

BAB 5

PENUTUP

5.1 KESIMPULAN

1. Waktu Respon: Openlitespeed menunjukkan waktu respon yang lebih baik dibandingkan dengan Nginx pada sebagian besar jumlah koneksi yang diuji. Misalnya, pada jumlah koneksi 1000, Nginx memiliki waktu respon sebesar (0,162 s), sedangkan Openlitespeed memiliki waktu respon sebesar (0,176 s). Perbedaan ini menunjukkan bahwa Openlitespeed mampu memberikan respons yang lebih cepat dalam merespon permintaan pengguna.
2. CPU *usage*: Openlitespeed memiliki CPU *usage* yang lebih rendah dibandingkan Nginx pada setiap jumlah koneksi yang diuji. Kenaikan CPU *usage* sejalan dengan jumlah koneksi yang harus diakomodasi oleh web server
3. *Memory usage*: *memory usage* Openlitespeed relative lebih rendah sedikit dibandingkan Nginx, walaupun selisihnya relative kecil. Kenaikan *memory usage* sejalan dengan jumlah koneksi yang harus diakomodasi oleh web server.
4. *Throughput*: Openlitespeed menunjukkan *throughput* yang lebih tinggi dibandingkan dengan Nginx pada setiap jumlah koneksi yang diuji. Misalnya, pada jumlah koneksi 1000, Nginx memiliki *throughput* sebesar (6,033 Mbps), sedangkan Openlitespeed memiliki *throughput* sebesar (11,275 Mbps) dan juga kedua web server termasuk kategori sangat bagus sesuai standar TIPHON.
5. *Packet Loss* : Nginx memiliki *packet loss* yang relatif lebih tinggi sedikit dibandingkan Openlitespeed pada setiap jumlah koneksi yang diuji. Walaupun perbedaannya sangat kecil, namun kedua web server termasuk kategori sangat bagus sesuai standar TIPHON.
6. *Delay*: variasi dalam tingkat *delay* antara Nginx dan Openlitespeed pada setiap jumlah koneksi yang diuji relatif serupa. Tidak terdapat

perbedaan yang signifikan dalam variasi *delay* antara kedua web server dan juga termasuk kategori sangat bagus sesuai standar TIPHON.

5.2 SARAN

1. Pada pengembangan selanjutnya, bisa dilakukan pengujian dengan jumlah koneksi semakin bervariasi, sehingga data pengujian semakin banyak, karena belum ada penelitian tersebut.
2. Pada pengembangan selanjutnya, mungkin bisa dicoba dalam lingkungan kubernetes dengan *container runtime* yang lain.
3. Pada pengembangan selanjutnya, perlu ditambahkan *monitoring* kondisi kesemua pod yang meliputi *CPU usage*, *memory usage*, kondisi *traffic* pada pod yang ada di kubernetes, sehingga data tidak hanya berpacu pada hasil *tool* penelitian ini