BAB V KESIMPULAN dan SARAN

5.1 Kesimpulan

Berdasarkan pengujian dan analisis komparatif yang telah dilakukan terhadap algoritma Round Robin, Least Connection, dan IP Hash pada web server berbasis NGINX dalam lingkungan virtualisasi, dapat disimpulkan bahwa algoritma Round Robin menunjukkan kinerja paling superior pada kondisi lalu lintas data ringan hingga normal (100 concurrent users). Kesederhanaan mekanisme rotasi tanpa adanya beban kalkulasi status peladen (overhead) membuat algoritma ini mampu memproses permintaan dengan throughput tertinggi, yaitu 6,77 req/s, dan latensi terendah sebesar 11.783 ms dibandingkan algoritma lainnya. Namun, ketika sistem dihadapkan pada skenario beban menengah (500 concurrent users), algoritma Least Connection terbukti memiliki stabilitas dan reliabilitas tertinggi. Mekanisme distribusi beban yang mempertimbangkan jumlah koneksi aktif secara real-time sangat efektif dalam mencegah penumpukan antrean pada satu node server. Hal ini dibuktikan dengan tingkat kesalahan (Error Rate) yang berhasil ditekan ke angka 43,81%, jauh lebih rendah dibandingkan Round Robin yang melonjak tajam hingga 89,91% maupun IP Hash sebesar 57,61%. Temuan ini membuktikan bahwa algoritma dinamis jauh lebih adaptif dalam menjaga ketersediaan layanan saat terjadi fluktuasi trafik yang masif.

Di sisi lain, algoritma IP Hash menunjukkan kinerja yang paling konservatif dengan tingkat throughput efektif terendah dan latensi tertinggi di hampir seluruh skenario pengujian normal. Mekanisme hashing yang membatasi distribusi trafik secara spesifik demi mempertahankan persistensi sesi (session persistence) justru menciptakan hambatan (bottleneck) kinerja. Penggunaan CPU yang rendah pada algoritma ini (khususnya pada beban ringan) bukanlah penanda efisiensi komputasi, melainkan dampak langsung dari rendahnya volume permintaan aktual yang berhasil disalurkan dan diproses oleh sistem.

Selanjutnya, pada kondisi beban puncak yang ekstrem (1.000 concurrent users), ditemukan bahwa seluruh algoritma mengalami kelumpuhan sistem (system failure) dengan tingkat kesalahan mendekati angka absolut (di atas 99,7%). Lonjakan nilai throughput dan penurunan waktu respons secara drastis pada fase ini dikonfirmasi sebagai angka semu, yang terjadi akibat server yang telah kelebihan beban (overload) menolak koneksi secara instan (connection refused atau timeout). Hal ini mengindikasikan bahwa pada titik saturasi infrastruktur, pemilihan jenis algoritma load balancing tidak lagi memberikan dampak yang signifikan. Kapasitas pemrosesan perangkat keras (CPU dan RAM) serta batas antrean sistem telah terlampaui, yang secara tegas menegaskan batasan fisik maksimal dari lingkungan uji yang digunakan.

5.2 Saran

Berdasarkan temuan tersebut, disarankan bagi administrator sistem untuk memilih algoritma berdasarkan karakteristik trafik aplikasi yang dikelola. Penggunaan algoritma *Round Robin* sangat direkomendasikan untuk aplikasi dengan beban statis dan relatif rendah karena efisiensi dan kecepatannya, sedangkan untuk aplikasi skala menengah hingga besar dengan pola trafik fluktuatif, sebaiknya beralih ke algoritma *Least Connection* guna meminimalisir risiko *downtime* dan meratakan beban antrean. Implementasi *IP Hash* sebaiknya dibatasi hanya pada aplikasi yang benar-benar mewajibkan konsistensi sesi tanpa adanya penyimpanan sesi terpusat, atau digantikan dengan mekanisme *sticky cookie* pada lapisan aplikasi untuk menghindari ketimpangan beban akibat distribusi alamat IP klien yang tidak merata.

Untuk pengembangan penelitian selanjutnya, disarankan menggunakan topologi infrastruktur fisik yang terpisah atau layanan komputasi awan (*cloud computing*) nyata guna menghilangkan bias pengukuran akibat pembagian sumber daya (*resource sharing*) pada virtualisasi lokal. Studi komparasi juga perlu diperluas dengan melibatkan algoritma lanjutan seperti *Weighted Round Robin* atau *Weighted Least Connection* untuk menguji efektivitas distribusi beban

pada peladen dengan spesifikasi perangkat keras heterogen. Terakhir, penelitian masa depan diharapkan dapat memasukkan analisis keamanan siber dengan menguji ketahanan algoritma terhadap serangan seperti *Distributed Denial of Service* (DDoS), untuk mengetahui algoritma mana yang paling tangguh dalam mempertahankan ketersediaan layanan di bawah tekanan serangan jahat.