

## **BAB 5**

### **PENUTUP**

#### **5.1 KESIMPULAN**

Berdasarkan hasil dari pembahasan mengenai analisis unjuk kerja *load balancing web server* menggunakan virtualisasi berbasis *container docker swarm* maka dapat diperoleh beberapa kesimpulan sebagai berikut:

1. Hasil pengukuran parameter *response time* pada *load balancing web server* menggunakan algoritma *round robin* menghasilkan nilai yang lebih baik daripada algoritma *least connection*. Yaitu 6,28 *second* pada 500 koneksi dengan 100 konkurensi, 19,83 *second* pada 2000 koneksi dengan 400 konkurensi, dan 30,19 *second* pada 5000 koneksi dengan 1000 konkurensi.
2. Hasil pengukuran parameter *CPU Utilization* pada *node manager* dan *node worker* menggunakan algoritma *round robin* menghasilkan nilai yang lebih tinggi dibandingkan dengan algoritma *least connection*. Yaitu *node manager* 62,56 % dan *node worker* 61,99 % pada 500 koneksi dengan 100 konkurensi, 63,00 % *CPU Utilization node manager* dan 62,15 % *node worker* pada 2000 koneksi dengan 400 konkurensi, 78,55 % *CPU Utilization node manager* dan 62,32 % *node worker* pada 5000 koneksi dengan 1000 konkurensi.
3. Hasil pengukuran parameter *memory usage* pada *node manager* dan *node worker* menggunakan algoritma *round robin* menghasilkan nilai yang lebih rendah dibandingkan dengan algoritma *least connection*. Yaitu *node manager* 344,10 MB dan *node worker* 319,72 MB pada 500 koneksi dengan 100 konkurensi, 543,77 MB *memory usage node manager* dan 523,38 MB *node worker* pada 2000 koneksi dengan 400 konkurensi, 602,44 MB *memory usage node manager* dan 586,70 MB *node worker* pada 5000 koneksi dengan 1000 konkurensi.
4. Algoritma *round robin* bekerja lebih cepat dan efisien daripada algoritma *least connection*.

#### **5.2 SARAN**

Dengan melihat hasil penelitian ini, maka saran yang diajukan untuk penelitian selanjutnya dengan topik yang sama adalah:

1. Penggunaan aplikasi *virtualbox* memakan *resource* yang besar, sehingga disarankan menggunakan komputer dengan spesifikasi yang tinggi.
2. Sistem yang dibuat dapat diimplementasikan menggunakan *server* nyata.
3. *Node manager* dan *node worker* dapat ditambahkan pada satu kluster yang sama atau dengan kluster yang berbeda.
4. Variasi jumlah koneksi dan banyaknya percobaan dapat ditambahkan agar variasi data dapat diperoleh dengan lebih detail.