

## ANALISIS PERFORMASI FRAMEWORK *CODEIGNITER* DAN *LARAVEL* MENGUNAKAN WEB SERVER *APACHE*

(*PERFORMANCE ANALYSIS FRAMEWORK CODEIGNITER AND LARAVEL USING  
APACHE WEB SERVER*)

**Ruli Erinton<sup>1</sup>, Ridha Muldina Negara<sup>2</sup>, Danu Dwi Sanjoyo<sup>3</sup>**

<sup>1,2,3</sup> Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom, Bandung

<sup>1</sup>[rulierin@student.telkomuniversity.ac.id](mailto:rulierin@student.telkomuniversity.ac.id), <sup>2</sup>[ridha@telkom](mailto:ridha@telkom), <sup>3</sup>[danu@telkom](mailto:danu@telkom)

---

### Abstrak

Untuk membangun sebuah website dibutuhkan sebuah framework, yang dapat membantu pembangunan dan pengembangan website itu sendiri sehingga *developer* tidak perlu membangun sebuah website dari awal lagi jika ingin memperbarui fitur-fitur pada website yang sudah ada sehingga sumberdaya yang dibutuhkan tidak terlalu banyak. Saat ini terdapat lebih dari satu jenis framework yang dapat digunakan sebagai tool untuk membuat website, Untuk itu dilakukan analisis performasi anatar dua framework, yang kemudian hasil analisis dapat digunakan sebagai bahan evaluasi dan masukan bagi *developer* untuk pemilihan framework dari sisi performasinya.

Framework adalah sekumpulan fungsi, class, dan aturan-aturan. Berbeda dengan *library* yang sifatnya untuk tujuan tertentu saja, framework bersifat menyeluruh mengatur bagaimana kita membangun aplikasi. Framework memungkinkan kita membangun aplikasi dengan lebih cepat karena sebagai *developer* kita akan lebih memfokuskan pada pokok permasalahan.

aplikasi web yang menggunakan framework *CodeIgniter* lebih baik dari sisi performasinya dibandingkan dengan aplikasi web yang menggunakan framewrok *Laravel*. nilai *time* pada *CodeIgniter* 150,5 ms lebih rendah dibnadingkan nilai *time* pada *Laravel* 254,5 ms. Nilai *error* terting didapat pad *Laravel* 79,7. Pada parameter QoS nilai throughput tertinggi 6,227 Mbps, *packet loss* 0%, *retransmission* terendah 1, *delay* terendah 91,46 dengan klasifikasi sangat baik berdasarkan standar ITU-T.

**Kata kunci : Performasi, Analisis, Framework CodeIgniter, Framework Laravel, Web Server Apache.**

---

### Abstrack

At this time information is growing so fast and now that is becoming a trend in the community is the culture of information and social network search using internet media, and one of the important components in the internet world is the website, but the creation of the website from the beginning is very time consuming. Therefore, building a website requires a framework, which can help the development and development of the website itself so that developers do not need to build a website from scratch again if you want to update the features on the existing website so that the resources needed are not too much.

Framework is a set of functions, classes, and rules. Unlike libraries that are for a specific purpose only, the framework is comprehensive in regulating how we build applications. Framework allows us to build applications more quickly because as a developer we will focus more on the subject matter.

Web apps using the CodeIgniter framework are better on the performance side than web apps that use Laravel framework. The time value on CodeIgniter 150.5 ms lower than the time value of Laravel 254.5 ms. The value of the terting error is found in Laravel 79.7. At QoS parameter the highest throughput value is 6,227 Mbps, packet loss 0%, retransmission lowest 1, lowest delay 91,46 with very good classification based on ITU-T standard.

