Sistem Pengawasan Kinerja Jaringan Server Web Apache dengan *Log Management System* ELK (Elasticsearch, Logstash, Kibana)

Claudia Tarigan [#1], Ventje Jeremias Lewi Engel [*2], Dina Angela [#3]

[#] *Departemen Teknologi Informasi, Institut Teknologi Harapan Bangsa*
Jl. Dipatiukur no. 80-84, Bandung, Jawa Barat, Indonesia

[1]claudia.adyan9@gmail.com

[2]ventje@ithb.ac.id

[3]*dina_angela@ithb.ac.id*

Abstract*— A server is a software that has a duty and responsibility to provide information to the web. Any process that occurs within the web server will be recorded in a log. A log is a file that contains a list of actions, events (activities) that have been going on in a computer system. By using the log files on a web server, a lot of things that can be done by system administrators to monitor the performance of the web server. However, the log on the web server are difficult to understand and read. Log management system is a system that can handle data services log in large numbers and generate detailed log information. Logstash is the application log management system that can help the system administrator in the performance monitor log of the web server. Logstash will be combined with the Elasticsearch serves as data storage media and Kibana as visualization. ELK (Elasticsearch, Logstash, Kibana) is used to display and keep an eye on the log data from web server so, system administrators can know the performance of the web server.

***Keywords*— web server, log, log management system, Logstash, Elasticsearch, Kibana.**

***Abstrak*— Server sebagai penyedia layanan di jaringan memastikan agar semua aktivitas yang berkaitan dengannya dicatat dalam log. Log adalah file yang berisi daftar aktivitas dan waktu yang terjadi dalam server. Pemantauan kinerja server dengan memanfaatkan log akan meningkatkan ketahanan dan tingkat layanan sebuah server. Masalahnya adalah log yang tercatat tersebar di beberapa tempat dan susah untuk langsung diolah karena bentuknya bukan seperti tabel. Log management system dapat menangani data log dalam jumlah besar, menghasilkan detail informasi aktivitas, dan visualisasi data log yang lebih mudah dimengerti pengguna. Perangkat lunak ELK (Elasticsearch, Logstash, Kibana) bisa dikombinasikan dan diintegrasikan dalam server web untuk mengelola, manampilkan, dan mengawasi data-data log yang ada sehingga administrator jaringan dapat mengetahui kinerja yang terdapat pada server web. Implementasi dan uji coba menunjukkan pengawasan kinerja server web Apache yang lebih mudah dimengerti pengguna.**

***Kata Kunci*— web server, log, log management system, Logstash, Elasticsearch, Kibana**

I. PENDAHULUAN

Server adalah sebuah perangkat lunak yang memiliki tugas dan tanggung jawab untuk menyediakan layanan informasi kepada web. Informasi ditampilkan berdasarkan data yang tersimpan di dalam web server. Fungsi utama dari web server adalah untuk men-*transfer* atau memindahkan berkas yang diminta oleh pengguna melalui protokol komunikasi tertentu. Setiap proses yang terjadi di dalam web server akan dicatat dalam *log*. *Log* adalah sebuah file yang berisi daftar tindakan, kejadian (aktivitas) yang telah terjadi di dalam suatu sistem komputer. *Log* sangat diperlukan untuk membantu sistem administrator dalam mengatasi hal-hal yang berpotensi menimbulkan masalah, ataupun melacak masalah yang sudah atau seterjadi, baik itu masalah keamanan maupun yang lainnya, karena *log* memuat laporan rinci tentang semua aktifitas yang terjadi pada web server. Dengan memanfaatkan *file log* pada web server, banyak hal yang dapat dilakukan oleh sistem administrator untuk memonitoring kinerja web server. Namun, *log* yang terdapat pada web server sulit untuk dipahami dan dibaca. Hal ini membuat web server membutuhkan aplikasi yang dapat membantu sistem administrator dalam memonitoring kinerja web server. Oleh karena itu, web server membutuhkan *log management system*, untuk membantu mengelola dan menganalisis hasil *log* web server.

Log management system merupakan suatu sistem yang dapat menangani data dalam jumlah yang besar dan menghasilkan detail informasi dari *log*, sehingga *log management* sangat dibutuhkan oleh sistem administrator untuk dapat mengelola data *log*. Oleh karena itu, perlu adanya sentralisasi yang dapat mengelola file *log* tersebut. Sentralisasi *log* dapat sangat berguna ketika sistem administrator mencoba untuk mengidentifikasi masalah pada sebuah server, agar dapat membantu sistem administrator mencari semua *log* di dalam satu tempat.

Penelitian ini akan mengimplementasikan aplikasi *log management system* yaitu, Logstash yang dipadukan dengan Elasticsearch sebagai media penyimpanan data dan Kibana sebagai visualisasi. ELK (Elasticsearch, Logstash, Kibana) digunakan untuk menampilkan dan mengawasi data-data *log*

dari web server. Hasil akhir dari penelitian ini berupa pelaporan kinerja web server berdasarkan *log* yang akan dilihat dari performa indikator waktu, *utilization*, dan *error*.

