

# Analisis Perbandingan Performa Website Berbasis *Laravel* dengan *CodeIgniter* menggunakan Web Server *NginX*

Muhammad<sup>1\*</sup>, M.Khadafi<sup>2</sup>, Amirullah<sup>3</sup>

<sup>1,2,3</sup> Jurusan Teknologi Informasi dan Komputer Politeknik Negeri Lhokseumawe  
Jln. B.Aceh Medan Km.280 Buketrata 24301 INDONESIA

<sup>1\*</sup>muhammad31032001@gmail.com

<sup>2</sup>M.Khadafi@pnl.ac.id

<sup>3</sup>amirullah@pnl.ac.id

**Abstrak**—Dalam mengembangkan aplikasi berbasis web para perancang membutuhkan suatu framework untuk mempermudah proses kerja. Pada perkembangannya framework telah banyak diciptakan dengan fitur-fitur yang mempermudah proses kinerja suatu aplikasi, oleh karena itu diperlukan analisis pemilihan framework yang tepat. Framework yang akan dianalisis dalam penelitian ini adalah CodeIgniter dan Laravel, analisis pengujian menggunakan inspect element load test dan Webserver Stress Tools. Pada framework CodeIgniter dan Laravel memiliki hasil performa time dan speed yang berbeda. CodeIgniter memiliki nilai rata-rata time dan speed yang lebih besar dibandingkan Laravel. Nilai time pada CodeIgniter adalah 1369 ms untuk me-load halaman dengan speed 332,68 byte/ms dengan ukuran file sebesar 484,3 Kbytes, sedangkan nilai time pada Laravel adalah 1984 ms untuk me-load halaman dengan speed 323,45 bit/ms dengan ukuran file sebesar 693,9 Kbytes Berdasarkan dari hasil pengujian load test Inspect element didapatkan kesimpulan bahwa aplikasi web yang menggunakan framework CodeIgniter lebih baik dari sisi performanya dibandingkan dengan aplikasi web yang menggunakan framework Laravel, dan menurut hasil stress test Web Server Stress Tool pada pengujian stability CodeIgniter lebih stabil dengan total error 205 user dalam 10 kali percobaan dibandingkan Laravel dengan total error 2096 user dalam percobaan yang sama. pada pengujian scalability Laravel dapat menangani permintaan load user maksimal sampai 1500 user sedangkan CodeIgniter dapat menangani load user maksimal 1000 user.

**Kata kunci**—Letakkan 5 – 6 kata kunci Anda di sini, kata kunci dipisahkan dengan koma.

**Abstract**—In developing web-based applications, designers need a framework to simplify the work process. In its development, frameworks have been created with many features that simplify the performance process of an application, therefore it is necessary to analyze the selection of the right framework. The frameworks to be analyzed in this study are Codeigniter and Laravel, test analysis using Inspect Element Load Test and Web Server Stress Tools. In the Codeigniter and Laravel frameworks, they have different performance results in time and speed. Codeigniter has a greater average time and speed value than Laravel. The time value in CodeIgniter is 1369 ms to load a page at a speed of 332,68 bytes/ms with a file size of 484,3 Kbytes, while the time value in Laravel is 1984 ms to load a page at a speed of 323,45 bytes/ms with a file size of 693,9 Kbytes, Based on the results of the Inspect element load test, it was concluded that web applications that use the CodeIgniter framework are better in terms of performance compared to web applications that use the Laravel framework, and according to the results of the stress test the web server stress tool in CodeIgniter stability testing is more stable with a total error of 205 users in 10 trials compared to Laravel with a total error of 2096 users in the same trial. in scalability testing, Laravel can handle a maximum user load request of up to 1500 users, while CodeIgniter can handle a maximum user load of 1000 users.

**Keywords**—Include 5 – 6 keywords or phrases, keywords are separated by a comm

## I. PENDAHULUAN

Dalam dunia teknologi informasi pengembangan *website* menjadi perhatian yang terus berkembang bagi organisasi dan perusahaan di dunia. Proses pengembangannya melibatkan banyak orang dibidang rekayasa perangkat lunak dan multimedia yang membuat pertumbuhan sistem berbasis *website* menjadi sangat cepat. Hingga saat ini perkiraan jumlah *website* yang tersebar di dunia mencapai lebih dari 400 miliar *website*, beberapa peneliti pun telah mencatat pertumbuhan *website* untuk tujuan komersial telah dimulai sejak tahun 1990-an [1].

Dengan adanya *website* dapat membantu perusahaan dalam menata suatu informasi yang dibutuhkan, sehingga keberadaan sistem berbasis *website* sangat menguntungkan dalam memajukan suatu perusahaan. oleh karena itu pembuatan sistem berbasis *website* meningkat jumlah permintaannya sehingga membuat para perancang perlu melakukan pekerjaan secara sistematis dan terencana agar *website* yang sedang dirancang dapat dikerjakan dan dimengerti secara cepat oleh para perancang *website*. Salah satu cara mudah dalam pembuatan suatu *website* adalah dengan menggunakan *framework PHP (Hypertext Preprocessor)*.