**Keywords : Performance, Analysis, CodeIgniter Framework, Laravel Framework, Apache Web Server.**

---

## 1. Pendahuluan

### 1.1 Latar Belakang

Saat ini dunia informasi berkembang begitu cepat dan saat ini yang sedang menjadi tren di masyarakat adalah budaya pencarian informasi dan social network menggunakan media internet, dan salah satu komponen penting dalam dunia internet adalah website. Untuk membangun sebuah website dibutuhkan sebuah framework, yang dapat membantu pembangunan dan pengembangan website itu sendiri sehingga *developer* tidak perlu membangun sebuah website dari awal lagi jika ingin memperbarui fitur-fitur pada website yang sudah ada. Namun saat ini terdapat lebih dari satu jenis framework yang dapat digunakan sebagai tool untuk membuat website. Untuk itu dilakukan analisis performansi antara dua framework, yang kemudian hasil analisis dapat digunakan sebagai bahan evaluasi dan masukan bagi *developer* untuk pemilihan framework dari sisi performasinya.

Framework adalah sekumpulan fungsi, class, dan aturan-aturan. Berbeda dengan *library* yang sifatnya untuk tujuan tertentu saja, framework bersifat menyeluruh mengatur bagaimana kita membangun aplikasi. Framework memungkinkan kita membangun aplikasi dengan lebih cepat karena sebagai *developer* kita akan lebih memfokuskan pada pokok permasalahan, sedangkan untuk hal-hal penunjang lainnya seperti koneksi ke database, form validation, GUI dan security umumnya telah disediakan oleh framework.

Pada penelitian sebelumnya telah dilakukan perbandingan analisis framework *Zend* dan *CakePHP* pada web perwalian didapatkan hasil bahwa performansi framework *Zend* lebih unggul dalam hal performansi karena dapat menanggung request sampai 1000 sedangkan pada *CakePHP* hanya 460 saja<sup>[5]</sup>. Namun kali ini penulis akan mengkaji dengan framework yang berbeda yaitu framework *CodeIgniter* dan *Laravel* menggunakan web server apache yang sudah terdapat xampp dengan menganalisis performansi pada *streaming service* dan *FTP service* (download) berdasarkan QoS (*Quality of Service*) yang ditetapkan ITU-T.

## 2. Dasar Teori

### 2.1 Website

Website adalah salah satu aplikasi yang berisikan dokumen multimedia (teks, gambar, suara, animasi, video) di dalamnya yang menggunakan protokol HTTP (hypertext transfer protocol) dan untuk mengaksesnya menggunakan perangkat lunak yang disebut browser.

## 2.2 Framework *Laravel*

*Laravel* merupakan *framework* PHP yang menekankan pada kesederhanaan dan fleksibilitas pada desainnya. *Laravel* dirilis dibawah lisensi MIT dengan sumber kode yang disediakan di Github. Sama seperti *framework* PHP lainnya, *Laravel* dibangun dengan basis *MVC* (*Model-View-Controller*). *Laravel* dilengkapi *command line tool* yang bernama “Artisan” yang bisa digunakan untuk *packaging* bundle dan instalasi bundle. Menurut survey yang dilakukan oleh Sitepoint.com pada Desember 2013 dalam popularitas *framework* PHP, *Laravel* menduduki urutan teratas.

## 2.3 Framework *CodeIgniter*

*CodeIgniter* adalah sebuah web application framework yang digunakan untuk membangun aplikasi PHP dinamis yang dibangun menggunakan konsep *Model View Controller development pattern*. *CodeIgniter* menyediakan berbagai macam *library* yang dapat mempermudah dalam pengembangan dan termasuk *framework* tercepat dibandingkan dengan *framework* lainnya.

## 2.4 PHP

PHP (Hypertext Preprocessor) adalah suatu server-side scripting yang menyatu dengan HTML (Hyper Text Markup Language) untuk membuat halaman website yang dinamis. Server-side scripting sendiri berarti suatu pemrograman yang pengeksekusiannya berada di sisi server. Seluruh proses di dalam bahasa pemrograman PHP dilakukan di sebuah server. Proses dilakukan di server tetapi hasil akan ditampilkan di browser.

