

BAB V

KESIMPULAN DAN SARAN

Berdasarkan hasil percobaan lapangan yang sudah penulis lakukan. Penulis menyimpulkan dan mengajukan saran–saran yang dapat diberikan pada penelitian berikutnya yang mengenai penelitian yang telah dilakukan dan ditulis pada laporan ini.

5.1. Kesimpulan

Kesimpulan yang dapat diambil dari penelitian terkait pembuatan sistem keamanan web server menggunakan arsitektur containerization terhadap serangan denial of service adalah sebagai berikut:

1. Arsitektur *containerization* bisa digunakan untuk membangun layanan web server yang aman dari serangan Denial of Service.
2. Load balancing yang menggunakan algoritma *round robin* dapat menyebarkan beban data kepada semua *node*, sehingga tidak ada server yang mengalami *overload*.
3. Kontainer dapat digunakan sebagai tempat untuk mengemas konfigurasi, kode, pustaka, dan *runtime*, mengemasnya dan menjalankannya dimana saja.
4. Dibutuhkan beberapa paket tambahan untuk dapat menjalankan arsitektur *containerization* yaitu, Docker, Prometheus, Grafana, cAdvisor dan Node Exporter

5. Penggunaan Arsitektur Containerization membutuhkan Swarm untuk saling menghubungkan antar node atau server sehingga dapat saling terkoneksi.
6. Hasil perbandingan dan pengujian layanan yang dilakukan dengan serangan denial of service membuktikan beban memory yang digunakan web server dengan arsitektur containerization lebih sedikit dibandingkan web server konvensional
7. Hasil pengujian juga dilakukan pada kinerja perangkat keras yaitu CPU. Web server konvensional lebih berat hingga menyentuh 0,00015% dibandingkan web server dengan arsitektur containerization yang paling tinggi hanya menyentuh angka 0,00010%.
8. Hasil pengujian pada beban prosesor juga membuktikan bahwa beban prosesor pada web server konvensional lebih berat daripada web server dengan arsitektur containerization.

5.2. Saran

Adapun saran untuk penelitian selanjutnya dari penulis adalah sebagai berikut:

1. Gunakan arsitektur containerization untuk membangun web server, ini dimaksudkan membagi kinerja *resource* yang ada.
2. Gunakan arsitektur containerization untuk melakukan perawatan pada web aplikasi karena mempermudah kolaborasi antar node.