*Framework* merupakan sebuah kerangka kerja yang dibuat untuk memudahkan pembuatan sebuah *website* sehingga perancang tidak perlu lagi membangun suatu sistem dari nol dikarenakan *framework* memiliki komponen dan *library dan fungsi-fungsi* yang dibutuhkan para perancang *website* untuk memudahkan pembacaan kode, perencanaan, pengujian dan pemeliharaan. Dalam perkembangannya *framework* yang dipublikasi secara umum dimulai pada tahun 2004, yaitu *framework Prado 1*. Lalu disusul pada tahun 2005 terdapat 3 *framework* yang mendunia yaitu *CakePHP 1*, *Symfony 1* dan *Prado 2*, hingga pada saat ini tercatat sudah lebih dari 30 *framework* yang mendunia dengan bermacam-macam fungsi spesifik [2].

Dalam buku karya Aminudin yang berjudul cara efektif belajar *framework Laravel* menjelaskan, *Laravel* tergolong sebagai *framework PHP* pendatang baru, namun sejak awal tahun 2014, *framework laravel* berhasil menduduki peringkat pertama di jajaran *framework PHP* terpopuler, jauh mengungguli *CodeIgniter*, *yii*, *cakePHP*, *symfony*, *zend* dan *phalcon*. Selain kecepatan akses dan *update*, *Laravel* juga menawarkan *sintaks* kode yang *clean*, *simple*, dan *expressive*, sehingga memudahkan pengguna dalam mempelajarinya dengan kode yang lebih ringkas, singkat dan langsung bisa dipahami [3]. Dan menurut buku karya Ibnu Daqiqil yang berjudul *framework CodeIgniter* sebuah panduan dan *best practice* menjelaskan *CodeIgniter* juga cepat, mudah di kostumisasi dan integrasikan dengan *library* serta menjadi perhatian pembuat *PHP-Rasmus Lerdorf* “*I like CodeIgniter because it is faster, lighter and like a framework.*” [4].

Pada penelitian sebelumnya telah dilakukan perbandingan analisis *framework zend* dan *CakePHP* pada *web* perwalian didapatkan hasil bahwa performasi *framework Zend* lebih unggul dalam hal performasi karena dapat menanggung request sampai 1000 sedangkan pada *CakePHP* hanya 460 saja [5]. Pada penelitian ini penulis mencoba membandingkan *framework Laravel* dan *CodeIgniter* menggunakan *web server Nginx* dengan menganalisis perfoma melalui metode komparasi. Komparasi merupakan penelitian dengan maksud untuk melakukan perbandingan kondisi yang ada dan untuk mengetahui apakah dua atau beberapa kondisi tersebut memiliki persamaan dan perbedaan [6].

Oleh karena itu berdasarkan uraian diatas, penulis bertujuan untuk melakukan studi komparasi pada *website* yang menggunakan *framework Laravel* dan *CodeIgniter* untuk mengetahui perbedaan dalam hal performa pada masing-masing *framework* tersebut sehingga menghasilkan kesimpulan *framework* mana yang terbaik antara keduanya dalam segi performa.

## II. METODOLOGI PENELITIAN

### A. Framework

*Framework* merupakan kerangka kerja pemrograman yang mencakup *library* ataupun perpustakaan kode, model perangkat lunak, *Application Programming Interfaces (API)*, serta bermacam elemen lain yang bisa memudahkan proses pemrograman. Suatu *framework* bisa digunakan buat membangun suatu aplikasi berbasis *web* ataupun perangkat lunak [8].

### B. Web Server NginX

*Web server* adalah sistem komputer yang menyimpan, memproses, dan mengirim file *website* ke *web browser* [14]. *Nginx* adalah *web server* dengan performa yang bagus dan mempunyai beberapa fitur canggih lain yang mudah untuk *dikonfigurasi*. Alhasil *Nginx* mampu membuat *website* menjadi lebih *powerfull* dan canggih. Pada awal munculnya, *Nginx* hanya dipakai untuk *server HTTP* saja. Seiring perkembangannya teknologinya, sekarang *web server Nginx* juga dipakai sebagai *HTTP cache*, *load balancer (HTTP, TCP, dan UDP)*, dan *server proxy (IMAP, POP3, dan SMTP)* [15].

### C. Inspect Element

*Inspect element* adalah fitur *browser* yang memungkinkan Anda untuk memeriksa dan menguji kode HTML dan CSS pada suatu *website*. *inspect element* juga memungkinkan untuk mengubah sementara semua kode HTML dan CSS di halaman tersebut [7].

