

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan analisis hasil pengujian yang telah dilakukan terhadap Spring Boot, Flask, Express.js, Laravel FrankenPHP, dan Gin dalam menangani lima metode HTTP utama, dapat disimpulkan bahwa:

1. Pengujian dilakukan pada lima *framework backend*, yaitu Express.js, Flask, Spring Boot, Gin, dan Laravel FrankenPHP, menggunakan metode HTTP GET, GET ID, POST, PUT, dan DELETE. Setiap metode diuji dengan pendekatan beban 20 *virtual user* selama 10 menit, serta parameter performa yang diamati meliputi *response time*, *throughput*, dan *resource usage* (CPU dan Memory).
2. Spring Boot menunjukkan performa paling konsisten dan unggul pada hampir seluruh metode pengujian, yaitu GET dengan 100 data, GET ID, POST, PUT, dan DELETE. Spring Boot memiliki *response time* tercepat, *throughput* tertinggi, dan tidak menghasilkan *error* sama sekali. Meskipun penggunaan *memory* cukup besar, performanya sangat stabil dan cocok untuk pengembangan sistem skala besar dan *enterprise*.
3. Gin menjadi *framework* yang unggul pada pengujian GET dengan jumlah data sedang pada 1.000–500.000 data. *Response time* dan *throughput* pada Gin relatif sangat baik di skala tersebut, menjadikannya pilihan tepat untuk sistem yang memproses data dalam jumlah menengah. Namun, konsumsi CPU cukup tinggi dan kurang efisien untuk data berskala besar.
4. Flask memiliki keunggulan dalam *skalabilitas* data karena berhasil menangani GET dengan jumlah data hingga 2 juta dan 3 juta, yang tidak mampu dijalankan oleh *framework* lain. Namun, *response time* mencapai 29 detik dan *error rate* mencapai 93–100%. Meskipun skalabel, Flask tidak stabil untuk kebutuhan performa tinggi. Selain itu, stabilitas dari Flask kurang baik karena terdapat banyak fluktuasi dan lonjakan selama pengujian. Hal ini menunjukkan Flask lebih cocok untuk prototipe atau pengujian internal, bukan sistem produksi dengan *traffic* tinggi.

5. Express.js dan Laravel FrankenPHP memperlihatkan performa yang solid dalam penggunaan *resource*. Express.js unggul dalam efisiensi CPU, sementara Laravel FrankenPHP memiliki penggunaan *memory* paling rendah di beberapa skenario. Meskipun tidak selalu menempati posisi teratas dalam hal performa mentah, keduanya layak digunakan untuk aplikasi ringan hingga menengah yang mengutamakan efisiensi *resource*.

Secara keseluruhan, Spring Boot adalah *framework* terbaik dari segi performa umum, dengan kestabilan, kecepatan, dan efisiensi yang sangat seimbang. Gin menjadi pilihan terbaik untuk data menengah, sementara Flask unggul dalam skalabilitas mentah. Express.js dan FrankenPHP cocok digunakan pada sistem dengan kebutuhan *resource* minimal, namun tetap memiliki performa kompetitif. Hasil penelitian ini memberikan acuan bagi pengembang dalam memilih *framework* yang sesuai dengan kebutuhan dan tujuan pengembangan REST API.

5.2. Saran

Beberapa saran yang dapat penulis pertimbangkan untuk penelitian yang lebih lanjut adalah:

1. Menambahkan aspek pengujian seperti *security* dan *maintanability*.
2. Pengujian dilakukan juga terhadap versi terbaru dari masing-masing *framework*, mengingat perkembangan zaman terutama pada *framework* cukup pesat. Hal ini dilakukan agar dapat menilai apakah pembaruan tersebut berdampak signifikan terhadap performa atau *resource usage*.
3. Optimalisasi pada tiap *framework* serta *database* untuk mendapat hasil yang lebih baik pada *load* yang lebih tinggi.