JTIM : Jurnal Teknologi Informasi dan Multimedia
Vol. 2, No. 1, Mei 2020, hlm. 12-20
p-ISSN : 2715-2529
e-ISSN : 2684-9151

Log Event Management Server Menggunakan *Elastic Search Logstash Kibana* (ELK Stack)

(*Log Event Management Server Using Kibana's Elastic Search Logstash*)

Walidatush Sholihah[1]*, Sangga Pripambudi[2], Anggi Mardiyono[3]

[1],[2]Teknik Komputer, Sekolah Vokasi IPB University
E-mail: walidah@apps.ipb.ac.id, anggabudi02@gmail.com
[3]Teknik Informatika, Politeknik Negeri Jakarta
E-mail: anggi.mardiyono@tik.pnj.ac.id

KEYWORDS:
ELK Stack, Kibana, Log, Server, SSH

ABSTRACT
This study aims to build an Event Management Server Log using ELK Stack (Elastic search Logstash Kibana) which can make it easier to read and analyze log services on the server. The Event Management Server log in this study uses CentOS 7 as the Central Server and CentOS 7 as a client-server with ssh services installed. This research consists of five stages. The stages are analysis, network design, server configuration, client configuration, and testing. The experimental results show that all ssh log services that occur on the client-server sent in realtime to the central server. Even though the contents of the log file on the client-server has deleted. In This study, in addition to sending logs, it can also display a percentage of success references.

KATA KUNCI:
ELK Stack, Kibana, Log, Server, SSH

ABSTRAK
Penelitian ini bertujuan untuk membuat Log Event Management Server menggunakan ELK Stack (Elastic search Logstash Kibana) yang dapat mempermudah dalam membaca dan menganalis log services pada server. Log Event Management Server pada penelitian kali ini menggunakan CentOS 7 sebagai server Pusat, dan CentOS 7 sebagai server client dengan ssh services yang sudah terpasang. Penelitian ini terdiri atas lima tahap. Tahapannya yaitu analisis, desaintopologi jaringan, konfigurasi server, konfigurasi client dan pengujian. Hasil percobaan menunjukkan bahwa semua log services ssh yang terjadi pada server client dapat dikirimkan secara realtime ke server utama. Sekalipun isi file log pada server client tersebut telah dihapus. Pada penelitian kali ini selain dapat mengirimkan log, juga dapat menampilkan presentase acuan keberhasilan.

I. PENDAHULUAN

Server adalah sebuah perangkat lunak yang memiliki tugas dan tanggung jawab untuk menyediakan layanan informasi kepada web [1]. Tugas server yaitu melayani seluruh yang terhubung ke jaringannya dan merupakan perangkat utama dalam sistem komunikasi jaringan berfungsi sebagai penyedia layanan [2] dan memiliki *log* yang sangat banyak. Masalah yang sering dialami seorang administrator jaringan yaitu harus secara manual untuk melakukan pembacaan *log service*. Aktivitas yang berjalan di sistem operasi server dicatat oleh *log service*. Untuk memeriksa *log service*, seorang administrator seringkali harus langsung berinteraksi dengan *server*. Hal ini tentunya kurang efektif dan efisien. Sebuah server tentunya harus selalu berjalan agar sistem dapat bekerja dengan baik. Ketika server selalu bekerja, maka log aktivitas yang dicatat akan sangat banyak. *Log* adalah sebuah *file* yang berisi daftar tindakan, kejadian (aktivitas) yang telah terjadi di dalam suatu sistem komputer [1]. Hampir

* Penulis Korespondensi (Walidatush Sholihah)
Email : walidah@apps.ipb.ac.id

semua aplikasi dan sistem perangkat lunak menghasilkan *file log*.

Manajemen *log* yang efektif, penting untuk keamanan. *File log* (terdiri atas: pemantauan, dokumentasi, dan analisis peristiwa sistem) merupakan komponen penting. Perangkat lunak manajemen *log* mengotomatiskan banyak proses yang terlibat. *Event Log Manager* (ELM), misalnya, melacak perubahan dalam infrastruktur jaringan. Belum tersedianya *Log Event Management Server* membuat data *log* pada *server* menjadi tidak teroganisasi dengan baik. Berdasarkan permasalahan di atas, maka perlu dibuatnya suatu *log event management server* yang dapat membaca sekaligus menganalisis *log service* pada server. *Elasticsearch* merupakan mesin pencari dan analitik yang membuat data mudah untuk dijelajahi [3]. *Elastic search platform open source* yang terdistribusi [4]. *Logstash* digunakan untuk mengumpulkan *log* atau data dari sumber yang berbeda, kemudian menyaring dan memproses data itu sesuai dengan kebutuhan dan mengirimkannya ke masing-masing ke tujuan tertentu [5]. ELK Stack merupakan kumpulan dari tiga alat yaitu *Elasticsearch*, Logstash dan Kibana [4]. Kelebihan dari ELK stack yaitu [3]:

- skalabilitas: ELK memiliki kemampuan untuk berkembang.
- Keandalan: elasticsearch membantu deteksi kegagalan node dan mendistribusikan data secara otomatis agar data tetap dapat dinilai dan diamankan.
- Otomatis: ELK menyimpan dan mengindeks JSON secara otomatis
- Ramah pengguna: ELK memvisualisasikan semua jenis data sumber yang diindeks ke elasticsearch.

