

## **BAB V**

### **IMPLEMENTASI DAN PENGUJIAN**

Bab ini akan berisikan tentang hasil implementasi serta pengujian dari rancangan dashboard monitoring dan analisa serangan aplikasi web menggunakan ELK Stack yang telah dibuat. Pada tahapan implementasi akan mencakup segala hal yang telah dilakukan oleh penulis lakukan, mulai dari instalasi, keberhasilan dan pengujian dari ELK Stack sebagai dashboard monitoring dan analisa serangan aplikasi web. Tahapan ini juga akan menghasilkan suatu output yang nantinya akan digunakan untuk kesimpulan dari laporan tugas akhir penulis.

#### **5.1 Impementasi System**

Pada tahapan implementasi penulis akan membahas detail mengenai pengimplementasian ELK Stack sebagai Dashboard monitoring dan analisa serangan aplikasi web, dari persiapan hingga instalasi serta konfigurasi yang dibutuhkan.



*Gambar 5. 1* Development Program

#### **5.2 Persiapan**

Pada tahapan ini, penulis akan membuat persiapan kebutuhan dan dependensi yang nantinya akan diperlukan untuk instalasi dari snort dan ELK Stack. Persiapannya mulai dari upgrade repo OS yang digunakan yaitu OS Ubuntu 20.04, Proses pengecekan Java version karena apabila versionnya masih dibawah 11 atau bahkan belum terpasang Java, maka diperlukan untuk melakukan proses instalasi Java Version 11 agar salah satu komponen dari ELK Stack dapat berjalan optimal, serta disusul dengan proses instalasi komponenlainnya.

### 5.3 Update Repositori Ubuntu

Sebelum ke tahap penginstalan sistem ada perlunya untuk memastikan repo pada OS yang digunakan pada penelitian ini sudah pada version latest. Update repo Ubuntu pada persiapan ini bertujuan agar nantinya aplikasi atau sistem yang dijalannya terhindar dari corrupt sistem, berikut command untuk melakukan update repo ubuntu

```
$ sudo apt-get update && apt-get upgrade -y
```

### 5.4 Instalasi Perangkat IDS

Tahap pertama yang perlu dilakukan pada instalasi sistem ini adalah menginstalasi snort version 3. Proses menginstalasi snort versi 3 ini tentunya berbeda dengan versi sebelumnya dikarenakan banyak perubahan juga dalam sistem dari snort yang tentunya juga jadi berbeda dari cara penggunaannya.

#### 5.4.1 Instalasi Dependensi Snort Tzdata

Pada tahap ini tzdata diperlukan untuk sistem operasi serta sistem yang digunakan untuk menyesuaikan waktu dengan waktu yang digunakan oleh penulis saat ini, berikut command yang harus dijalankan

```
$ sudo apt install tzdata  
$ sudo dpkg-configure tzdata
```

#### 5.4.2 Instalasi dan Konfigurasi Snort

Tahapan ini yaitu tahapan untuk melakukan instalasi dan konfigurasi snort hingga snort dapat bekerja sebagai IDS pada penelitian ini berikut Langkah – Langkah yang harus dilakukan

Proses pembuatan directory untuk menyimpan file sistem snort

```
$ mkdir snort-source-file
```

Setelah terbuat, lalu masuk kedalam directory tersebut untuk melakukan proses selanjutnya, berikut commandnya

```
$ cd snort-source-file
```

Tahapan selanjutnya yaitu proses menginstalasi prasyarat yang dibutuhkan snort, tentunya prasyarat ini untuk menunjang micro aplikasi yang terdapat pada snort, berikut commandnya

```
$ sudo apt-get install -y build-essential autotools-dev  
libdumbnet-dev liblua5.1-dev libpcap-dev zlib1g-dev  
pkg-config libhwloc-dev cmake liblzma-dev openssl  
libssl-dev cputest libsqlite3-dev libtool uuid-dev git  
autoconf bison flex libcmocka-dev libnetfilter-queue-dev  
libunwind-dev libmnl-dev ethtool libjemalloc-dev
```

Tahapan selanjutnya yaitu proses instalasi snort 3 nya, sebelumnya pastikan sudah masuk didalam directory yang sebelumnya dibuat.

```
$ wget  
https://github.com/snort3/snort3/archive/refs/tags/3.1.  
18.0.tar.gz -O snort3-3.1.18.0.tar.gz  
$ tar -xzf snort3-3.1.18.0.tar.gz  
$ cd snort3-3.1.18.0.tar.gz
```

Lalu untuk proses instalasi adalah sebagai berikut

```
$ ./configure_cmake.sh --prefix=/usr/local --enable-  
tcmalloc --enable-jemalloc  
$ cd build  
$ make  
$ sudo make install
```

setelah itu untuk mengecek apakah snort sudah terpasang lalu jalankan command berikut

```
$ /usr/local/bin/snort -v
root@labs:~# /usr/local/bin/snort -v
-----
o")~  Snort++ 3.1.49.0
-----
Network Policy : policy id 0 :
-----
Inspection Policy : policy id 0 :
-----
pcap DAQ configured to passive.
-----
host_cache
  memcap: 8388608 bytes
-----
Snort successfully validated the configuration (with 0 warnings).
o")~  Snort exiting
```

Gambar 5. 2 Status snort running

#### 5.4.2.1 Konfigurasi Snort

Pada tahapan ini penulis akan mendetailkan hal hal apa saja yang harus dilakukan konfigurasi pada snort agar perangkat IDS ini dapat bekerja sesuai dengan rancangan yang sebelumnya dibuat.

Tahapan pertama yang itu melakukan konfigurasi pada file konfigurasi snort yang bernama snort.lua, untuk file konfigurasi ini berbeda dengan snort versi 2. File snort.lua ini berada pada directory /usr/local/etc/snort. Oleh karena itu agar memudahkan penulis, sebelum merubah konfigurasi alangkah baik nya perlu masuk kedalam directory tersebut dengan command

```
$ cd /usr/local/etc/snort
```

setelah masuk kedalam directory tersebut lalu masuk kedalam file dengan menggunakan text editor, disini penulis akan menggunakan VI sebagai text editornya

```
$ vi snort.lua
```

```
6. [AWS] ELK - New Server x 7. [AWS] ELK - New Server x +
-- 1. configure defaults
-----

-- HOME_NET and EXTERNAL_NET must be set now
-- setup the network addresses you are protecting
HOME_NET = '54.179.12.27'

-- set up the external network addresses.
-- (leave as "any" in most situations)
EXTERNAL_NET = 'any' EXTERNAL_NET = '!$HOME_NET'

include 'snort_defaults.lua'
--include 'file_magic.lua'
```

Gambar 5. 3 Snort network configure

```
6. [AWS] ELK - New Server x 7. [AWS] ELK - New Server x +
-- 7. configure outputs
-----

-- event logging
-- you can enable with defaults from the command line with -A <alert_type>
-- uncomment below to set non-default configs
--alert_csv = { }
--alert_fast = { }
--alert_full = { }
--alert_sfsocket = { }
--alert_syslog = { }
--unified2 = { }
alert_json = { file = true, limit = 100, fields = 'timestamp sid src_addr src_port msg dst_addr dst_port ' }

-- packet logging
-- you can enable with defaults from the command line with -L <log_type>
--log_codecs = { }
--log_hex = { }
--log_pcap = { }

-- additional logs
--packet_capture = { }
--file_log = { }
```

Gambar 5. 4 Snort alert configure

berikut penjelasan terkait hal apa saja yang ada didalam konfigurasi network dan alert yang nantinya akan digunakan dalam peneliatan ini.

