p- ISSN : 2407 – 1846
e-ISSN : 2460 – 8416
Website : jurnal.umj.ac.id/index.php/semnastek

DESAIN DAN IMPLEMENTASI LOG EVENT MANAGEMENT SERVER MENGGUNAKAN ELASTICSEARCH LOGSTASH KIBANA (ELK STACK)

Muhamad Nur Arifin, Sugiartowo, Emi Susilowati
Teknik Informatika, Universitas Muhammadiyah Jakarta, Jl. Cempaka Putih Tengah 27, 10510
contact.arifin@gmail.com

Abstrak

Penggunaan server yang harus berjalan selama 24 jam dan services yang berjalan pada server tersebut pasti menghasilkan sebuah log yang cukup banyak. Hal ini mengharuskan seorang sistem administrator dalam pengecekannya masih harus berinteraksi langsung dengan server tersebut. Dalam penelitian ini bermaksud untuk melakukan suatu perancangan untuk membangun Log Event Management Server menggunakan ELK Stack (Elasticsearch Logstash Kibana) yang dapat memudahkan dalam membaca sekaligus menganalis log services pada server. Implementasi Log Event Management Server dalam penelitian kali ini menggunakan CentOS 7 Server, dan Ubuntu 14.04 sebagai server client dengan SSH services yang terpasang. Dari hasil pengujian ELK Stack sebagai Log Event Management yang telah dibangun dengan tingkat keberhasilan 100% menunjukan bahwa semua log services SSH yang terjadi pada server client dapat dikirimkan secara realtime ke server utama ELK Stack sekalipun isi file log pada server client tersebut dihapus.

Kata kunci: Server, Log, ELK Stack, SSH Services.

Abstract

The use of servers that have to run continuously for 24 hours and services that run produce a lot of log. This requires that the system administrator in his checks still have to do with the server. In this research intends to conduct a design to build Log Event Management Server using ELK Stack (Elasticsearch Logstash Kibana) that can make it easier to read and analyze log services on the server. The implementation in this research uses CentOS 7 Server, and Ubuntu 14.04 as the client server with SSH services built in. From the results of ELK Stack testing as a Log Event Management that has been built with a success rate of 100%, it shows that all SSH log services that occur on the client server can be sent in realtime to the ELK Stack main server even though the log file contents on the client server are deleted.

Keywords: Server, Log, ELK Stack, SSH Services.

PENDAHULUAN

Perkembangan dunia teknologi saat ini sangat cepat dan selalu ada inovasi-inovasi terbaru yang hadir. Salah satu inovasi yang bisa kita rasakan adalah hadirnya sebuah sistem komunikasi jaringan komputer melalui sebuah perangkat server. Server adalah suatu unit komputer yang berfungsi untuk menyimpan informasi dan untuk mengelola suatu jaringan komputer. Server akan melayani seluruh client atau workstation yang terhubung ke jaringannya dan merupakan perangkat utama dalam sebuah sistem komunikasi jaringan yang berfungsi sebagai penyedia layanan atau service [4]. Sebagai penyedia layanan, sistem yang berjalan pada server harus mampu berjalan selama 24 jam penuh, sehingga untuk memantau jalannya service pada server diperlukan pencatatan dalam bentuk log event management yang bersifat realtime untuk mencatat aktifitas service yang berjalan pada server.

Kendala umum yang terjadi adalah seorang administrator jaringan harus harus secara manual untuk melakukan pembacaan log service pada server dan harus berinteraksi langsung dengan server yang memakan waktu cukup lama. Masalah selanjutnya yaitu server

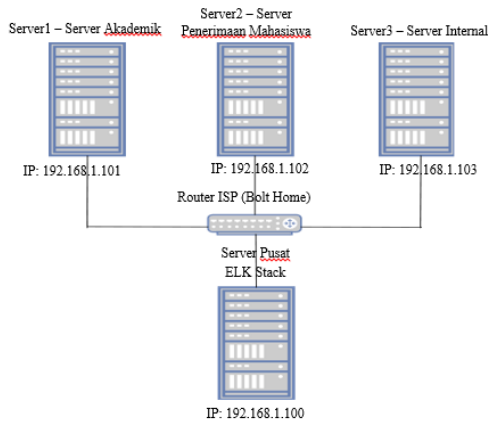
p- ISSN : 2407 – 1846
e-ISSN : 2460 – 8416
Website : jurnal.umj.ac.id/index.php/semnastek

yang harus berjalan 24 jam penuh menghasilkan log service dalam jumlah banyak.

