RABIT : Jurnal Teknologi dan Sistem Informasi Univrab
Volume 8 No. 1 | Januari 2023 : Hal : 124-134
ISSN CETAK : 2477-2062
ISSN ONLINE : 2502-891X

MONITORING LOG SERVER DENGAN ELASTICSEARCH, LOGSTASH DAN KIBANA (ELK)

[1] Aviecenna Yudhistira, [2] Yuli Fitrisia
1) Program Studi Teknik Informatika, Politeknik Caltex Riau
2) Program Studi Teknologi Rekayasa Komputer, Politeknik Caltex Riau
1,2)Jl. Umban Sari No. 1, Rumbai, Pekanbaru – Riau - Indonesia
E-mail : 1) aviecenna.yudhistira@alumni.pcr.ac.id, 2)uli@pcr.ac.id

ABSTRAK
Menggunakan server yang harus berjalan terus menerus selama 24 jam dan menjalankan layanan menghasilkan banyak log. Ini mengharuskan administrator sistem yang di cek masih memiliki hubungan dengan server. Penelitian ini bermaksud merancang Monitoring Log Server dengan menggunakan ELK Stack (Elasticsearch Logstash Kibana) yang dapat mempermudah dalam membaca dan menganalisa service log di server. Implementasi dalam penelitian kasus ini menggunakan Ubuntu 18.04 Server dan client dengan layanan SSH Putty. Dari hasil penelitian ELK Stack, didapatkan solusi berupa Monitoring Log Server untuk memudahkan administrator mengecek setiap access log. Berdasarkan hasil pengujian skenario pertama dan kedua, diperoleh hasil yang berhasil menampilkan data setiap log yang dapat dibaca oleh administrator dalam jangka waktu tertentu berdasarkan data per hari dan per minggu. Untuk skenario ketiga, administrator dapat melihat tingkat akurasi log akses masing-masing. Dan skenario terakhir adalah hasil untuk memudahkan administrator mengaudit atau memeriksa setiap log akses.

Kata Kunci: Elasticsearch, Kibana, Log, Logstash, Server

ABSTRACT
Using a server that has to be running continuously for 24 hours and running services generates a lot of logs. This requires that the administrator of the system being checked still has a relationship with the server. This study intends to design a Monitoring Log Server using the ELK Stack (Elasticsearch Logstash Kibana) which can make it easier to read and analyze service logs on the server. The implementation in this case study uses Ubuntu 18.04 Server and client with the PuTTY SSH service. From the results of the ELK Stack research, a solution was obtained in the form of a Monitoring Log Server to make it easier for administrators to check each access log. Based on the results of testing the first and second scenarios, the results obtained successfully display data for each log that can be read by the administrator within a certain period based on data per day and per week. For the third scenario, administrators can see the level of accuracy of each access log. And the last scenario is the result of making it easier for administrators to audit or check each access log.

Keyword: *Elasticsearch, Kibana, Log, Logstash, Server.*

PENDAHULUAN

Latar Belakang

Perkembangan dunia teknologi saat ini sangat cepat dan selalu ada inovasi-inovasi terbaru yang hadir. Salah satu inovasi yang bisa semua 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 jaringan dan merupakan perangkat utama dalam sebuah sistem komunikasi jaringan yang berfungsi sebagai penyedia layanan atau service. 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 secara manual untuk melakukan pembacaan log service pada

DOI : https://doi.org/10.36341/rabit.v8i1.2975

server dan harus berinteraksi langsung dengan server yang memakan waktu cukup lama. Masalah selanjutnya yaitu server yang harus berjalan 24 jam penuh menghasilkan log service dalam jumlah banyak. Berdasarkan permasalahan di atas, maka perlu dibuat suatu log event management yang mampu meringankan dan memudahkan administrator dalam membaca sekaligus menganalisis log service pada server.

Dalam hal ini Elasticsearch Logstash Kibana (ELK Stack) sebagai salah satu pilihan dalam membangun log event management yang dapat memberi insight kepada sistem administrator mengenai tren, statistik, dan anomali yang terjadi. ELK Stack di rancang untuk digunakan sebagai solusi terintegrasi.

Berdasarkan permasalahan tersebut maka pada peneletian kali ini akan dibahas bagaimana implementasi Log Event Management Server menggunakan Elasticsearch Logstash Kibana (ELK Stack) dengan pengujian pada Web server yang nantinya akan di pantau berdasarkan log.

Studi Literatur

Penelitian yang terkait Monitoring Log Server Dengan Elasticsearch Logstash Dan Kibana (ELK) telah dilakukan oleh beberapa peneliti terdahulu. Penelitian terdahulu ini berguna untuk memberi masukan terkait dengan penelitian yang sedang dilaksanakan.

Penelitian yang dilakukan oleh Arifin[1] membahas tentang suatu log event management yang mampu meringankan dan memudahkan dalam membaca sekaligus menganalisis log service pada server. Dalam hal ini log event management yang digunakan adalah Elasticsearch Logstash Kibana (ELK) Stack.