- a. **HOME\_NET** merupakan variabel untuk mentukan IP Home yang nantinya akan didetect, home net juga bisa dibilang merupakan jaringan yang aman atau datangnya dari internal. Namun disini penulis akan menggunakan IP dari server aktivitas yang akan dilakukan proses pemantauan
- b. **EXTERNAL\_NET** merupakan variable untuk menentukan IP External yang dianggap tidak aman atau IP yang dicuriagi, namun pada kali atau alangkah baiknya untuk dapat mendeteksi banyak sumber serangnya maka hanya akan diisi “any” yang artinya semua dari sumber serangan akan di deteksi

- c. **Alert JSON** merupakan variable yang digunakan untuk mengumpulkan seluruh alert snort menjadi satu file berformat txt dan bersifat JSON. Didalam variable ini nantinya ada beberapa konfigurasi lagi untuk dapat mengaktifkan dan mendapatkan kalimat sesuai dengan yang dibutuhkan.
- d. **File = True** merupakan syntac untuk membuat file JSON
- e. **Limit = 100** merupakan syntac untuk mebuat limit maksimum ini yaitu sebesar 100 Mb
- f. **Fields** merupakan syntac untuk menentukan isi dari file alert ini
- g. **Timestamp** digunakan untuk mencatat waktu kejadian
- h. **Sid** digunakan untuk mencatat id spesifik dari jenis serangan
- i. **Src\_addr** digunakan untuk mencatat sumber serangan berasal dari mana
- j. **Src\_ap** fungsinya juga sama dengan src\_addr
- k. **Src\_port** digunakan untuk mencatat port yang digunakan oleh penyerang
- l. **Msg** digunakan untuk mencatat pesan dari serangan, contohnya “*SQL Injection Attempt*”
- m. **Dst\_addr** digunakan untuk mencatat tujuan yang diserang
- n. **Dst\_ap** fungsinya juga sama dengan dst\_addr
- o. **Dst\_port** digunakan untuk mencatat port mana yang terserang

#### 5.4.2.2 Konfigurasi Rule Snort

Untuk dapat mencatat serangan apa saja yang nantinya akan didetect maka akan dibutuhkan rule. Rule sendiri teradapat pada luar directori snort.lua, rule terdapat pada directori sebelum snort yaitu /usr/local/etc/rules dan untuk file rule tersebut yaitu local.rule. Berikut rule yang akan digunakan :

Tabel 5. 1 Tabel rule snort

No	Jenis serangan	Rule snort
1	<i>SQL Injection</i>	- alert tcp any any -> any 80 (msg:" <i>SQL INJECTION ATTEMP</i> "; content: "%270%27"; sid:1000001; rev: 1;)

		- alert tcp any any -> any 80 (msg:" <i>SQL INJECTION UNION ATTEMP</i> "; content:"UNION"; pcr:"^w*(\\ \\%28)+[^]*[^%29]*(\\ \\%29)+/i"; sid:1000002; rev: 1;)
2	DDOS Attack	- alert tcp any any -> any 80 (msg:"DDOS ATTACK ATTEMP"; flags: S+; detection_filter: track by_src, count 50, seconds 30; sid:1000003; rev: 1;)
3	<i>Cross Site Scripting</i>	- alert tcp any any -> any any (msg: " <i>XSS ATTACK ATTEMP</i> "; content:"<script>"; content:"</script>"; sid:1000004; rev: 1;)

Rule yang akan digunakan terdiri dari rule untuk mendeteksi serangan *SQL Injection*, Ddos Attack, dan *Cross Site Scripting* (XSS).

#### 5.4.2.3 Menjalankan Snort Dengan Menggunakan Rule

Tahapan ini akan menjelaskan bagaimana cara memanggil rule snort dan menjalankan snort nya agar dapat bekerja sesuai dengan fungsinya yaitu sebagai perangkat IDS, berikut command yang harus dijalankan

```
$ sudo snort -c /usr/local/etc/snort/snort.lua -R
/usr/local/etc/rules/local.rules -s 65535 -k none -l
/var/log/snort -i eth0 -m 0x1b
```

```

search engine
  instances: 2
  patterns: 3
  pattern chars: 23
  num states: 23
  num match states: 3
  memory scale: KB
  total memory: 2.91113
  pattern memory: 0.137695
  match list memory: 0.273438
  transition memory: 2.25
-----
pcap DAQ configured to passive.
Commencing packet processing
++ [0] eth0
02/27-01:21:52.745812 [**] [1:1000004:1] "XSS ATTACK ATTEMP" [**] [Priority: 0] {TCP} 172.31.13.47:80 → 180.252.93.108:63157
02/27-01:21:55.147256 [**] [1:1000004:1] "XSS ATTACK ATTEMP" [**] [Priority: 0] {TCP} 172.31.13.47:80 → 180.252.93.108:63158
02/27-01:22:19.844594 [**] [1:1000001:1] "SQL INJECTION ATTEMP" [**] [Priority: 0] {TCP} 180.252.93.108:63159 → 172.31.13.47:80
02/27-01:22:28.081117 [**] [1:1000001:1] "SQL INJECTION ATTEMP" [**] [Priority: 0] {TCP} 180.252.93.108:63161 → 172.31.13.47:80
02/27-01:22:33.163531 [**] [1:1000001:1] "SQL INJECTION ATTEMP" [**] [Priority: 0] {TCP} 180.252.93.108:63162 → 172.31.13.47:80
02/27-01:22:33.167492 [**] [1:1000004:1] "XSS ATTACK ATTEMP" [**] [Priority: 0] {TCP} 172.31.13.47:80 → 180.252.93.108:63162
02/27-01:22:37.353758 [**] [1:1000004:1] "XSS ATTACK ATTEMP" [**] [Priority: 0] {TCP} 172.31.13.47:80 → 180.252.93.108:63162
02/27-01:22:56.005078 [**] [1:1000002:1] "SQL INJECTION UNION ATTEMP" [**] [Priority: 0] {TCP} 180.252.93.108:63164 → 172.31.13.47:80
02/27-01:23:09.423058 [**] [1:1000002:1] "SQL INJECTION UNION ATTEMP" [**] [Priority: 0] {TCP} 180.252.93.108:63165 → 172.31.13.47:80

```

Gambar 5. 5 Snort running with rule

command ini memiliki arti yaitu menjalannya snort dengan library C lalu memanggil configurasinya yang berada pada path /usr/local/etc/snort/snort.lua dan memanggil rule yang berada pada path /usr/local/etc/snort/snort.lua lalu akan melakukan pencatatan log pada file alert\_json.txt yang beradap pada path /var/log/snort dan melakukan scanning pada eth0 sebagai nomor dari socket dari jaringannya. Apabila tidak ada error makan tampilannya akan seperti berikut ini

## 5.5 Instalasi Komponen ELK Stack

Tahapan ini akan menerangkan terkait sistem apa saja yang dibutuhkan ELK Stack agar dapat berjalan dengan baik, ELK Stack memiliki beberapa komponen penting yang harus diinstalasi. Berikut komponen yang akan diinstalasi

### 5.5.1 Instalasi Java

pada proses instalasi dari aplikasi ELK Stack nantinya akan dibutuhkan komponen dari java dengan version 11 agar aplikasi ELK dapat dijalankan, sebelum proses instalasi maka diperlukan untuk mengecek dari version javanya. Berikut command untuk mengecek version java

```
$ java --version
```

apabila muncul perintah untuk proses penginstalasi maka berarti pada OS tersebut belum memiliki java. Berikut command menginstalasi java

```
$ sudo apt install default-jre
```