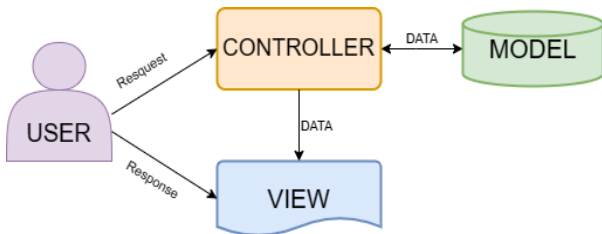
### D. Web Server Stress Tool

*Web server stress tool* merupakan suatu aplikasi atau *tools* yang digunakan untuk menguji atau bisa mengukur *performance* dari *HTTP client/server* serta dapat mengukur *performance* dari suatu *website* sehingga banyak di

gunakan untuk keperluan tes performa suatu *website*. Contoh aplikasi *web server stress tool* ialah *Paessler*. *Paessler* ini dapat digunakan pada *windows 2003*, *windows vista*, dan *windows 10* [16].

**E. Perancangan Sistem**

Perancangan sistem adalah tahap pertama yang dilakukan untuk melakukan analisis performa. Pada tahap ini akan dirancang dua aplikasi berbasis *website* yang sama-sama menggunakan arsitektur MVC (*Model Views Controller*) agar dapat dilakukan Analisis performa antara *framework* *Laravel* dan *CodeIgniter*.

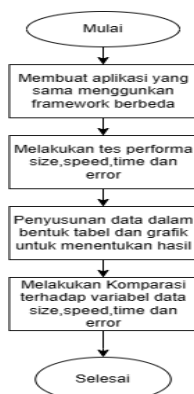


Gambar 1 Gambaran umum sistem

Berdasarkan Gambar 1, dapat dijelaskan bahwa sistem terbagi menjadi tiga bagian utama yaitu *model*, *view*, dan *controller* atau juga dapat disebut memisahkan antara *business logic* dan *presentation logic* sehingga membuat lebih terstruktur. Rancangan sistem dibangun untuk memberikan rancangan awal dari proses analisis performa sehingga dua aplikasi yang dibangun harus sama dari bentuk *view*, *controller*, dan *model* (*database*).

**F. Metode dan Variabel Penelitian**

Analisis performa antara *framework* *Laravel* dengan *framework* *CodeIgniter* dilakukan menggunakan metode penelitian komparasi untuk mengambil kesimpulan *framework* yang lebih bagus dalam segi performa antara dua *framework* yang dibandingkan. Variabel perbandingan yang digunakan ada 3, yaitu *speed*, *size*, *time scalability*, dan *stability*.



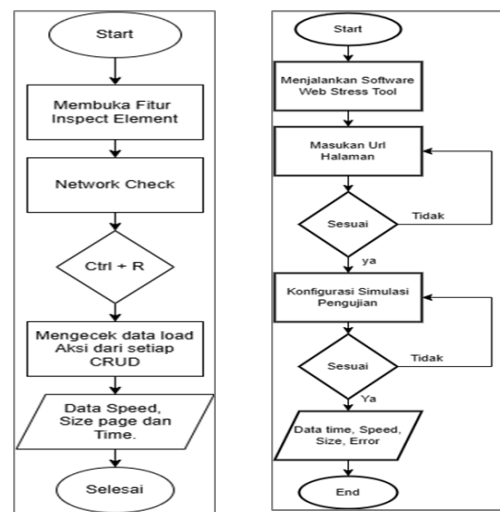
Gambar 2 Flowchart Metode Penelitian

Berdasarkan Gambar 2, tahapan metode penelitian komparasi terbagi menjadi 4 tahap, yaitu :

Pembuatan aplikasi *web* yang sama tetapi menggunakan *framework* yang berbeda. Pada tahap pertama dilakukan proses perancangan dan pembuatan dua aplikasi *web* yang sama tetapi menggunakan *framework* berbeda, yaitu *Laravel* dan *CodeIgniter*. Pengetesan performa *speed*, *scalability* dan *stability*. Setelah membuat aplikasi, dilakukan tes performa aplikasi menggunakan *Webserver Stress Tool* sehingga mendapatkan data variabel *size*, *speed*, *time* dan *error*. Penyusunan data dalam bentuk tabel dan grafik. Setelah variabel data didapatkan, data disusun dalam bentuk tabel dan grafik sehingga dapat memudahkan dalam melakukan perbandingan. Komparasi terhadap variabel data. Hasil akhir dari proses komparasi akan dijadikan kesimpulan mengenai performa *framework* yang lebih unggul antara *framework* *Laravel* dan *CodeIgniter*.

**G. Perancangan Pengujian**

Perancangan aplikasi *web* diawali dengan pembuatan alur sistem aplikasi dan dilanjutkan dengan pengujian performa. Gambar 3 adalah *flowchart* atau skenario pengujian performa secara manual pada aplikasi *web* menggunakan *framework* *Laravel* dan *CodeIgniter*. Pengetesan performa pada tugas akhir ini dilakukan menggunakan fitur dari *inspect element*. Dan pada Gambar 4 adalah *flowchart* atau skenario pengujian performa secara otomatis pada aplikasi *web* menggunakan *framework* *Laravel* dan *CodeIgniter*. Pengetesan performa otomatis dilakukan menggunakan *Web server Nginx*.



