

## BAB 5

### KESIMPULAN

#### 5.1 Kesimpulan

1. Secara umum, HAProxy menunjukkan kinerja yang lebih unggul dibandingkan nginx dalam menangani beban koneksi konkuren, meskipun keunggulan kinerja tersebut diiringi dengan penggunaan RAM yang secara konsisten lebih tinggi (22%) dibanding nginx. Pada protokol HTTP/1.1, HAProxy menghasilkan rata-rata *requests per second* (RPS) 4,3% lebih tinggi dibanding nginx, di mana keunggulan kinerjanya lebih signifikan pada rentang beban 32 hingga 316 koneksi dengan RPS 9-12% lebih tinggi. Keunggulan HAProxy semakin dominan pada protokol HTTP/2 dengan rata-rata RPS 13% lebih tinggi, yang secara spesifik mencatatkan keunggulan RPS sebesar 15-24% lebih tinggi pada rentang beban 10 hingga 316 koneksi. Sementara itu, pada protokol HTTP/3, perbedaan kinerja yang paling signifikan terlihat dari aspek stabilitas, di mana HAProxy menunjukkan ketahanan yang jauh lebih baik dibanding nginx pada beban tinggi. Hal ini ditunjukkan dari hasil pengujian pada tingkat beban 1000 koneksi konkuren, di mana HAProxy secara rata-rata hanya mengalami *error rate* 89% lebih rendah dibanding nginx.
2. Pada pengujian dengan simulasi jaringan terganggu, seluruh versi protokol di setiap *load balancer* mengalami penurunan kinerja. HTTP/1.1, seperti yang diharapkan, menjadi yang paling terpengaruh karena terbatas pada *singlestream*. Sementara itu, pada HTTP/2, nginx dan HAProxy

menunjukkan kinerja yang relatif sebanding pada kasus pengujian ini. Kemudian, pada HTTP/3, meskipun dukungan protokol ini pada kedua *load balancer* masih dalam status eksperimental, implementasi pada HAProxy terbukti jauh lebih andal dibanding nginx. Pada seknario pengujian dengan simulasi *latency* (25 ms dan 50 ms), implementasi HTTP/3 pada HAProxy mampu menghasilkan RPS rata-rata 86% lebih tinggi dan *response time* 38% lebih cepat dibanding nginx. Selanjutnya, pada kondisi jaringan dengan simulasi *packet loss* 2%, HAProxy kembali menunjukkan keunggulan atas nginx dengan menghasilkan rata-rata RPS 91% lebih tinggi dan *response time* 47% lebih cepat. Keunggulan HAProxy tersebut juga konsisten pada tingkat *packet loss* 4%. Selain itu, pada kondisi yang sama, kinerja HTTP/3 pada HAProxy juga lebih baik dibanding HTTP/2 dengan rata-rata RPS 16% lebih tinggi pada pengujian *multistream* dan bahkan 160% lebih tinggi pada pengujian *singlestream*. Hal ini, kemungkinan disebabkan karena desain implementasi QUIC pada HTTP/3 yang memang dirancang untuk mengatasi masalah *head-of-line blocking* yang melekat pada protokol TCP. Namun, hal yang sama tidak teramati pada nginx, implementasi HTTP/3-nya masih menunjukkan kinerja yang lebih rendah dibanding HTTP/2-nya.

## 5.2 Saran

1. Penelitian ini hanya berfokus pada *load balancer* nginx dan HAProxy. Penelitian selanjutnya dapat diperluas dengan menyertakan perangkat lunak *load balancer* sumber terbuka lainnya, seperti Envoy dan Traefik

yang menawarkan integrasi *native* dengan *container orchestrator*, untuk memberikan perbandingan yang lebih komprehensif dalam menangani protokol HTTP modern pada layanan web berbasis *container* dan lingkungan *cloud*.

2. Mengingat dukungan QUIC dan HTTP/3 pada nginx dan HAProxy masih dalam tahap eksperimental, penelitian serupa dapat dilakukan di masa mendatang saat implementasi protokol tersebut sudah lebih matang dan stabil.
3. Mempertimbangkan bahwa penelitian ini hanya mengevaluasi kinerja dengan algoritma *congestion control default* (Cubic), penelitian selanjutnya dapat dilakukan untuk membandingkan implementasi algoritma lain yang lebih modern seperti BBR. Terutama karena fleksibilitas protokol QUIC yang memungkinkan penyesuaian di tingkat *user space*, penting untuk menentukan konfigurasi *congestion control* yang paling efisien bagi *load balancer* HTTP/3.