Berdasarkan permasalahan diatas, maka perlu dibuatnya suatu log event management yang mampu meringankan dan memudahkan dalam membaca sekaligus menganalisis log service pada server. Dalam hal ini Elasticsearch Logstash Kibana (ELK Stack) merupakan komponen yang tepat dalam membangun log event management yang dapat memberi insight kepada sistem administrator mengenai tren, statistik, dan anomali yang terjadi. ELK Stack dirancang untuk digunakan sebagai solusi terintegrasi [1]. Elasticsearch adalah sebuah platform berbasis opensource yang dibangun diatas Apache Lucene, distributable [1], dan merupakan mesin pencari dan pengindeksan [5]. Proses pencarian pada elasticsearch dibatasi oleh alamat url dan kueri nya hanya berbeda dalam kondisi filter [3]. Logstash bertugas untuk mengumpulkan dan mengurai data dan mengirim hasilnya ke elasticsearch untuk pengindeksan [2]. Kibana adalah sebuah platform yang dirancang untuk visualisasi layer pada komposisi ELK Stack dan bertugas menampilkan serta mencari data pada elasticsearch [6]. Pada peneletian kali ini akan dibahas bagaimana Desain dan Implementasi Log Event Management Server Menggunakan Elasticsearch Logstash Kibana (ELK Stack) dengan pengujian pada services SSH.

HASIL DAN PEMBAHASAN

Berikut ini adalah topologi jaringan pada perancangan Log Event Management Server.



Gambar 1. Topologi Jaringan ELK-Stack

Pada penilitan ini menggunakan 1 buah server utama untuk ELK Stack dan 3 buah server yang menjadi server-client atau server yang nantinya akan dipantau log nya. Perancangannya akan dibahas menjadi 2 bagian, yaitu pertama perancangan pada Server ELK-Stack dan kedua perancangan pada Server-Client.

a. Perancangan Server ELK-Stack

Pada konfigurasi *server* ELK-Stack atau *server* pusat yang nantinya akan menjadi *server* utama dari topologi *Log Event Management Server*, tahapan awal adalah kita *install* terlebih dahulu paket-paket aplikasi yang kita gunakan, dengan menuliskan kode otomatis *install* berikut ini.

```
#!/bin/bash
 #script automate install ELK Stack

 #import GPG-Key Elastic Stack
 rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch
 printf "\n\n==========Success Import GPG-Key!==========\n\n"
 #make repo Elastic Stack
 touch /etc/yum.repos.d/elasticsearch.repo
 #input repo Elastic Stack
 printf "[elasticsearch-6.x]\nname=Elasticsearch repository for 6.x packages\nbaseurl=https://artifacts.elastic.co/packages/6.x/yum\ngpgcheck=1\ngpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch\nenabled=1\nautorefresh=1\ntype=rpm-md" > /etc/yum.repos.d/elasticsearch.repo
 printf "\n\n==========Success Add Repo!==========\n\n"
 #install Elasticsearch
 yum install -y elasticsearch
 printf "\n\n==========Success Install Elasticsearch!==========\n\n"
 #install Logstash
 yum install -y logstash
 printf "\n\n==========Success Install Logstash!==========\n\n"
 #install Kibana
 yum install -y kibana
 printf "\n\n==========Success Install Kibana!==========\n\n"
```

Lalu simpan script tersebut dengan nama “install.sh” dan kemudian jalankan script dengan perintah berikut:

“bash install.sh”

p- ISSN : 2407 – 1846
e-ISSN : 2460 – 8416
Website : jurnal.umj.ac.id/index.php/semnastek

Selanjutnya kita mulai konfigurasi dari Elasticsearch terlebih dahulu. Pertama kita konfigurasi file "elasticsearch.yml" pada direktori aplikasi Elasticsearch seperti berikut

network.host: localhost
http.port: 9200
bootstrap.memory_lock: true

Kemudian konfigurasi lagi pada file "elasticsearch.service" untuk membuat Elasticsearch Limit Memory nya unlimited seperti berikut ini.

LimitMEMLOCK=infinity

Lalu kita konfigurasi juga pada *file* "sysconfig/elasticsearch", yang tujuan sama untuk membuat *Limit Memory* nya *unlimited* seperti berikut.