Elasticsearch Logstash Kibana (ELK *Stack*) merupakan komponen yang tepat dalam membangun log event management. ELK *Stack* dapat memberi informasi kepada sistem administrator mengenai tren, statistik, dan anomali yang terjadi.

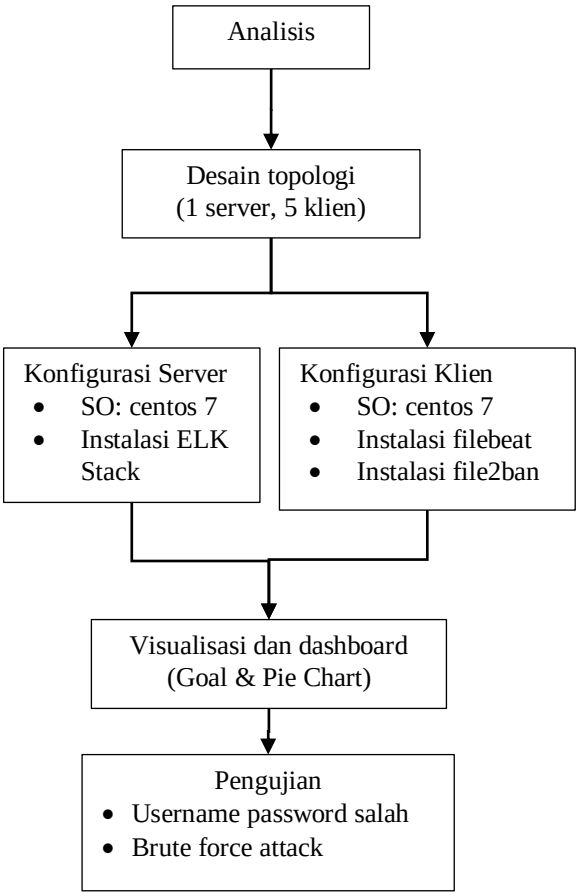
Tujuan dari penelitian ini adalah sebagai berikut :

- Menyediakan fitur mengenai tren, statistik serta anomali yang terjadi
- Menyediakan fitur monitoring log secara *real time*
- Mengelompokkan dan menganalisa log file dengan jumlah yang sangat banyak.

II. METODOLOGI

Metode yang digunakan pada penelitian ini terdiri atas lima tahap yang digambarkan pada Gbr. 1. Tahapan-tahapan tersebut yaitu: Analisis, Desain topologi, Konfigurasi server, Konfigurasi klien dan Pengujian.

Tahap analisis merupakan tahap awal untuk mengidentifikasi permasalahan dan kemungkinan solusi untuk menyelesaikannya. Tahap kedua yaitu desain topologi. Pada tahap ini dibuat desain topologi jaringan yang sesuai untuk menyelesaikan maslah pada tahap awal. Tahap berikutnya yaitu konfigurasi server (ELK *stack*) dan klien. Di sisi klien dipasang file beat sebagai log shipper. Tugas dari filebeat ini adalah meneruskan log dari server-klien ke logstash. Tahap akhir yaitu pengujian. Proses yang dilakukan untuk menguji elastic search kibana ini yaitu pengujian kesalahan *login* password ssh pada setiap server. Kedua, pengujian kesalahan *login user* SSH pada server. Ketiga, melihat tingkat akurasi waktu pada setiap kejadian *login* ssh pada *server-client* ke ELK *Stack*. Keempat, penghapusan log pada server dan pengujian serangan menggunakan *brute-force* serta pencegahannya menggunakan *fail2ban*.