Berdasarkan latar belakang masalah yang telah diuraikan, maka ditetapkan rumusan masalah sebagai berikut:

1. Bagaimana cara mengintegrasikan pengelolan *log* Apache web server ke dalam *log management system* ELK (Elasticsearch, Logstash, Kibana)?
2. Bagaimana cara mengidentifikasi dan mendapatkan *performance indicator* Apache web server dengan *log management system* ELK (Elasticsearch, Logstash, Kibana)

Tujuan dari penelitian ini adalah:

Bagaimana cara mengintegrasi, mengidentifikasi, dan mendapatkan *performance indicator* Apache web server dengan menggunakan *log management system* ELK (Elasticsearch, Logstash, Kibana).

Hal-hal yang menjadi batasan dari penelitian ini adalah:

1. Penelitian ini hanya berfokus pada server web Apache.
2. Penelitian ini hanya menggunakan informasi dari *log* Apache web server.

Metodologi yang digunakan dalam penelitian ini adalah, pertama rumusan masalah yang telah diperoleh, selanjutnya dilakukan kegiatan pembelajaran materi melalui sumber pustaka, kemudian menganalisis/mengidentifikasi aspek-aspek kinerja, merancang alur dan skenario dari percobaan yang akan dilakukan pada Apache web server dengan menggunakan aplikasi *log management system* ELK (Elasticsearch, Logstash, Kibana). Tahap selanjutnya akan dilakukann implementasi dan pengujian pada kedua aplikasi, yaitu Apache web server dan *log management system*.

Hasil dari penelitian ini adalah mengintegrasikan *log management system* dengan Apache web server, untuk memudahkan sistem administrator dalam mengidentifikasi dan mengawasi kinerja web server menggunakan *log management system* ELK.

Manfaat penelitian ini adalah membantu sistem administrator dalam mengelola, mengoptimalkan, dan menganalisis *log* kinerja Apache web server dengan menggunakan *log management system*.

Sistematika penulisan artikel ilmiah ini terdiri atas empat bagian. Bagian Pendahuluan membahas mengenai latar belakang masalah, rumusan masalah batasan masalah, tujuan penelitian, metodologi penelitian, keluaran dan manfaat penelitian, serta sistematika penulisan. Pada bagian kedua yaitu Landasan Teori, dibahas kajian pustaka tentang pengertian, arsitektur, cara kerja Apache *web server* dan *log management system*, serta kinerja perfoma indicator. Pada bagian ketiga yaitu Hasil dan Pembahasan dijelaskan tentang analisis kebutuhan perancangan, dan integrasi sistem dilanjutkan dengan implementasi dan integrasi kedua sistem, kemudian dilakukan pengujian dan perhitungan performa indikator. Pada bagian keempat, yaitu Simpulan akan dituliskan kesimpulan dari proses penelitian yang telah dilakukan serta saran-saran untuk bahan pengembangan penelitian ke depan.

II. LANDASAN TEORI

A. Apache Web Server

Web server merupakan sebuah perangkat lunak yang menerima informasi layanan data. Web server berfungsi menerima permintaan HTTP atau HTTPS dari klien yang dikenal dengan web browser dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman web yang umumnya berbentuk dokumen HTML. Salah satu jenis web server yang paling banyak digunakan ialah Apache [1]. Apache didukung oleh sejumlah antarmuka pengguna berbasis grafik (GUI) yang memungkinkan penanganan server menjadi mudah. Apache merupakan *open source software* yang dikembangkan oleh komunitas terbuka dibawah naungan Apache Software Foundation [2].

Gambar 1 menunjukkan arsitektur Apache web server yang terdiri dari inti yang relatif kecil, yaitu dengan sejumlah modul. Modul disusun secara statis ke dalam server. Tujuan khusus modul MPM (*Multi-Processing Module*) adalah berfungsi untuk mengoptimalkan Apache untuk sistem operasi. MPM biasanya hanya untuk modul dalam mengakses pada sistem operasi melalui APR [6].

Apache web server bekerja seperti mesin, dimana aplikasi atau *software* beroperasi dalam mendistribusikan *web page* sesuai dengan permintaan pengguna. Gambar 2 menjelaskan, pada saat browser meminta data *web page* ke server, maka instruksi permintaan data oleh browser tersebut dikemas didalam TCP yang merupakan *protocol transport* dan dikirim ke alamat yang dalam hal ini merupakan protokol berikutnya, yaitu Hyper Text Transfer Protocol (HTTP). HTTP tersebut merupakan protokol yang digunakan dalam World Wide Web (WWW) antar komputer yang terhubung dalam jaringan di Dunia [7].

B. Log Management

Log merupakan sebuah *file* yang berisi catatan peristiwa-peristiwa yang terjadi di dalam sistem komputer. *Log* berfungsi untuk mempermudah dalam melakukan evaluasi sistem dan meningkatkan kinerja sistem. *Log management* merupakan sistem yang digunakan untuk melakukan penanganan data yang terjadi di log. Menurut The National Institute for Standards and Technology (NIST) dalam Special Publication SP800-92 definisi dari sistem *log management* adalah proses untuk menghasilkan, mentransmisikan, menyimpan, menganalisis, dan mengatur keamanan data pada computer [3].