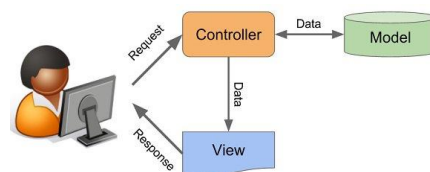
## 2.5 Xampp

XAMPP adalah software web server apache yang di dalamnya tertanam server MySQL yang didukung dengan bahasa pemrograman PHP untuk membuat website yang dinamis. XAMPP sendiri mendukung dua system operasi yaitu windows dan Linux. Untuk linux dalam proses penginstalnya menggunakan command line sedangkan untuk windows dalam proses penginstalnya menggunakan interface grafis sehingga lebih mudah dalam penggunaan XAMPP di Windows dibanding dengan Linux. Didalam XAMPP ada 3 komponen utama yang di tanam di dalamnya yaitu web server Apache, PHP, dan MySQL.

## 3. Perancangan

### 3.1 Perancangan Sistem

Tahap pemodelan adalah tahap pertama yang dilakukan untuk melakukan analisis performansi. Dimana pada tahap ini akan dirancang sebuah sistem agar dapat melakukan analisis performansi antara *framework CodeIgniter* dan *Laravel* dengan membangun sebuah aplikasi berbasis website dengan menggunakan arsitektur *MVC*.

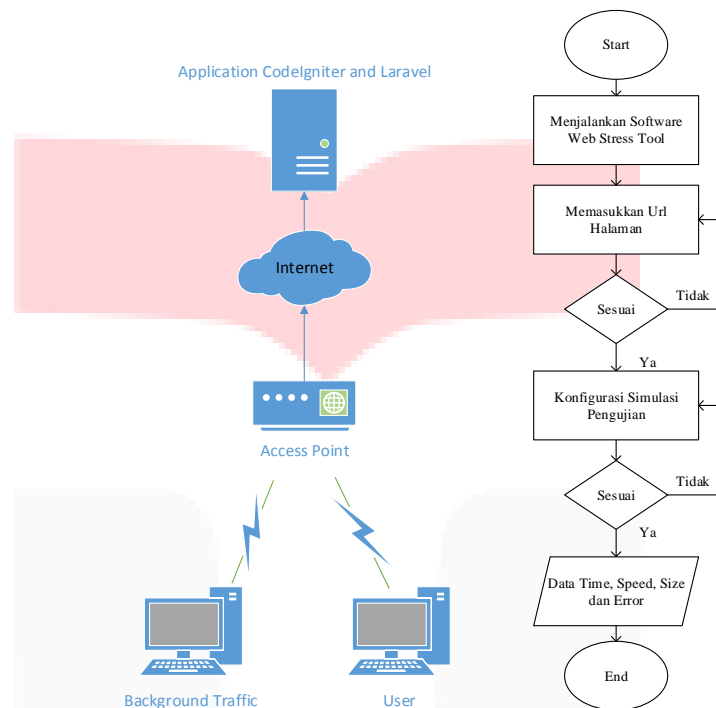


Gambar 3.1 Gambaran umum sistem

Pada gambar 3.2 menjelaskan sistem akan menekankan kepada pembagian dari komponen komponen program menjadi tiga bagian utama yaitu *model*, *view*, dan *controller*, atau juga dapat disebut memisahkan antara *business logic* dan *presentation logic*, sehingga membuat lebih terstruktur.

### 3.2 Skenario Pengujian

Perancangan aplikasi android diawali dengan pembuatan alur sistem aplikasi. Berikut adalah *flowchart* alur kerja dari aplikasi android secara umum yang telah dibuat pada tugas akhir ini.