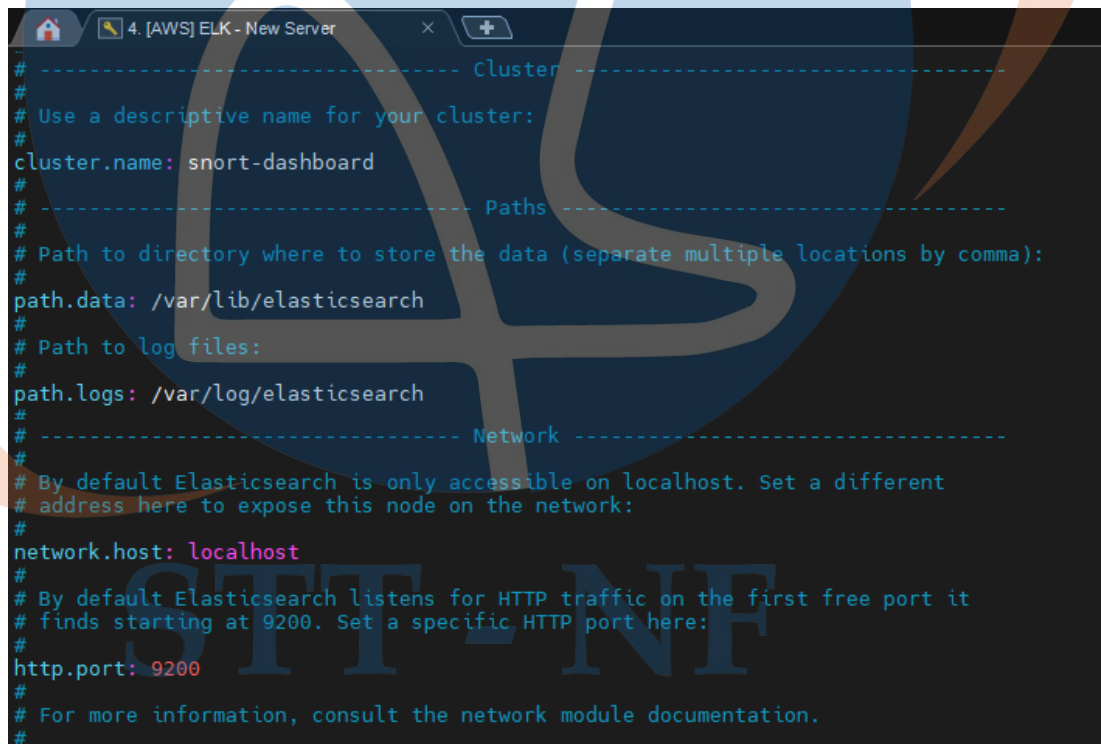


### 5.5.2 Instalasi dan Konfigurasi Elasticsearch

Tahapan ini akan menjelaskan detail proses dari penginstalasi Elasticsearch serta disusul dengan hal apa saja yang harus dikofigurasi. Untuk melakukan proses instalasi elasticsearch ada sebagai berikut

```
$ curl -L -O
https://artifacts.elastic.co/downloads/elasticsearch/el
asticsearch-7.14.2-amd64.deb
$ dpkg -i elasticsearch-7.14.2-amd64.deb
```

agar dapat di remote dari jarak jauh, atau bisa diakses maka perlu perubahan konfigurasi pada file elasticsearch.yml. berikut konfigurasi nya



```
# ----- Cluster -----
#
# Use a descriptive name for your cluster:
#
cluster.name: snort-dashboard
#
# ----- Paths -----
#
# Path to directory where to store the data (separate multiple locations by comma):
#
path.data: /var/lib/elasticsearch
#
# Path to log files:
#
path.logs: /var/log/elasticsearch
#
# ----- Network -----
#
# By default Elasticsearch is only accessible on localhost. Set a different
# address here to expose this node on the network:
#
network.host: localhost
#
# By default Elasticsearch listens for HTTP traffic on the first free port it
# finds starting at 9200. Set a specific HTTP port here:
#
http.port: 9200
#
# For more information, consult the network module documentation.
#
```

Gambar 5.6 Elasticsearch Configuratio

IP yang ditetapkan pada network.host diambil dari lokal dari server untuk menjaga lalu lintas jaringan tetap aman tanpa ada *intecept* dar pihak asing atau pihak yang tidak dikenali

### *Running Service Elasticsearch*

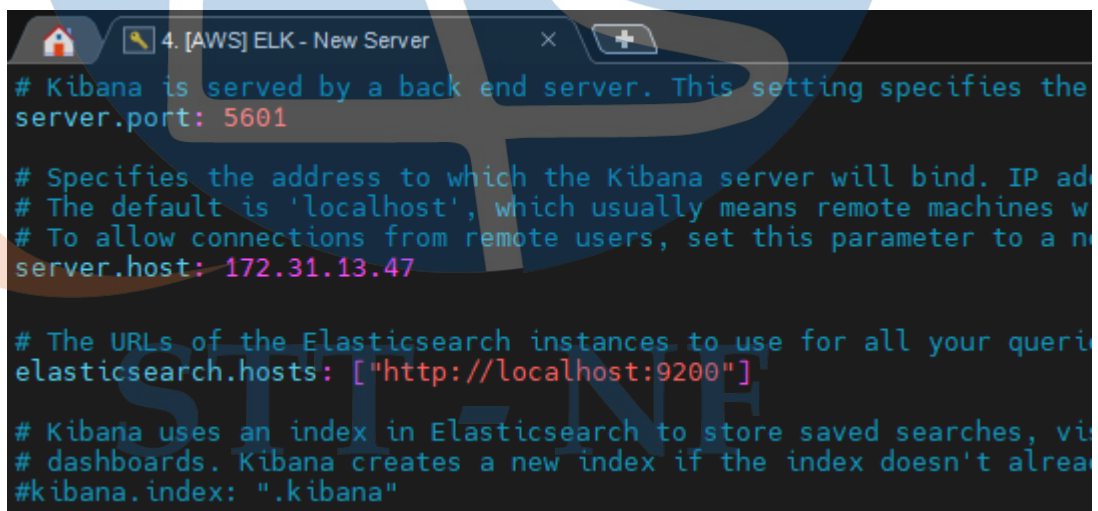
```
$ systemctl enable elasticsearch
$ systemctl start elasticsearch
```

### **5.5.3 Instalasi dan Konfigurasi Kibana**

Tahapan ini akan menjelaskan detail dari proses instalasi dan proses konfigurasi kibana agar dapat menjadi dashboard monitoring. Berikut cara melakukan instalasi kibana dengan menjalankan command berikut ini

```
$ curl -L -O
https://artifacts.elastic.co/downloads/kibana/kibana-
7.14.2-amd64.deb
$ dpkg -i kibana-7.14.2-amd64.deb
```

setelah proses ini selesai maka perlu dilakukan konfigurasi pada file kibana.yml berikut untuk configurasinya

A screenshot of a terminal window titled "4. [AWS] ELK - New Server". The terminal displays the configuration for kibana.yml. The visible text includes: "# Kibana is served by a back end server. This setting specifies the server.port: 5601", "# Specifies the address to which the Kibana server will bind. IP address. The default is 'localhost', which usually means remote machines will not be able to connect to this service. To allow connections from remote users, set this parameter to a non-loopback address. server.host: 172.31.13.47", "# The URLs of the Elasticsearch instances to use for all your queries. elasticsearch.hosts: [\"http://localhost:9200\"]", and "# Kibana uses an index in Elasticsearch to store saved searches, visualizations and dashboards. Kibana creates a new index if the index doesn't already exist. #kibana.index: \".kibana\"".

```
# Kibana is served by a back end server. This setting specifies the
server.port: 5601

# Specifies the address to which the Kibana server will bind. IP address.
# The default is 'localhost', which usually means remote machines will
# not be able to connect to this service. To allow connections from remote
# users, set this parameter to a non-loopback address.
server.host: 172.31.13.47

# The URLs of the Elasticsearch instances to use for all your queries.
elasticsearch.hosts: [\"http://localhost:9200\"]

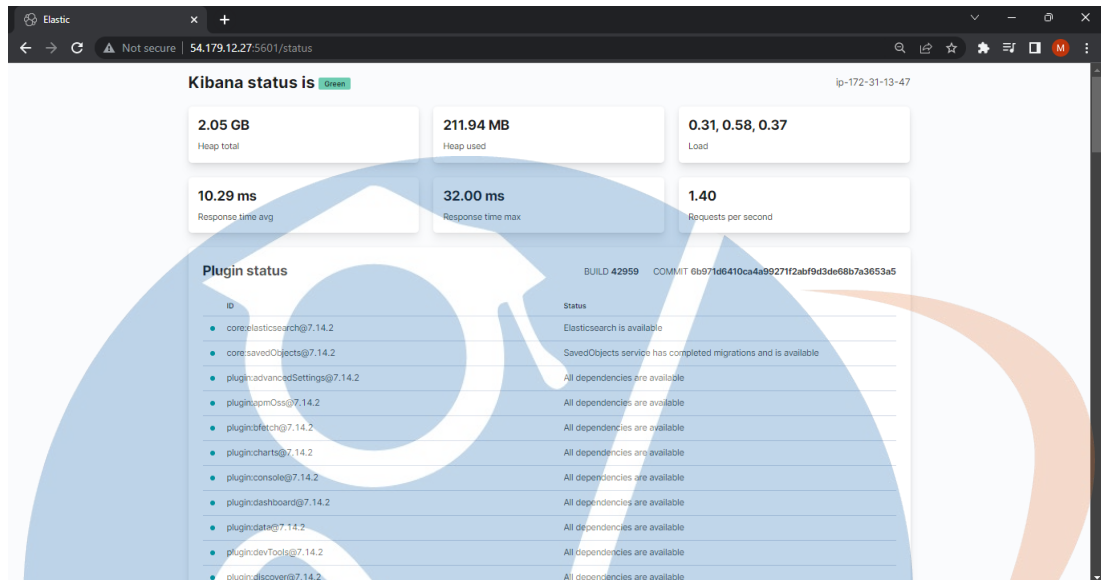
# Kibana uses an index in Elasticsearch to store saved searches, visualizations
# and dashboards. Kibana creates a new index if the index doesn't already
# exist.
#kibana.index: \".kibana\"
```