Salah satu *open source log management system* yang digunakan untuk memantau dan memonitor trafik di sebuah jaringan adalah ELK. Gambar 3 menjelaskan, tentang ELK yang terdiri dari Elasticsearch, Logstash, dan Kibana. Logstash merupakan sebuah perangkat lunak *open source* yang berfungsi untuk mengumpulkan dan mem-*parsing log*, Elasticsearch yang dipergunakan untuk menyimpan *log*. *Shipper* (*beat*) akan mengirim *log* ke logstash, yang memiliki fungsi sebagai *shipping agent* untuk *log*, dan Kibana menampilkan hasil visualisasi *log* [8].

C. *Apache Log*

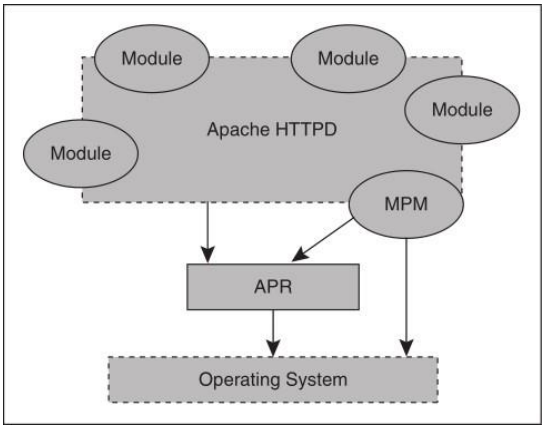
Apache web server menyimpan *file log* yang mana terdapat dua jenis *log* yaitu, *error.log* dan *access.log*. *Error.log* menyimpan pesan kesalahan pada web server, sedangkan *access.log* menyimpan data-data yang berupa IP akses web, *timestamp*, *status code*, besar *bandwith* yang diakses, keterangan lokasi yang diakses, dan engine yang digunakan [17].

- Error Log

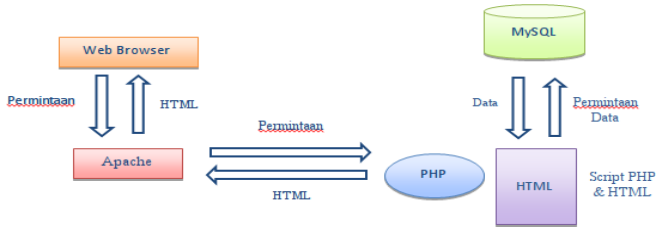
Error.log merupakan *file* yang sangat penting untuk mengelola debug dan misconfiguration yang terjadi di *server*. *Error.log* dapat mempermudah sistem administrator dalam mencari kesalahan pada web *server*. Gambar 4 menunjukkan contoh dari *error.log*. Item pertama pada entri log ([Wed Oct 11 14:32:52 2000]) merupakan tanggal dan waktu pesan. *Item* kedua ([error]) adalah daftar tingkat *error* yang dilaporkan. Direktif log level digunakan untuk mengontrol jenis kesalahan yang dikirim ke *error.log* dengan membatasi tingkat fatal kesalahan. Item ketiga ([client 127.0.0.1]) memberikan alamat IP dari klien yang menghasilkan ke *error. Item* selanjutnya merupakan pesan itu sendiri yang dalam hal ini menunjukkan bahwa *server* telah dikonfigurasi untuk menolak akses klien [9].

- Access Log

Access.log bertugas untuk mencatat semua proses *request* yang terjadi di server. Menyimpan informasi dalam *access.log* adalah hanya awal dari *log manajement.* Langkah berikutnya adalah menganalisis informasi untuk menghasilkan statistik proses yang terjadi di server. Gambar 5 merupakan contoh format *access.log* dalam bentuk *string yang* dapat dikonfigurasi [9].



Gambar 1 Arsitektur *Web Server*



Gambar 2 Cara Kerja Apache *Web Server*

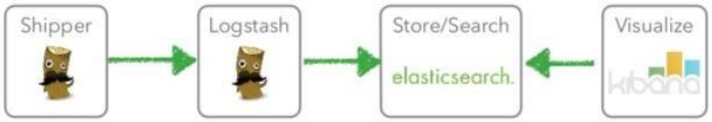
D. *Kinerja Web Server*

Apa Memonitoring kinerja sebuah web server adalah manajemen yang berguna untuk menganalisis dan memantau server yang bekerja secara optimal. Beberapa hal yang perlu dilakukan untuk memonitorng kinerja web server ialah melakukan pengujian terhadap waktu, utilization, dan error [7].

Waktu terdiri dari ART, PRT, dan Uptime. ART (Average Response Times) mengacu pada jumlah waktu yang diperlukan server untuk mengembalikan permintaan ke pengguna. ART sangat dipengaruhi oleh faktor-faktor seperti bandwidth jaringan, jumlah pengguna, dan jumlah/jenis permintaan yang dilakukan PRT (Peak Response Times) memiliki fungsi yang sama seperti ART, yang membedakannya adalah pengambilan *data request* yang maksimal. *Uptime* adalah lama waktu sebuah *server* dapat memberikan layanan kepada sistem administator, yang berfungsi untuk melihat jumlah proses waktu yang telah berjalan dengan baik pada server [4].

