

BAB 5

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil data dan analisis yang telah dilakukan, maka kesimpulan terkait penelitian ini adalah hasil pengujian sistem LMS menunjukkan hasil yang positif. Hasil pengujian *throughput* menunjukkan nilai yang hampir mendekati satu sama lain dengan rentang 9,42 – 12,12 Mbps (kategori sangat bagus menurut standar THIPON). Selain itu, nilai *delay* dari semua jenis permintaan pada *server* LMS juga sangat bagus dengan rentang 0,7 ms hingga mendekati 1 ms. Dalam pengujian *packet loss*, nilai yang diperoleh berada dalam rentang 0,2%-0,32%, (kategori bagus). Pengujian *jitter* juga menunjukkan bahwa pada 1000 permintaan sebesar 1,6 ms, menunjukkan optimalitas *server* LMS dalam melayani permintaan tersebut. Bahkan, saat permintaan ditingkatkan, *jitter* masih tetap terjaga untuk peningkatan permintaan hingga 10000. Selain itu, hasil pengujian CPU *usage* menunjukkan bahwa setiap *server* LMS mendapatkan beban kerja yang sesuai dengan permintaan yang dikirimkan, tidak ada *server* yang menganggur. Dengan demikian, sistem LMS mampu mengelola permintaan dengan baik dan memanfaatkan sumber daya secara efisien.

B. Saran

Karena terdapat beberapa kekurangan dalam penelitian ini, maka diajukan beberapa saran yang dapat dilakukan untuk melakukan pengembangan pada penelitian ini.

- 1) Menambah jumlah *server* LMS bertujuan untuk meningkatkan kualitas pelayanan karena dengan jumlah *server* yang lebih banyak, sistem dapat menangani lebih banyak permintaan dari pengguna dengan lebih cepat dan lebih efisien. Hal ini dapat meningkatkan *throughput*, mengurangi *jitter* dan *delay* serta *packet loss*.
- 2) Untuk pengujian dapat ditambahkan variasi seperti jumlah permintaan dan jumlah *request* per detik supaya lebih maksimal dalam mendapatkan data yang diperoleh pada saat pengujian.

- 3) Menerapkan replikasi *container* yang dapat mendistribusikan *docker container* ke bebrapa *host* atau mesin virtual untuk meningkatkan ketersediaan aplikasi. Replikasi *container* dapat dilakukan dengan menggunakan fitur *orchestration* pada *Docker*, seperti *Docker Swarm* atau *Kubernetes*.