

Perancangan Dan Implementasi Monitoring Web Server Menggunakan Elastic Stack Dengan Infrastruktur Microsoft Azure

1st Mario Fanza Munzafa

Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia

mariofanza@student.telkomuniversity.ac.id

2nd Muhammad Iqbal

Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia

miqbal@tass.telkomuniversity.ac.id

3rd Agus Ganda Permana

Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia

gandapermana@tass.telkomuniversity.ac.id

Abstrak — Dalam era digital, ketersediaan dan keandalan layanan web menjadi faktor penting bagi perusahaan dan organisasi. Web server sebagai komponen inti penyedia layanan digital memerlukan sistem monitoring andal untuk memantau performa, penggunaan sumber daya, dan potensi gangguan secara real time. Proyek ini merancang dan mengimplementasikan sistem monitoring web server berbasis Elastic Stack (Elasticsearch, Logstash, Beats, Kibana) yang diintegrasikan pada infrastruktur cloud Microsoft Azure. Sistem dibangun dengan men-deploy web server Nginx pada virtual machine Azure, mengumpulkan log aktivitas dan metrik performa menggunakan Filebeat dan Metricbeat, lalu mengirimkannya ke Elasticsearch untuk dianalisis dan divisualisasikan melalui dashboard Kibana. Sistem dapat menampilkan informasi seperti trafik jaringan, status HTTP response code, penggunaan CPU dan memori, serta deteksi anomali. Hasil pengujian menunjukkan CPU server 3,05% saat idle dan meningkat seiring penambahan request (50–200), serta mampu mendeteksi kode 200 dan 400 dengan akurasi 100%. Elastic Stack dinilai efektif karena menyediakan sistem terpadu dibandingkan alat monitoring lain seperti Prometheus dan Grafana yang memerlukan konfigurasi terpisah.

Kata kunci—elasticstack, monitoring web server, microsoft azure, nginx, kibana, filebeat, metricbeat.

I. PENDAHULUAN

Di era digital saat ini, keberadaan web server yang stabil dan dapat diakses dengan cepat sangat penting bagi kelangsungan operasional berbagai jenis bisnis dan layanan. Web server bertugas untuk melayani permintaan dari pengguna di internet, baik itu untuk situs web, aplikasi web, maupun layanan lainnya. Namun, seperti halnya sistem teknologi informasi lainnya, web server juga rentan terhadap masalah teknis, seperti kegagalan perangkat keras, overload, atau bahkan serangan siber. Oleh karena itu, pemantauan kinerja dan kondisi web server secara real-time menjadi hal yang sangat penting untuk memastikan bahwa layanan yang diberikan dapat berjalan dengan baik, efisien, dan tanpa gangguan.

Masih banyak organisasi maupun perusahaan yang menghadapi kendala dalam hal visibilitas terhadap kondisi sistem mereka secara real-time. Saat ini sistem monitoring

yang tradisional dan manual masih bersifat terpisah, sulit diintegrasikan, dan tidak mampu memberikan analisis yang mendalam terhadap log aktivitas yang dihasilkan server. Hal ini akan menjadi penghambat untuk sistem tersebut karena tidak ada penanganan yang cepat maupun efisien. Oleh karena itu kebutuhan monitoring sistem yang modern sangat diperlukan.

Monitoring web server pada umumnya melibatkan pengumpulan dan analisis berbagai data kinerja seperti penggunaan CPU, memori, disk I/O, waktu respons, hingga tingkat trafik pengunjung. Beberapa metode monitoring bisa dilakukan salah satunya penerapan Elastic Stack sebagai tools alternatif pemantauan jaringan. Penggunaan Elastic Stack bertujuan untuk memonitoring trafik, host serta menganalisis log yang bersifat open-source. Monitoring trafik yang ditampilkan merupakan data yang berasal dari pengolahan log yang selanjutnya divisualisasikan untuk dilakukan analisis sesuai kebutuhan yang diperlukan. Elastic Stack terdiri dari Elasticsearch, Logstash, dan Kibana untuk mendukung web server log monitoring pada proyek akhir ini. Elastic Stack dibantu oleh Beats untuk proses pengiriman dan pengelolaan data.

II. KAJIAN TEORI

A. Infrastruktur Cloud Microsoft Azure

Microsoft Azure adalah salah satu layanan cloud computing dari Microsoft yang menyediakan berbagai layanan komputasi seperti penyimpanan data, virtual machine, database, jaringan maupun server. Azure memungkinkan pengguna untuk membangun, serta mengelola aplikasi di cloud tanpa harus memiliki dan mengelola infrastruktur sendiri. Dalam Microsoft Azure pengguna dibebaskan untuk membangun, mengelola, dan menyebarkan perangkat lunak pada jaringan global yang masif menggunakan Tools Favorite dan Frameworks. Microsoft Azure menyediakan Software as a Service (SaaS), Platform as a service (PaaS) dan Infrastructure as a Service (IaaS) dan mendukung banyak bahasa pemrograman, tools, dan Framework yang berbeda, termasuk perangkat lunak dan perangkat lunak pihak ketiga khusus dan Microsoft.

B. Elastic Stack

Elastic Stack adalah kumpulan perangkat lunak open-source yang digunakan untuk mengumpulkan, menyimpan, menganalisis, dan memvisualisasikan data dalam jumlah besar secara real-time. *Elastic Stack* terdiri dari tiga produk *open-source* yaitu *Kibana*, *Elasticsearch*, dan *Logstash*. Mereka dirancang untuk bisa saling terintegrasi satu sama lain untuk memantau sekaligus mengatasi masalah yang ada pada server dan memastikan lingkungan aman.