Utilization terdiri dari CPU dan Memori. Me-*monitoring* CPU merupakan hal yang perlu dilakukan untuk mengetahui jumlah waktu CPU yang digunakan pada proses permintaan (request). CPU berfungsi untuk mengukur seberapa besar kapasitas prosesor yang digunakan CPU untuk menangani proses *request*. Memori *utilization* mengacu pada jumlah memori yang digunakan oleh aplikasi web saat memproses permintaan. Memori berfungsi untuk mengukur seberapa besar memori yang digunakan pada proses *request* [5].

Error merupakan tingkat kesalahan yang dapat terjadi selama proses server berjalan. Tingkat kesalahan atau *error* biasanya dihitung sebagai persentase dari masalah relatif terhadap semua permintaan, dan itu menggambarkan berapa banyak kode status Respon HTTP menunjukkan kesalahan yang terjadi pada server [4].



Gambar 3 ELK Arsitektur [8]

```
[Wed Oct 11 14:32:52 2000] [error] [client 127.0.0.1] client denied
by server configuration:  /export/home/live/ap/htdocs/test
```

Gambar 4 Contoh Error Log [9]

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
CustomLog logs/access.log common
```

Gambar 5 Contoh Access Log [9]

III. HASIL DAN PEMBAHASAN

A. Analisis Sistem

Analisis sistem terdiri dari dua yaitu kebutuhan fungsional dan non-fungsional. Kebutuhan fungsional, sistem harus dapat berjalan pada Sistem Operasi Ubuntu 16.06, sistem dapat mengumpulkan, menyimpan, dan menampilkan data log. Kebutuhan non-fungsional terbagi dua, yaitu kebutuhan perangkat keras dan lunak. Pada kebutuhan perangkat keras, sistem menggunakan spesifikasi komputer dengan RAM 4GB, CPU 2Core, dan Memori 500MB. Kebutuhan lunak, sistem monitoring harus berjalan pada *Apache* web server, harus terintegrasi satu dengan yang lainnya antara server *Apache* dengan sistem *management log*, sistem dapat menangani lebih dari 1000 *request* per menit, dan Hasil kinerja pada sistem akan ditampilkan dalam bentuk visualisasi.

B. Analisis Kebutuhan Kinerja

Analisis kinerja sistem yang berjalan dengan baik memerlukan beberapa kebutuhan yang akan dicerminkan dalam aspek-aspek berikut:

1) Waktu

- *Average Response Time* (ART) untuk mengukur berapa lama waktu yang dibutuhkan *server* untuk mengembalikan *request*.
- *Peak Response Time* (PRT) untuk mengukur berapa lama waktu yang dibutuhkan server untuk mengembalikan *request* yang paling maksimal.
- *Uptime* untuk mengukur kinerja proses selama *server* berjalan.

2) *Utilization*

- CPU berfungsi untuk mengukur seberapa besar kapasitas prosesor yang digunakan CPU untuk menangani proses *request*.
- Memori berfungsi untuk mengukur seberapa besar memori yang digunakan pada proses *request*.

3) *Error* untuk melihat tingkat kesalahan yang dapat terjadi selama server berjalan.

4) Performa Indikator (PI)

Performa indikator merupakan ukuran akhir yang memberikan informasi tingkat kinerja web server. Pada Tabel I, dapat dilihat keterangan performa indikator yang akan digunakan pada penelitian ini sebagai hasil akhir dari pengujian terhadap Apache web server. Dengan menggunakan hasil *survey* Ka I. Pun 2012 tentang indikator kinerja web server[7] maka, dapat ditentukan bahwa waktu yaitu, PRT dan Uptime menjadi indikator kinerja utama, kemudian *error* menjadi indikator kedua yang penting, dan sisa indikator lain akan dianggap setara. Performa indikator merupakan nilai gabungan dari ketiga indikator tersebut.

TABEL I

PERFORMA INDIKATOR

No.	Performa Indikator	Keterangan	Bobot
1.	Average Response Time (ART)	Waktu yang diperlukan untuk memproses permintaan client (ms)	10
2.	Peak Response Time (PRT)	Waktu (max) yang diperlukan untuk memproses permintaan client (ms)	25
3.	Uptime	Jumlah kinerja proses selama server berjalan	25
4.	CPU	Jumlah pemakaian CPU untuk memproses data	10
5.	Memori	Jumlah pemakaian memori yang digunakan	10