Penelitian terdahulu dilakukan oleh Putra[2]salah mahasiswa Teknik Informatika dan Komputer Politeknik Negeri Jakarta. Pada penelitian ini penulis membuat studi kasus pada Pusat PT.XYZ, Pusat PT.XYZ merupakan sebuah lembaga yang melaksanakan tugas berada di bawah dan bertanggung jawab kepada Menteri Keuangan melalui Sekretaris Jenderal. Bidang KIKTIK (Keamanan Informasi dan kelangsungan TIK) sebagai salah satu divisi yang bertanggung jawab dalam mengelola keamanan TIK. Pada bidang KIKTIK terdapat beberapa perangkat server dalam mengelola keamanan informasi di PT.XYZ. Dalam hal ini jika ada kendala apapun atau serangan pada server tersebut akan menyulitkan administrator untuk mendeteksi sumber dan penyebab kendala di server tersebut karena banyak log yang terlihat secara terus menerus. Administrator juga akan sulit mengidentifikasi jika terdapat masalah pada server. Untuk meminimalisir hal tersebut, diperlukan suatu sistem management log server berupa software yang dapat melakukan pengiriman log dari server. Dengan di terapkannya manajemen log ini diharapkan memudahkan administrator untuk membaca log dan mengidentifikasi masalah pada server dan memiliki sistem pelaporan yang mudah dianalisis.

Penelitian selanjutnya dilakukan oleh Sartika & Cahyono[3] dari Fakultas Teknologi Industri Universitas Islam Indonesia. Pada penelitian ini penulis membahas Portal PPSDM memiliki basis data dengan volume data yang besar. Adapun data–data yang di simpan dalam basis data Portal PPSDM berupa: dokumen pembelajaran, video pembelajaran, audio pembelajaran, data peserta, nilai peserta, dan sertifikat tiap peserta. Seluruh data ini tidak hanya di simpan, namun juga di olah agar menjadi sebuah informasi. Contoh informasi yang didapatkan dari pengolahan data ini antara lain: pelatihan dengan kelulusan terbanyak, sertifikasi dengan kelulusan terbanyak, sepuluh peserta dengan nilai pelatihan terbaik. Dalam Portal PPSDM, kecepatan pengolahan data

DOI : https://doi.org/10.36341/rabit.v8i1.2975 *Corresponding Author* : Yuli Fitrisia

merupakan hal yang sangat diperlukan. Kecepatan pengolahan data sangat berkaitan erat dengan keakuratan data, karena data yang tidak akurat dapat disebabkan oleh waktu jeda pengolahan data yang lama. Solusi untuk menangani waktu jeda yang lama adalah dengan penerapan Elasticsearch Logstash Kibana Stack (ELK Stack) pada Portal PPSDM. ELK Stack merupakan sekumpulan aplikasi opensource yang digunakan dalam pengolahan data dengan volume yang besar. ELK Stack terdiri dari tiga aplikasi opensource yaitu: Elasticsearch, Logstash, dan Kibana.

Penelitian selanjutnya dilakukan Sholihah dkk[4], dari Teknik Komputer, Sekolah Vokasi IPB University. Membahas tentang Log Event Management Server Menggunakan Elastic Search Logstash Kibana (ELK Stack). 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, desain topologi 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 di hapus. Pada penelitian kali ini selain dapat mengirimkan log, juga dapat menampilkan presentase acuan keberhasilan.

Penelitian selanjutnya dilakukan oleh Putra dan Saptono[5] dari Teknik Informatika, Sekolah Tinggi Teknologi Terpada Nurul Fikri. Membahas tentang Penerapan Log Analyzer Log Untuk Mengetahui Lalu Lintas Jaringan Berbasis Elasticsearch, Logstash, Dan Kibana. Dalam penelitian ini bermaksud untuk melakukan implementasi analyzer log dan data lalu lintas jaringan berbasis ELK Stack (Elasticsearch Logstash Kibana) yang dapat melakukan pengiriman log, mengumpulkan log dan memvisualisasi log dari server. Implementasi analyzer log dan data lalu lintas jaringan dalam penelitian kali ini menggunakan Ubuntu 18.04 Server, dan Ubuntu 18.04 sebagai client. Dari hasil pengujian implementasi analyzer log dan data lalu lintas jaringan yang telah dilakukan dengan tingkat keberhasilan 100% menunjukan bahwa semua pengujian log terjadi pada server dapat dikirimkan secara realtime ke server utama ELK Stack.

Tinjauan Pustaka

Menurut [6] Elasticsearch adalah *open source*, gratis untuk digunakan, dan memiliki tujuan atau fungsi yang terukur sebagai mesin pencari dan mesin analisis teks. *Elasticsearch* memudahkan untuk menyimpan, mengambil, dan menganalisis data dari kumpulan data besar dengan cepat dan realtime. *Elasticsearch* juga merupakan salah satu database taksonomi *NoSQL* yang berfokus pada *database* mesin pencari.