MAX_LOKCKED_MEMORY=unlimited

Terakhir kita buat Elasticsearch auto-run ketika server mengulang kembali, dan setelah itu kita jalankan services Elasticsearch dengan perintah berikut ini

"systemctl daemon-reload"
"systemctl enable elasticsearch"
"systemctl start elasticsearch"

Tahap selanjutnya konfigurasi pada Kibana agar memvisualisasi semua log server yang ada. Pertama-tama kita konfigurasi file "kibana.yml" pada direktori sistem Kibana seperti berikut ini.

server.port: 5601
server.host: "localhost"
elasticsearch.url: "http://localhost:9200"

Lalu kita buat Kibana auto-run ketika server mengulang kembali, dan setelah itu jalankan services Kibana dengan perintah berikut ini.

"systemctl enable kibana"
"systemctl start kibana"

Selanjutnya kita *install* Nginx untuk dapat memudahkan kita mengakses halaman dari Kibana dengan menggunakan perintah berikut ini

"yum -y install epel-release nginx httpd-tools"

Setelah itu konfigurasi file "nginx.conf" pada direktori sistem dari Nginx, dengan memberi pagar disetiap konfigurasi. Lalu buat file konfigurasi baru untuk Kibana dengan nama "kibana.conf" pada direktori Nginx, dengan konfigurasinya seperti berikut.

```
server {
        listen 80;

        server_name IP;

        auth_basic "Restricted Access";
        auth_basic_user_file /etc/nginx/.kibana-user;

        location / {
        proxy_pass http://localhost:5601;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
        }
}
```

Kemudian buat 2 user yang dapat mengakses halaman Kibana dengan perintah berikut:

"sudo htpasswd -c /etc/nginx/.kibana-user itmanager"
"sudo htpasswd -c /etc/nginx/.kibana-user admin"

Terakhir kita buat Nginx auto-run ketika server mengulang kembali dan jalankan services Nginx dengan perintah berikut

"nginx –t"
"systemctl enable nginx"
"systemctl start nginx"

Tahap finalisasi untuk konfigurasi pada sisi server elk-stack adalah membuat konfigurasi untuk Logstash agar bisa input, parsing, dan output data dari log server client. Pertama kita buat file konfigurasi Logstash untuk input pada direktori sistem Logstash dengan nama

p- ISSN : 2407 – 1846
e-ISSN : 2460 – 8416
Website : jurnal.umj.ac.id/index.php/semnastek

"filebeat-input.conf", lalu konfigurasinya seperti berikut

```
#FILE KONFIG INPUT FILEBEAT
input {
  beats {
        port => 5443
        ssl => true
        ssl_certificate => "/etc/pki/tls/certs/logstash-forwarder.crt"
        ssl_key => "/etc/pki/tls/private/logstash-forwarder.key"
  }
}
```

Kemudian kita buat file konfigurasi Logstash untuk filter dengan nama "ssh-filter.conf", lalu konfigurasi nya seperti ini

```
#FILE KONFIG FILTER SSH
filter {
  if [type] == "syslog" {
  grok {
    match => { "message" => "%{SYSLOGTIMESTAMP:timestamp} Message forwarded from %{HOSTNAME:host_target}: sshd\[%{BASE10NUM}\]: Failed password for invalid user %{USERNAME:username} from %{IP:src_ip} port %{BASE10NUM:port} ssh2" }
    add_tag => "ssh_failed_login"
    add_tag => "invalid_user"
    add_tag => "ssh_login_event"
    }
  }

  if ("ssh_failed_login" not in [tags]) {
  grok {
    match => { "message" => "%{SYSLOGTIMESTAMP:timestamp} Message forwarded from %{HOSTNAME:host_target}: sshd\[%{BASE10NUM}\]: Failed password for %{USERNAME:username} from %{IP:src_ip} port %{BASE10NUM:port} ssh2" }
    add_tag => "ssh_failed_login"
    add_tag => "invalid_password"
    add_tag => "ssh_login_event"
    }
  }

  if ("ssh_successful_login" not in [tags]) {
  grok {
    match => { "message" => "%{SYSLOGTIMESTAMP:timestamp} Message forwarded from %{HOSTNAME:host_target}: sshd\[%{BASE10NUM}\]: Accepted password for %{USERNAME:username} from %{IP:src_ip} port %{BASE10NUM:port} ssh2" }
    add_tag => "ssh_successful_login"
    add_tag => "ssh_login_event"
    }
  }

  if ("ssh_successful_login" not in [tags]) {
  grok {
    match => { "message" => "%{SYSLOGTIMESTAMP:timestamp}
    %{SYSLOGHOST:syslog_hostname}

%{DATA:syslog_program}(?:\[%{POSINT:syslog_pid}\])?:
    %{GREEDYDATA:syslog_message}" }
    add_field => [ "received_at", "%{@timestamp}" ]
    add_field => [ "received_from", "%{host}" ]
    add_tag => "syslog_event"
    }
  }
}

output {
  elasticsearch { hosts => ["localhost:9200"]
  manage_template => false
  index => "%{[@metadata][beat]}-%{+YYYY.MM.dd}"
  document_type => "%{[@metadata][type]}"
  }
}
```