Gambar 3 Flowchart Pengujian Performa

**III. HASIL DAN PEMBAHASAN**

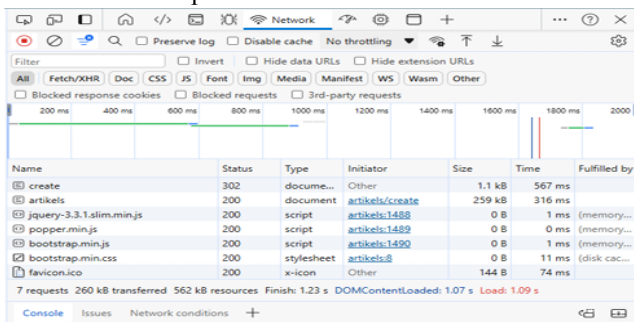
**A. Analisis Hasil Pengujian**

Setelah tahap implementasi selanjutnya adalah tahap pengujian. Pada tahap ini dilakukan analisis terhadap hasil pengujian yang telah dilakukan menggunakan beberapa skenario untuk mendapatkan nilai tertentu pada setiap parameter yang di uji dan sebagai perbandingan antara kedua *framework*.

a. Pengujian *Load Test Manual*

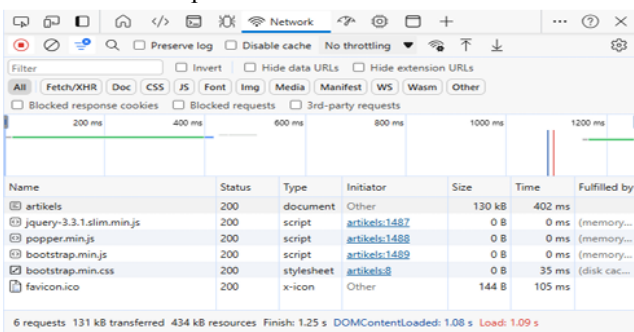
Parameter yang dihasilkan dari pengujian yang dilakukan adalah time, page size dan speed. Page size (dalam KB), merupakan besarnya page yang harus di-load, time (dalam ms) adalah waktu yang dibutuhkan untuk memunculkan halaman text pada web browser atau juga dapat dikatakan untuk client untuk mengirimkan request/permintaan ke server hingga selesai mendapat answer/jawaban dari server, speed (dalam Kbyte/s) merupakan kecepatan me-load halaman web.

1. *Inspect Element Network Create*: menjelaskan tampilan salah satu contoh load test Create dengan menggunakan inspect element dari masing-masing framework saat diakses. Tampilan ini memuat informasi tentang time (ms), page size (KB) dan speed (Kbyte/s), maka dilakukan 10 kali pengujian inspect element dengan penambahan 100 sampai 1000 data artikels pada aksi dari *Create*.



Gambar 4 *Inspect Element Network Test Create*

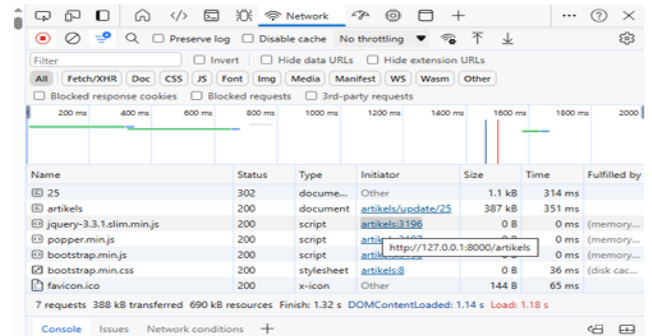
2. *Inspect Element Network Read*: menjelaskan tampilan salah satu contoh load test Read dengan menggunakan inspect element dari masing-masing framework saat diakses. Tampilan ini memuat informasi tentang time (ms), page size (KB) dan speed (Kbyte/s), maka dilakukan 10 kali pengujian inspect element dengan penambahan 100 sampai 1000 data artikels pada aksi dari *Read*.



Gambar 5 *Inspect Element Network Test Read*

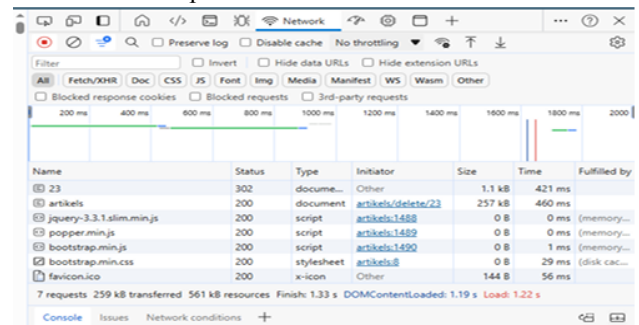
3. *Inspect Element Network Update*: menjelaskan tampilan salah satu contoh load test Update dengan menggunakan inspect element dari masing-masing

framework saat diakses. Tampilan ini memuat informasi tentang time (ms), page size (KB) dan speed (Kbyte/s), maka dilakukan 10 kali pengujian inspect element dengan penambahan 100 sampai 1000 data artikels pada aksi dari *Update*.