Menurut [7] *Logstash* adalah salah satu produk inti *Elastic Stack*. *Logstash* digunakan untuk mengumpulkan dan memproses data sebelum mentransfer ke *Elasticsearch*. Data yang di terima *Logstash* dapat di ambil dari *database*, *Apache Kafka, Amazon* S3, dan *Beats*. Data ini diteruskan ke *Elasticsearch* untuk pengindeksan. *Logstash* memerlukan dokumen konfigurasi untuk mendapatkan data *input* yang benar.

Menurut [7] *Kibana* merupakan sebuah alat visualisasi dan manajemen data untuk *Elasticsearch*. Data yang divisualisasikan dapat berupa *grafik, metric,* tabel, ataupun Gambar.

DOI : https://doi.org/10.36341/rabit.v8i1.2975 *Corresponding Author* : Yuli Fitrisia

Kibana juga menyediakan fitur *dashboard*, fitur ini berfungsi untuk mengumpulkan data yang telah divisualisasikan ke dalam satu halaman *dashboard*. *Dashboard* yang telah di buat dapat di-embed ke dalam tampilan sistem. untuk membuat visualisasi dari data yang telah di-index oleh *Elasticsearch.*

Menurut [8] *Ubuntu* adalah salah satu di stribusi *Linux* yang berbasis *Debian* dan didistribusikan menjadi perangkat lunak sistem operasi yang bebas atau di sebut juga sistem operasi yang berbasiskan *Linux Debian*.

Menurut [9] *Web Server* menyediakan halaman *web* dan data yang terkait dengan situs *web* yang buat, memungkinkan orang lain untuk mengakses dan melihat data tersebut. Segera setelah permintaan di kirim dari *browser, server web* memproses permintaan dan memberikan hasil proses dalam format data yang diinginkan untuk ditampilkan di *browser*.

Menurut [10] *Server* adalah sistem komputer yang merupakan bagian dari jaringan komputer untuk melayani pengguna yang di sebut *client.* Di balik penggunaannya, *server* telah melalui banyak proses untuk memenuhi tuntutan *client.* Akibatnya, *server* sering menghadapi masalah karena tidak memiliki sumber daya yang cukup untuk memenuhi tuntutan tersebut. Hal ini menyebabkan layanan *server* berhenti tiba-tiba karena kernel telah memutuskan untuk menonaktifkan layanan *server* intensif sumber daya.

Menurut [6] *Log* berarti bisa menjadi catatan peristiwa. *Log* ini di simpan oleh sistem dalam *file log* yang dapat berisi informasi tentang perubahan perangkat, *driver* perangkat, perubahan sistem, peristiwa, operasi, dan banyak lagi. *Log* sistem memiliki kegunaan yang sangat penting dalam membuat sebuah sistem untuk mendeteksi dan menyelesaikan masalah yang ditemukan pada komputer, dan digunakan sebagai media untuk menyimpan dan memulihkan data tentang perubahan item data yang akan dilakukan. Itu juga dapat digunakan sebagai peringatan dari sistem. Ini dapat digunakan untuk memprediksi masalah system.

Menurut [11] *Filebeat* adalah pengirim ringan untuk meneruskan dan memusatkan data *log*. Di pasang sebagai agen di *server*, *Filebeat* memantau file *log* atau lokasi yang akan tentukan, mengumpulkan peristiwa *log*, dan meneruskannya ke *Elasticsearch* atau *Logstash* untuk di indeks.

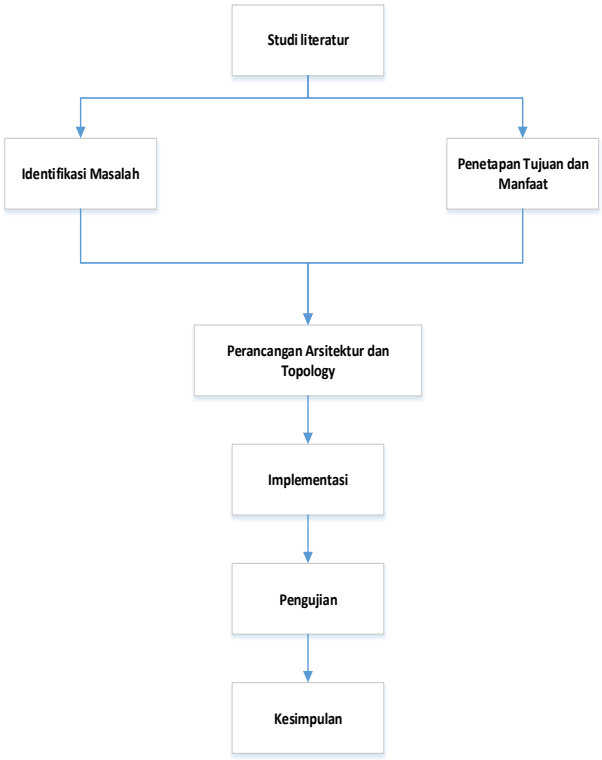
Menurut [12] Nginx adalah *software opensource* yang memiliki kinerja tinggi sebagai *server* HTTP dan *reverse proxy*. *Nginx* dengan cepat memberikan konten statis dengan penggunaan efisien sumber daya sistem. Hal ini dapat menyebarkan dinamis HTTP konten di jaringan menggunakan *FastCGI handler* untuk *script*, dan dapat berfungsi sebagai perangkat lunak yang sangat mampu penyeimbang beban. *Nginx* dapat dijalankan dan tersedia untuk *platform Unix, Linux*, varian dari *BSD, MacOS X*, *Solaris*, dan *Microsoft Windows*).