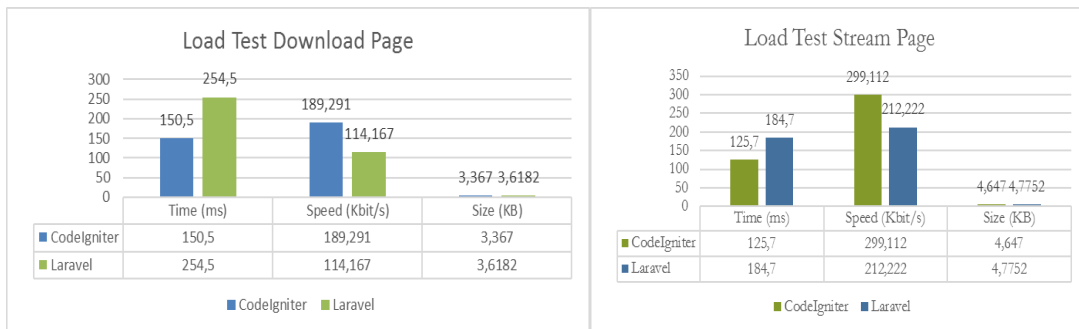
Gambar 3.2 *Flowchart* Skenario Pengujian

## 4. 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 diuji dan sebagai perbandingan antara kedua framework.

### 4.1 Pengujian Load Test

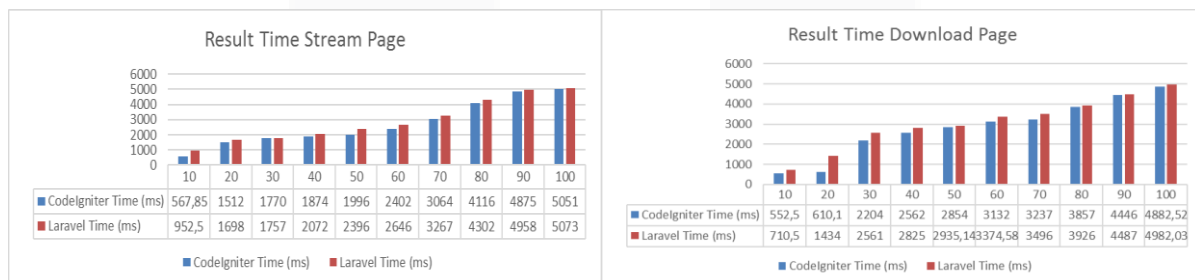
Pengujian *Load Test* pada tugas akhir ini dilakukan dengan menggunakan bantuan *software web stress tool*. Parameter yang dihasilkan dari pengujian yang dilakukan adalah *time*, *page size* dan *speed*. *Page size* (dalam KB), merupakan besarnya *page* yang harus di-load, *time* (dalam ms) adalah waktu 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 Kbit/s) merupakan kecepatan me-load halaman web pada skenario ini akan dilakukan dengan menggunakan jaringan internet dengan kecepatan *download* berdasarkan hasil [www.speedtest.net](http://www.speedtest.net) adalah 3,13 Mbps.