Gbr. 1 Metodologi penelitian

III. HASIL DAN PEMBAHASAN

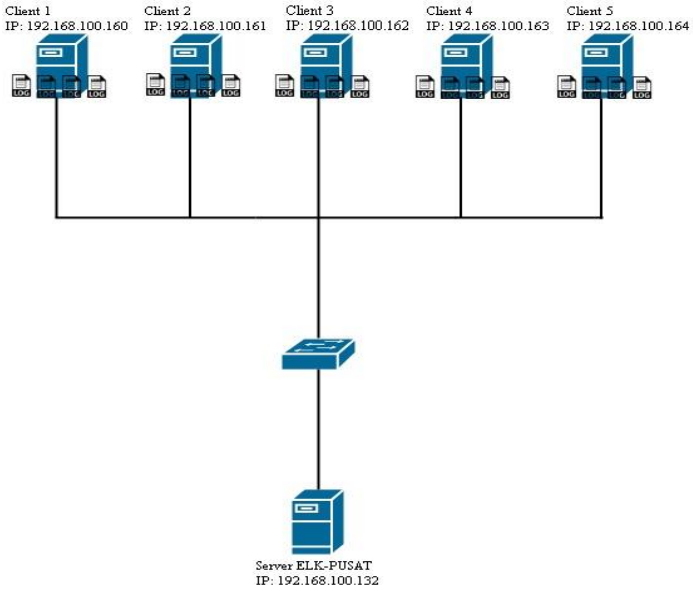
A. Perancangan

Pada penelitian ini, perangkat keras dan perangkat lunak yang digunakan disajikan pada Tabel I

TABEL I
PERANGKAT YANG DIGUNAKAN

No.	Nama Perangkat	Jumlah	Keterangan
1	Laptop	1	Berfungsi sebagai penerapan simulasi.
2	Centos 7 + *Repository*	6	Untuk simulasi satu sever dan lima klien
3	*Elasticsearch* 6.7	-	Berfungsi sebagai *search engine.*
4	*Logstash* 6.5	-	Berfungsi sebagai *filtering log* yang dianalisa.
5	*Kibana* 6.7	-	Berfungsi sebagai visualiasi *log* yang telah di *filtering.*
6	VMware Workstation Pro 10	1	Berfungsi sebagai mesin virtual dimana server dan klien di instal.
7	*Filebeat* 5.1	-	Berfungsi sebagai *log parshing* atau pengirim *log.*
8	*Nginx*	-	Berfungsi sebagai *reverse proxy*.
9	*Java 8*	-	bahasa pemrograman yang digunakan.
10	*Fail2ban*	-	aplikasi pencegahan serangan.

Desain topologi jaringan dibuat menggunakan aplikasi draw.io. Topologi jaringan dapat dilihat pada Gbr. 2. pada penelitian ini digunakan satu buah server ELK pusat dan lima buah server klien. Server ELK pusat dibangun dengan spesifikasi Centos 7, dengan RAM 4 GB,VCPU 1 Core dan penyimpanan Disk (SCSI) 50 GB. Untuk server client dibangun dengan spesifikasi Centos 7, dengan RAM 512 MB,VCPU 1 Core dan penyimpanan Disk (SCSI) 10 GB.



Gbr. 2 Topologi jaringan

Server klien dipantau aktivitas log serta visualisasinya. Gbr. 2 memperlihatkan bahwa *log services SSH* di server-klien dikirim oleh *filebeat* ke *logstash. Filebeat* berguna sebagai *log shipper. Log shipper* merupakan gerakan perubahan yang tidak sinkron dari satu server ke server lain dan dapat terjadi dengan perubahan data yang ditransfer ke beberapa database. *Logstash* menjadi gerbang awal ketika *log* server-klien masuk ke Server Pusat ELK Stack. *Logstash* sendiri berfungsi sebagai filterisasi log yang sudah dikirim. Tujuannya untuk membagi beberapa kategori. Setelah diproses pada *logstash*, log tersebut akan dikirim ke Elasticsearch untuk

ditampung dan terakhir divisualisasikan menggunakan Kibana. Agar aplikasi ELK dapat berjalan lancar, pengembang dari ELK *Stack* menyarankan untuk menggunakan aplikasi dengan nomor versi utama yang sama, agar aplikasi dapat bekerja dengan baik.

B. Konfigurasi Server

Tahap berikutnya yaitu konfigurasi server. Konfigurasi yang dilakukan saat instalasi Elasticsearch Logstash Kibana (ELK Stack) yaitu instalasi paket-paket aplikasi ELK-Stack yang sudah dibuatkan simple bash script untuk otomatis instal dengan Nginx. Konfigurasi tersebut seluruhnya dilakukan pada virtual private server menggunakan CLI (Command Line Interfaces). Server ELK Stack memiliki spesifikasi sebagai berikut: sistem operasi Centos 7, RAM 4 GB,VCPU 1 Core dan Disk (SCSI) 20 GB. Sistem operasi Centos 7 merupakan aplikasi open source. Kapasitas memori minimum untuk instalasi ELK stack adalah RAM 4 GB. Kapasitas harddisk 20 GB diharapkam dapat menampung banyak request dari banyak klien. Fitur SElinux perlu dinonaktifkan (Gbr. 3). SElinux (Security Enhanced Linux) merupakan salah satu fitur yang secara default dimiliki oleh Linux pada distro Red Hat, CentOS, Fedora dan turunan Red Hat lainnya yang menyediakan mekanisme untuk mendukung kebijakan keamanan kontrol aplikasi akses. SELinux berfungsi sebagai pengaman pada Sistem, mengamankan aplikasi dari modifikasi/akses yang tidak diinginkan.

```
$ vim /etc/sysconfig/selinux

SELINUX=disabled

$ reboot
```