C. Elasticsearch

Elasticsearch adalah mesin pencari dan analitik terdistribusi yang digunakan untuk menyimpan dan mengindeks data dalam jumlah besar dengan waktu akses yang cepat. Dalam konteks *monitoring*, *Elasticsearch* berfungsi sebagai penyimpan utama log dan metrik performa server yang kemudian bisa dicari atau dianalisis lebih lanjut.

Elasticsearch sangat cocok untuk pengolahan data *real-time* karena memiliki kemampuan *full-text search* dan dukungan *scalability* yang tinggi, terutama dalam lingkungan jaringan dan riset edukasi.

D. Logstash

Logstash adalah *data processing pipeline* yang digunakan untuk mengumpulkan, memfilter, dan mentransformasi data sebelum dikirim ke *Elasticsearch*. Komponen ini mendukung banyak *input* dan *output plugins*, serta mampu menyesuaikan data log agar bisa dianalisis dengan lebih efektif. *Logstash* memungkinkan pemrosesan data log dari berbagai sumber seperti server, aplikasi, dan sistem jaringan secara otomatis dan mendalam.

E. Kibana

Kibana adalah antarmuka visualisasi dari *Elastic Stack* yang memungkinkan pengguna untuk membuat *dashboard*, grafik, dan visualisasi lainnya berdasarkan data yang disimpan di *Elasticsearch*. *Kibana* membantu dalam interpretasi data dan mempercepat proses pengambilan keputusan. *Kibana* sangat efektif digunakan dalam sistem *monitoring* untuk menampilkan data log secara *real-time*, serta memberikan kemudahan dalam mencari *error* atau anomali pada sistem layanan web.

F. Metricbeat

Kibana adalah antarmuka visualisasi dari *Elastic Stack* yang memungkinkan pengguna untuk membuat *dashboard*, grafik, dan visualisasi lainnya berdasarkan data yang disimpan di *Elasticsearch*. *Kibana* membantu dalam interpretasi data dan mempercepat proses pengambilan keputusan. *Kibana* sangat efektif digunakan dalam sistem *monitoring* untuk menampilkan data log secara *real-time*, serta memberikan kemudahan dalam mencari *error* atau anomali pada sistem layanan web

G. Filebeat

Filebeat adalah komponen *Elastic Stack* yang dirancang khusus untuk mengekstrak dan mengirimkan log dari berbagai file teks dari sebuah aplikasi yang ada di server contohnya *nginx*. *Filebeat* akan membaca file *access.log* dan *error.log* yang ada pada *nginx* dengan menggunakan modul bawaan yang akan sekaligus melakukan parsing data

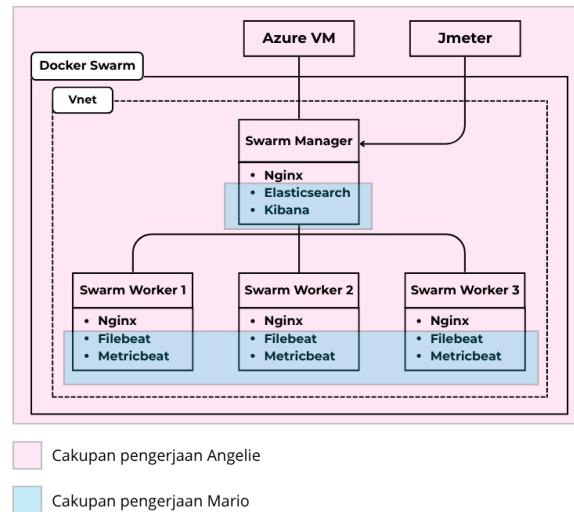
terhadap log tersebut secara otomatis, sehingga data yang dikirim bisa terstruktur dan mudah dianalisis. *Filebeat* adalah komponen *Elastic Stack* yang dirancang khusus untuk mengekstrak dan mengirimkan log dari berbagai file teks dari sebuah aplikasi yang ada di server

H. Infrastruktur Cloud Microsoft Azure

Microsoft Azure adalah salah satu layanan cloud computing dari Microsoft yang menyediakan berbagai layanan komputasi seperti penyimpanan data, virtual machine, database, jaringan maupun server. Azure memungkinkan pengguna untuk membangun, serta mengelola aplikasi di cloud tanpa harus memiliki dan mengelola infrastruktur sendiri. Dalam Microsoft Azure pengguna dibebaskan untuk membangun, mengelola, dan menyebarkan perangkat lunak pada jaringan global yang masif menggunakan Tools Favorite dan Frameworks. Microsoft Azure menyediakan Software as a Service (SaaS), Platform as a service (PaaS) dan Infrastructure as a Service (IaaS) dan mendukung banyak bahasa pemrograman, tools, dan Framework yang berbeda, termasuk perangkat lunak dan perangkat lunak pihak ketiga khusus dan Microsoft.