Gambar 6 *Inspect Element Network Test Update*

4. *Inspect Element Network Delete*: menjelaskan tampilan salah satu contoh load test Delete dengan menggunakan inspect element dari masing-masing framework saat diakses. Tampilan ini memuat informasi tentang time (ms), page size (KB) dan speed (Kbyte/s), maka dilakukan 10 kali pengujian inspect element dengan penambahan 100 sampai 1000 data artikels pada aksi dari *Delete*.



Gambar 7 *Inspect Element Network Test Delete*

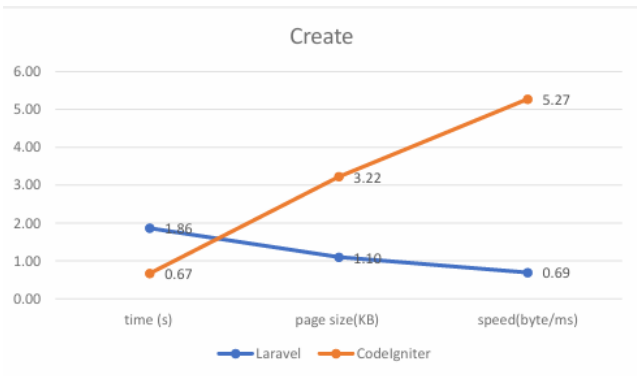
b. Hasil Perbandingan *Load Test Manual*

Parameter yang dihasilkan dari pengujian yang dilakukan melalui inspect element adalah time, page size dan speed. Oleh karena itu, nilai rata rata diambil sebagai data perbandingan create, read, update dan delete pada kedua framework. *Microsoft Excel*, dapat dilihat pada:

1. Perbandingan *Inspect Element Network Create*: berdasarkan Tabel 1 dan Gambar Grafik 8 hasil perbandingan aksi create dari tes manual menggunakan inspect element dapat dinyatakan bahwa CodeIgniter lebih unggul. nilai time pada CodeIgniter adalah 617,50 ms untuk me-load halaman dengan speed 5,27 byte/ms dengan ukuran file sebesar 3,220 bytes, sedangkan nilai time pada Laravel adalah 1862 ms untuk me-load halaman dengan speed 0,69 byte/ms dengan ukuran file sebesar 1100 bytes.

Tabel 1 Data Perbandingan Aksi Create

Perbandingan Create			
Perbandingan	time (ms)	page size (byte)	speed (byte/ms)
Laravel	1862.00	1100.00	0.69
CodeIgniter	617.50	3220.00	5.27



Gambar 8 Grafik Perbandingan Aksi Create

- Perbandingan *Inspect Element Network Read*: berdasarkan Tabel 2 dan Gambar Grafik 9, hasil perbandingan aksi *read* dari tes manual menggunakan *inspect element* dapat dinyatakan bahwa CodeIgniter sedikit lebih unggul. nilai *time* pada CodeIgniter adalah 1369 ms untuk *me-load* halaman dengan *speed* 332,68 *byte/ms* dengan ukuran file sebesar 484,3 *Kbytes*, sedangkan nilai *time* pada Laravel adalah 1984 ms untuk *me-load* halaman dengan *speed* 323,45 *byte/ms* dengan ukuran file sebesar 693,9 *Kbytes*.

Tabel 2 Data Perbandingan Aksi Read

Perbandingan Read			
Perbandingan	time (ms)	page size (byte)	speed (byte/ms)
Laravel	1984.00	693900.00	323.45
CodeIgniter	1369.50	484300.00	332.68



Gambar 9 Grafik Perbandingan Aksi Read

- Perbandingan *Inspect Element Network Update*: berdasarkan Tabel 3 dan Gambar Grafik 10, hasil perbandingan aksi *update* dari tes manual

menggunakan *inspect element* dapat dinyatakan bahwa CodeIgniter lebih unggul. Nilai *time* pada CodeIgniter adalah 597 ms untuk *me-load* halaman dengan *speed* 6,08 *byte/ms* dengan ukuran file sebesar 3600 *bytes*, sedangkan nilai *time* pada Laravel adalah 1740 ms untuk *me-load* halaman dengan *speed* 1,08 *byte/ms* dengan ukuran file sebesar 1400 *bytes*.

Tabel 3 Data Perbandingan Aksi Update

Perbandingan Update			
Perbandingan	time (ms)	page size (byte)	speed (byte/ms)
Laravel	1740.10	1400.00	1.08
CodeIgniter	597.30	3600.00	6.08

