

BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan analisis hasil penelitian yang telah dilakukan, maka dapat disimpulkan bahwa.

1. Untuk melakukan pengujian performa dari aplikasi *chatting* yang menggunakan *websocket* maupun *webtransport*, pengujian dapat dilakukan dalam beberapa langkah dan skenario yang telah dirancang. Pertama, lingkungan pengujian diatur dan diisolasi dalam docker container, untuk memastikan bahwa penggunaan resource dapat dipantau dengan baik dan tidak terganggu oleh aplikasi lain. Pengujian fungsionalitas dan antarmuka aplikasi dilakukan menggunakan metode Black Box Testing, yang hasilnya menunjukkan bahwa kedua aplikasi memenuhi standar fungsionalitas yang diharapkan. Untuk pengujian performa, metode Load Testing diterapkan dengan beberapa skenario uji. Karena terbatasnya tools yang mendukung pengujian load testing dari metode *webtransport*, skenario uji diimplementasikan menggunakan kode program yang dimodifikasi dari penelitian sebelumnya oleh Matt Tomasetti, sehingga mendukung penggunaan *websocket* dan *webtransport*.
2. Berdasarkan penelitian yang telah dilakukan, penggunaan *websocket* dalam pembuatan aplikasi *chatting* memiliki performa yang lebih baik dibanding penggunaan *webtransport*. *Websocket* memiliki waktu koneksi yang 483% lebih cepat dibanding *webtransport* untuk terhubung ke 10.000 user, *websocket* membutuhkan 3938ms dan *webtransport* membutuhkan waktu 19036ms. *Websocket* juga memiliki response time yang lebih rendah dan terlihat makin signifikan ketika jumlah konkuren user mencapai 500 dan 1000 user. Pada jumlah pengguna yang rendah (100 dan 200 pengguna), *websocket* menunjukkan waktu respons yang lebih cepat dengan masing-masing 410 ms dan 849 ms dibandingkan

webtransport yang memiliki waktu respons 1833 ms dan 3112 ms. Ketika jumlah pengguna meningkat menjadi 500 dan 1000, perbedaan waktu respons antara kedua teknologi ini semakin signifikan. *websocket* tetap menunjukkan kinerja yang lebih baik dengan waktu respons 2167 ms dan 4355 ms, sementara *webtransport* menunjukkan peningkatan waktu respons yang drastis menjadi 8166 ms dan 18075 ms. Selain itu, dalam hal penggunaan *resource* server, *websocket* menggunakan *resource* server yang lebih sedikit dibanding *webtransport* dengan penggunaan CPU tertinggi di 1000 konkuren user berada di angka 74% untuk *websocket* dan 130% untuk *webtransport* sedangkan penggunaan memory tertinggi berada di angka 241 MiB untuk *websocket* dan 551 MiB untuk *webtransport*. Hal ini bisa terjadi karena *webtransport* yang masih dalam tahap pengembangan, sehingga implementasinya belum optimal dan di Node.js pun *webtransport* belum disupport secara resmi hingga skripsi ini dibuat.

5.2 Saran

Berdasarkan hasil penelitian yang telah dijabarkan sebelumnya, maka saran yang dapat diberikan pada penelitian selanjutnya adalah mengembangkan pengujian dengan menggunakan metode komunikasi real time lain atau mengkombinasikan metode *websocket* maupun *webtransport* dengan metode yang lain. Karena saat skripsi ini dibuat, *webtransport* masih dalam tahap pengembangan, untuk penelitian selanjutnya bisa menggunakan implementasi protokol *webtransport* yang lebih baik atau melakukan pengujian *webtransport* dan *websocket* menggunakan objek penelitian yang lain seperti streaming video real-time dan game interaktif. Selain itu, dapat juga dilakukan pengujian performansi dengan metode pengujian yang lain serta pada skala yang lebih besar atau mendekati kasus nyata.