III. PEMODELAN DAN PERANCANGAN

A. Arsitektur Sistem



GAMBAR 1
Arsitektur Sistem

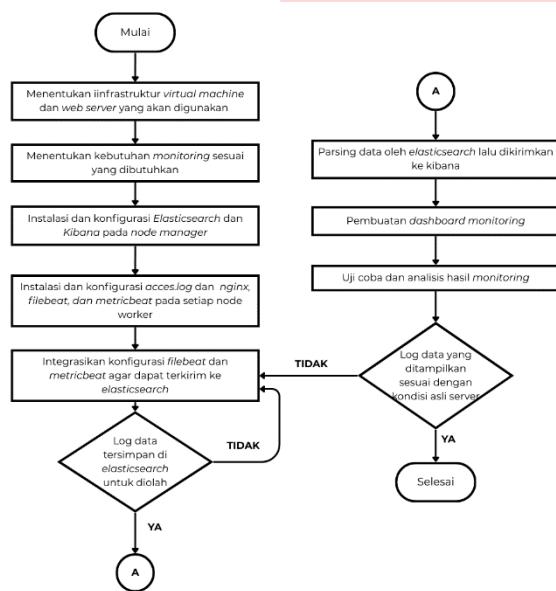
Pada arsitektur sistem yang digambarkan pada gambar di atas, perancangan sistem monitoring ini menggunakan arsitektur berbasis *Elastic Stack* yang terdiri dari beberapa layanan utama yaitu diantranya adalah *Elasticsearch*, *Kibana*, *Logstash*, *Metricbeat*, dan *Filebeat* yang semuanya berjalan di satu lingkungan *virtual machine* yang sama yaitu *Virtual Machine Azure*. Secara umum, konfigurasi serta penginstalan dari *Elasticsearch* dan *Kibana* berada di *node manager* agar bisa memantau kinerja dari ketiga *node manager*.

Elasticsearch ini memiliki sebuah peran utama untuk melakukan pengolahan dan penyimpanan data log serta metrik, *Elasticsearch* juga berfungsi sebagai mesin pencari log yang dikirimkan oleh *filebeat* dan *metricbeat*. Selanjutnya data data yang tersimpan tadi dikumpulkan

untuk ditampilkan di *dashboard* monitoring di dalam Kibana. Lalu penginstalan dari *filebeat* dan *metricbeat* dilakukan di setiap *node worker* agar kinerja dari setiap *node worker* beserta isinya berupa log bisa dikirimkan melalui *logstash* ke *elasticsearch* dan juga *kibana*. Informasi yang dikirimkan oleh *metricbeat* berupa penggunaan *CPU*, penggunaan memori, penggunaan *disk*, dan proses aktif yang berhubungan dengan kinerja server tiap *node worker*. Sementara informasi yang dikirimkan oleh *filebeat* adalah kinerja aplikasi yang berjalan di *node worker*, aplikasi yang digunakan disini adalah web server *nginx*.

Dengan perancangan ini, sistem mampu melakukan pemantauan secara real-time terhadap performa dan log dari setiap Worker dalam satu kesatuan visualisasi terpusat. Arsitektur ini juga dapat dikembangkan lebih lanjut untuk menyertakan alerting, analisis keamanan, dan otomasi respons insiden melalui integrasi fitur tambahan dari Elastic Stack.

B. Alur Perancangan



GAMBAR 2

Alur Perancangan

Tahap implementasi dimulai dengan menentukan infrastruktur virtual machine (VM) yang berperan sebagai node utama dan node pekerja, disesuaikan dengan kebutuhan performa dan skalabilitas, serta memilih web server yang akan dimonitor (Nginx). Selanjutnya dilakukan identifikasi kebutuhan monitoring, seperti log akses, log kesalahan, serta metrik CPU, memori, disk, dan jaringan, untuk menentukan konfigurasi modul pada agen Filebeat dan Metricbeat di setiap node pekerja. Node utama dikonfigurasi dengan Elasticsearch sebagai pusat penyimpanan dan pencarian data, serta Kibana untuk visualisasi interaktif. Setelah instalasi dan konfigurasi agen monitoring selesai, data log dan metrik dikirim otomatis ke Elasticsearch, diproses melalui pipeline indexing, dan divisualisasikan di dashboard Kibana, baik dengan modul bawaan maupun dashboard kustom. Uji coba dilakukan dengan memberikan beban akses ke server untuk memastikan seluruh komponen bekerja terintegrasi dengan baik. Jika hasil sesuai kondisi aktual, implementasi

dinyatakan berhasil, sedangkan jika ditemukan kendala dilakukan evaluasi ulang. Proses ini menghasilkan sistem monitoring berbasis Elastic Stack yang efektif memberikan visibilitas menyeluruh terhadap performa dan kondisi web server secara real time.

C. Perancangan Sistem

Tahapan penting dalam pembangunan sistem monitoring ini adalah melakukan konfigurasi pada dua komponen utama Elastic Stack, yaitu Elasticsearch dan Kibana. Keduanya dipasang dan dikonfigurasi pada Node Manager yang bertindak sebagai pusat pemrosesan, penyimpanan, dan visualisasi data hasil monitoring. Konfigurasi dilakukan sedemikian rupa agar sistem dapat berjalan secara optimal, aman, dan terintegrasi dengan komponen lain seperti Filebeat dan Metricbeat dari node worker.

