

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1. Kesimpulan**

Berdasarkan hasil pengujian performa yang telah dilakukan pada aplikasi *website AM Checklist* berbasis CodeIgniter 3 dan Laravel, dapat ditarik beberapa kesimpulan utama sebagai berikut.

1. Pada skenario uji beban, CodeIgniter 3 menunjukkan performa waktu respon dan stabilitas yang jauh lebih unggul dibandingkan Laravel. Arsitektur CodeIgniter yang minimalis terbukti lebih efisien dalam menangani peningkatan beban pengguna konkuren yang tercermin dari waktu respon yang secara konsisten lebih rendah daripada Laravel. Kedua *website* menghasilkan tingkat kegagalan 0% di semua skenario menandakan fungsional *website* masih dapat berjalan dengan baik ketika dihadapkan pada beban yang diujikan.
2. Pada skenario uji skalabilitas data, penelitian ini membuktikan bahwa teknik penyajian DataTables *server-side* jauh lebih unggul dibandingkan *client-side* untuk menangani data bervolume besar. Baik pada CodeIgniter 3 maupun Laravel, pendekatan *server-side* mampu mempertahankan waktu respon yang sangat cepat dan stabil bahkan saat total data mencapai 50.000 baris. Sebaliknya, pendekatan *client-side* terbukti kurang cepat dengan waktu respon yang sangat tinggi dan terus meningkat seiring bertambahnya data. Laravel dengan penggunaan ORM yang tepat (*Eager Loading*) dapat meningkatkan efisiensi server secara drastis dan memberikan performa yang sebanding mendekati dengan *Query Builder*.

3. Dalam memilih *framework* disarankan untuk mempertimbangkan bahwa CodeIgniter unggul jika prioritas utama adalah performa kecepatan *website*, sementara Laravel menawarkan fitur yang kaya dan memiliki optimasi yang modern di mana menawarkan nilai pengembangan jangka panjang.

## 5.2.Saran

Berdasarkan penelitian dan analisis yang telah dilakukan, berikut adalah saran yang didapatkan dari penelitian ini.

1. Pengujian aplikasi dapat dilakukan dengan menggunakan metode dan *tools* yang berbeda. Hal ini bertujuan untuk memperluas perspektif dengan membandingkan dan menganalisis konsistensi hasil pengujian masing-masing metode dan *tools*.
2. Pengujian dapat ditambahkan aspek konsumsi sumber daya *hardware* seperti penggunaan *CPU* dan penggunaan memori untuk menunjukkan seberapa besar beban yang diberikan pada *server*.
3. Mengingat *framework* modern seperti Laravel menyediakan teknik optimasi seperti contohnya *caching* dengan *Redis*, disarankan melakukan pengujian setelah menerapkan teknik tersebut pada aplikasi Laravel. Pengujian ini akan menunjukkan apakah performa *framework* modern yang telah dioptimalkan mampu menyamai atau bahkan melampaui performa mentah dari *framework* minimalis di bawah beban tinggi.