6.	Error	Jumlah error selama server proses request	20

C. Perancangan dan Integrasi Sistem

Tahap ini akan menjelaskan tentang cara merancang dan menitegrasikan antara kedua aplikasi yaitu Apache dan ELK (*Elasticsearch, Logstash, Kibana*). Integrasi adalah keterkaitan antar sub sistem sehingga data dari satu sistem secara rutin dapat melintas, menuju atau diambil oleh satu atau lebih sistem yang lain. Dalam penelitian ini, server *Apache* dan *management log system* harus saling terintegrasi satu dengan yang lainnya yang bertujuan untuk mengetahui kinerja sistem pada server *Apache*.

D. Perancangan Implementasi dan Pengujian

Implementasi merupakan tahap selanjutnya dari analisis dan perancangan sistem. Implementasi ini dilakukan untuk mengetahui kesiapan sistem dalam wujud sebenarnya agar dapat beroperasi dan dilakukan pengujian. Proses implementasi meliputi proses implementasi perangkat keras, implementasi perangkat lunak, dan implementasi integrasi. Dalam proses implementasi perangkat lunak menggunakan aplikasi untuk memantau kinerja web server ini. Sedangkan, untuk proses implementasi integrasi akan mengintegrasikan kedua aplikasi.

1) Implementasi Perangkat Keras

Perangkat keras yang digunakan untuk membangun sistem aplikasi ini yaitu dengan menggunakan satu buah komputer yang mempunyai spesifikasi seperti berikut:

- *Processor* Intel Core i3-2100 CPU 3.10 GHz
- RAM 6 GB
- *Harddisk* 500 GB

2) Implementasi Perangkat Lunak

Dalam pemantauan kinerja *web server* ini dibutuhkan perangkat lunak yang bertujuan membantu pengembangan sistem. Adapun beberapa perangkat lunak yang digunakan yaitu, seperti berikut:

- *Ubuntu* 16.04 64-bit sebagai sistem operasi.
- *Apache2* sebagai Web *ServerJava OpenJDK* atau *Oracle Java*
- *Logstash* sebagai pengumpul *log*
- Elasticsearch sebagai tempat penyimpanan log.
- Beat sebagai shipper.
- Kibana sebagai web interface atau visualisasi.
- Mozilla Firefox sebagai web browser.

C. Integrasi Perangkat Lunak

Setelah memastikan aplikasi telah terinstal dengan baik, tahap selanjutnya ialah melakukan intergrasi sistem.

D. Pengujian Sistem

Pengujian merupakan tahap penting yang harus dilakukan untuk mengetahui proses-proses dari tahap yang sudah dilakukan sebelumnya. Pengujian ini dilakukan bertujuan untuk mengetahui kinerja dari Apache *web server*. Ada dua tahap pengujian yang akan dilakukan untuk memonitoring kinerja Apache *web server* yaitu, pengujian umum dan khusus. Pengujian umum akan memonitoring kinerja ART, PRT, CPU, dan Memori, sedangkan pengujian khusus akan memonitoring kinerja Uptime, dan Error. Tahap pengujian ini menggunakan aplikasi pendukung yaitu Crontab, yang berfungsi untuk mengirim perintah yang dijadwalkan secara *real-time*.

1. Pengujian Umum

Pada tahap pengujian umum, dilakukan lima kali pengujian dengan interval waktu selama satu jam dan mengirim 1.000 request per menit. Setiap satu proses pengujian, dilakukan dengan jeda waktu selama 30 menit sebelum proses dilanjutkan kembali. Pengujian ART dan PRT, akan mengukur rata-rata, dan range untuk mengetahui kinerja web server, sedangkan pengujian CPU dan Memori akan dilihat seberapa banyak penggunaanya selama proses request. Crontab akan mengirim data sesuai dengan waktu yang telah dijadwalkan/diatur ke Apache web server.

- ART

ART duji untuk mengetahui jumlah waktu yang dibutuhkan oleh *server* pada proses *request.* Tabel II menunjukkan hasil pengujian ART.

Pada Gambar 5, dapat dilihat hasil rata-rata dari pengujian ART. Proses pada pengujian pertama menunjukkan hasil proses waktu pengiriman *request* tinggi yang berarti proses *request* lambat. Pada proses pengujian kedua dan ketiga, hasil

