

BAB V

KESIMPULAN

5.1 Kesimpulan

Hasil yang didapatkan oleh penulis dari penelitian ini ialah konfigurasi *load balancing* dapat memberikan optimasi dalam respon pada *webserver*. Dari konfigurasi yang dilakukan oleh penulis kedua algoritma bekerja sesuai dengan penggunaannya dan memberikan daya salur beban yang cukup optimal.

Kedua algoritma bekerja dengan membagi beban *request* berdasarkan alur algoritmanya masing masing untuk mencapai titik optimal dari sebuah sistem jaringan *load balance*. Dari hal tersebut dicapai hasil Analisa yaitu:

- *Round Robin* lebih unggul serta konsisten hampir di semua metrik.
- Tidak konsistennya *least connection* terlihat pada percobaan pertama dan kedua pada 3000 *thread* dimana keunggulan tersebut disebabkan anomali yang dapat dilihat dari data yang melompat terlalu tinggi dibanding hasil percobaan sebelumnya.
- *Round Robin* memiliki algoritma yang lebih simpel dimana hanya melakukan pembagian secara siklik tanpa melakukan monitoring jumlah request yang masuk pada tiap *server*.

Dari keunggulan diatas dalam algoritma *round robin* memiliki beberapa kekurangan yaitu :

- Bekerja paling optimal hanya dalam penanganan request yang seragam dan tidak dinamis seperti pada percobaan ini.
- Bekerja paling optimal pada *server* dengan spesifikasi yang seragam, dikarenakan hanya membagi secara siklik tanpa melihat koneksi atau antrian yang sedang terjadi di tiap servernya.

Sedangkan dalam algoritma *least connection* memiliki beberapa kelebihan yaitu :

- Dapat bekerja dengan baik dalam pembagian *load* yang lebih dinamis.
- Lebih toleran pada spesifikasi *hardware* perangkat *server* dikarenakan pembagian *load* sesuai dengan kemampuan dalam menangani antrian.

Oleh karena itu *Least Connection* dapat bekerja dengan lebih baik jika spesifikasi server yang tidak seragam ataupun di skenario beban kerja yang bervariasi, dimana pada test ini dilakukan dengan spesifikasi seragam yang mana *least connection* kurang cocok, namun masih bisa mengimbangi yang membuat algoritma tetap dapat diandalkan walaupun dalam kondisi *traffic* yang seragam.

Kedua algoritma tidaklah berbeda jauh dalam hasil yang didapatkan pada penelitian ini, namun *least connection* dapat menyesuaikan *load* sesuai dengan penggunaan *node server* secara *realtime*. Adanya algoritma ini bertujuan untuk penggunaan pada *node server* dengan spesifikasi per-*node* yang berbeda, dimana salah satu *node* atau beberapa *node* memiliki spesifikasi lebih rendah dibanding *node* yang lainnya, menjadikan utilisasi *node* tidak berjalan secara optimal. Sedangkan pada algoritma *round robin* bekerja paling baik jika diterapkan pada *node* dengan spesifikasi seragam serta *traffic* yang seragam, dimana *least connection* lebih toleran terhadap *traffic* yang dinamis.

5.2 Saran

Berdasarkan kesimpulan sebelumnya, penulis menyarankan untuk menyediakan *load balancing* bila memiliki *web* atau *server* yang memiliki *traffic* yang tinggi, terutama seperti *website-website* krusial seperti perbankan, pemerintahan, ataupun perguruan.

Untuk penelitian selanjutnya, penulis menyarankan untuk melakukan pengujian dengan *traffic* yang dinamis dan *load* yang lebih tinggi serta penggunaan *environment* yang beragam meliputi penggunaan di lingkungan yang seragam maupun tidak seragam.