Menurut [13] *Docker* adalah sebuah aplikasi yang bersifat *opensource* yang berfungsi sebagai wadah/*container* untuk mengepak/memasukkan sebuah *software* secara lengkap beserta semua hal lainnya yang dibutuhkan oleh *software* tersebut dapat berfungsi. Pengaturan *software* beserta *file*/hal pendukung lainnya akan menjadi sebuah *Image* (istilah yang diberikan oleh *Docker*). Kemudian sebuah *instan* dari *Image* tersebut kemudian disebut *Containe.*

METODE

Adapun metode yang dipakai dalam penelitian ini adalah mengikuti tahapan kerangka kerja seperti pada Gambar 1.

DOI : https://doi.org/10.36341/rabit.v8i1.2975 *Corresponding Author* : Yuli Fitrisia



Gambar 1. Kerangka Kerja Penelitian

Studi Literatur

Berdasarkan literatur-literatur yang telah diterbitkan baik dari buku, jurnal dan proceeding, maka data–data yang mendukung akan dikumpulkan sehingga menjadi suatu pengetahuan yang saling mendukung antara satu dengan yang lainnya. Data ini akan menjadi dasar pengetahuan untuk merancang sistem yang akan dibangun.

Identifikasi Masalah

Berdasarkan latar belakang yang telah diuraikan, maka penulis merumuskan masalah yaitu:

1) Bagaimana cara pemantauan log pada service yang berjalan di server secara realtime.
2) Bagaimana cara menampilkan log service ke dashboard untuk divisualisasikan

Penetapan Tujuan dan Manfaat

Tujuan dari penelitian ini adalah implementasi monitoring server dengan Elasticsearch, Logstash dan Kibana (ELK Stack). Sedangkan hasil dari penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1) Sistem lebih mudah di *monitoring* dengan mengimplementasikan ELK *stack*.
2) Admin lebih dimudahkan dengan adanya tampilan *dashboard log server*.

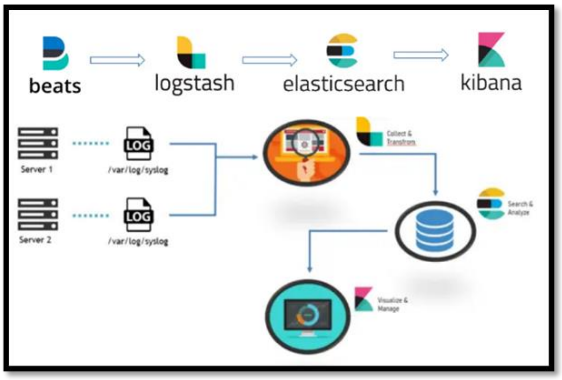
Perancangan Arsitektur dan Perancangan Topology

Pada penelitian ini, salah satu solusi manajemen *log* yang tersedia adalah ELK stack, solusi berbasis *opensource* yang dapat memelihara *log Real-time* untuk semua perangkat di infrastruktur IT. Kumpulan ELK terdiri dari empat komponen berbeda, yaitu:

1) *Beats, shipping agent* dengan kemampuan untuk mengirim data *log* ke *Logstash*.
2) *Logstash*, digunakan untuk mengumpulkan dan menganalisis data *log* dan *log* indeks yang di simpan di *Elasticsearch.*
3) *Elasticsearch, Search* dan *Analytics Engine* berbasis *Apache Lucene* yang bersifat *Open Source*.
4) *Kibana, Web Interface* yang memungkinkan untuk memvisualisasikan data dari *Elasticsearch* dalam berbagai format *grafik.*

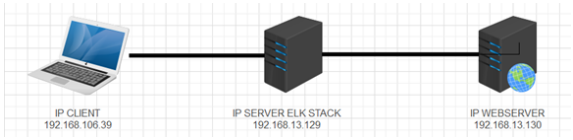
Keempat komponen di atas akan saling bekerjasama untuk *memonitor* dan mengamankan infrastruktur IT secara *Real-time* seperti pada Gambar 2 rancangan arsitektur sistem.