Pengujian menunjukkan penurunan yang cukup jauh dari proses pertama. Hal ini dikarenakan sistem mengalami titik jenuh. Pada pengujian ketiga dan kelima sistem mengalami peningkatkan kembali. Grafik tersebut menunjukkan terjadinya proses naik dan turun ART dalam melakukan proses *request* selama pengujian. Hasil ini menunjukkan bahwa server selama proses pengujian berjalan dengan baik dan semakin efisien.

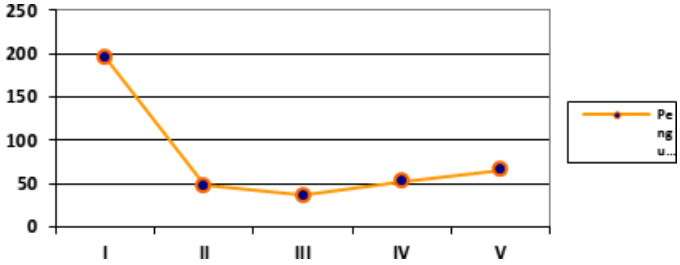
- PRT

PRT diuji untuk mengetahui jumlah waktu maksimum yang dibutuhkan oleh *server* pada proses *request.* Tabel III menunjukkan hasil pengujian PRT.

Hasil yang ditunjukkan pada Gambar 7, dapat dilihat untuk proses pengujian PRT tidak berbeda dengan ART, yang membedakannya hanyalah proses *request* yang terdapat pada PRT ialah nilai maksimal dari kecepatan waktu proses pengiriman request ke server. Proses pada pengujian pertama menunjukkan hasil proses waktu pengiriman request yang tinggi yang berarti proses request lambat. Pada proses pengujian kedua dan ketiga, hasil pengujian menunjukkan penurunan yang cukup jauh dari proses pertama. Hal ini dikarenakan sistem mengalami titik jenuh.

TABEL II
HASILPENGUJIAN ART

Pengujian Ke-	ART (ms)
I	195,11
II	47,70
III	36,37
IV	52,40
V	66,17
Rata-rata	**79,55**
Range	**158,74**



Gambar 6

Hasil Pengujian ART

TABEL III
HASIL PENGUJIAN ART

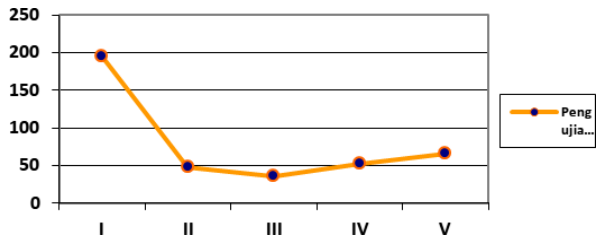
Pengujian Ke-	PRT (ms)
I	240,83
II	45,58
III	46,27
IV	55,67
V	67,58
Rata-rata	**91,186**
Range	**195,25**

Pada pengujian keempat dan kelima sistem mengalami peningkatkan kembali. Grafik tersebut menunjukkan terjadinya proses naik dan turun PRT dalam melakukan proses request selama pengujian. Hasil ini menunjukkan bahwa server selama proses pengujian berjalan dengan baik dan semakin efisien.

- CPU

Memonitoring CPU berguna untuk mengetahui seberapa banyak prosesor yang digunakan server selama proses *request.* Tabel IV menunjukkan hasil pengujian CPU.

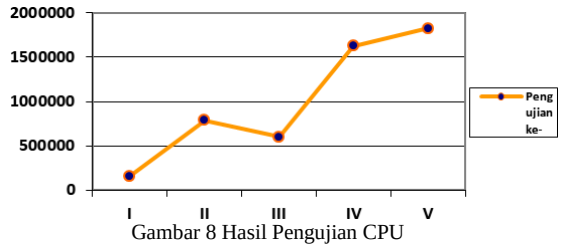
Pada Gambar 8, dapat dilihat grafik penggunaan prosesor CPU selama proses pengujian. Terlihat pada gambar, bahwa grafik mengalami peningkatan penggunaan prosesor selama proses pengujian dari pengujian pertama sampai kelima. Peningkatan ini disebabkan semakin banyak penerimaan request yang diproses oleh CPU sehingga, prosesor yang digunakan pun semakin meningkat.



Gambar 7 Hasil Pengujian PRT

TABEL IV
HASIL PENGUJIAN CPU

Pengujian Ke-	CPU (Hz)
I	161.317
II	791.077
III	600.172
IV	1.624.053
V	1.823.148
Rata-rata	**999,953**
Range	**1.661.183**



Gambar 8 Hasil Pengujian CPU

- Memori

Memonitoring memori berguna untuk mengetahui jumlah memori yang digunakan selama proses request. Tabel V menunjukkan hasil dari kelima pengujian yang dilakukan pada Apache web server.

Pada Gambar 9, memori yang digunakan selama proses request menggunakan memori yang cukup banyak. Selama proses kelima pengujian, memori juga mengalami naik dan penurunan yang disebabkan banyaknya request yang diproses.

2. Pengujian Khusus

Pada tahap pengujian ini, yang akan diuji adalah Uptime dan Error. Pengujian Uptime akan dilakukan hanya satu kali pengujian yang diproses selama 24 jam dengan mengirim 1.000 *request* per menit. Pengujian Error dilakukan selama satu jam dengan mengirim 100.000 *request* per menit. *Error* terjadi karena halaman yang diminta tidak ada. Crontab akan mengirim data sesuai dengan waktu yang telah dijadwalkan/diatur ke Apache web server.

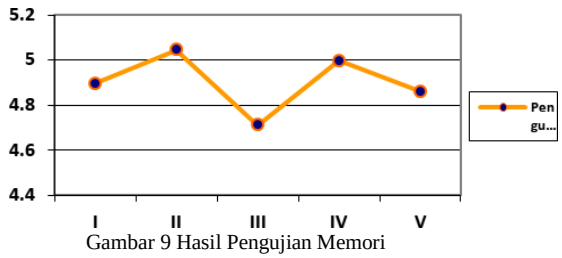
- Uptime

Uptime diuji untuk mengetahui jumlah waktu yang dibutuhkan bahwa *server* telah terjaga dan berjalan dengan baik. Hasil pengujian *Uptime* diharapkan berjalan dengan baik dengan presentase 100%. Pada Gambar 6 menunjukkan hasil kinerja *Uptime* pada Apache *web server* yang telah diuji selama 24 jam melalui visualisasi Kibana.