*Gambar 5. 7. Kibana Configuration*

### *Running Service Kibana*

```
$ systemctl enable kibana
$ systemctl start kibana
```

Setelah berhasil running maka perlu dites pada halaman browser dengan memasukan ip public dan spesifik port kibana yang sebelumnya sudah ditentukan



Gambar 5. 8 Kibana Dashboard Status

#### 5.5.4 Instalasi dan Konfigurasi Logstash

Tahapan ini akan menjelaskan detail dari proses instalasi dan proses konfigurasi logstash agar dapat menjadi sistem yang dapat melakukan manajemen log yang diterima dari snort lalu di kirimkan ke elasticsearch. Berikut cara melakukan instalasi logstash dengan menjalankan command berikut ini

```
$ curl -L -O  
https://artifacts.elastic.co/downloads/logstash/logstash-7.14.2-amd64.deb  
$ dpkg -i logstash-7.14.2-amd64.deb
```

Setelah proses ini selesai maka perlu dilakukan konfigurasi logstash dengan melakukan perubahan pada file logstash.yml

```
# ----- HTTP API Settings -----
# Define settings related to the HTTP API here.
#
# The HTTP API is enabled by default. It can be disabled, but features that rely
# on it will not work as intended.
# http.enabled: true
#
# By default, the HTTP API is bound to only the host's local loopback interface,
# ensuring that it is not accessible to the rest of the network. Because the API
# includes neither authentication nor authorization and has not been hardened or
# tested for use as a publicly-reachable API, binding to publicly accessible IPs
# should be avoided where possible.
#
http.host: 172.31.13.47
#
# The HTTP API web server will listen on an available port from the given range.
# Values can be specified as a single port (e.g., '9600'), or an inclusive range
# of ports (e.g., '9600-9700').
#
http.port: 5044
#
```

Gambar 5. 9 Logstash Configuration

selanjutnya membuat file konfigurasi dengan format .conf. disini penulis akan membuat file .conf tersebut dengan cara masuk kedalam path /etc/logstash/conf.d/ lalu mencalankan perintah

```
$ touch snort.conf
```

Lalu file tersebut diisi dengan konfigurasi sebagai berikut

```
input {
  file {
    path => "/var/log/snort/alert_json.txt"
    start_position => "beginning"
    sincedb_path => "/dev/null"
  }
}

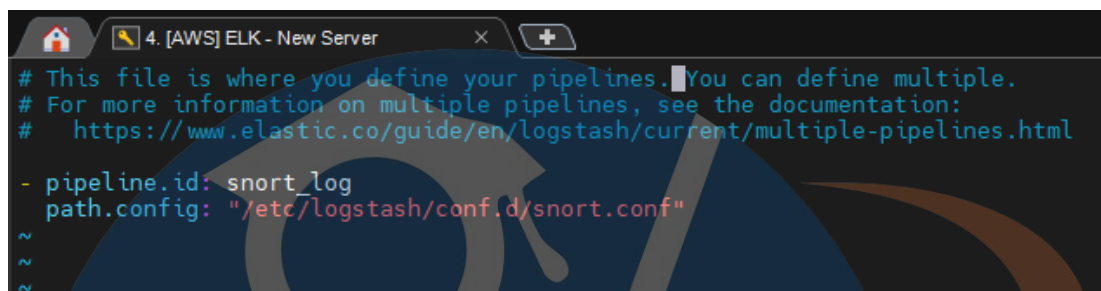
filter {
  json {
    source => "message"
  }
  mutate {
    convert => {
      "pkt_num" => "integer"
      "pkt_len" => "integer"
      "src_port" => "integer"
      "dst_port" => "integer"
      "priority" => "integer"
    }
    gsub => ["timestamp", "\d{3}$", ""]
  }
  date {
    match => [ "timestamp", "yy/MM/dd-HH:mm:ss.SSS" ]
  }
  geoip { source => "src_addr" }
}

output {
  elasticsearch {
    hosts => ["localhost:9200"]
    index => snort_log
  }
  stdout { }
}
"/etc/logstash/conf.d/snort.conf" 37L, 676C
```

Gambar 5. 10 Snort Logstash Configuration

Penulis juga akan melakukan konfigurasi pada file pipeline.yml agar sistem pipeline dapat berfungsi dengan baik dan bekerja sesuai dengan konfigurasi yang dibutuhkan. Untuk file pipeline terletak pada path /etc/logstash/pipeline.yml.

Berikut untuk konfigurasinya



```
# This file is where you define your pipelines. You can define multiple.
# For more information on multiple pipelines, see the documentation:
# https://www.elastic.co/guide/en/logstash/current/multiple-pipelines.html

- pipeline.id: snort_log
  path.config: "/etc/logstash/conf.d/snort.conf"

~
~
~
```

Gambar 5. 11 Logstash Pipeline Configuration

#### Running Service Logstash

```
$ systemctl enable logstash
$ systemctl start logstash
```

### 5.6 Pengujian

Tahapan ini penulis akan memberikan hasil dari proses pengujian yang telah dilakukan setelah implementasi Snort dan ELK Stack. Penulis akan melakukan pengujian Efektifitas dan pengujian terhadap macam – macam log yang terkirim sesuai dengan yang sudah ditentukan.

Proses pengujian ini, konfigurasi sistem dilakukan pada 1 jaringan yang sama namun dengan alamat yang berbeda antara aktivitas yang diserangan dengan aktivitas dashboard monitoring serangan, serangan aplikasi web dilakukan penyerangan terhadap aktivitas simulasi yaitu DVWA atau biasa dikenal dengan aktivitas damn vulnerable web application. Berikut untuk lebih jelasnya :

- Web DVWA  
URL : <http://4.193.135.227/>
- Dashboard monitoring serangan aplikasi web  
URL : <http://4.193.135.227/5601>

Kedua sistem ini dipisahkan sistem keamanan firewall dengan membedakan port yang digunakan.

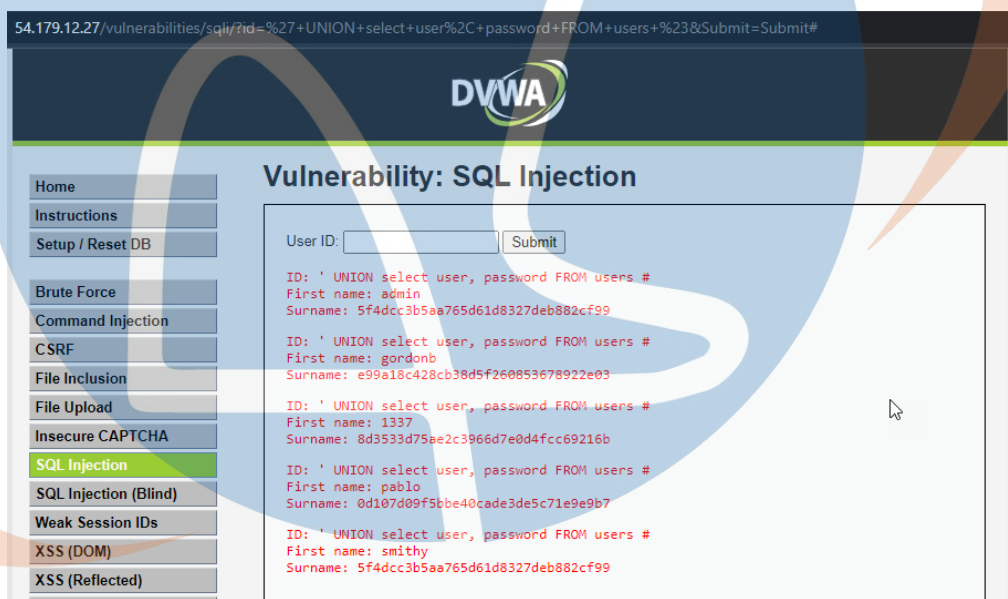
### 5.6.1 Pengujian pendeteksi serangan *SQL Injection*

Pada pengujian pendeteksi serangan *SQL Injection* penguji ingin memastikan apakah serangan yang *SQL Injection* disedang dilakukan dapat terdeteksi oleh perangkat IPS/IDS yang digunakan. Berikut hasil dari pengujian nya.