Kemudian kita generate sertifikat openSSL yang akan digunakan pada server client dengan perintah berikut

> "*openssl req -config /etc/pki/tls/openssl.cnf -x509 -days 3650 -batch -nodes -newkey rsa:2048 -keyout /etc/pki/tls/private/logstash-forwarder.key -out /etc/pki/tls/certs/logstash-forwarder.crt*"

b. Perancangan Server-Client

Konfigurasi pada sisi server client yaitu dengan memasang filebeat sebagai logshipper. Filebeat sendiri bertugas meneruskan log dari

p- ISSN : 2407 – 1846
e-ISSN : 2460 – 8416
Website : jurnal.umj.ac.id/index.php/semnastek

server-client ke logstash [1]. Untuk installasi aplikasi filebeat dengan perintah berikut:
"wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add –"
"wget https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-5.1.1-amd64.deb
dpkg -i filebeat-5.1.1-amd64.deb"

Kemudian konfigurasikan filebeat pada file "filebeat.yml" pada direktori sistem Filebeat.

```
# Paths that should be crawled and fetched. Glob based paths.
  paths:
    - /var/log/auth.log
# Exclude lines. A list of regular expressions to match. It drops the lines that are
  # matching any regular expression from the list.
  #exclude_lines: ['^DBG']
document-type: syslog
output.logstash:
  # The Logstash hosts
  hosts: ["192.168.1.97:5443"]
  bulk_max_size: 1024
  ssl.certificate_authorities: ["/etc/pki/tls/certs/logstash-forwarder.crt"]
  template.name: "filebeat"
  template.path: "filebeat.template.json"
  template.overwrite: false
```

Selanjutnya tinggal dijalankan services dari aplikasi Filbeat tersebut dengan perintah "*service filebeat start*"

Setelah perancangan selesai semua, selanjutnya pengujian hasil dari penelitian ini. Pengujian akan dibagi dalam beberapa tahap yaitu pertama pengujian kesalahan login password ssh pada setiap server, kedua pengujian kesalahan login user ssh pada server-ubuntu2 dan server-ubuntu3, ketiga melihat tingkat akurasi waktu pada setiap kejadian login ssh pada server-client ke ELK-Stack, dan terakhir penghapusan log pada server-ubuntu1

I. Pengujian Pertama

Pengujian pertama ini bertujuan untuk melihat hasil kesalahan login password ssh pada setiap server-client dan hasil yang terjadi.

```
C:\Users\Arifin>ssh user-ubuntu1@192.168.1.101
user-ubuntu1@192.168.1.101's password:
Permission denied, please try again.
user-ubuntu1@192.168.1.101's password:
Permission denied, please try again.
user-ubuntu1@192.168.1.101's password:
user-ubuntu1@192.168.1.101: Permission denied (publickey,password).
```

Gambar 2. Pengujian kesalahan login ssh server-1

```
C:\Users\Arifin>ssh user-ubuntu2@192.168.1.102
user-ubuntu2@192.168.1.102's password:
Permission denied, please try again.
user-ubuntu2@192.168.1.102's password:
Permission denied, please try again.
user-ubuntu2@192.168.1.102's password:
user-ubuntu2@192.168.1.102: Permission denied (publickey,password).
```

Gambar 3. Pengujian kesalahan login ssh server-2

```
C:\Users\Arifin>ssh user-ubuntu3@192.168.1.103
user-ubuntu3@192.168.1.103's password:
Permission denied, please try again.
user-ubuntu3@192.168.1.103's password:
Permission denied, please try again.
user-ubuntu3@192.168.1.103's password:
user-ubuntu3@192.168.1.103: Permission denied (publickey,password).
```

Gambar 4. Pengujian kesalahan login ssh server-3

Hasil yang terjadi bisa dilihat pada gambar dibawah ini. Terlihat semua kegagalan login dapat lansung masuk ke dalam dashboard kibana.