Port	Protocol	External Port	Container Port	Protocol	External IP	Container IP	Allow
3000	SSH	22	22	TCP	Any	Any	<input checked="" type="checkbox"/>
3200	HTTPS	443	443	TCP	Any	Any	<input checked="" type="checkbox"/>
3400	HTTP	80	80	TCP	Any	Any	<input checked="" type="checkbox"/>
4000	Awesomeness	8080	8080	TCP	Any	Any	<input checked="" type="checkbox"/>
10000	Awesomeness	8080	8080	TCP	Any	Any	<input checked="" type="checkbox"/>
10200	Awesomeness	8080	8080	TCP	Any	Any	<input checked="" type="checkbox"/>
10300	Awesomeness	8080	8080	TCP	Any	Any	<input checked="" type="checkbox"/>
10400	Awesomeness	8080	8080	TCP	Any	Any	<input checked="" type="checkbox"/>
10500	Awesomeness	8080	8080	TCP	Any	Any	<input checked="" type="checkbox"/>
30000	Awesomeness	8080	8080	TCP	Any	Any	<input checked="" type="checkbox"/>
30100	Awesomeness	8080	8080	TCP	Any	Any	<input checked="" type="checkbox"/>
30200	Awesomeness	8080	8080	TCP	Any	Any	<input checked="" type="checkbox"/>
30300	Awesomeness	8080	8080	TCP	Any	Any	<input checked="" type="checkbox"/>
30400	Awesomeness	8080	8080	TCP	Any	Any	<input checked="" type="checkbox"/>

GAMBAR 3 Konfigurasi Port

Tahap pertama dalam konfigurasi atau penginstalan kita harus mengatur Network Security Group pada Azure yang berfungsi sebagai lapisan kontrol lalu lintas jaringan (network traffic) yang mengizinkan atau memblokir koneksi ke virtual Pemachine berdasarkan aturan tertentu, seperti nomor port, alamat IP, dan protokol. Untuk mengizinkan akses ke Elasticsearch (Port:9200) dan Kibana (Port:5601), perlu dilakukan penambahan aturan masuk (inbound rules) pada NSG yang terhubung dengan virtual machine atau network interface dari Node Manager.

D. Konfigurasi Elastic dan Kibana

Proses konfigurasi sistem monitoring dimulai dengan menyiapkan dua komponen inti dari Elastic Stack, yaitu Elasticsearch dan Kibana, pada Node Manager. Kedua layanan ini diinstal dan dijalankan menggunakan teknologi container berbasis Docker, dengan cara meng-clone langsung dari repository resmi Elastic yang tersedia di GitHub. Metode ini dipilih karena memberikan kemudahan dalam pengelolaan versi, konsistensi lingkungan, serta fleksibilitas dalam proses deployment.

1. Tahap ini dilakukan di dalam node manager oleh karena itu harus melakukan login ssh ke dalam IP Node manager menggunakan perintah “ssh <namaserverver>@IP_Public”

```
PS C:\Users\ACER NITRO 5> ssh swarm@20.2.82.132
Last login: Thu Jul 24 22:09:33 2025 from
180.253.252.57
example@DockerManager:~$
```

2. Selanjutnya melakukan penginstalan elastic menggunakan docker untuk mendapatkan file master dan file config yang diperlukan.

```
git clone https://github.com/elastic/stack-docker.git
```

3. Setelah itu masuk kedalam direktori “docker-elk” untuk melakukan konfigurasi file .env dengan tujuan supaya menyimpan variabel lingkungan (environment variables) secara terpusat dan terpisah dari file konfigurasi utama serta memberikan fleksibilitas, keamanan, dan kemudahan dalam manajemen konfigurasi.

```
ELASTIC_VERSION=9.0.4
## Password untuk layanan ELK
ELASTIC_PASSWORD='changeme'
LOGSTASH_INTERNAL_PASSWORD='changeme'
KIBANA_SYSTEM_PASSWORD='changeme'
```

4. Mengkonfigurasi file “docker-compose.yml” untuk mendefinisikan dan menjalankan beberapa layanan Docker container secara bersamaan.

```
services:
  setup:
    profiles:
      - setup
    build:
      context: setup/config
      args:
        ELASTIC_VERSION: 9.0.4
    init: true
    volumes:
      - ./setup_elastic/entrypoint.sh
      - ./setup_elastic/lib.sh:/lib.sh
      - ./setup_elastic/roles:/roles
    environment:
      ELASTIC_PASSWORD: "changeme"
      ELASTIC_VERSION: 9.0.4
    command:
      - bin/kibana-encryption-keys
      - generate
    network_mode: none
  elasticsearch:
    build:
      context: elasticsearch/config
      args:
        ELASTIC_VERSION: 9.0.4
    volumes:
      - elasticsearch:/usr/share/elasticsearch/data
    ports:
      - 9200:9200
    networks:
      - elk
  restart: none
```

```
depends_on:
  - elastic_search
restart: none
kibana:
  build:
    context: kibana/config
    args:
      ELASTIC_VERSION: 9.0.4
  volumes:
    - ./kibana/config/kibana.yml:/usr/share/kibana/config/kibana.yml
  ports:
    - 5601:5601
  environment:
    KIBANA_PASSWORD: "changeme"
  networks:
    - elk_network
  depends_on:
    - elastic_search
  restart: none
networks:
  elk-network:
    drive: bridge
  volumes:
    elastic_search:
    network_mode: service:elastic_search
    depends_on:
      - elastic_search
  kibana-genkeys:
    profiles:
      - setup-conf
    build:
      context: kibana/config
      args:
        ELASTIC_VERSION: 9.0.4
    command:
      - bin/kibana-encryption
      - generate
    network_mode: none
```

5. Terakhir menjalankan semua layanan Elastic yang sudah terinstal di docker container secara bersamaan.

```
docker compose up -d
```

E. Konfigurasi Metricbeat dan Filebeat

Setelah layanan inti Elasticsearch dan Kibana dikonfigurasi pada Node Manager, tahap berikutnya adalah menginstal dan mengonfigurasi dua agen monitoring utama pada node worker, yaitu Metricbeat dan Filebeat. Kedua komponen ini berperan penting dalam proses pengumpulan data dari sistem, yang kemudian dikirim ke Elasticsearch untuk dianalisis dan divisualisasikan melalui Kibana.