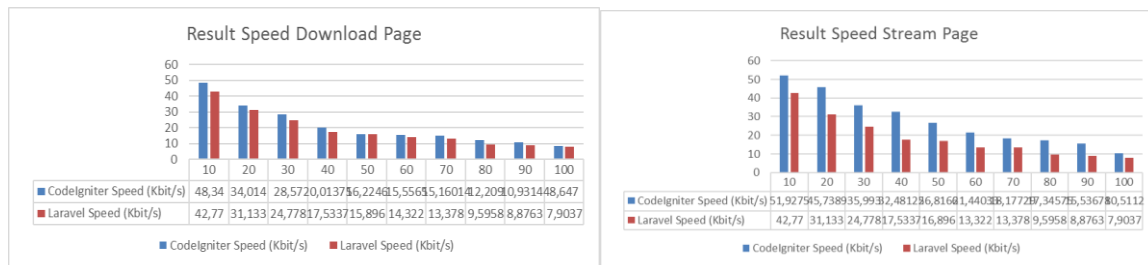
Berdasarkan grafik di dapatkan nilai rata-rata dari pengujian sebanyak 10 kali, nilai *time* pada *CodeIgniter* 150,5 ms untuk me-load halaman dengan *speed* 189,291 Kbit/s pada *download file* dengan file sebesar 3,367 KB dan , nilai *time* pada *Laravel* 254,5 ms untuk me-load halaman dengan *speed* 114,167 Kbit/s pada *download file* dengan file sebesar 3,6182 KB, pada tabel 4.2 nilai page size yang berubah-ubah dan lebih besar dari *CodeIgniter* disebabkan pada framework *Laravel* me-load library yang lebih banyak dibandingkan dengan *CodeIgniter*. Sehingga dari data tersebut didapatkan bahwa aplikasi web yang menggunakan framework *CodeIgniter* memiliki performance yang lebih baik dari aplikasi web yang menggunakan framework *Laravel*. Berdasarkan grafik stream page di dapatkan nilai rata-rata dari pengujian sebanyak 10 kali, nilai *time* pada *CodeIgniter* 125,7 ms untuk me-load halaman dengan *speed* 299,112 Kbit/s pada *download file* dengan file sebesar 4,647 KB dan , nilai *time* pada *Laravel* 184,7 ms untuk me-load halaman dengan *speed* 212,222 Kbit/s pada

## 4.2 Pengujian Stress Test

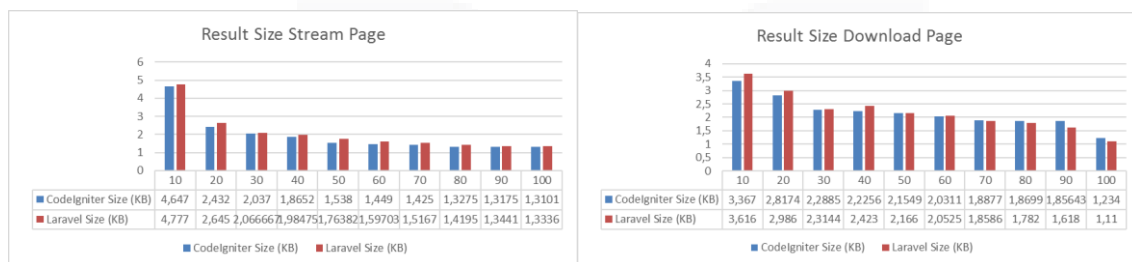
Pengujian *Stress Test* pada tugas akhir ini dilakukan dengan menggunakan bantuan *software web stress tool*, pengujian yang dilakukan dengan mensimulasikan variasi *user* yang mengakses aplikasi web secara bersamaan, jumlah *user* yang digunakan adalah 100.



Berdasarkan gambar grafik semakin bertambahnya nilai *user* yang mengakses suatu halaman mengakibatkan bertambah lama waktu pengaksesan pada masing-masing *user*, hal tersebut disebabkan karena server akan memproses atau melayani *request* yang lebih banyak sehingga waktu proses yang dibutuhkan akan lebih lama untuk memberikan layanan pada *user*. pada hasil pengujian yang didapat dari jumlah *user* 10 sampai dengan 100 performansi yang dihasilkan oleh aplikasi web menggunakan framework *CodeIgniter* masih lebih unggul terlihat pada grafik ketika jumlah *user* diatas 10 nilai *time* yang terdapat pada framework *Laravel* masih lebih besar yaitu 2935 ms pada saat *user* 50 dan 4982 ms pada saat *user* 100 dibandingkan dengan framework *CodeIgniter* yaitu 2854 ms pada saat *user* 50 dan 4882 pada saat *user* 100, namun terdapat kesamaan pada kedua framework yang mengalami kenaikan *time* yang signifikan ketika jumlah *user* bertambah saat mengakses halaman secara bersamaan, hal ini disebabkan karena ketika *user* semakin bertambah akan mengakibatkan *size* page yang dikirimkan oleh web server akan semakin besar.