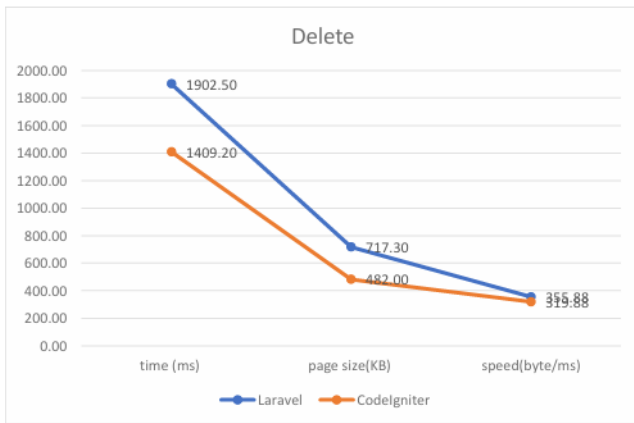


Gambar 10 Grafik Perbandingan Aksi Update

- Perbandingan *Inspect Element Network Delete*: berdasarkan Tabel 4 dan Gambar Grafik 10,, hasil perbandingan aksi *delete* dari tes manual menggunakan *inspect element* dapat dinyatakan bahwa CodeIgniter sedikit lebih unggul. Nilai *time* pada CodeIgniter adalah 1409 ms untuk *me-load* halaman dengan *speed* 319 *byte/ms* dengan ukuran file sebesar 482 *Kbytes*, sedangkan nilai *time* pada Laravel adalah 1902 ms untuk *me-load* halaman dengan *speed* 355 *byte/ms* dengan ukuran file sebesar 717,3 *Kbytes*.

Tabel 4 Data Perbandingan Aksi Delete

Perbandingan Delete			
Perbandingan	time (ms)	page size (byte)	speed (byte/ms)
Laravel	1902.50	717300.00	355.88
CodeIgniter	1409.20	482000.00	319.88



Gambar 10 Grafik Perbandingan Aksi Delete

c. Pengujian Load Test Otomatis

Pada tahap ini dilakukan analisis terhadap hasil pengujian user secara otomatis menggunakan Webserver Stress Tools dalam menguji batas kekuatan dari kedua framework. Pengujian dilakukan menggunakan beberapa skenario untuk mendapatkan nilai dari speed, time, page size sehingga dapat ditentukan scalabytey dan stability setiap parameter yang diuji dan sebagai perbandingan antara kedua framework.

1. Perbandingan *Load Test* Webserver Stress Tool: parameter yang dihasilkan dari pengujian yang dilakukan dengan Webserver Stress Tool adalah *time*, *page size* dan *speed*. Oleh karena itu, nilai rata-rata diambil sebagai data perbandingan pada kedua framework. Berdasarkan Tabel 5, hasil dari perbandingan pengujian menggunakan Webserver Stress Tool pada framework Laravel dan Codeigniter berdasarkan time, page size, dan speed dalam 10 kali percobaan dengan data dan user bertambah didapatkan hasil rata-rata, Dimana Page Size Laravel 224.1 MB, dengan speed 8.142 KB/s dalam waktu 5 menit. Dan Page size CodeIgniter 1884 MB dengan speed 125.623 KB/s dalam waktu 5 menit.

Tabel 5 Data Perbandingan *Load test* Otomatis

Test	Page Size CI (Kb)	Page Size LA (Kb)	Speed CI (Kb/s)	Time CI (menit)	Speed LA (Kb/s)	Time LA (menit)
test 1	573905	128584	20700	5	3396	5
test 2	1188631	319227	36689	5	7887	5
test 3	940021	294122	31084	5	8869	5
test 4	1575177	215541	59491	5	7708	5
test 5	2029801	157573	273773	5	5574	5
test 6	1416268	213950	52065	5	9504	5
test 7	2680742	386462	327790	5	13135	5
test 8	2688635	163139	157816	5	10130	5
test 9	2778774	171566	143697	5	8049	5
test 10	2973445	191558	153134	5	7170	5
Hasil	1884539	224172	125623	5	8142	5

2. Perbandingan *Pengujian Batas Stability*: pengujian Stabilitas atau kestabilan dalam proses untuk menangani penambahan beban yang diberikan, atau potensinya untuk ditingkatkan hingga didapatkan Tingkat kestabilan dari aplikasi. Maka di dilakukan pengujian menggunakan Web server strees tool hingga di temukan kestabilan dari kemampuan sistem di mulai dengan 100 data dan 100 user terus ditambahkan sampai ditemukan kestabilan dari aplikasi.

Tabel 11 Data *Stability* Sistem Laravel

Hasil Rata Rata Load Test Otomatis Laravel						
No	User	Data	time (m)	page size(Kbit)	speed(Kb/ms)	Error
1	100	100	5.00	128584	3396	0
2	200	200	5.00	319227	7887	7
3	400	400	5.00	215541	7708	200
4	600	600	5.00	213950	9504	426
5	800	800	5.00	163139	10130	643
6	1000	1000	5.00	191558	7170	820
7	1500	2000	5.00	263458	7928	856
8	2000	2000	5.00	-	-	-
9	2500	4000	5.00	-	-	-
10	4000	5000	5.00	-	-	-
Total Hasil			5.00	213636.71	7674.71	2096

Berdasarkan Tabel 11, hasil dari Data *Stability* Sistem Laravel menggunakan Webserver Stress Tool pada *framework* Laravel berdasarkan *time*, *page size*, dan *speed* dan *error* dalam 10 kali percobaan dengan data dan *user* bertambah didapatkan hasil rata-rata, Dimana kestabilan Laravel berada pada 2952 *error*.