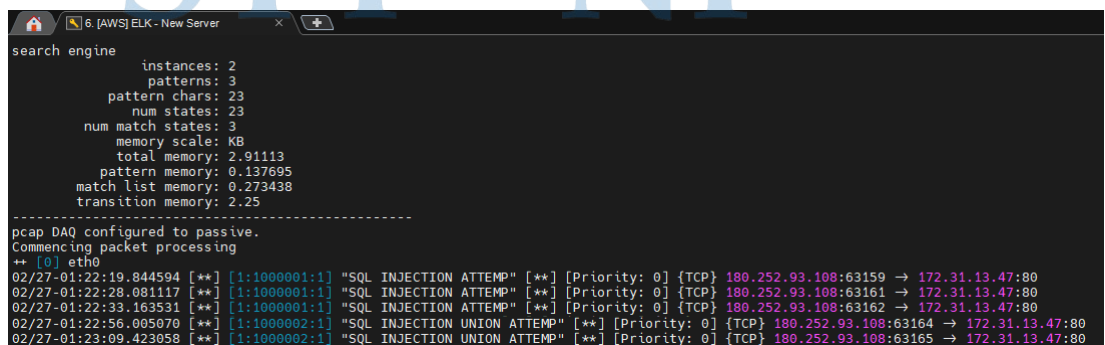
- a. Serangan *SQL Injection* basic dan yang menggunakan UNION untuk memanfaatkan celah keamanannya

Berikut contoh syntax yang digunakan untuk serangan

```
- ' UNION select user, password FROM users #  
- %' or '0'='0
```



Gambar. Attack Result *SQL Injection*



Gambar 5. 12 Snort receiving Logs *SQL Injection*

Tabel 5. 2 Hasil pengujian serangan *SQL Injection*

Pengujian Ke	Jumlah Serangan	Jumlah log yang diterima
Ke 1	50	50
Ke 2	50	50
Ke 3	50	50
Ke 4	50	50

Pengujian dilakukan dengan 4 waktu yang beda dan masing - masing pengujian dilakukan 50 kali serangan *SQL Injection*, hal ini bertujuan untuk memastikan apakah snort dapat mendeteksi secara serangan secara terus menerus atau tidak. Berdasarkan dengan pengujian yang dilakukan bahwa snort dapat mendeteksi serangan ditiap – tiap waktu yang berbeda dan selanjutnya log ini akan diuji juga saat dilakukan shipping log kedalam ELK Stack.

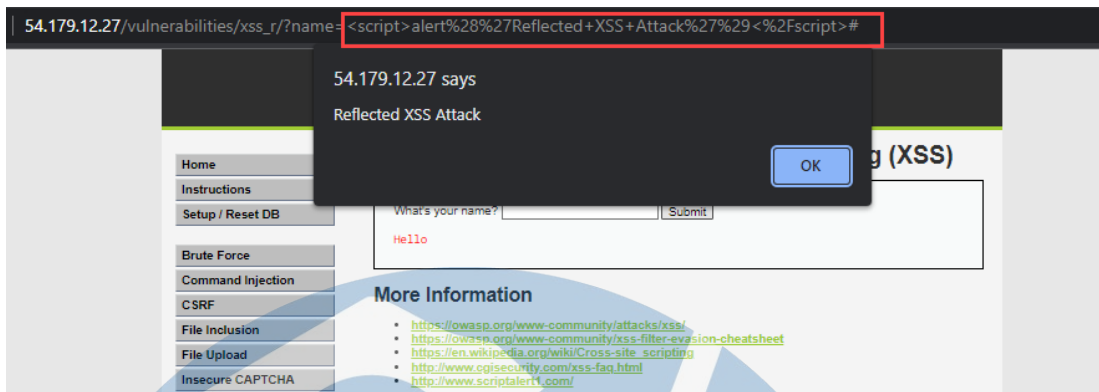
### 5.6.2 Pengujian pendeteksian serangan *Cross Site Scripting*

Pada pengujian pendeteksian serangan *Cross Site Scripting*, penguji ingin memastikan apakah serangan yang sedang dilakukan dapat terdeteksi oleh perangkat IPS/IDS yang digunakan. Berikut hasil dari pengujian nya.

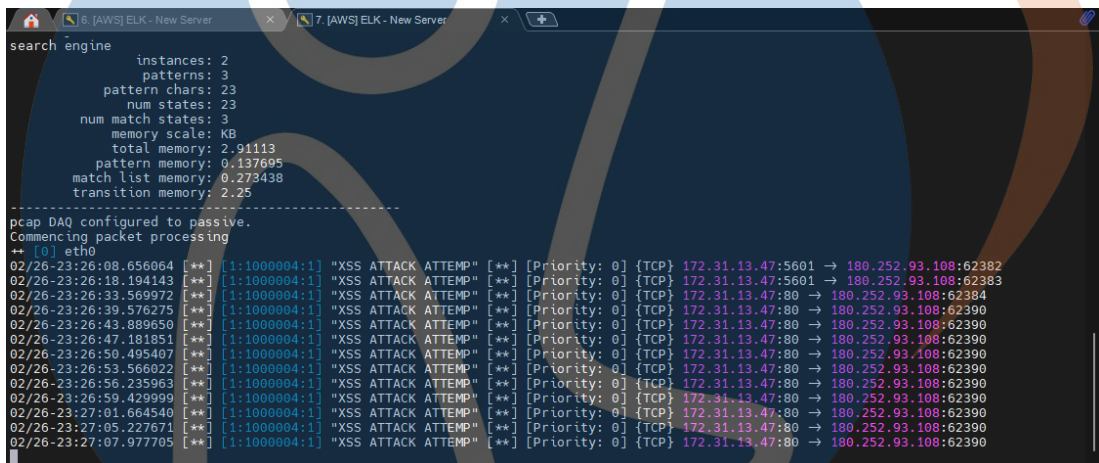
- a. Berikut serangan XSS yang sedang dilakukan, untuk contoh syntax nya adalah sebagai berikut dengan memanfaatkan celah keamanan dari method GET dari aktivitas.

```
- <script>alert('Reflected XSS Attack')</script>
```

Tag script biasanya digunakan untuk melakukan perintah popup dalam html dan pada serangan XSS ini tag script biasanya dimanfaatkan penyerang untuk menyisipkan sebuah link phising atau link berbahaya untuk serangan yang lebih berbahaya.



Gambar 5. 13 Attack Result Cross Site Scripting



Gambar 5. 14 Snort receiving Logs XSS

Tabel 5.3 Hasil pengujian serangan Cross Site Scripting

Pengujian Ke	Jumlah Serangan	Jumlah log yang diterima
Ke 1	50	50
Ke 2	50	50
Ke 3	50	50
Ke 4	50	50

Pengujian dilakukan dengan 4 waktu yang beda dan masing - masing pengujian dilakukan 50 kali serangan *Cross Site Scripting*, hal ini bertujuan untuk memastikan apakah snort dapat mendeteksi secara serangan secara terus menerus atau tidak. Berdasarkan dengan pengujian yang dilakukan bahwa snort dapat mendeteksi