Gbr. 3 Menonaktifkan fitur SElinux

Selain SElinux, fitur lain yang dinonaktifkan adalah firewall. Hal ini dilakukan agar tes jaringan dan aplikasi yang memerlukan izin keluar masuk server dapat berjalan bebas. Kemudian perlu instalasi java pada server. Perintah elasticsearch dan logstash menggunakan bahasa pemrograman java. Konfigurasi server ELK stack dapat dilihat pada

```
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\ngp

gcheck=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"
```

Gbr. 4 Instalasi ELK stack

Port url dan host elasticsearch perlu diatur dan diproteksi. Port yang digunakan yaitu port 9200. Hal ini dilakukan agar tidak ada pihak lain yang membaca atau mematikan elasticsearch melalui HTP API. Selain itu, memory swapping untuk elasticsearch perlu dinonaktifkan.

Kibana dikonfigurasi agar dapat memvisualisasikan semua log server yang ada. File "kibana.xml" pada direktori sistem Kibana dikonfigurasi menggunakan server web Nginx. Kibana menggunakan alamat IP localhost dan Nginx bertindak sebagai reverse proxy untuk aplikasi Kibana. Reverse proxy adalah salah satu jenis dari proxy. Biasanya reverse proxy digunakan sebagai perantara antara client dengan web server. Tanda komentar pada baris konfigurasi untuk server.port, server.host dan elasticsearch.url dihapus. Kibana ditambahkan agar berjalan saat *boot* dan *start service*. Kibana berjalan di *port* 5601 sebagai *node application*. *Node application* adalah *platform* untuk membangun aplikasi server yang cepat dan dapat diskalakan menggunakan *JavaScript*. *Node.js* adalah *runtime* dan *npm* adalah *Package Manager* untuk modul *Node.js*. Setelah *Elasticsearch* berjalan, *port* yang terbuka di server, yang merupakan *'state'* untuk port 9200, adalah *LISTEN*.

Setelah instalasi kibana selesai, *Nginx* dikonfigurasi sebagai *reverse proxy* agar bisa

mengakses *Kibana* dari alamat IP publik. Repositori *Epel* menyediakan Nginx. *Epel-release* di instal dengan *yum*. Kemudian, instalasi paket *Nginx* dan *httpd-tools*. Paket *httpd-tools* berisi *tool* untuk server web yang menggunakan otentikasi dasar *htpasswd* untuk *Kibana*.

Pada file nginx.conf di direktori sistem nginx, diberi tanda pagar di setiap konfigurasi. Kibana.conf merupakan file konfigurasi baru untuk kibana. Langkah ini untuk membuat konfigurasi kustom yang disesuaikan dengan kebutuhan pada ELK stack.

Nginx ditambahkan untuk dijalankan pada saat *boot* dan *start*. *Nginx auto-run* ketika server mengulang kembali dan menjalankan *services*. *Nginx* berjalan di *port* 80 sebagai *nginx: master*. Tujuan utama dari proses *master* adalah untuk membaca dan mengevaluasi konfigurasi dan memelihara proses pekerja.

Proses terakhir dari konfigurasi server adalah konfigurasi logstash. konfigurasi pada *Logstash* ini terbagi menjadi 3 bagian, yaitu *Input*, *Filter*, dan *Output*. Pertama, konfigurasikan untuk *Input* pada *Logstash* yang bertujuan menerima *Input* dari *Filebeat* dari masing-masing server-client. Pertama buat file konfigurasi *Logstash* dibuat untuk *Input* pada direktori sistem *Logstash* dengan nama *"filebeat-input.conf".*

Perintah untuk konfigurasi output pada logstash yaitu *vim conf.d/output.conf*. Terdapat *Elasticsearch* merupakan rujukan untuk mengarah ke *port Elasticsearch* yaitu *Manage_template Manage_template* diatur ke *false* untuk menonaktifkan fitur ini. Jika memerlukan kontrol lebih besar atas pembuatan *template*, (seperti membuat indeks secara dinamis berdasarkan nama bidang), *manage_template* diatur menjadi *false* dan *REST API* digunakan untuk menerapkan template secara manual. Indeks untuk menulis bisa dinamis menggunakan *sintaks*% {foo}. Nilai *default* mempartisi indeks berdasarkan hari sehingga dapat lebih mudah menghapus data lama atau hanya mencari rentang tanggal tertentu. Konfigurasi *remove-grokparse.conf* ini bertujuan untuk menghapus file *_grokparsefailure*. *Grokparsefailure* merupakan kesalahan pada saat *log* tersebut dikirim. Konfigurasi *removebeatscodec.conf* ini bertujuan untuk menghapus beats yang tidak dibutuhkan. Dapat diketahui bahwa setiap beats yang dikirimkan oleh klien menghasilkan *beats_input_codec_plain_applied.* Dengan demikian dibutuhkan filter yang berfungsi sebagai penghapus beats_input_codec_plain.