1. Pada tahap pertama, melakukan login ssh kedalam tiap node worker, dengan menggunakan perintah “ssh <username>@<IP_WORKER>”
Khusus untuk worker 3, diakses dengan menggunakan ip private.

2. Instalasi filebeat di setiap node worker menggunakan paket resmi dari Elastic.

```
curl -L -O
https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-8.12.2-amd64.deb
sudo dpkg -i filebeat-9.0.4-amd64.deb
```

3. Selanjutnya, konfigurasi file yml dari metricbeat dengan menggunakan perintah “sudo nano /etc/filebeat/filebeat.yml”

```
filebeat.inputs:
- type: filestream
  id: my-filestream-id
  enabled: enable
  paths:
    - /nginx-logs/access.log
    - /nginx-logs/error.log

filebeat.config.modules:
  path: ${path.config}/modules.d/*.yml
  reload.enabled: false

setup.template.settings:
  index.number_of_shards: 1

setup.kibana:
  host: "20.2.81.132:5601"

output.elasticsearch:
  hosts: ["20.2.81.132:9200"]
  preset: balanced
  username: "elastic"
  password: "changeme"

processors:
  - add_host_metadata:
    when.not.contains.tags: forwarded
  - add_cloud_metadata: ~
  - add_docker_metadata: ~
  - add_kubernetes_metadata: ~
```

4. Instalasi metricbeat di setiap node worker menggunakan paket resmi dari Elastic.

```
curl -L -O
https://artifacts.elastic.co/downloads/beats/metricbeat/metricbeat-9.0.4-amd64.deb
sudo dpkg -i metricbeat-8.12.2-amd64.deb
```

5. Selanjutnya, konfigurasi file yml dari metricbeat dengan menggunakan perintah “sudo nano /etc/metricbeat/metricbeat.yml”

```
metricbeat.config.modules:
  path: ${path.config}/modules.d/*.yml
  reload.enabled: false

setup.template.settings:
  index.number_of_shards: 1
  index.codec: best_compression
  #_source.enabled: false

setup.kibana:
  host: "http://20.2.81.132:5601"
  output.elasticsearch:
    hosts: ["http://20.2.81.132:9200"]
    preset: balanced
    username: "elastic"
    password: "changeme"
  processors:
    - add_host_metadata: ~
    - add_cloud_metadata: ~
    - add_docker_metadata: ~
```

6. Menjalankan kedua sistem tersebut di masing masing node worker, dari masing masing layanan.

```
sudo systemctl start metricbeat
```

```
sudo systemctl start filebeat
```

7. Terakhir harus konfigurasi log dari kedua layanan tersebut agar bisa muncul index di layanan pemantauan kibana

```
sudo filebeat setup --dashboards
```

```
sudo metricbeat setup --dashboards
```

Setelah Metricbeat dan Filebeat dijalankan, proses selanjutnya adalah tahap pengujian dan verifikasi layanan dengan membuka antarmuka Kibana, lalu mengecek apakah index pattern metricbeat-* dan filebeat-* telah terbentuk. Apabila berhasil, data dari masing-masing node worker akan mulai terlihat dalam bentuk dashboard dan visualisasi metrik secara real-time.

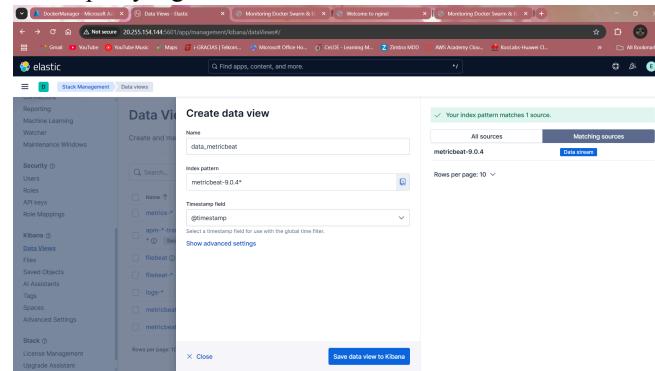
IV. PENGUJIAN DAN HASIL

Pengujian dilakukan untuk memastikan bahwa seluruh komponen dalam sistem monitoring telah terpasang dan berfungsi sebagaimana mestinya. Terdapat tiga fokus utama dalam proses pengujian ini. Pertama, dilakukan pengujian untuk memastikan bahwa data yang dikirim oleh Filebeat dan Metricbeat dari setiap node worker dapat diterima dengan baik dan ditampilkan di Kibana melalui dashboard serta fitur Discover. Pengujian ini menjadi indikator keberhasilan integrasi antara agen pengumpul data dan pusat penyimpanan serta visualisasi (Elasticsearch dan Kibana). Umumnya pengujian dibuat dalam bentuk tabel, terdapat nama pengujian, tujuan pengujian, hasil pengujian dan kesimpulan hasil penelitian, kemudian didukung grafik

dilanjutkan deskripsi naratif [10 pts]. Berikan kemungkinan pengembangan atau penelitian ke depan terkait penelitian ini

