

BAB V PENUTUP

A. KESIMPULAN

Berdasarkan hasil data dan analisis yang telah dilakukan, maka kesimpulan terkait penelitian ini adalah sebagai berikut.

- 1) Hasil pengujian *throughput* pada saat 1000 permintaan yang dikirimkan sebesar 1012 kbps. Terjadi penurunan *throughput* ketika permintaan ditingkatkan menjadi 1500, 2000, 2500 dan 3000 permintaan. Penurunan *throughput* terjadi disebabkan meningkatnya jumlah permintaan ke LMS, sehingga *server* LMS akan menerima lebih banyak permintaan yang akan menurunkan performa *server* dan otomatis akan menurunkan *throughput* yang dihasilkan.
- 2) Hasil pengujian *response time* menunjukkan adanya peningkatan waktu yang dibutuhkan ketika jumlah permintaan ditingkatkan. Ketika meningkatnya jumlah permintaan ke HTTP, maka menyebabkan jumlah permintaan ke *server*. Hal ini menyebabkan *server* menjadi lebih sibuk dan waktu yang dibutuhkan menjadi lebih besar. Sebagai contoh pada saat 1000 permintaan waktu yang dibutuhkan apabila tanpa menggunakan *load balancing* sebesar 86.166 s. Hal ini menunjukkan bahwa dengan penggunaan *load balancing*, *response time* yang dibutuhkan menjadi rendah dibandingkan tanpa *load balancing*.
- 3) Hasil pengujian *packet loss* ketika 1000 permintaan sebesar 0% sehingga *server* LMS masih sangat optimal ketika melayani 1000 permintaan. Ketika permintaan ditingkatkan ke 1500 permintaan *packet loss* sebesar 29.90%, hal ini menunjukkan bahwa ketika permintaan ditingkatkan menjadi 1500 permintaan *server* LMS tidak optimal dalam melayani permintaan. Kemudian menjadi sangat tidak optimal ketika permintaan kembali ditingkatkan menjadi 2000, 2500 dan 3000 permintaan karena *packet loss* sebesar 42.92%, 50.88, dan 56.46%. Hasil pengujian menunjukkan bahwa hasil pengujian *packet loss* masih sangat optimal ketika 1000 permintaan dan menjadi sangat tidak optimal ketika permintaan ditingkatkan. Peningkatan permintaan mengakibatkan jumlah permintaan HTTP juga meningkat. Hal ini mengakibatkan *server* menjadi lebih sibuk dan jumlah permintaan yang ditolak lebih besar.

- 4) Hasil pemantauan CPU *usage* ketika 1000 permintaan sebesar 39.76% dan mengalami peningkatan ketika permintaan ditingkatkan menjadi 1500, 2000, 2500, dan 3000 permintaan sebesar 41.45%, 42.36, 43.18, 46.63. Hasil pemantauan menunjukkan bahwa setiap *server* LMS mendapatkan beban kerja ketika permintaan dikirimkan dan tidak ada *server* yang *idle* (menggantung).
- 5) Hasil pengujian menunjukkan bahwa algoritma *least connection* masih belum stabil ketika 1500, 2000, 2500 dan 3000 permintaan dikirimkan. Hal ini disebabkan *least connection* tidak memperhitungkan kemampuan *server*, jarak antara klien dan *server* serta faktor lainnya.

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 yang akan digunakan, sehingga jumlah *server* yang akan melayani permintaan akan lebih banyak. Dengan menambah jumlah *server* akan meningkatkan spesifikasi CPU, storage dan RAM pada *server* yang akan meningkatkan kinerja *server*. Hal ini dapat dilakukan dengan logis, karena *cloud computing* dapat meningkatkan kapasitas *server* tanpa perlu menambah perangkat fisik.
- 2) Menerapkan *Kolla Ansible* untuk *deploy Openstack* yang memiliki kelebihan *scalable, fast, reliable, dan upgradable*. *Openstack* dijalankan pada *container* yang dapat menyimpan data secara persistant dan dapat direstore untuk melindungi dari kesalahan.
- 3) Menggunakan *service load balancing* yang terdapat *Openstack*, yaitu *Openstack Octavia*. Fitur penskalaan *horizontal* pada *Openstack Octavia* akan disesuaikan dengan permintaan ini yang membedakan *Octavia* dengan *load balance* lainnya, sehingga *Octavia* merupakan solusi yang tepat untuk *cloud computing*.
- 4) Menerapkan algoritma *weight least connection* yang menentukan beban berdasarkan bobot *server*. Sehingga *server* yang akan menangani permintaan lebih dulu adalah *server* yang memiliki bobot lebih besar. Sehingga algoritma ini akan lebih baik dalam mendistribusikan beban.