

BAB V

SIMPULAN, IMPLIKASI, DAN REKOMENDASI

5.1 Simpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Secara keseluruhan, performa REST API pada *framework* Express lebih unggul dibandingkan Next.js dalam pengujian *load*, *spike*, dan *stress testing*. Express lebih unggul dibandingkan Next.js pada aspek *response time*, *throughput*, *CPU usage*, dan *memory usage*. Pada aspek *response time*, Express lebih cepat pada 14 pengujian dari 15 pengujian yang dilakukan. Pada aspek *throughput*, Express lebih unggul pada 11 pengujian dari 15 pengujian yang dilakukan. Pada aspek *CPU usage*, Express lebih unggul pada 11 pengujian dari 15 pengujian yang dilakukan. Terakhir, pada aspek *memory usage*, Express mengungguli Next.js pada seluruh pengujian. Oleh karena itu, berdasarkan hasil pengujian yang telah dilakukan, didapati bahwa implementasi REST API pada *framework* Express memiliki performa yang lebih baik dibandingkan pada *framework* Next.js.
2. Hasil pengujian dalam penelitian ini menyatakan bahwa implementasi REST API pada *framework* Express lebih unggul dibandingkan Next.js. Hasil pengujian menyatakan bahwa Express lebih baik dibandingkan Next.js pada aspek *response time*, *throughput*, *CPU usage*, dan *memory usage*. Adapun hasil analisa yang diperoleh adalah sebagai berikut:
 - a. *Response time* Express lebih cepat dibandingkan Next.js pada mayoritas pengujian dipengaruhi oleh perbedaan cara kerja *framework*, terutama dalam hal mempersiapkan koneksi ke database. Pada penelitian ini, MongoDB digunakan sebagai database. Hal ini dilakukan untuk mensimulasikan REST API yang berinteraksi dengan database. Dalam *framework* Express, ketika aplikasi yang menggunakan Express dijalankan, Express akan langsung melakukan koneksi ke database. Hal ini menyebabkan ketika klien melakukan permintaan HTTP ke API pada aplikasi Express yang mengharuskan mengambil data dari database, aplikasi Express dapat langsung mengambil data tersebut oleh karena telah terdapat

koneksi antara aplikasi dengan database. Sedangkan pada aplikasi Next.js, koneksi pada database dilakukan ketika klien pertama kali melakukan permintaan HTTP ke server. Hal ini mengakibatkan response time aplikasi Next.js lebih lambat pada saat klien melakukan permintaan HTTP ke server oleh karena aplikasi perlu mempersiapkan koneksi dengan database.

- b. *Throughput* yang dimiliki oleh Express lebih besar dibandingkan Next.js pada mayoritas pengujian dapat dipengaruhi oleh karakteristik dari *framework* Express. *Framework* Express merupakan *framework* yang ringan dan minimalis serta dibuat murni untuk pengembangan aplikasi *back-end*. Sedangkan Next.js merupakan *full-stack framework* yang dibuat untuk pengembangan aplikasi secara keseluruhan, baik itu pengembangan sisi *front-end* maupun *back-end*. Oleh karena itu, *throughput* Express lebih unggul dibandingkan Next.js pada pengujian dalam penelitian ini dapat dipengaruhi oleh ringannya *framework* Express.
- c. *CPU usage* dan *memory usage* dari Express lebih baik dibandingkan oleh Next.js oleh karena karakteristik *framework* Express yang ringan dan minimalis. Hal ini menyebabkan Express memakan sumber daya yang lebih sedikit dibandingkan Next.js yang merupakan *full-stack framework*.

5.2 Implikasi

Implikasi dari penelitian ini adalah performa implementasi REST API pada Express lebih baik dibandingkan pada Next.js. Hal ini dapat dilihat dari hasil pengujian *load*, *spike*, dan *stress testing* yang telah dilakukan dalam penelitian ini. Express lebih unggul dibandingkan Next.js pada aspek *response time*, *throughput*, *CPU usage*, dan *memory usage*. Berdasarkan hal tersebut, jika *developer* ingin mengembangkan REST API yang menggunakan metode HTTP GET, POST, PUT, atau DELETE serta memiliki performa *response time*, *throughput*, *CPU usage*, dan *memory usage* yang baik, maka Express dapat menjadi *framework* yang tepat untuk digunakan. Sedangkan Next.js dapat dipilih jika *developer* ingin mengembangkan REST API yang menggunakan metode HTTP GET, POST, PUT, dan DELETE secara bersamaan serta memiliki *CPU usage* yang rendah. Diharapkan penelitian

ini dapat menjadi bahan pertimbangan bagi pengembang dalam memilih *framework* ketika mengembangkan REST API.

5.3 Rekomendasi

Pada penelitian ini penulis menyadari masih ada banyak hal yang dapat dilakukan dalam melakukan pengujian performa implementasi REST API. Berikut ini adalah rekomendasi dari penulis untuk penelitian selanjutnya:

1. Melakukan pengujian performa implementasi REST API menggunakan metode pengujian yang berbeda dengan apa yang dilakukan dalam penelitian ini.
2. Menggunakan *framework* lain untuk implementasi REST API.
3. Menggunakan *database* SQL pada implementasi REST API.