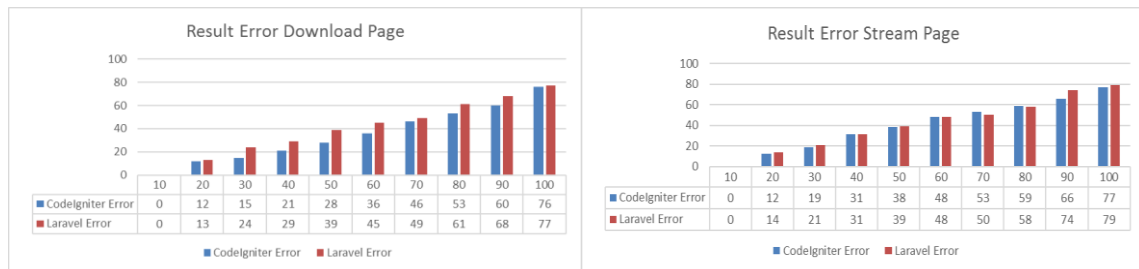


Pada grafik 4.11 hasil pengujian yang didapat dari jumlah *user* 10 sampai dengan 100 performasi yang dihasilkan oleh aplikasi web menggunakan framework *CodeIgniter* masih lebih unggul terlihat pada grafik nilai *speed* pada framework *CodeIgniter* lebih besar yaitu 51,927 Kbit/s pada saat *user* 10 dibandingkan dengan nilai *speed* pada framework *Laravel* 42,77 Kbit/s pada saat *user* 10, berdasarkan grafik terdapat kesamaan pada kedua framework yang mengalami penurunan *speed* yang signifikan ketika jumlah *user* mengakses halaman secara bersamaan, dilihat dari 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 mengakibatkan nilai *speed* akan berkurang.



Pada gambar grafik hasil pengujian size yang didapat dari jumlah *user* 10 sampai dengan 100 *user* oleh aplikasi web menggunakan framework terdapat kesamaan pada kedua framework yang mengalami penurunan jumlah *size* yang signifikan ketika jumlah *user* mengakses halaman secara bersamaan bertambah, dilihat dari *page size*nya ketika jumlah *user* 10 sampai dengan 100 nilai rata-rata *page size* tiap *user* yang dihasilkan mengalami penurunan, hal ini disebabkan ketika jumlah *user* mengakses suatu halaman diatas 10 *user* secara bersamaan beban pada server akan meningkat sehingga akan terjadi penolakan system dalam memberikan paket data secara utuh kepada sejumlah *user* sehingga menyebabkan nilai *size* akan menurun. Berdasarkan hasil uji aplikasi web menggunakan framework *CodeIgniter* menghasilkan *page* yang lebih kecil dengan jumlah 3,367 KB dibandingkan dengan aplikasi web menggunakan framework *Laravel* dengan jumlah 3,616 KB, semakin besar *size* yang dihasilkan akan mempengaruhi waktu *load time* pada saat *user* mengakses.





Pada hasil pengujian yang didapat dari jumlah *user* 10 sampai dengan 100 performansi yang dihasilkan oleh aplikasi web menggunakan framework *CodeIgniter* masih lebih unggul, terlihat pada gambar grafik 4.9 ketika *user* 100 jumlah *error* rata-rata yang terdapat pada framework *Laravel* masih lebih banyak yaitu 79 pada saat *user* dibandingkan dengan framework *CodeIgniter* ketika *user* 100 yaitu 77 pada saat *user*, namun terdapat kesamaan pada kedua framework yang mengalami kenaikan *error* yang signifikan ketika jumlah *user* mengakses halaman secara bersamaan semakin bertambah, hal ini disebabkan ketika jumlah *user* mengakses suatu halaman diatas 10 *user* semakin besar *user* mengakses halaman secara bersamaan beban pada server akan semakin meningkat sehingga akan terjadi penolakan system dalam memberikan paket data kepada sejumlah *user*.