Tabel 12 Data *Stability* Sistem CodeIgniter

Hasil Rata Rata Load Test Otomatis CodeIgniter						
No	User	Data	time (m)	page size(Kbit)	speed(Kb/ms)	Error
1	100	100	5.00	573905	20700	0
2	200	200	5.00	1188631	36689	0
3	400	400	5.00	1575177	59491	74
4	600	600	5.00	1416268	52065	0
5	800	800	5.00	2688635	157816	45
6	1000	1000	5.00	2973445	153134	89
7	1500	2000	5.00	-	-	-
8	2000	3000	5.00	-	-	-
9	2500	4000	5.00	-	-	-
10	4000	5000	5.00	-	-	-
Total Hasil			5.00	1736010.17	79982.50	205

Berdasarkan Tabel 11, hasil dari Data *Stability* Sistem CodeIgniter menggunakan Webserver Stress Tool pada *framework* CodeIgniter berdasarkan *time*, *page size*, dan *speed* dan *error* dalam 10 kali percobaan dengan data dan *user* bertambah didapatkan hasil rata-rata, Dimana kestabilan Laravel berada pada 205 *error*.

3. Perbandingan *Pengujian Batas Scalability*: pengujian *Skalabilitas* atau keterluasan dalam proses untuk menangani penambahan beban yang diberikan, atau potensinya untuk ditingkatkan hingga didapatkan *limit*/batas kemampuan dari aplikasi. Maka di dilakukan pengujian menggunakan Web server stress

tool hingga ditemukan batas akhir dari kemampuan sistem dimulai dengan 100 data dengan 100 user terus ditambahkan sampai ditemukan batas akhir dari aplikasi.

Tabel 12 Data Scalability Sistem Laravel

Hasil Rata Rata Load Test Otomatis Laravel						
No	User	Data	time (m)	page size(Kbit)	speed(Kb/ms)	Error
1	100	100	5.00	128584	3396	0
2	200	200	5.00	319227	7887	7
3	400	400	5.00	215541	7708	200
4	600	600	5.00	213950	9504	426
5	800	800	5.00	163139	10130	643
6	1000	1000	5.00	191558	7170	820
7	1500	2000	5.00	263458	7928	856
8	2000	2000	5.00	-	-	-
9	2500	4000	5.00	-	-	-
10	4000	5000	5.00	-	-	-
Total Hasil			5.00	213636.71	7674.71	2096

Berdasarkan Tabel 12, hasil dari Data Scalability Sistem Laravel menggunakan Webserver Stress Tool pada framework Laravel berdasarkan *time*, *page size*, dan *speed* dan *error* dalam 10 kali percobaan dengan data dan user bertambah didapatkan hasil rata-rata, Dimana batas kemampuan Laravel Berada pada tets 8 2000 user dengan 2000 data error.

Tabel 13 Data Scalability Sistem CodeIgniter

Hasil Rata Rata Load Test Otomatis CodeIgniter						
No	User	Data	time (m)	page size(Kbit)	speed(Kb/ms)	Error
1	100	100	5.00	573905	20700	0
2	200	200	5.00	1188631	36689	0
3	400	400	5.00	1575177	59491	74
4	600	600	5.00	1416268	52065	0
5	800	800	5.00	2688635	157816	45
6	1000	1000	5.00	2973445	153134	89
7	1500	2000	5.00	-	-	-
8	2000	3000	5.00	-	-	-
9	2500	4000	5.00	-	-	-
10	4000	5000	5.00	-	-	-
Total Hasil			5.00	1736010.17	79982.50	205

Berdasarkan Tabel 13, hasil dari Data Scalability Sistem Laravel menggunakan Webserver Stress Tool pada framework Laravel berdasarkan *time*, *page size*, dan *speed* dan *error* dalam 10 kali percobaan dengan data dan user bertambah didapatkan hasil rata-rata, Dimana batas kemampuan CodeIgniter Berada pada tets 7, 1500 user dengan 2000 data.

#### IV. KESIMPULAN

Berdasarkan analisis yang telah dilakukan terhadap aplikasi web yang dibangun menggunakan framework *CodeIgniter* dan *Laravel* dengan berbagai skenario yang dibuat. Pengujian *load test inspect element* didapatkan kesimpulan bahwa aplikasi web yang menggunakan framework *CodeIgniter* lebih baik dari sisi performanya dibandingkan dengan aplikasi web yang menggunakan framework *Laravel*, dan menurut hasil stress test web