File konfigurasi *Logstash* untuk *filter* diberi nama *"sshfilter.conf"*. *Filter Grok* dikirimkan dengan berbagai ekspresi dan pola reguler untuk tipe data umum dan ekspresi yang dapat ditemui dalam *log* (IP, nama pengguna, email, nama *host*). Ketika *Logstash* membaca *log*, menggunakan pola ini untuk menemukan elemen semantik pesan log yang ingin diubah menjadi bidang terstruktur. Pada dasarnya tujuan dilakukan *filter* ini adalah ketika *log* tersebut masuk *elasticsearch.* Pada elasticsearch sudah ada beberapa tag membedakan tiap log. Log bisa divisualisasikan pada Kibana.

Konfigurasi yang berfungsi sebagai generate sertifikat *openSSL* (Security Socket Layer), digunakan pada server *client*. Sebelum mendapatkan sertifikat *ssl*, perlu dilakukan *generate csr* dari server yang dilakukan proses install dengan *SSL Certificates*. Proses *Generate CSR* dari server akan membawa informasi-informasi yang diperlukan untuk disertakan pada sertifikat *SSL* seperti *Common Name*, *Company Name*, *Country*, *City*, *Province* dan Organization Unit Name sekaligus dengan informasi identitas atas Server yang didapat ketika melakukan *Generate CSR*. Kemudian perlu diketahui saat telah mendapatkan Sertifikat *SSL*, bahwa Sertifikat *SSL* tersebut untuk diinstal pada server yang sebelumnya telah dibuat CSR. Hal ini karena informasi yang telah submit ketika mengirim sertifikat. Segala informasi yang terdapat pada *CSR* dari server telah disertakan pada sertifikat *SSL* yang telah dikirim. Dan kemudian dapat lakukan instalasi pada Server berikutnya apabila ingin menggunakan sertifikat *SSL* pada banyak server. proses menyalin *logstash-forwarder.crt* dari */etc/pki/tls/certs/logstash-forwarder.crt* ke */root/logstash-forwarder.crt* . Langkah ini berfungsi untuk memberi sertifikat ke server *client*.

C. Konfigurasi klien

Konfigurasi client dilakukan dengan memasang *filebeat* sebagai *log shipper*. *Filebeat* sendiri bertugas meneruskan *log* dari server-client ke *logstash*. Konfigurasi tersebut seluruhnya dilakukan pada *virtual private server* menggunakan CLI (*Command Line Interfaces*). Konfigurasi sisi server client Centos yaitu dengan memasang *filebeat* sebagai *logshipper*. *Filebeat* sendiri bertugas meneruskan log dari server-client ke *logstash*. *Filebeat* adalah pengirim data, agen ringan yang dapat diinstal pada node klien untuk mengirim sejumlah besar data dari mesin klien ke server *Logstash* atau *Elasticsearch*. Server client sendiri memiliki spesifikasi sistem operasi Centos 7, RAM 512 GB, VCPU 1 Core dan Disk (SCSI) 10 GB. Sistem operasi Centos 7 dipilih dengan mempertimbangkan sisi *open source* karena dapat secara bebas melakukan perubahan di dalamnya. Pada saat menginstal dan mengkonfigurasi *'Filebeat'* untuk mentransfer *file log* data ke server *Logstash* melalui koneksi SSL.

Tahap awal yang harus dilakukan pada saat instalasi server client adalah mengizinkan beberapa port jika menggunakan layanan *firewalld.Firewall* memang sangat dibutuhkan untuk pertahanan server dari akses luar. Namun adakalanya ketika ingin melakukan pengetesan jaringan, atau ketika ada sebuah aplikasi yang membutuhkan perizinan service tertentu untuk masuk/keluar server, maka biasanya *firewall* dimatikan terlebih dahulu. Apabila dalam jaringan sudah ada *dedicated firewall*, maka firewall dapat dinonaktifkan. Setelah itu, Selinux dinonaktifkan.

File sertifikat *SSL*(Security Socket Layer) disalin dari elastic server ke server client1. Fungsi menyalin file sertifikat adalah untuk mengamankan komunikasi dua komputer yang terhubung oleh jaringan internet. Pertukaran data seperti mengunjungi website, pengiriman dan penerimaan serta data-data penting perusahaan dienkripsi agar tidak ada oknum tak bertanggung jawab yang berusaha menggunakan data tersebut secara ilegal. Pada client1 dibuat direktori certs. File sertifikat tadi dipindahkan ke dalam direktori ini. Pada client1 dilakukan proses *imporelastic key*. File berisi aktivitas yang terjadi pada ssh dikonfigurasi. Selain itu juga dilakukan konfigurasi untuk log server menggunakan perintah file /var/log/message. Setelah itu diatur jenis file yang digunakan, yaitu file *syslog*. *Syslog* adalah protocol untuk *computer message logging*. Syslog ini membolehkan untuk mengirimkan *system message* sebuah komputer melewati *network* ke *syslog* server untuk ditampilkan. Sebuah server yang menyimpan data syslog berbagai macam perangkat komputer dan jaringan secara terpusat disebut syslog server. Syslog server harus memiliki kapasitas yang tinggi untuk melayani penyimpanan syslog setiap perangkat komputer dan jaringan [6].