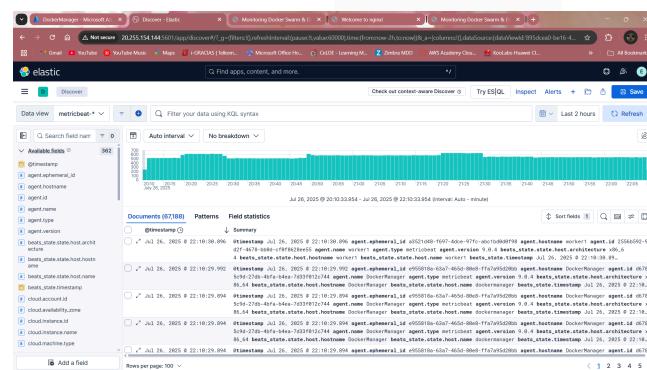
A. Log Data Kibana

Pengujian ini dilakukan untuk memastikan bahwa data log yang dikumpulkan oleh Filebeat dari masing-masing node worker berhasil terkirim ke Elasticsearch dan dapat divisualisasikan melalui Kibana. Fokus utama dari pengujian ini adalah mengevaluasi apakah proses pengiriman log berjalan dengan baik serta memastikan bahwa log yang dikirim mencerminkan aktivitas aktual dari web server Nginx, seperti permintaan HTTP yang masuk dan respon yang diberikan



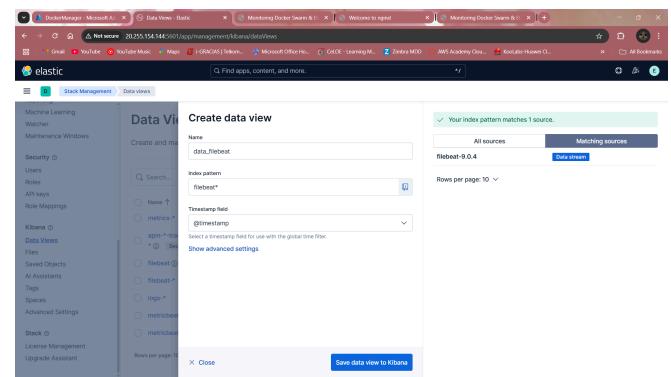
GAMBAR 4
Log Data Metricbeat

Pada gambar di atas terlihat bahwa *index pattern* “metricbeat-*” sudah muncul di menu *Data View* Kibana dan sudah bisa dipakai untuk menampilkan keseluruhan data dari *metricbeat*



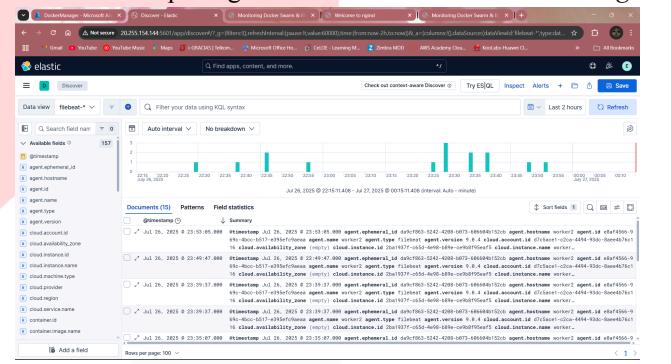
GAMBAR 5
Tampilan Discover Metricbeat

Bisa dilihat pada gambar di atas bahwa *data log* dari *index metricbeat* sudah bisa muncul secara lengkap dan bisa digunakan untuk menampilkan data di *dashboard* kibana.



GAMBAR 6
Log Data Filebeat

Pada gambar di atas terlihat bahwa *index filebeat* sudah muncul dan dapat digunakan untuk *dashboard monitoring*



GAMBAR 7
Tampilan Discover Filebeat

Pada gambar di atas menunjukkan bahwa log data dari *index filebeat* sudah muncul seutuhnya, terlihat data berupa berapa banyak jumlah akses di sebuah *web server*. Hasil pengujian menunjukkan bahwa *log* dari setiap *node worker* berhasil dikirim dan ditampilkan secara real-time di *Kibana*. Hal ini menunjukkan bahwa proses pengumpulan data oleh *Filebeat* berjalan dengan baik, serta koneksi antara *Filebeat*, *Elasticsearch*, dan *Kibana* telah terjalin dengan benar. Selain itu, modul *nginx* pada *Filebeat* juga terbukti aktif karena dapat memetakan log menjadi format terstruktur yang mudah dianalisis di *Kibana dashboard*.

B. Monitoring Server

Pengujian ini berfokus terhadap kinerja server, yang mencakup pemantauan penggunaan sumber daya sistem seperti CPU, memori, disk, dan network traffic. Pengujian ini bertujuan untuk mengevaluasi apakah server dalam kondisi stabil dan mampu menjalankan beban monitoring secara kontinu tanpa hambatan. Dalam pengujian ini juga dilakukan 4 skenario jumlah request, yaitu 50,100,150, dan 200 dalam 1 detik, dan juga saat kondisi idle.



GAMBAR 8
Dashboard Monitoring saat Idle

Berdasarkan hasil pemantauan server menggunakan Metricbeat yang ditampilkan pada dashboard di Kibana, dapat diambil data sebagai berikut :

- Penggunaan CPU – 3.05%
- Penggunaan Memori – 41.6%
- Load Gauge (5m Load) – 0.42
- Inbound Traffic – 16 KB/s (Total: 1,003.7 MB)
- Outbound Traffic – 25 KB/s (Total: 1,009.7 MB)



GAMBAR 9
Dashboard monitoring saat 50 Request

Berdasarkan hasil pengujian 50 *request* dan melakukan pemantauan server menggunakan Metricbeat yang ditampilkan pada dashboard di Kibana, dapat diambil data sebagai berikut :