## 5. Kesimpulan

Berdasarkan analisis yang telah dilakukan terhadap aplikasi web yang dibangun menggunakan framework *CodeIgniter* dan *Laravel* dengan berbagai skenario yang dibuat, maka dihasilkan kesimpulan sebagai berikut.

1. Berdasarkan dari hasil pengujian *load test* dan *stress test* didapatkan kesimpulan bahwa aplikasi web yang menggunakan framework *CodeIgniter* lebih baik dari sisi performasinya dibandingkan dengan aplikasi web yang menggunakan framework *Laravel*.
2. Dari 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* meload lebih banyak *library* dari pada framework *CodeIgniter*, maka mengakibatkan *time* yang dihasilkan framework *Laravel* lebih besar.
3. Setelah *stress testing* dilakukan disimpulkan dengan bertambahnya *user* yang mengakses suatu halaman, mengakibatkan bertambah lama waktu akses dari masing-masing *user* dan ketika *user* jumlah *user* diatas 10 terjadi puncak transaksi yang menyebabkan penolakan system dalam memberikan packet 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 mengakibatkan nilai *speed* akan berkurang.

## 6. Saran

Tugas akhir ini masih sangat memungkinkan untuk dapat dilakukan pengembangan. Pengembangan yang dapat dilakukan untuk tugas akhir selanjutnya adalah:

1. Menganalisis dari sisi keamanan pada database masing-masing framework.
2. Menganalisis pengaruh jaringan 3G dan 4G pada delay dan *response time*.
3. Menambah service dengan streaming secara live atau video conference.
4. Menggunakan web server yang berbeda.

## Daftar Pustaka

- [1] Ahmad, Hidayatullah. 2015. Analisis Dan Perancangan Toko Online Geraihana Berbasis Seo Menggunakan *Laravel*. Tugas Akhir. Yogyakarta : Amikom Yogyakarta.
- [2] Anbazhagan Mani and Arun Nagarajan, "Understanding quality of service for Web services", [Online] Availabel: [www.ibm.com/developerworks/library/ws-quality/index.html](http://www.ibm.com/developerworks/library/ws-quality/index.html). [Akses 16 Oktober 2016].
- [3] Cecep Mahbub, "Benchmark Performa Web Server dengan ApacheBench (ab)", [Online] Availabel: <http://ngadimin.com/2009/09/27/benchmark-performa-web-server-dengan-apachebench-ab/>. [Akses 6 Oktober 2016].
- [4] D. Ibnu, "Ebook Framework *CodeIgniter*: Sebuah Panduan dan Best Practice," 2011
- [5] Ardhana, YM Kusuma. 2013. Pemograman PHP *CodeIgniter* Black Box. Jaskom.
- [6] Hustinawati, Kurnia Himawan. 2014. Performance Analysis Framework *Zend* and *CakePHP* in Website Creation. Jurnal. Bekasi: Universitas Gunadarma.
- [7] Laksono, Prasetyo Singgih. 2015. Perancangan Dan Implementasi Sistem Informasi Kependudukan dan Surat-Menyurat (Studi Kasus Kelurahan Kadokan). Bandung: Telkom University.
- [8] Hadiansyah, Reno Muktiaji. 2015. *Quality Of Service (Qos)* Layanan Video Coference Pada Jaringan *High Speed Packet Access (HSPA)* Menggunakan Emulator *Graphic Network Simulator (GNS)*. Bandung: Telkom University.
- [9] Saleh, Agung Rahmat. 2010. Analisis Dan Perbandingan Framework *Zend* Dan *Yii*. Tugas Akhir. Bandung: Telkom University.