Gambar 5. Hasil visualisasi kesalahan login

II. Pengujian Kedua

Pengujian yang kedua yaitu membuat kesalahan login ssh dengan username yang tidak terdapat pada server-ubuntu2 dan server-ubuntu3 sehingga akan terjadi kesalahan login user dan username yang akan digunakan yaitu "asal-asal" & "coba-coba".

```
C:\Users\Arifin>ssh asal-asal@192.168.1.102
asal-asal@192.168.1.102's password:
Permission denied, please try again.
asal-asal@192.168.1.102's password:
Permission denied, please try again.
asal-asal@192.168.1.102's password:
asal-asal@192.168.1.102: Permission denied (publickey,password).

C:\Users\Arifin>ssh coba-coba@192.168.1.103
coba-coba@192.168.1.103's password:
Permission denied, please try again.
coba-coba@192.168.1.103's password:
Permission denied, please try again.
coba-coba@192.168.1.103's password:
coba-coba@192.168.1.103: Permission denied (publickey,password).
```

Gambar 6. Percobaan kesalahan login user

Hasil yang terlihat pada kibana server-elkstack adalah bertambahnya username di

p- ISSN : 2407 – 1846
e-ISSN : 2460 – 8416
Website : jurnal.umj.ac.id/index.php/semnastek

grafik ssh-gagal-login-user dengan user asal-asal dan coba-coba seperti gambar berikut.

Gambar 7. Hasil pengujian kesalahan login user

III. Pengujian Ke-tiga

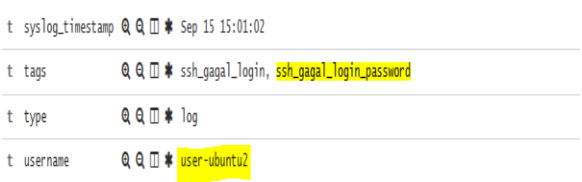
Pengujian yang ketiga melihat tingkat akurasi waktu antara terjadi nya kejadian kesalahan login ssh dan masuk log nya pada ELK-Stack.

```
C:\Users\Arifin>ssh user-ubuntu2@192.168.1.102
user-ubuntu2@192.168.1.102's password:
Permission denied, please try again.
user-ubuntu2@192.168.1.102's password:
Permission denied, please try again.
user-ubuntu2@192.168.1.102's password:
user-ubuntu2@192.168.1.102: Permission denied (publickey,password).

C:\Users\Arifin>time
The current time is: 15:01:06,68
```

Gambar 8. Pengujian kesalahan dan waktu kejadian

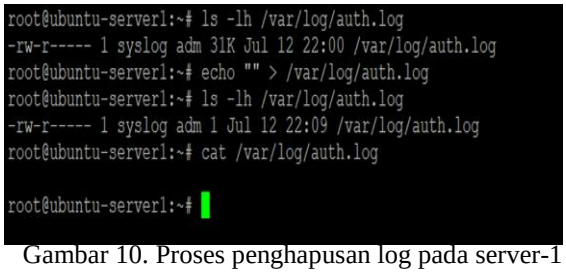
Hasil yang terlihat pada kibana server-elkstack adalah waktu nya sama dengan waktu kejadian kesalahan login ssh tersebut.



Gambar 9. Hasil pada Kibana Server-ELKStack

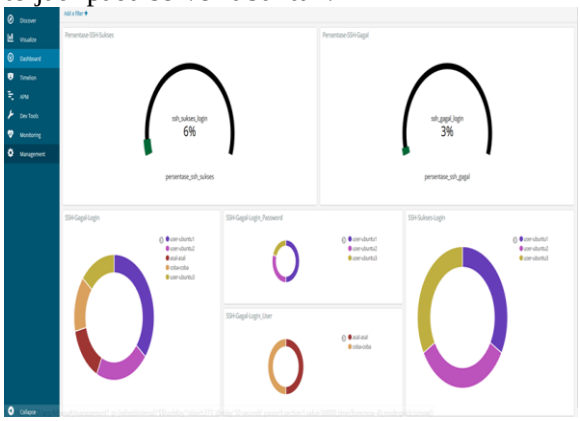
IV. Pengujian Ke-empat

Pengujian yang terakhir yaitu kita hapus semua log yang terjadi pada server-ubuntu1 dan melihat apakah ada pengaruh nya pada visualisasi yang terjadi di server-elkstack.