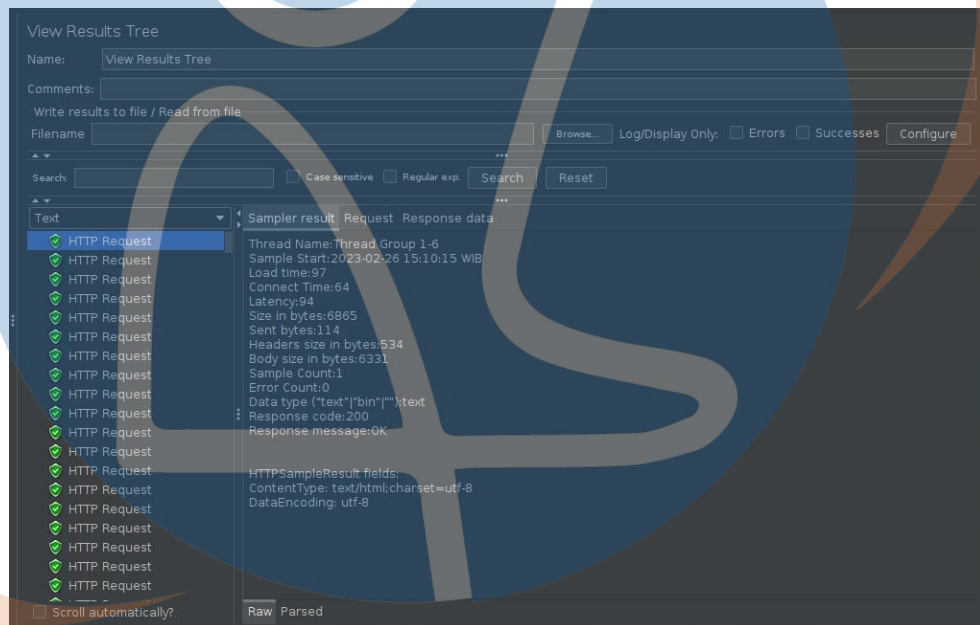


serangan di tiap – tiap waktu yang berbeda dan selanjutnya log ini akan diuji juga saat dilakukan shipping log kedalam ELK Stack.

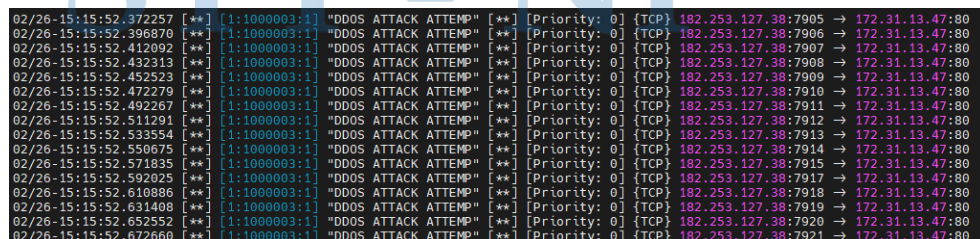
### 5.6.3 Pengujian pendeteksian serangan DDoS Attack

Pada pengujian pendeteksian serangan DDoS penulis ingin memastikan apakah serangan yang sedang dilakukan dapat terdeteksi oleh perangkat IPS/IDS yang digunakan. Berikut hasil dari pengujian nya.

- a. Untuk pengujian penguji menggunakan sebuah tools tambahan yaitu apache jmeter, berikut saat melakukan proses serangan ddos



Gambar 5. 15 Attacking Process DDoS



Gambar 5. 16 Snort receiving Logs DDOS

Tabel 5. 4 Hasil pengujian serangan DDOS

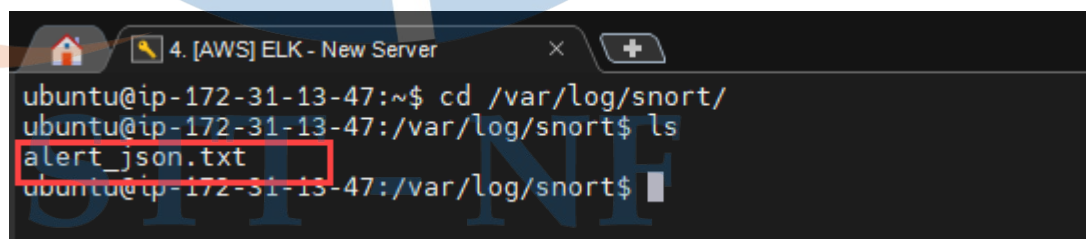
Pengujian Ke	Jumlah Serangan	Jumlah log yang diterima
Ke 1	50	50
Ke 2	50	50
Ke 3	50	50
Ke 4	50	50

Pengujian dilakukan dengan 4 waktu yang beda dan masing - masing pengujian dilakukan 50 kali serangan DDOS, hal ini bertujuan untuk memastikan apakah snort dapat mendeteksi secara serangan secara terus menerus atau tidak. Berdasarkan dengan pengujian yang dilakukan bahwa snort dapat mendeteksi serangan ditiap – tiap waktu yang berbeda dan selanjutnya log ini akan diuji juga saat dilakukan shipping log kedalam ELK Stack.

#### 5.6.4 Pengujian pengiriman log snort kedalam ELK Stack

Pada pengujian ini, penulis akan melakukan pengujian apakah log yang sudah dihasilkan dari snort apakah dapat terkirim kedalam logstash, lalu dari logstash apakah bisa berintegrasi dengan elasticsearch dan kibana. Berikut hasil nya

- a. File log snort yaitu file untuk menyimpan log serangan yang terdeteksi oleh snort dan berlokasi di path /var/log/snort/alert\_json.txt



Gambar 5. 17 Location snort log file

- b. Configurasi snort logstash yang berada pada path /etc/logstash/conf.d/snort.conf

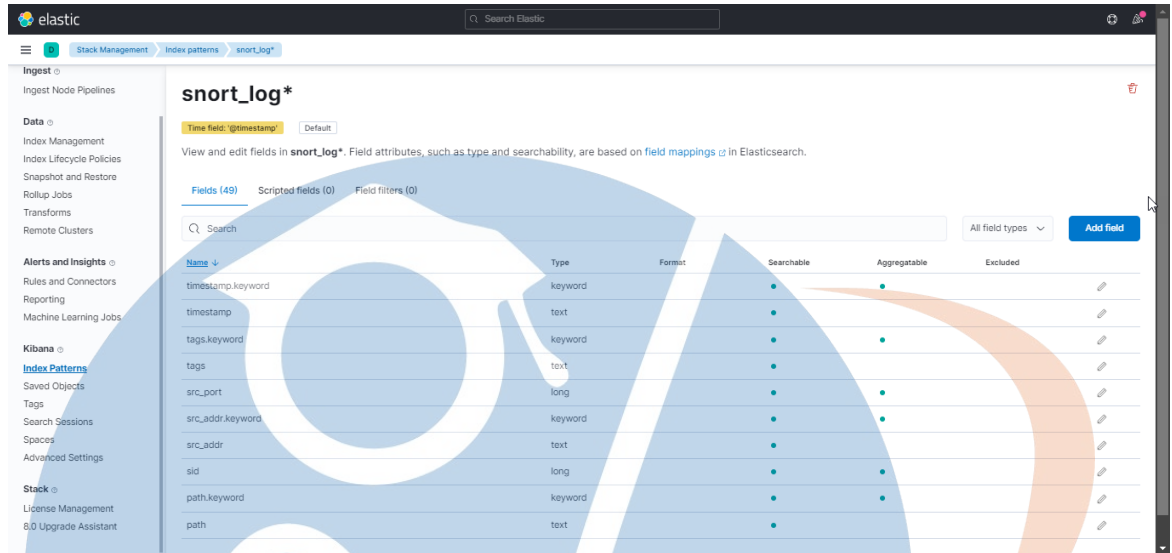
```
input {  
  file {
```

```

    path => "/var/log/snort/alert_json.txt"
    start_position => "beginning"
    since_db_path => "/dev/null"
  }
}
filter {
  json {
    source => "message"
  }
  mutate {
    convert => {
      "pkt_num" => "integer"
      "pkt_len" => "integer"
      "src_port" => "integer"
      "dst_port" => "integer"
      "priority" => "integer"
    }
    gsub => ["timestamp", "\d{3}$", ""]
  }
  date {
    match => [ "timestamp", "yy/MM/dd-
HH:mm:ss.SSS" ]
  }
  geoip { source => "src_addr" }
}
output {
  elasticsearch {
    hosts => ["localhost:9200"]
    index => snort_log
  }
  stdout { }}

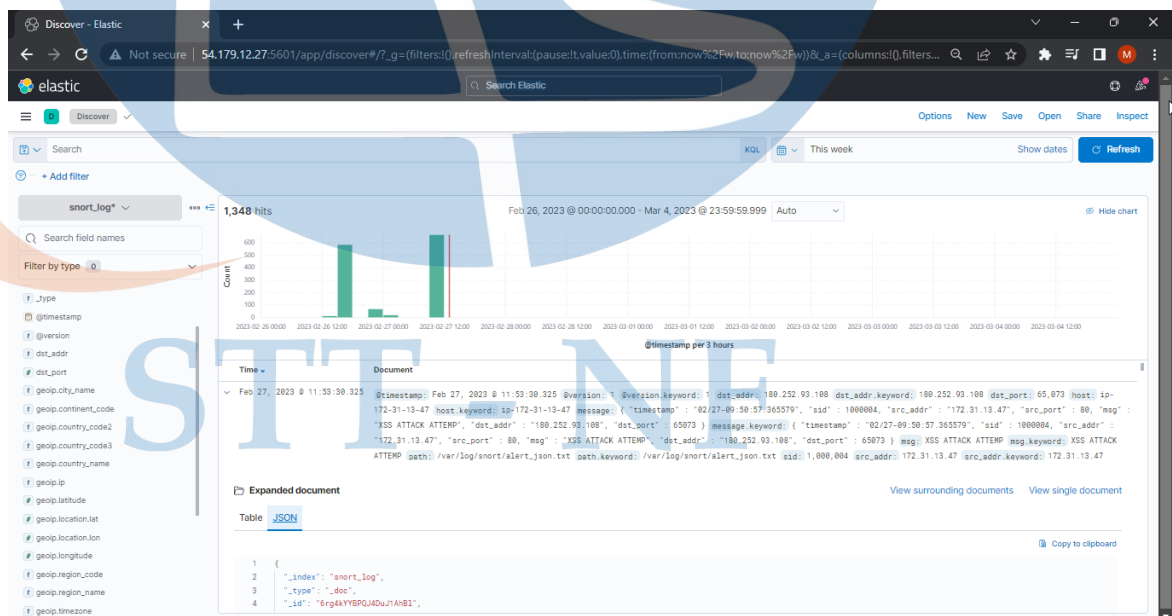
```