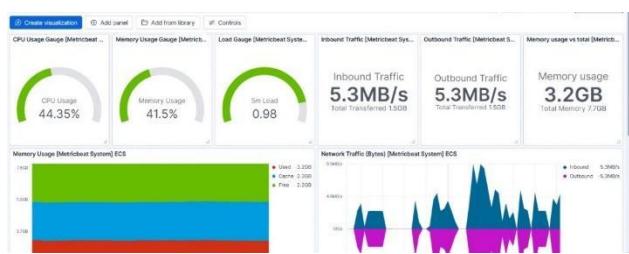
- Penggunaan CPU – 26.6%
- Penggunaan Memori – 41.2%
- Load Gauge (5m Load) – 0.62
- Inbound Traffic – 2.6 MB/s (Total: 941.5 MB)
- Outbound Traffic – 2.6 MB/s (Total: 947.2 MB)



GAMBAR 10
Dashboard Monitoring saat 100 request

Berdasarkan hasil pengujian 100 *request* dan melakukan pemantauan server menggunakan Metricbeat yang ditampilkan pada dashboard di Kibana, dapat diambil data sebagai berikut :

- Penggunaan CPU – 32.75%
- Penggunaan Memori – 41.6%
- Load Gauge (5m Load) – 0.48
- Inbound Traffic – 4.1 KB/s (Total: 1,1 GB)
- Outbound Traffic – 4.1 KB/s (Total: 1,1 GB)



GAMBAR 11 Dashboard Monitoring saat 150 Request

Berdasarkan hasil pengujian 150 *request* dan melakukan pemantauan server menggunakan Metricbeat yang ditampilkan pada dashboard di Kibana, dapat diambil data sebagai berikut :

- Penggunaan CPU – 44.35%
- Penggunaan Memori – 41.5%
- Load Gauge (5m Load) – 0.98
- Inbound Traffic – 5.3 MB/s (Total: 1,5 GB)
- Outbound Traffic – 5.3 MB/s (Total: 1,5 GB)



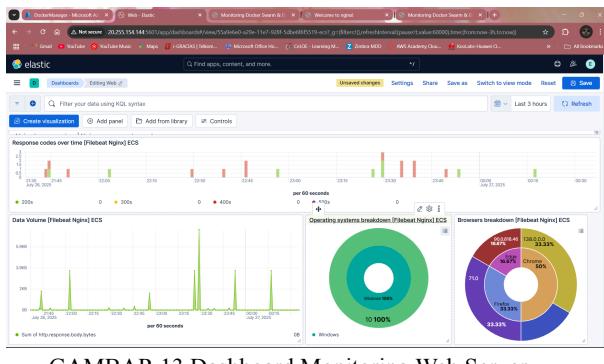
GAMBAR 12
Dashboard Monitoring saat 200 Request

Berdasarkan hasil pengujian 200 *request* dan melakukan pemantauan server menggunakan Metricbeat yang ditampilkan pada dashboard di Kibana, dapat diambil data sebagai berikut :

- Penggunaan CPU – 59.5%
- Penggunaan Memori – 42%
- Load Gauge (5m Load) – 2.45
- Inbound Traffic – 13.4 MB/s (Total: 2,4 GB)
- Outbound Traffic – 13.7 MB/s (Total: 2,4 GB)

C. Monitoring Web Server

Tahap pengujian ini berfokus pada kinerja layanan web server (Nginx) yang dijalankan di setiap node worker. Pengujian ini meliputi pengamatan terhadap jumlah permintaan yang diproses, serta analisis status kode HTTP yang dihasilkan. Tujuan dari pengujian ini adalah untuk memastikan bahwa layanan web server dapat berjalan optimal dan data aktivitasnya tercatat dengan baik dalam sistem monitoring.



GAMBAR 13 Dashboard Monitoring Web Server

Pemantauan aktivitas web server (Nginx) pada setiap node worker dilakukan menggunakan Filebeat dengan modul nginx, yang secara otomatis mengambil serta mengirimkan log akses dan error dari layanan Nginx ke Elasticsearch, kemudian menampilkan dalam bentuk visualisasi di Kibana. Hasil visualisasi pada dashboard Filebeat Nginx menunjukkan beberapa metrik utama yang menjadi acuan penilaian performa dan aktivitas layanan. Grafik *Response Codes Over Time* memperlihatkan bahwa kode respons 200 (OK) muncul secara konsisten, menandakan server mampu memproses permintaan klien dengan baik, sementara kemunculan beberapa kode 400-an mengindikasikan adanya permintaan tidak valid, seperti URL salah atau akses ditolak. Grafik *Data Volume* menunjukkan lonjakan data pada waktu tertentu, misalnya sekitar pukul 23:30, yang menandakan peningkatan aktivitas akses halaman tanpa adanya gangguan atau error. Diagram *Operating Systems Breakdown* mengungkap bahwa seluruh permintaan berasal dari perangkat dengan sistem operasi Windows, sedangkan *Browsers Breakdown* mencatat distribusi penggunaan browser dengan Chrome sebesar 50%, Firefox 33,33%, dan Edge 16,67%. Data ini juga mencatat alamat IP pengakses, mencerminkan keragaman klien yang terhubung. Secara keseluruhan, hasil monitoring membuktikan bahwa web server dapat memproses permintaan HTTP dengan baik, dan modul nginx pada Filebeat mampu melakukan parsing serta pengiriman log secara real-time, memberikan informasi teknis sekaligus wawasan pola penggunaan dan potensi anomali pada sistem.