Gambar 10. Proses penghapusan log pada server-1

Hasil yang terjadi pada visualisasi kibana adalah tidak ada perubahan sedikitpun pada grafiknya, sehingga menandakan tidak berpengaruhnya penghapusan log yang sudah terjadi pada server-ubuntu1.



Gambar 11. Hasil visualisasi setelah log dihapus

Berikut ini tabel hasil pengujian yang sudah dilakukan diatas.

Tabel 1. Hasil Pengujian

No.	Pengujian	Hasil	Kesimpulan
1.	Kesalahan login password ssh pada setiap server-client	Semua kejadian kegagalan login ssh terlihat pada Kibana	Berhasil
2.	Kesalahan login user asal-asal dan coba-coba pada server-ubuntu2 dan server-ubuntu3	Semua kejadian kegagalan login ssh terlihat pada Kibana	Berhasil
3.	Waktu kejadian kesalahan login ssh dan masuk ke ELK-Stack Server	Tidak ada jeda antara kedua nya (real-time)	Berhasil
4.	Penghapusan log pada server-ubuntu1	Tidak terjadi perubahan pada ELK-Stack server	Berhasil

Dari hasil pengujian ELK Stack sebagai Log Event Management Server yang telah dibangun, tingkat keberhasilan nya yaitu 100% dari semua percobaan yang dilakukan.

SIMPULAN DAN SARAN

Simpulan

Dari hasil analisa penelitian ini, dapat disimpulkan sebagai berikut.

1. Setiap log yang terjadi pada server client dapat langsung dikirimkan ke server utama dengan ELK Stack.
2. Perubahan-perubahan yang ada secara real-time, sehingga tidak butuh waktu yang lama untuk melihat perubahan apa yang sedang terjadi.
3. Penghapusan log yang sudah terjadi pada server client tidak merubah hasil visualisasi pada server elk-stack utama.
4. ELK-Stack terbukti dapat menjadi solusi untuk implementasi sebuah Log Management Server yang membantu dalam memvisualisasikan hasil log pada server client

p- ISSN : 2407 – 1846
e-ISSN : 2460 – 8416
Website : jurnal.umj.ac.id/index.php/semnastek

dan membuatnya terekam walau terjadi penghapusan log.

Saran

Dalam membangun Log Management Server dengan ELK-Stack, dibawah ini merupakan saran-saran untuk penelitian selanjutnya:

1.Perlu adanya pengujian lebih lanjut terhadap services yang lain bukan hanya pada ssh server.
2.Penelitian lebih lanjut untuk mengoptimalkan ELK-Stack sebagai Log Management Server.

DAFTAR PUSTAKA

[1] Bajer, Marcin. 2017. Building an IoT Data Hub with Elasticsearch, Logstash and Kibana. 5th International Conference on Future Internet of Things and Cloud Workshops. doi:10.1109/FiCloudW.2017.101.

[2] Bagnasco, S., Berzano, D., Guarise, A., Lusso, S., Masera, M., & Vallero, S. 2015.Towards Monitoring-as-a-service for Scientific Computing Cloud applications using the ElasticSearch ecosystem. In Journal of Physics: Conference Series (Vol. 664, No. 2, p. 022040). IOP Publishing.

[3] Kononenko, Oleksii, Olga Baysal, Reid Holmes, and Michael W. Godfrey. 2014. Mining modern repositories with elasticsearch. Proceedings of the 11th Working Conference on Mining Software Repositories, pp.328-331.

[4] Sembiring, Jhony H.2010.Jaringan Komputer Berbasis Linux.Elex Media Komputindo, Jakarta.

[5] Talas, Andrei, Florin Pop, Gabriel Neagu. 2017.Elastic Stack in Action for Smart Cities: Making Sense of Big Data.IEEE.doi:10.1109/ICCP.2017.8117049.

[6] Zou, Q. 2015.A Novel Open Source Approach to Monitor EZproxy Users' Activities. Code4lib Journal, 29.