server stress tool pada pengujian *stability CodeIgniter* lebih stabil dengan total *error 205 user* dalam 10 kali percobaan dibandingkan *Laravel* dengan total *error 2096 user* dalam percobaan yang sama. pada pengujian *scalability Laravel* dapat menangani permintaan *load user* maksimal sampai 1500 user sedangkan *CodeIgniter* dapat menangani *load user* maksimal 1000 user. Ukuran *page size* pada saat pengujian *load test*, aplikasi web yang menggunakan *framework Laravel* menghasilkan *page size* yang lebih besar dan berubah-ubah karena pada *framework Laravel me-load* lebih banyak *library* dari pada *framework CodeIgniter*, maka mengakibatkan *time* yang dihasilkan *framework Laravel* lebih besar. Setelah *stress testing* dilakukan disimpulkan dengan bertambahnya *user* yang mengakses suatu halaman, mengakibatkan bertambah lama waktu akses dari masing-masing *user* dan ketika jumlah *user* diatas 1500 terjadi puncak transaksi yang menyebabkan penolakan sistem dalam memberikan paket secara utuh kepada sejumlah *user*. Jumlah *user* yang semakin bertambah akan mengakibatkan nilai *speed* akan semakin berkurang, hal ini disebabkan karena semakin banyak *user* yang mengakses suatu halaman akan mengakibatkan beban pada *server* akan semakin besar.

#### REFERENSI

- [1] T. Yunita and A. Muhajir, "Analisis dan Perancangan Website Sebagai Sarana Informasi pada Lembaga Bahasa Kewirausahaan dan Komputer Akmi Baturaja," 2016.
- [2] A. Haniefardy, M. B. A. Fadhillah and S. Rochimah, "Tinjauan Literatur Sistematis: Pengaruh Penggunaan Framework Khusus dalam Proses Pengembangan Web dan Pembuatan Web," *Matrix : Jurnal Manajemen Teknologi Dan Informatika*, vol. 9, no. 2, pp. 68-73, 2019.
- [3] Aminudin, *Cara Efektif Belajar Framework Laravel*, Yogyakarta: Lokomedia, 2015.
- [4] I. Daqiqil, *Framework CodeIgniter Sebuah Panduan dan Best Practice*, Pekanbaru: Leanpub, 2011.
- [5] Hustinawati and K. Himawan, *Performance Analysis Framework Zend and CakePHP in Website Creation*, Bekasi: Universitas Gunadarma, 2014.
- [6] J. Manhas, "Initial Framework for Website Design and Development," *International Journal of Information Technology*, vol. 9, no. 4, pp. 363-375, 2017.
- [7] J. Asmara, "Rancang Bangun Sistem Informasi Desa Berbasis Website (Studi Kasus Desa Netpala)," *Jurnal STIKOM Uyelindo Kupang*, 2019.
- [8] F. Hafizulhaq, "Mengenal Framework Pengertian, Fungsi, Jenis dan Contohnya," 21 September 2021. [Online]. <https://www.exabytes.co.id>. [Accessed 02 Maret 2024]. Available: <https://www.exabytes.co.id>.
- [9] C. Ariata, "8 Framework PHP Terbaik untuk Developer," 29 Januari 2024. [Online]. Available: <http://www.hostinger.co.id>. [Accessed 20 Juli 2024].
- [10] R. Puspa, "Laravel adalah: Pengertian, Kelebihan, dan Fitur," 23 Maret 2022. [Online]. Available: <https://academy.alterra.id>. [Accessed 20 Juli 2024].
- [11] N. B. Afrizal, "Apa Itu Codeigniter Pengertian, Keunggulan & Cara Kerja," 05 Februari 2023. [Online]. Available: <https://www.jagoanhosting.com>. [Accessed 21 Juli 2023].
- [12] A. Pupuh, "Laragon: Pengertian, Fungsi, Download, dan Cara Instalasinya," 03 November 2023. [Online]. Available: <https://blog.rumahweb.com>. [Accessed 21 Juli 2024].
- [13] A. Amarulloh, "Analisis Perbandingan Performa Web Service Rest Menggunakan Framework Laravel, Django, Dan Node Js Pada

- Aplikasi Berbasis Website," Jurnal Teknik Informatika, vol. 09, no. 01, pp. 12-17, 2023.
- [14] Faradilla, "Apa itu web server pengertian, fungsi, dan cara kerjanya," 01 Maret 2023. [Online]. Available: <https://www.hostinger.co.id>. [Accessed 22 Juli 2023].
- [15] Y. Kurniawan, "Apa itu Nginx dan cara kerjanya," 21 Juli 2019. [Online]. Available: <https://www.niagahoster.co.id>. [Accessed 22 Juli 2023].
- [16] Sutiono, "Web stress tool pengertian contoh dan fungsinya," 02 Maret 2022. [Online]. Available: <https://dosenit.com>. [Accessed 23 Juli 2023].
- [17] B. Fahlevi, "Pengujian aplikasi web dengan webserver stress tool," 28 Juni 2010. [Online]. Available: <https://bobyfahlevi.wordpress.com>. [Accessed 24 Juli 2023].
- [18] Rahmah, "E-Book Implementasi dan Pengujian Sistem," Sistem Informasi Unisi, 2020.
- [19] I. R. Dhaifullah and dkk, "Survei Teknik Pengujian Software," Jacis: Journal Automation Computer Information System, vol. 02, no. 01, 2022.