Berikutnya, ditambahkan konfigurasi *output logstash* baru. Komen pada konfigurasi *output logstash* dihapus dan semua nilai ke konfigurasi yang ditujukan diubah. *Host* menentukan server *logstash* dan *port* (5443) tempat logstash dikonfigurasi untuk mengamati koneksi *beats* yang masuk. *Bulk_max_size* jumlah maksimum yang dikelompokkan dalam satu permintaan *logstash* standarnya adalah 2048. *Ssl.certificate_authorities* mengonfigurasi *filebeat* untuk memberi sertifikat apa pun yang ditandatangani. *Template.name* defaultnya adalah *filebeat*. Versi *filebeat* selalu ditambahkan ke nama yang diberikan, jadi nama akhirnya adalah *filebeat*. *Filebeat* secara otomatis memuat *file template* yang direkomendasikan, *filebeat.template.json* jika *output Elasticsearch* diaktifkan. Hal ini agar *filebeat* memuat *template* yang berbeda dengan menyesuaikan opsi templat.name dan *templat.path* dalam *file filebeat.yml*. *Filebeat* diatur untuk memulai saat *boot* dan *start service*. Konfigurasi ini merupakan tahap akhir pada bagian konfigurasi klien.

File2ban diinstal pada setiap server klien untuk pencegahan *brute force attack*. Lokasi *Fail2Ban default* terletak di */etc/fail2ban/jail.conf*. Konfigurasi tidak boleh dilakukan dalam file itu, karena dapat dimodifikasi dengan peningkatan paket. Tetapi lebih baik menyalinnya sehingga dapat membuat perubahan dengan aman. Fail2ban disalin ke file jail.local. Setelah *file* disalin, langkah

selanjutnya membuat semua perubahan di dalam *file jail.local* baru. Banyak layanan yang membutuhkan perlindungan sudah ada dalam *file*. Masing-masing terletak dibagiannya sendiri, perlu diatur dan dimatikan

Konfigurasi *jail.local IgnoreIP* memungkinkan daftar alamat IP tertentu dan memastikan bahwa mereka tidak terkena *banned*. Termasuk alamat menjamin bahwa tidak secara tidak sengaja memblok server sendiri. Parameter banaction ditambahkan untuk memastikan penggunaan *iptables* untuk konfigurasi *firewall*. Langkah selanjutnya adalah memutuskan sebuah *bantime.* Jumlah detik yang dimiliki suatu host diblokir dari server jika mereka terbukti melanggar aturan. Ini sangat berguna dalam kasus bot, yang pernah dilarang, hanya beralih ke target berikutnya. Standarnya diatur selama 10 menit dan dapat dinaikkan menjadi satu jam (atau lebih tinggi) jika dibutuhkan. *Maxretry* adalah jumlah upaya *login* yang salah yang mungkin dilakukan oleh pemilik sebelum mereka dilarang selama jangka waktu larangan. *Findtime* mengacu pada jumlah waktu *host*. Pengaturan *default* adalah 10 menit. Hal ini berarti bahwa jika suatu host mencoba, dan gagal, untuk *login* lebih dari jumlah *maxretry* kali dalam 10 menit yang ditentukan, maka host diblokir. File2ban diatur juga untuk selalu restrat setelah ada perubahan perngaturan.

Setelah konfigurasi selesai semua, selanjutnya pengujian hasil dari penelitian ini. Pengujian dibagi dalam beberapa tahap yaitu pertama pengujian kesalahan *login* password ssh pada setiap server client, kedua pengujian kesalahan *login* user ssh pada serverclient1, serverclient2, serverclient3, serverclient4, serverclient5 ketiga melihat tingkat akurasi waktu pada setiap kejadian *login* ssh pada server-client1 ke ELK-*Stack*, kempat penghapusan log pada serverclient1,kelima percobaan penyerang menggunakan *brute-force* dengan tidak menggunakan pengaman *fail2ban* dan terakhir penyerangan dengan brute-force tetapi menggunakan pengaman *fail2ban.*