DOI : https://doi.org/10.36341/rabit.v8i1.2975 *Corresponding Author* : Yuli Fitrisia



Gambar 2. Arsitektur Sistem

Pada penelitian ini, Perancangan topologi yang akan digunakan adalah seperti Gambar 3. Pada topologi Gambar 3, Client berfungsi sebagai administrator yang akan mengakses server ELK Stack, Server ELK akan menampilkan hasil access log pada web server nginx yang akan digunakan.



Gambar 3. Perancangan Topology

Implementasi

Implementasi hasil perancangan dapat dilihat pada bagian HASIL yang menjelaskan tentang konfigurasi Docker io dan konfigurasi Elasticsearch, Logstash dan Kibana.

Pengujian

Pengujian hasil implementasi dapat dilihat pada Hasil Pengujian. Terdapat 4 skenario pengujian yang dilakukan yaitu pengujian log per hari, pengujian log per minggu, pengujian akurasi waktu masuknya Log ke ELK dan pengujian monitoring pada log web.

Kesimpulan

Kesimpulan dari penelitian yang dilakukan dapat dilihat pada akhir penelitian pada bagian Kesimpulan.

HASIL

Implementasi yang dilakukan berupa konfigurasi Docker io dan konfigurasi Elasticsearch, Logstash dan Kibana seperti dibawah ini:

Konfigurasi Docker io

Adapun tahapan konfigurasi Docker io yaitu lakukan *update* dan *upgrade* terlebih dahulu. *Update* dan *Upgrade* digunakan untuk menginstal versi terbaru dari semua paket saat ini yang diinstall pada sistem dari sumber-sumber yang disebutkan dalam /etc/apt/sources.list. Perintah untuk melakukan update dan upgrade yaitu:

```
#sudo apt update

#sudo apt upgrade
```

Lalu tuliskan perintah *install docker.io* untuk melakukan penginstalan. Pada perintah *Apt Install Docker.io* ini digunakan untuk menginstall aplikasi *docker*, jika dalam *linux ubuntu* salah satu opsi untuk menginstall sebuah aplikasi bisa menggunakan *command apt install* berikut:

```
#sudo apt install docker.io
```

Kemudia lakukan *enable* pada *docker*. Perintah *systemctl* adalah utilitas yang bertanggung jawab untuk memeriksa dan mengendalikan sistem systemd dan manajer layanan. Adapun perintah yang digunakan yaitu:

```
#systemctl enable docker
```

Kemudian cek status *docker*, Perintah systemctl status ini akan memberi tahu tentang keadaan sebuah layanan. Berdasarkan output dari perintah tersebut dapat dilihat bahwa status docker sudah aktif atau belum.

DOI : https://doi.org/10.36341/rabit.v8i1.2975 *Corresponding Author* : Yuli Fitrisia

```
#systemctl status docker
```

Konfigurasi Elasticsearch, Logstash dan Kibana

Adapun tahapan konfigurasi Elasticsearch, Logstash dan Kibana yaitu membuat *network* untuk *Docker* terlebih dahulu. Pada *command docker network create elastic* ini, *docker* akan membuat jaringan tersendiri yang bernama *elastic* agar saat penginstallan nanti dipermudah. Adapun perintah yang digunakan yaitu:

```
#docker network create elastic
```

Kemudian lihat list Network yang sudah dibuat. Perintah docker network list ini akan menampilkan list dari setiap network yang ada dan aktif.

```
#docker network list
```

Lalu install Elasticsearch ke dalam docker. Adapun perintah untuk install elasticsearch ini yaitu command –name itu akan memberi nama elk stacknya dan pada command -p 192.168.13.129:9200 dan 9300 itu dimana nanti setelah penginstallan dan pull image IP yang telah pasangkan seperti Gambar diatas dapat access.

```
#docker run -d --name es-node-01
--net elastic --rm -p
192.168.13.129:9200:9200 -p
192.168.13.129:9300:9300 -e
"discovery.type=single-node"
docker.elastic.co/elasticsearch
/elasticsearch:7.15.2
```

Kemudian install Kibana ke dalam Docker. Name yang berikan ialah kibana dan IP yang akan pasangkan sama dengan IP sebelumnya yang pasangkan.

```
#docker run -d --name kibana-01
--net elastic --rm -p
192.168.13.129:5601:5601 -e
"ELASTICSEARCH_HOSTS=http://es-
node-01:9200"
docker.elastic.co/kibana/kibana
:7.15.2
```

Kemudian install NginxLogGenerator. Pada perintah ini akan menginstall NginxLogGenerator ke dalam docker yang akan memberikan inisial, memberikan inisial itu agar setiap yang akan install dikenali oleh mesin.

```
#docker run --name
nginxLogGenerator --rm --label
type=nginxLog -d -e "RATE=10"
kscarlett/nginx-log-
generator:sha-5416ec2
```

Install Filebeat ke dalam *Docker*. *Filebeat* yang akan diinstal akan diletakkan di directory dan juga diletakkan didalam *container docker.*

```
#docker run -d \
 --name=filebeat \
 --net=elastic \
 --user=root \
 --rm \
 --
volume="$(pwd)/filebeat.docker.
yml:/usr/share/filebeat/filebea
t.yml:ro" \
 --
volume="/var/lib/docker/contain
ers:/var/lib/docker/containers:
ro" \
 --
volume="/var/run/docker.sock:/v
ar/run/docker.sock:ro" \

docker.elastic.co/beats/filebea
t:7.15.2 -e -strict.perms=false
```

