptabilitas algoritma load balancing dalam skenario nyata.
3. Penelitian terfokus pada Proxmox *Virtual Environment* sebagai platform virtualisasi. Untuk generalisasi hasil, akan lebih kuat jika penelitian juga menguji pada platform lain seperti VMware ESXi, KVM native, atau *container-based* (Docker Swarm / Kubernetes) agar dapat melihat konsistensi performa antar lingkungan virtualisasi.
4. Selain menggunakan Apache Benchmark, disarankan menggunakan alat uji atau workload generator yang dapat meniru lalu lintas jaringan nyata seperti JMeter atau Locust. Ini akan memberikan hasil yang lebih relevan terhadap implementasi di lingkungan produksi.
5. Penelitian dilakukan menggunakan dua algoritma, disarankan menambahkan satu atau dua algoritma load balancing lain seperti *Least Response Time* atau *Consistent Hashing* sebagai pembanding. Hal ini penting untuk melihat sejauh mana performa WRR dan WLC bersaing dengan algoritma lain di berbagai skenario trafik.