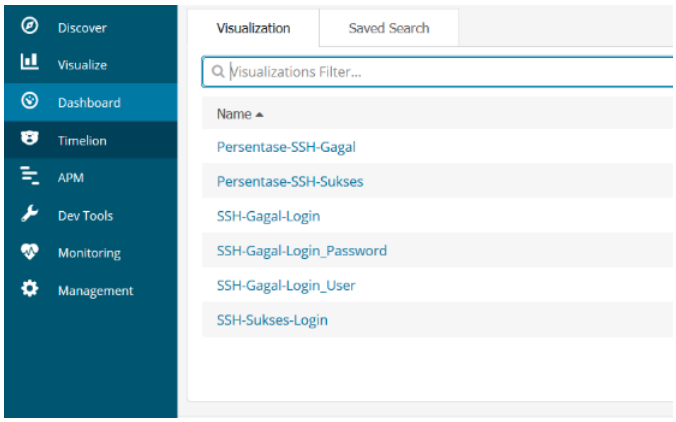
D. Visualisasi dan Dashboard

Konfigurasi selanjutnya adalah Visualisasi dan *Dashboard*, konfigurasi ini bertujuan agar lebih mudah dalam menampilkan informasi tentang *log service* yang sebelumnya telah dikirimkan dari *client*. Kibana memiliki fitur untuk membuat visualisasi pada log service. Visualisasi yang ditampilkan pada Kibana dapat dilihat pada TABEL .

TABEL II
VISUALISASI PADA KIBANA

No	Nama Visualisasi	Keterangan
1	Presentase SSH Gagal	Hasil visualisasi menampilkan presentase kegagalan SSH *login* pada *client*.
2	Presentase SSH Sukses	Hasil visualisai menampilkan presentase keberhasilan SSH *login* pada client.
3	SSH Gagal *Login*	Hasil visualisasi menampilkan *username* pada *client* yang menggalami kegagalan *login* SSH.
4	SSH Sukses *Login*	Hasil visualisasi menampilkan *username* pada *client* yang berhasil *login* SSH.
5	SSH Gagal *Login User*	Hasil visualisasi yang menampilkan *username* pada *client* yang mengalami kegagalan *login* SSH pada *username* yang digunakan.
6	SSH Gagal *Login Password*	Hasil visualisasi yang menampilkan *username* pada *client* yang mengalami kegagalan *login* SSH pada password yang digunakan.

Visualisasi yang dipilih yaitu Goal chart dan Pie chart. Visualisasi ini bertujuan untuk memudahkan dalam menampilkan hasil atau *result* dari *log* yang dikirimkan oleh *client* sehingga cepat dianalisa [7]. Tampilan dashboard kibana setelah dilakukan pengaturan visualisasi dapat dilihat pada Gbr. 5.



Gbr. 5 Dashboard kibana

E. Pengujian

Pengujian dilakukan dengan memasukkan username dan password yang salah ke semua komputer klien secara berulang-ulang. Gbr. 6 merupakan hasil dari kegagalan tersebut ditandai oleh proses permintaan untuk meminta *passsword* yang benar secara berulang-ulang. Setelah terindentifikasi letak salahnya kemudian kegagalan *login* dapat masuk ke *dashboard Kibana*.

Gbr. 6 Tampilan dashboard kibana saat gagal login

Berikutnya pengujian dilakukan dengan melakukan penghapusan log pada klien untuk melihat pengaruhnya pada hasil visualisasi. Hasilnya, tidak ada perubahan pada grafik visualisasi. Log harus dihapus untuk mengurangi beban *memory* yang terdapat pada server ELK Pusat dikarenakan banyaknya *log* yang sudah tidak dibutuhkan dapat mempengaruhui kinerja dari server ELK Pusat. Dengan melakukan perintah *remove log* yang ingin dihapus pada gambar ini *log* yang ingin dihapus adalah *tags* ssh sukses login, *tags* ssh gagal *login password* dan ssh gagal *login.*

Pengujian berikutnya yaitu brute force attack. Brute force attack adalah eksperimen kata sandi yang menggunakan campuran karakter ASCII yang memungkinkan dalam isolasi atau dalam kombinasi [8]. Metode serangan brute force yaitu serangan bertubi-tubi ke server. Pada penelitian ini, sistem dibanjiri dengan password sebanyak 100 baris. Dengan adanya file2ban, serangan brute force ini tidak berhasil. Fail2Ban dapat mengurangi tingkat upaya otentikasi yang salah namun tidak dapat menghilangkan risiko yang disajikan oleh otentikasi yang lemah [9]. Fail2ban adalah alat pencatatan jaringan yang menawarkan deteksi dan respons intrusi otomatis [10]. Setelah Fail2ban mengidentifikasi IP sumber aktivitas mencurigakan, fail2ban kemudian memperbarui aturan firewall untuk menolak alamat IP dari mesin sumber tersebut [10].

Setelah melakukan banyak percobaan yang telah di lakukan,maka untuk membuatnya lebih rinci dibuatkan tabel yang berisi hasil dari berbagai percobaan yang telah dilakukan yang terdapat pada Tabel III .