Gambar 10 menunjukkan proses pengujian Uptime yang telah dilakukan pada pukul 04.00 (12 Juli 2017) sampai pukul 04.00 (13 Juli 2017) kinerja server berjalan 100% dengan baik, ini dibuktikan dengan grafik yang terus naik dan tidak terjadi penurunan/*down.*

TABEL IV
HASIL PENGUJIAN MEMORI

Pengujian Ke-	Memori (GB)
I	4.869
II	5.046
III	4.71
IV	4.996
V	4.861
Rata-rata	**4.048**
Range	**4.575**



Gambar 9 Hasil Pengujian Memori

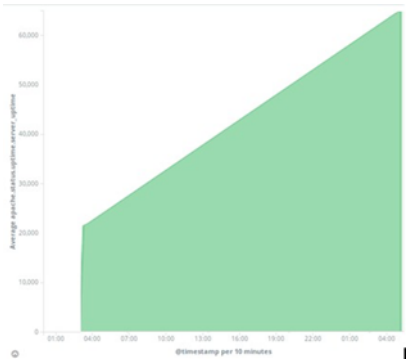
- Error

Gambar 11 menunjukkan tahap pengujian *error*. Pengujian ini akan dilihat dari tingkat kesalahan yang terjadi pada setiap proses request. Pengujian dilakukan untuk melihat seberapa banyak kesalahan yang terjadi pada server. Pengujian error disebabkan karena halaman yang diminta tidak ada.

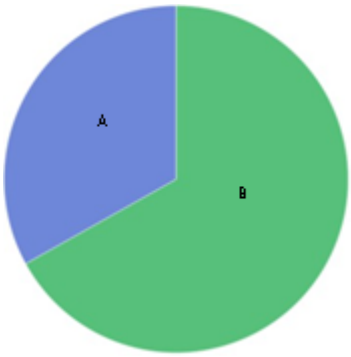
Gambar 11 menunjukkan jumlah error yang terjadi selama proses pengujian 100.000 request per menit selama satu jam. Dari 100.000 request yang dikirim, terdapat 33.780 error yang terjadi. Gambar 12 merupakan hasil error dalam bentuk jumlah (angka).

Semua grafik dan tabel harus terletak di posisi rata tengah. Grafik dan tabel yang besar dapat direntangkan pada kedua kolom. Setiap tabel atau grafik yang mencakup lebar lebih dari 1 kolom harus diposisikan di bagian paling atas atau di bagian paling bawah halaman.

Grafik diperbolehkan berwarna. Grafik tidak boleh menggunakan pola titik-titik karena ada kemungkinan tidak dapat dicetak sesuai aslinya. Gunakan hanya warna-warna solid yang kontrasnya baik untuk tampilan di layar komputer



Gambar 10 Hasil Pengujian Uptime



Gambar 11 (A) Error Log dan (B) Access Log

source: Descending	Count
/var/log/apache2/access.log	68,290
/var/log/apache2/error.log	33,780

Gambar 12 Hasil Error Log

3. Perhitungan Performa Indikator

Perhitungan performa indikator merupakan pengujian yang dilakukan untuk mendapatkan skor akhir dari kinerja Apache web server. Ada tiga tahap yang akan dilakukan dalam menghitung performa indikator yaitu, nilai indikator performa, menormalisasi nilai indikator performa, dan perhitungan indikator performa.

a. Nilai Indikator Performa

Nilai indikator performa ini diperoleh dari hasil pengujian ketiga kinerja web server dan mengambil nilai rata-rata dari hasil pengujian tersebut. Hasil dari nilai tersebut, yaitu:

- ART = 79,55
- PRT = 91,186
- Uptime = 43.000
- CPU = 999,953
- Memori = 27.200.000
- Error = 33.780

b. Normalisasi Indikator Performa

Tahap normalisasi indikator performa ini dilakukan untuk membuat nilai-nilai indikator performa menjadi scala terdekat, yang berfungsi agar hasil nilai ketiga indikator kinerja tidak jauh berbeda. Teknik normalisasi ini biasa disebut *feature scaling.*

Rumus:

$$Z^1 = \frac{X - \min(x)}{x(\max) - x(\min)} \quad (1)$$

Ket:

X= hasil dari rata-rata pengujian

min(x)= nilai minimum pengujian

max(x)= nilai maksimum pengujian

Z= hasil normalisasi

Tabel VI menunjukkan hasil dari normalisasi setiap kinerja indicator performa.

TABEL VI
HASIL INDIKATOR PERFORMA

ART	PRT	Uptime	CPU	Memori	Error
0,272	0,234	0,5	0,505	0,781	0,51

TABEL VIII
HASIL PENGUJIAN MEMORI

Performa Indikator	Bobot	Nilai Indikator	Hasil
Uptime	25	0,5	12,5
Error	20	0,51	10,2
Memori	10	0,781	7,81
PRT	25	0,234	5,85
CPU	10	0,505	5,05
ART	10	0,272	2,72

c. Perhitungan Indikator Performa

Perhitungan indikator performa dilakukan dengan mengkalikan bobot yang ditentukan dengan nilai indikator yang telah dinormalisasi, sehingga diperoleh hasil nilai keseluruhan kinerja.

Tabel 7 menunjukkan hasil perhitungan dari performa indikator dari setiap perfoma web *server*, yaitu waktu, utilization, dan error. Hasil tersebut diperoleh dari bobot yang dikalikan dengan nilai indikator yang telah dinormalisasi, sehingga diperoleh hasil tersebut. Dapat diilihat, hasil dari kinerja Uptime mendapatkan hasil yang terbesar yaitu 12,5, sedangkan ART mendapatkan hasil terkecil yaitu 2,72.