DOI : https://doi.org/10.36341/rabit.v8i1.2975 *Corresponding Author* : Yuli Fitrisia

Pada command docker images ialah menampilkan isi dari images yang telah install seperti elasticsearch, kibana, filebeat, dan nginx.

```
#docker images
```

Adapun tampilan Docker Container ELK yang sudah install dan running. Pada perintah ini adalah cara menampilkan docker container yang sedang berjalan. Container yang berjalan diatas adalah images yang telah di download seperti elasticsearch, kibana, filebeat dan nginx.

```
#docker ps -a
```

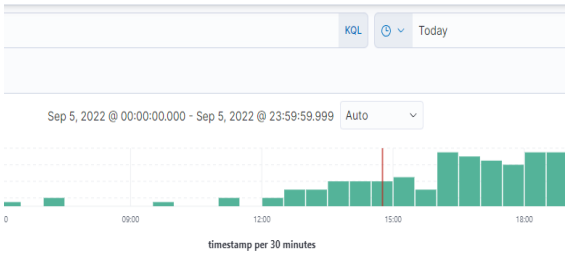
Kemudian ketikkan IP Address dari server ELK Stack pada Browser untuk menjalankan Dashboard ELK Stack. Setelah itu Dashboard ELK stack.

Hasil Pengujian

Adapun pengujian dilakukan dalam empat scenario yaitu:

1. Hasil Pengujian Log Perhari

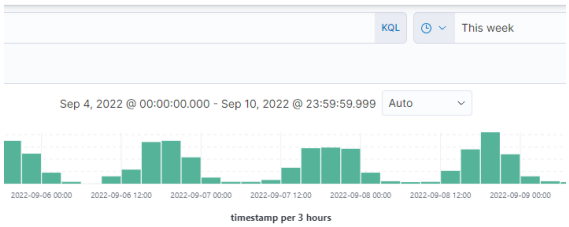
   Pengujian pertama ini bertujuan untuk melihat hasil *log* yang masuk Per-hari nya. Jika berhasil melakukan pengujian tersebut maka dapat dilihat pada *dashboard Kibana*. Pada Gambar 4 dapat dilihat tampilan halaman dashboard kibana yang menampilkan hasil dari proses *log* untuk jangka waktu perharinya. Selain itu juga dapat dilihat pada tampilan halaman *dashboard* tersebut terdapat *chart diagram* batang untuk menampilkan hasil *log* berbentuk *chart* diagram yang memudakan admin dalam membaca hasil data *log*. Pada Gambar 4 akan terlihat *log* yang terbaca setiap melakukan aksi.



Gambar 4. Log Per Hari

2. Hasil Pengujian Log Per Minggu

Pengujian kedua ini bertujuan untuk melihat hasil *log* yang masuk Per-minggu nya. Jika berhasil melakukan pengujian tersebut maka dapat dilihat pada *dashboard* Kibana. Pada Gambar 5 dapat dilihat tampilan halaman *dashboard kibana* yang menampilkan hasil dari proses *log* untuk jangka waktu perminggunya. Selain itu juga dapat dilihat pada tampilan halaman *dashboard* tersebut ada *chart diagram* batang untuk menampilkan hasil *log* berbentuk *chart diagram* yang memudakan admin dalam membaca hasil data *log*. Pada Gambar 5 akan terlihat *log* yang terbaca setiap user melakukan aksi per minggu.



Gambar 5. Log Per Minggu

3. Hasil Pengujian Akurasi Waktu Masuknya Log ke ELK

Pengujian ketiga ini bertujuan untuk melihat tingkat akurasi waktu antara terjadinya akses ke server dengan masuknya log pada ELK. Jika berhasil akan terlihat pada dasboard Kibana. Pada Gambar 6 dapat dilihat bahwa tampilan halaman dashboard terdapat log yang

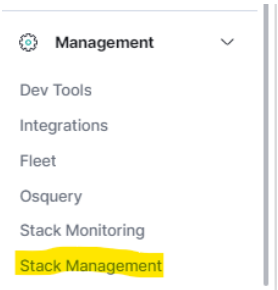
DOI : https://doi.org/10.36341/rabit.v8i1.2975 *Corresponding Author* : Yuli Fitrisia

masuk ke dalam table expanded document, setiap user melakukan aksi maka log berubah. Artinya ELK stack dapat menampilkan log secara real time. Log yang masuk tersebut dapat dilihat pada Gambar 6.