c. Hasil log yang diterima logstash dan diubah menjadi index oleh elasticsearch



Gambar 5. 18 Index pattern dalam kibana

d. Index diintergrasikan kedalam kibana, berikut log dalam menu discovery kibana dan belum diubah kedalam dashboard visualisasi



Gambar 5. 19 Discovery menu Kibana

### 5.5.5 Hasil Pengujian

Hasil pengujian didapatkan dari seluruh jumlah pengujian serangan mulai dari *SQL Injection*, *DDOS*, dan *Cross Site Scripting*. Nantinya data ini akan dihitung sesuai dengan rumusan untuk mengukur efektifitas yang sebelumnya telah dibuat . berikut hasil pengujian nya

Tabel 5. 5 Hasil pengujian efektifitas

Uji Ke	Jenis Serangan	Jumlah log dari Snort	Log Tersalin di ELK	Efektifitas
<b>Ke 1</b>	<b><i>SQL Injection</i></b>	<b>50</b>	<b>50</b>	<b>100 %</b>
Ke 2	<i>SQL Injection</i>	50	50	100 %
Ke 3	<i>SQL Injection</i>	50	50	100 %
Ke 4	<i>SQL Injection</i>	50	50	100 %
<b>Ke 1</b>	<b>DDOS Attack</b>	<b>50</b>	<b>50</b>	<b>100%</b>
Ke 2	DDOS Attack	50	50	100%
Ke 3	DDOS Attack	50	50	100%
Ke 4	DDOS Attack	50	50	100%
<b>Ke 1</b>	<b><i>Cross Site Scripting</i></b>	<b>50</b>	<b>50</b>	<b>100%</b>
Ke 2	<i>Cross Site Scripting</i>	50	50	100%
Ke 3	<i>Cross Site Scripting</i>	50	50	100%
Ke 4	<i>Cross Site Scripting</i>	50	50	100%
<b>Rata Rata</b>				<b>100 %</b>

Penjelasan pengujian Log serangan aplikasi web dari snort :

- Pengujian Ke 1 – 4 melakukan serangan aplikasi web menggunakan metode *SQL Injection* dengan setiap pengujian melakukan serangan sebanyak 50 kali, sehingga dari tahap ujian ke 1 hingga ke 4 menghasilkan 200 log, berhasil terbaca 200 log. Maka  $(200/200) \times 100\% = 100\%$
- Pengujian Ke 1 – 4 melakukan serangan aplikasi web menggunakan metode *DDOS Attack* dengan setiap pengujian melakukan serangan sebanyak 50 kali,

sehingga dari tahap ujian ke 1 hingga ke 4 menghasilkan 200 log, berhasil terbaca 200 log. Maka  $(200/200) \times 100\% = 100\%$

- Pengujian Ke 1 – 4 melakukan serangan aplikasi web menggunakan metode *Cross Site Scripting* dengan setiap pengujian melakukan serangan sebanyak 50 kali, sehingga dari tahap ujian ke 1 hingga ke 4 menghasilkan 200 log, berhasil terbaca 200 log. Maka  $(200/200) \times 100\% = 100\%$

Rata – rata efektifitas =  $(100\% + 100\% + 100\% + 100\%)/4 = 100\%$ , sehingga dapat dikatakan hasil dari pengujian Dashboard monitoring dan analisa serangan aplikasi web menggunakan ELK Stack memiliki tingkat efektifitas **100%**

### 5.5.6 Hasil Uji Visualisasi Kibana

Berikut hasil visualisas dari log yang sudah diubah kedalam index lalu dijadikan sebuah visualisasi ke bentuk macam – macam diagram didalam sebuah dashboard monitoring yang kibana.

Tabel 5. 6 Hasil Uji Visualisasi

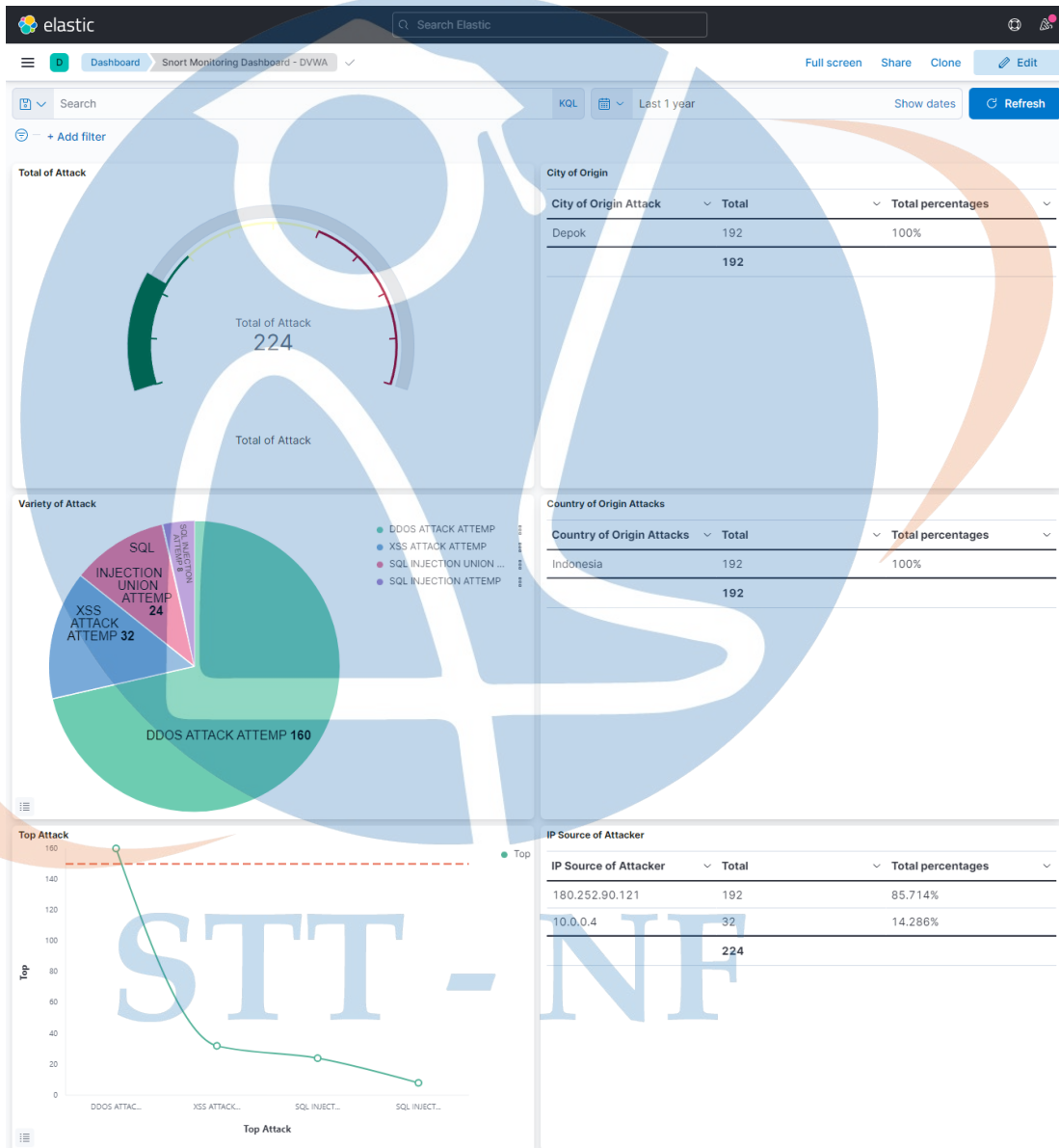
No	Nama Visualisasi	Jenis Visualisasi	Hasil Uji	Presentase
1	Total of Attacks	Gauge	Berhasil Tergambarkan	100 %
2	Variety of Attacks	Pie Chart	Berhasil Tergambarkan	100 %
3	City of Attacker	Table	Berhasil Tergambarkan	100 %
4	Country of Attacker	Table	Berhasil Tergambarkan	100 %
5	Top Attack	Vertical Bar	Berhasil Tergambarkan	100 %
6	Source IP of Attacker	Table	Berhasil Tergambarkan	100 %