Hasil tersebut menunjukkan bahwa memonitoring kinerja Apache menggunakan log management system perlu dilakukan untuk mengetahui pada waktu kapan saja web server banyak diakses oleh pengguna dan apakah selama proses web server berjalan tidak terdapat waktu down.

IV. KESIMPULAN

Berdasarkan hasil pengujian pada penelitian ini, dapat disimpulkan bahwa Apache web *server* dapat menghasilkan *log* yang berisi informasi lengkap, sehingga sistem administrator dapat memonitoring dan menganalisis kinerja Apache web server.

Aplikasi *open source log management system* ELK terdiri dari tiga aplikasi terpilih yaitu Elasticsearch, Logstash, dan Kibana yang saling terintegrasi satu dengan yang lainnya. Logstash berfungsi sebagai pengumpul data, yang dibantu oleh Beat mengirim data tersebut ke Elasticsearch. Elasticsearch yang mengirim data tersebut yang akan dikirim ke Kibana untuk di visualisasikan.

Memonitoring Apache web server dilakukan dengan mengamati kinerja ketiga indikator performa yaitu, waktu, utilization, dan error, yang diakhiri dengan melakukan pembobotan dan hasil nilai indikator performa.

Kekurangan yang dirasakan pada penelitian ini adalah pengawasan kinerja Apache web server dengan menggunakan log management system diharapkan dapat dikembang lebih luas lagi dengan memonitoring firewall, database, dan lainnya, serta ditambahnya dengan sistem alert.

Kekurangan yang lainnya yaitu penentuan kinerja indikator performa Apache web server dapat dibuat dengan sistem penscalaan untuk mendapatkan hasil apakah kinerja web server tersebut baik atau tidak.

DAFTAR REFERENSI

[1] A. Br, (2013). "*Apache"* [Online]. Available: http://sir.stikom.edu/218/6/BAB%20III.pdf [June 9, 2017].

[2] Apache, (2017). "*Apache: HTTP Server Project"* [Online]. Available: https://httpd.apache.org/ [January 22, 2017).

[3] C. Phillips, S. Kevin, et al. "*Logging and Log Management"*. Waltham: Elsevier, 2013, pp. 267-303.

[4] Load Storm. "Load Testing Metric Explained." Internet: https://loadstorm.com/load-testing-metrics/, 2017 [July 12, 2017].

[5] M. Anicas. "How to use Kibana Dashboard and Virtualizations." Internet: https://www.digitalocean.com/community/tutorials/how-to-use-kibana-dashboards-and-visualizations, March. 12, 2015 [November 18, 2016].

[6] NN. The Apache Platform and Architecture. 2016, pp. 22-23.

[7] P. I. Kai. "Key Performance Indicator for Traffic Intensive Web- Enable Business Process." Business Process Management Journal, Vol. 18, No. 2, pp. 10-11, 2012.

[8] R. Alexander. "Using Elasticsearch, Logstash, and Kibana to Create Realtime Dashboards". 2014.

[9] The Apache Software Foundation. "*Apache HTTP Server Documentation Version 2.2.* 2014", pp. 43-47.

Claudia Adyan Tarigan, kelahiran Medan 1994, menyelesaikan S1 Jurusan Teknologi Informasi bidang Media & Internet di Institut Teknologi Harapan Bangsa pada 2017. Bidang penelitian: web server dan *log management system.*

Ventje Jeremias Lewi Engel, kelahiran Ketapang, Kalimantan Barat tahun 1990 dan memperoleh gelar Sarjana Teknik dari Sistem dan Teknologi Informasi ITB pada tahun 2012 dan Magister Teknik dari Informatika ITB pada tahun 2013. Minat penelitian pada bidang analisis data, *internet of things*, dan keamanan informasi. Saat ini aktif sebagai staf pengajar di Departemen Sistem Komputer dan Teknik Elektro Institut Teknologi Harapan Bangsa.

Dina Angela, kelahiran Bandung 1974, menyelesaikan S1 Jurusan Teknik Elektro bidang Telekomunikasi di Universitas Kristen Maranatha pada 1999 dan S2 Jurusan Teknik Elektro bidang Telekomunikasi di Institut Teknologi Bandung pada 2003. Bidang penelitian: antena dan propagasi dan sistem komunikasi.