Gambar 6. Akurasi Waktu Masuknya Log ke ELK

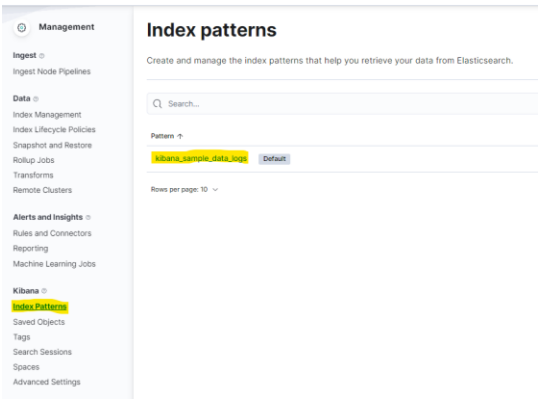
4. Hasil Pengujian Monitoring pada log web

Pengujian keempat ini bertujuan bagaimana cara melakukan *monitoring* pada *log web*. Untuk hasil yang di *monitoring* dapat dilihat pada *dashboard Kibana*. Adapun langkahnya yaitu: setelah tampil halaman *Dashboard Kibana*, lalu pada *Navigation Bar* dan Pilih *Stack Managent* seperti pada Gambar 7.

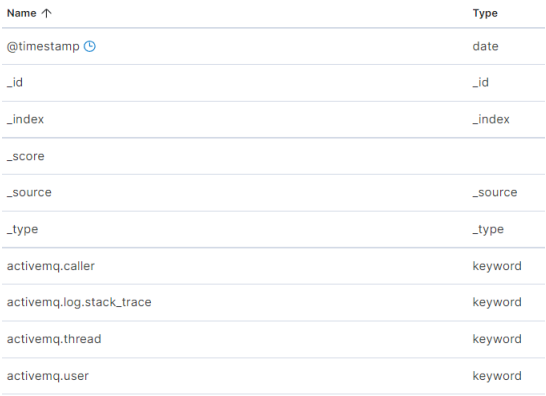


Gambar 7. Stack Management

Setelah Masuk Ke *Navigation Bar* terdapat pengaturan untuk *kibana* lalu pilih *Index Pattern* seperti pada Gambar 8. Lalu *Klik Index Pattern* dan lihat isi dari *Index Pattern* dari *Kibana* dan *log* yang akan *monitoring*. Lihat isi dari *index pattern* yang telah dibuat seperti apda Gambar 9.

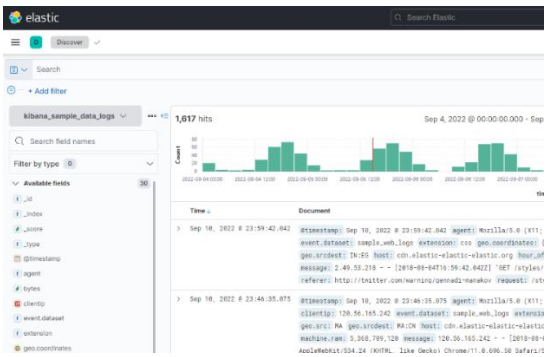


Gambar 8. Index Patterns



Gambar 9. Index Patterns

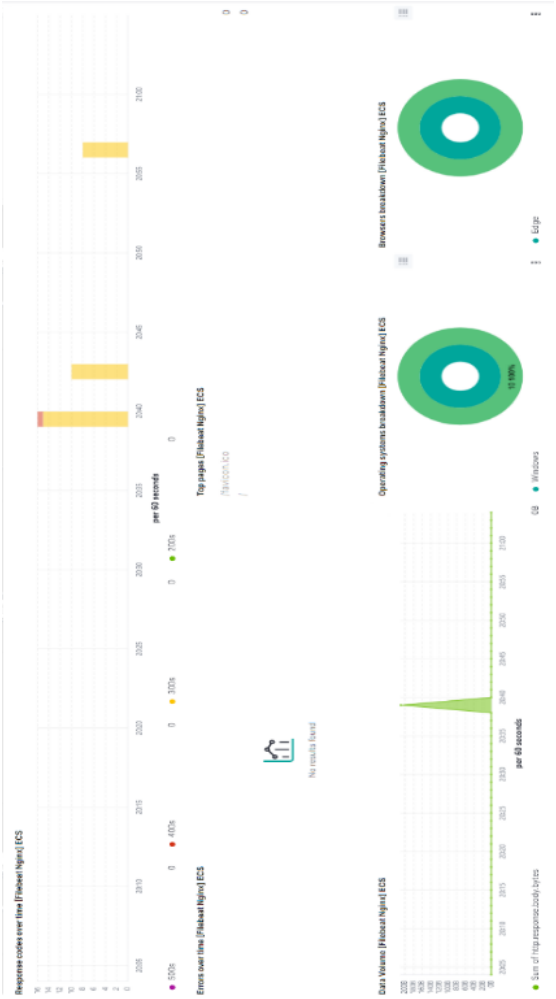
Setelah itu kembali ke halaman utama dan buka *navigation bar* dan pilih *Kibana – Discover*. Setelah itu bisa dilihat tampilanya seperti pada Gambar 10 untuk melakukan monitoring.