TABEL III
HASIL PENGUJIAN

No	Pengujian	Hasil	Keterangan
1	Kegagalan *login* SSH pada setiap server client	Semua kegagalan *login* SSH dapat di tampilkan oleh Kibana	Sukses
2	Keberhasil *login* SSH pada setiap server client	Semua keberhasilan *login* SSH dapat di tampilkan oleh Kibana	Sukses
3	Kesalahan *login user* sangga,pripambudi,teknik, komputer dan ipb pada setiap server client	Semua kesalahan *login* SSH yang di ikuti *user* dapat di tampilkan oleh Kibana	Sukses
4	Pencatatan waktu kejadian *login* SSH yang masuk ke server elk pusat	Tidak ditemukan jeda antara keduanya hal ini berarti (realtime)	Sukses
5	Pengujian dengan percobaan penghapusan log yang dilakukan oleh client dan server	Tidak menimbulkan pengaruh yang signifikan terhadap Kibana artinya tidak terjadi perubahan sediktpun. Pada	Sukes

No	Pengujian	Hasil	Keterangan
		server ketika dilakukan penghapusan log maka semua akan bernilai nol atau tidak terdapat log yang dikirimkan oleh client.	
6	Pengujian penyerangan terhadap client yang dilakukan dengan memanfaatkan serangan *brute-force*	Semua penyerangan dapat dilihat anomali pergerakan di kibana	Sukses
7	Pengujian penyerangan terhadap client yang dilakukan dengan pertahanan fail2ban	Tidak ditemukan penyerangan yang dilakukan	Sukses

IV. PENUTUP

ELK atau Elasticsearch, Kibana, Logstash, adalah aplikasi himpunan sebagai alat searching, indexing data dan log analysis. Log analysis diperlukan dalam memanipulasi data log yang sangat banyak jumlahnya sehingga sulit untuk memahami bahkan menganalisis kumpulan log yang sangat banyak. Big data sendiri tidak dapat lepas dari log analysis, dikarenakan suatu log terus menghasilkan data karena suatu proses yang terjadi pada sistem atau mesin secara terus menerus. Dalam menangani kumpulan log ini, digunakan ELK, sebuah aplikasi yang cepat dan efisien dalam memanipulasi sebuah big data pada konteks ini adalah log. Dari aplikasi ELK ini, Logstash berperan sebagai data *collector*, *forwarder* serta memanipulasi data log tersebut menjadi sesuai dengan format yang diinginkan.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Sekolah Vokasi IPB University atas dukungannya dalam penelitian ini. Ucapan terima kasih juga disampaikan kepada semua pihak yang telah banyak membantu penulis yang tidak dapat disebutkan satu-persatu. Semoga tulisan ini bermanfaat bagi masyarakat.

REFERENSI

[1] C. Tarigan, V. Jeremias, L. Engel, and D. Angela, "Sistem Pengawasan Kinerja Jaringan Server Web Apache dengan Log Management System ELK (Elasticsearch , Logstash , Kibana)," pp. 7–14, 2018.

[2] M. N. Arifin, E. Susilowati, and Sugiartowo, "Desain dan Implementasi Log Event Management Server Menggunakan Elasticsearch Logstash Kibana Elk)," in *Seminar Nasional Sains dan Teknologi*, 2018, pp. 1–7.

[3] J. N. Praneeth and M. Sreedevi, "Detecting and Analyzing the Malicious Windows Events using Winlogbeat and ELK Stack," *Int. J. Recent Technol. Eng.*, vol. 7, no. 6, pp. 156–160, 2019.

[4] M. Bajer, "Building an IoT Data Hub with Elasticsearch , Logstash and Kibana," in *INternational Conference on Future Internet of Things and Cloud Workshops*, 2017, pp. 63–68.

[5] M. Harikanth and P. Rajarajeswari, "Malicious Event Detection Using ELK Stack Through Cyber Threat Intelligence," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 7, pp. 882–886, 2019.

[6] T. H. Ditanaya, R. M. Ijtihadie, and M. Husni, "Rancang Bangun Sistem Log Server Berbasis Syslog dan Cassandra untuk Monitoring Pengelolaan Jaringan di ITS," *J. Tek. ITS*, vol. 5, no. 2, 2016.

[7] A. Chuvakin, K. Schmidt, and C. Philips, *Logging and Log Management 1st Edition*, 1st ed. Elsevier .inc, 2012.

[8] D. Stiawan, M. Y. Idris, R. F. Malik, S. Nurmaini, N. Alsharif, and R. Budiarto, "Investigating Brute Force Attack Patterns in IoT Network," *J. Electr. Comput. Eng.*, vol. 2019, 2019.

[9] "No Title," 2016. [Online]. Available: https://www.fail2ban.org/wiki/index.php/Main_Page. [Accessed: 25-Apr-2020].

[10] C. Lopez-Araiza and E. C. Cankaya, "A Comprehensive Analysis of Security Tools for Network Forensics," *J. Med. - Clin. Res. Rev.*, vol. 1, no. 3, pp. 1–9, 2017.