D. Perbandingan Elastic Stack dengan Prometheus+Grafana

Berdasarkan hasil pengujian di atas bisa saya simpulkan bahwa alat monitoring Elastic Stack memiliki beberapa kelebihan dibandingkan dengan alat monitoring Prometheus + Grafana yang ditampilkan pada tabel berikut :

TABEL 1

Perbandingan ElasticStack dengan Prometheus+Grafana

Aspek Perbandingan	Elastic Stack	Prometheus + Grafana
Jenis Data yang Dimonitor	Mendukung <i>metrics</i> , <i>logs</i> , dan <i>traces</i> dalam satu ekosistem (<i>observability lengkap</i>).	Fokus metrik time-series, log butuh tool tambahan (mis. Loki)
Pencarian Data	<i>Full-text search</i> pada log & metrik dengan filter detail	Tidak ada <i>full-text search</i> log bawaan

Integrasi Web Server (Nginx)	Modul siap pakai (Metricbeat/Filebeat) yang otomatis mem-parse log	Perlu konfigurasi manual untuk metrik, dan log harus pakai tool tambahan
Visualisasi	Memiliki <i>dashboard</i> bawaan serta bisa melakukan <i>drag and drop</i>	Mengharuskan membuat <i>dashboard</i> manual menggunakan query

V. KESIMPULAN

Berdasarkan hasil perancangan, implementasi, dan pengujian, dapat disimpulkan bahwa sistem monitoring berbasis Elastic Stack berhasil dibangun dan diimplementasikan pada lingkungan cluster yang terdiri dari satu node manager dan tiga node worker menggunakan layanan virtual machine di Microsoft Azure. Komponen utama Elastic Stack seperti Elasticsearch dan Kibana berhasil dikonfigurasi menggunakan Docker container dan berfungsi dengan baik dalam mengelola serta menampilkan data log dan metrik secara terpusat. Instalasi Filebeat dan Metricbeat pada setiap node worker mampu mengirimkan data log serta informasi performa sistem ke Elasticsearch, yang kemudian berhasil divisualisasikan secara real-time melalui dashboard Kibana. Hasil pengujian menunjukkan bahwa sistem mampu menampilkan informasi performa server, seperti penggunaan CPU sebesar 3,05% pada kondisi idle dan meningkat sesuai skenario beban 50, 100, 150, hingga 200, serta menampilkan **response code** 200 dan 400 dengan tingkat akurasi 100%. Elastic Stack juga memiliki keunggulan dalam visualisasi real-time yang efektif karena terintegrasi dalam satu kesatuan sistem, berbeda dengan sistem lain seperti Prometheus dan Grafana yang memerlukan konfigurasi terpisah.

REFERENSI

- [1] S. A. Indrarto and A. Basuki, "Penerapan Platform Visualisasi dan Analisis Trafik Jaringan menggunakan Elastic Stack," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 9, pp. 4562-4570, 2022.
- [2] H. Karmila and H. Saptono, "IMPLEMENTASI VISUALISASI LOG SERVER DENGAN ELK STACK: ANALISIS KASUS LOG AKSES PADA SISTEM ELENA DI SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI," *Jurnal Informatika Terpadu*, vol. 10, no. 1, pp. 59-60, 2024.
- [3] K. A. Ashari, I. Mardianto and D. Sugiarto, "Analisa Performa RStudio Server Berbasis Cloud Menggunakan Elastic Stack sebagai Sistem Manajemen Metrik," *Jurnal Edukasi dan Penelitian Informatika*, vol. 7, no. 3, p. 450, 2021.
- [4] P. N. K. Bayu, "Implementasi Server Log Monitoring System menggunakan Elastic Stack," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 4, pp. 1814-1824, 2022.

- [5] R. Hefiana and Y. Fernando, "Analisis Perbandingan Elastic Compute Service (ECS) Instance Alibaba Cloud dengan Virtual Machine Azure," *KLIK: Kajian Ilmiah Informatika dan Komputer*, vol. 4, no. 4, pp. 2158-2168, 2024.
- [6] K. S. R. Surya, A. Bhawiyuga and A. Basuki, "Implementasi Monitoring Platform untuk Microservice Architecture menggunakan Elastic Stack(Studi Kasus : Alterra Indonesia)," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 4, pp. 1668-1676, 2022.
- [7] E. Nurmiati, "ANALISIS DAN PERANCANGAN WEB SERVER PADA HANDPHONE," *Studia Informatika: Jurnal Sistem Informasi*, vol. 5, no. 2, pp. 1-17, 2012.
- [8] I. Mardianto, D. Sugiarto and K. A. Ashari, "Elastic Stack Ability Test Monitoring Slowloris Attack on Digital Ocean Server," *Jurnal Teknik Informatika*, vol. 13, no. 2, p. 120, 2021.
- [9] A. Setiyawan, A. Pinandito and W. Purnomo, "Pengembangan Sistem Informasi Log Management Server Monitoring Menggunakan ELK (Elastic Search, Logstash dan Kibana) Stack pada Aplikasi Padichain di PT. Bank Rakyat Indonesia," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 7, no. 5, pp. 2142-2151, 2023.
- [10] M. R. F. Fathin, A. Basuki and A. Bhawiyuga, "Penerapan Elastic Stacksebagai Platform Visualisasi dan Analisis Trafik pada Jaringan Riset dan Edukasi," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 6, pp. 2664-2672, 2022.
- [11] M. J. R. Putra and H. Saptono, "PENERAPAN LOGANALYZER LOGUNTUK MENGETAHUI LALULINTAS JARINGAN BERBASIS ELASTICSEARCH, LOGSTASH, DANKIBANA," *Jurnal InformatikaTerpadu*, vol. 8, no. 1, pp. 21-25, 2022.