BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil pengujian dan analisis yang telah dilakukan dari penelitian ini, disimpulkan bahwa:

1. Round Robin menunjukkan kinerja yang lebih stabil dalam menangani beban rendah hingga menengah, dengan waktu respons konsisten terutama pada 400 pengguna, menjadikannya efektif untuk beban trafik tidak terlalu tinggi. Sebaliknya, Least Connection unggul dalam menangani beban tinggi, meskipun waktu respons meningkat seiring bertambahnya jumlah koneksi.
2. Round Robin secara konsisten memproses 560 hingga 570 permintaan per detik dalam kondisi beban tinggi, menunjukkan stabilitas throughput yang baik. Least Connection mengalami fluktuasi throughput antara 405 hingga 574 permintaan per detik, tetapi menunjukkan peningkatan throughput yang signifikan saat permintaan meningkat dari 1000 ke 2000.
3. Keduanya, Round Robin dan Least Connection, menunjukkan peningkatan penggunaan CPU seiring bertambahnya jumlah koneksi. Namun, Least Connection memperlihatkan variasi penggunaan CPU yang lebih signifikan antara VM 1 dan VM 2, terutama pada tingkat permintaan yang lebih tinggi.
4. Optimasi server, seperti peningkatan timeout, terbukti efektif dalam menghilangkan kegagalan transaksi yang sebelumnya terjadi.
5. Hasil optimasi menunjukkan peningkatan signifikan dalam ketersediaan dan keandalan layanan. Semua pengujian setelah optimasi berjalan tanpa kegagalan, bahkan pada tingkat permintaan yang lebih tinggi, yang sebelumnya menyebabkan masalah.

5.2. Saran

Berlandaskan pada temuan dan analisis yang telah dilakukan, berikut adalah beberapa saran yang dapat digunakan untuk memperbaiki dan mengembangkan penelitian ini di masa depan:

1. Uji dan bandingkan performa algoritma load balancing seperti IP Hash, Weighted Round Robin, dan Least Response Time untuk menemukan solusi yang paling optimal.
2. Lakukan pengujian pada lingkungan server yang heterogen dan dengan skenario beban yang variatif, termasuk beban bursty, untuk menilai respons algoritma dalam kondisi dinamis.
3. Lakukan analisis mendalam terhadap penggunaan memori dan bandwidth, selain CPU, untuk mengidentifikasi potensi bottleneck dan area optimasi.
4. Uji algoritma dalam durasi jangka panjang untuk mengevaluasi performa dan stabilitas seiring waktu, serta respons terhadap perubahan beban.
5. Eksplorasi integrasi algoritma load balancing dengan teknik otomasi dan kecerdasan buatan untuk memilih algoritma yang optimal secara dinamis berdasarkan pola lalu lintas.