Gambar 10. Halaman Discover

DOI : https://doi.org/10.36341/rabit.v8i1.2975
Corresponding Author : Yuli Fitrisia

Berdasarkan hasil pengujian keempat ini yaitu mempermudah administrator dalam mengolah data *log* dan juga mempermudah untuk memonitoring sebuah *log* yang berbentuk grafik data *metric* dan data *log* yang dapat divisualisasikan ke dashboard seperti pada Gambar 11. Admin juga bisa membaca *log* dengan mudah dikarenakan setiap *log* yang masuk diberitahukan bahwa *user* menggunakan *web browser* apa yang mereka gunakan.



Gambar 11. Halaman Dashboard Data Metric dan Log

KESIMPULAN

Dari hasil pengujian dan analisis yang telah dilakukan, maka dapat diambil beberapa kesimpulan sebagai berikut:

a. Setiap *user* yang mengakses web server dapat ditampilkan pada *Dashboard Kibana* perhari ataupun perminggu.
b. ELK Stack Dashboard dapat menampilkan tingkat akurasi pada log berdasarkan akses pada web server secara realtime. Selain itu ELK Stack Dashboard dapat mempermudah administrator dalam monitoring log server dengan adanya tampilan pada Dashboard berupa grafik data *metric* dan data *log*.
c. Adapun pengembangan yang bisa dilakukan pada penelitian ini yaitu menggunakan ELK Stack Dashboard untuk memonitoring lebih dari satu jenis layanan server.

DAFTAR PUSTAKA

[1] M. N. Arifin, S. Sugiartowo, and E. Susilowati, "Desain dan Implementasi Log Event Management Server Menggunakan Elasticsearch Logstash Kibana (Elk)," *Semin. Nas. Sains dan Teknol.*, 2018, [Online]. Available: https://jurnal.umj.ac.id/index.php/semnastek/article/view/3451.

[2] P. H. Putra, "Implementasi Log Management Server Menggunakan Elk (Elastic Search , Logstash Dan Kibana) Stack Pada Server Web Snort Di Pt . Xyz," *J. Inform. Sunan Kalijaga*, vol. 4, 2020.

[3] E. P. Sartika and A. B. Cahyono, "Implementasi Elasticsearch Logstash Kibana Stack pada Sistem Portal Pengembangan dan Pembinaan Sumber Daya Manusia," 2020, [Online]. Available: https://journal.uii.ac.id/AUTOMATA/article/view/13872.

[4] W. Sholihah, S. Pripambudi, and A. Mardiyono, "Log Event Management Server Menggunakan Elastic Search Logstash Kibana (ELK Stack)," *J. Teknol. Inf. Dan Multimed.*, vol. Vol 2, No, 2020, [Online]. Available:

https://journal.sekawan-org.id/index.php/jtim/article/view/79.

[5] M. J. R. Putra and H. Saptono, "Penerapan Log Analyzer untuk Mengetahui Lalu Lintas Jaringan berbasis Elasticsearch, Logstash, dan Kibana," *J. Inform. Terpadu*, vol. Vol 8, No, 2022, [Online]. Available: https://journal.nurulfikri.ac.id/index.php/jit/article/view/388.

[6] D. Lintang, "Monitoring Aktivitas User Pada System Dengan Menggunakan Efk (Elasticsearch , Fluentd ," 2020.

[7] Elastic, "What is Elasticsearch ?," 2019. https://www.elastic.co/what-is/elasticsearch (accessed Nov. 11, 2022).

[8] M. S. S. Zakaria Husen, *Membangun Server dan Jaringan Komputer dengan Linux Ubuntu*. 2020.

[9] A. Tedyyana and R. Kurniati, "Membuat Web Server Menggunakan Dinamic Domain Name System Pada IP Dinamis," *J. Teknol. Inf. Komun. Digit. Zo.*, vol. 7, no. 1, pp. 1–10, 2016.

[10] A. Michael, "Sistem Monitoring Server Dengan Menggunakan SNMP," *Widyakala J.*, vol. 6, no. 2, p. 163, 2019, doi: 10.36262/widyakala.v6i2.218.

[11] S. Pripambudi, "Log Event Management Server Menggunakan Elastic Search Logstash Kibana (ELK Stack)," *JTIM J. Teknol. Inf. dan Multimed.*, vol. 2, no. 1, pp. 12–20, 2020, doi: 10.35746/jtim.v2i1.79.

[12] D. K. Hakim and D. Y. Yulianto, "Pengujian Algoritma Load Balancing pada Web Server Menggunakan NGINX," *JRST (Jurnal Ris. Sains dan Teknol.*, vol. 3, no. 2, p. 85, 2019, doi: 10.30595/jrst.v3i2.5165.

[13] M. Rexa, "Implementasi Load Balancing Server Web Berbasis Docker Swarm Berdasarkan Penggunaan Sumber Daya Memory Host," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 3, no. 4, pp. 3478–3487, 2019.