No	Nama Visualisasi	Jenis Visualisasi	Hasil Uji	Presentase
7	Filter	Filter Options	Berhasil Tergambarkan	100%

Penjelasan Table Hasil Uji Visualisasi :

1. Hasil pengujian ke 1 dengan menguji visualisasi “Total of Attacks” dalam bentuk diagram “Gauge” dengan mengambil parameter salah satu JSON yaitu “Count” berhasil tergambarkan seluruhnya dari total jumlah parameter tersebut yaitu 100%
2. Hasil pengujian ke 2 dengan menguji visualisasi “Variety of Attacks” dalam bentuk diagram “Pie chart” dengan mengambil parameter salah satu JSON yaitu “msg.keyboard” berhasil tergambarkan seluruhnya dari total jumlah parameter tersebut yaitu 100%
3. Hasil pengujian ke 3 dengan menguji visualisasi “City of Attacker” dalam bentuk diagram “Table” dengan mengambil parameter salah satu JSON yaitu “geoup.city\_name.keyword” berhasil tergambarkan seluruhnya dari total jumlah parameter tersebut yaitu 100%
4. Hasil pengujian ke 4 dengan menguji visualisasi “Country of Attacker” dalam bentuk diagram “Table” dengan mengambil parameter salah satu JSON yaitu “geoup.country\_name.keyword” berhasil tergambarkan seluruhnya dari total jumlah parameter tersebut yaitu 100%
5. Hasil pengujian ke 5 dengan menguji visualisasi “Top Attack” dalam bentuk diagram “Vertical Bar” dengan mengambil parameter salah satu JSON yaitu “msg.keyword” berhasil tergambarkan seluruhnya dari total jumlah parameter tersebut yaitu 100%
6. Hasil pengujian ke 6 dengan menguji visualisasi “Source IP of Attacker” dalam bentuk diagram “Table” dengan mengambil parameter salah satu JSON yaitu “src.addr.keyword” berhasil tergambarkan seluruhnya dari total jumlah parameter tersebut yaitu 100%

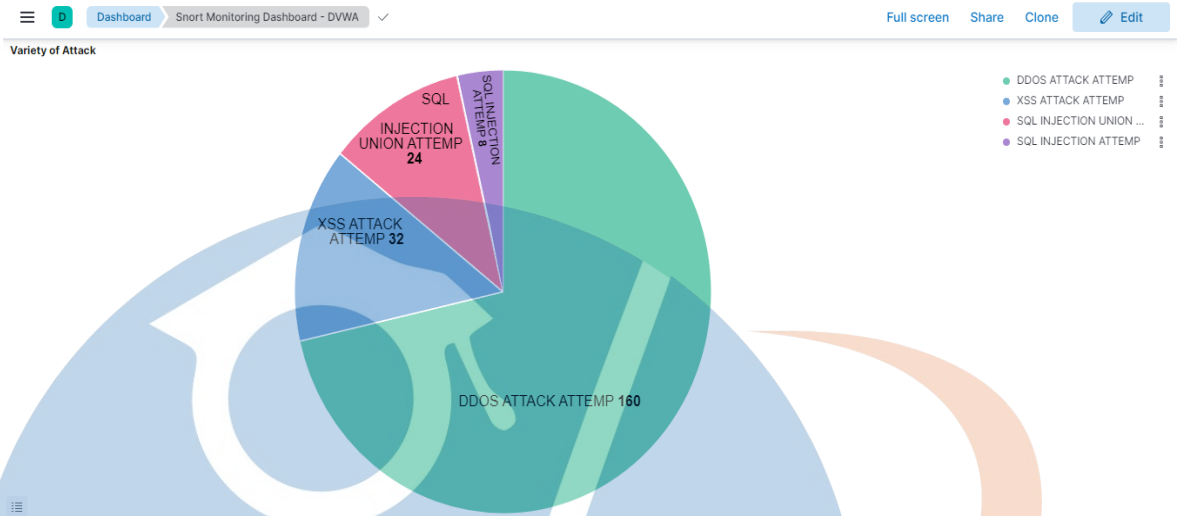
- Hasil pengujian ke 7 dengan memanfaatkan fitur filter didalam kibana untuk menampilkan data dengan waktu sesuai keinginan kapan dan hari apa jumlah serangan itu muncul, pengujian ini melakukan penentuan serta filter tanggal dan waktu dan berhasil menampilkan data sesuai dengan keinginan yaitu 100%



Gambar 5. 20 Dashboard Security Aktivitas Monitoring ELK Stack

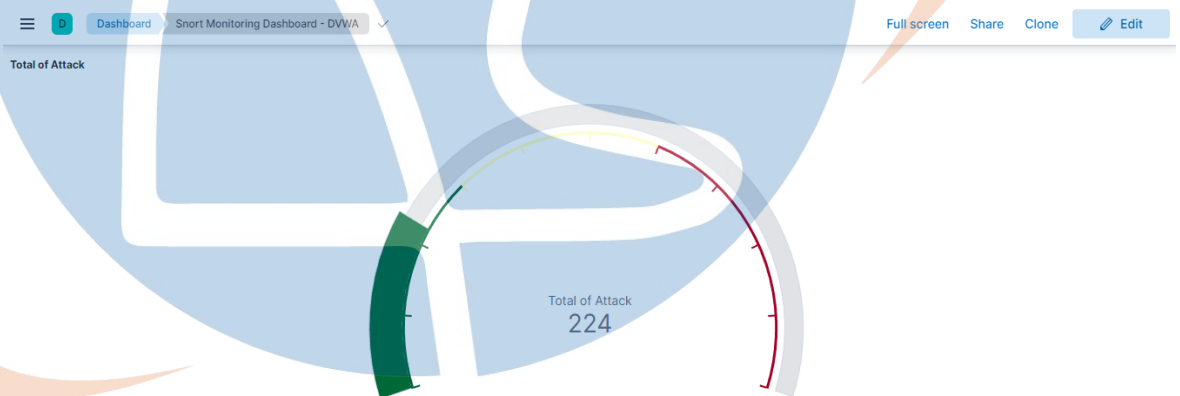


- Hasil visualisasi jenis serangan menggunakan diagram pie



Gambar 5. 21 Variasi Serangan

- Hasil visualisasi total serangan menggunakan visual Gauge



STT - NF  
Gambar 5. 22 Total Serangan

- Hasil visualisasi sumber IP yang melakukan penyerangan menggunakan visual data table

The screenshot shows a dashboard titled 'Editing Snort Monitoring Dashboard - DVWA'. The main content is a table titled 'IP Source of Attacker'. The table has three columns: 'IP Source of Attacker', 'Total', and 'Total percentages'. The data is as follows:

IP Source of Attacker	Total	Total percentages
180.252.90.121	192	85.714%
10.0.0.4	32	14.286%
	224	

Gambar 5. 23 Sumber IP Penyerang

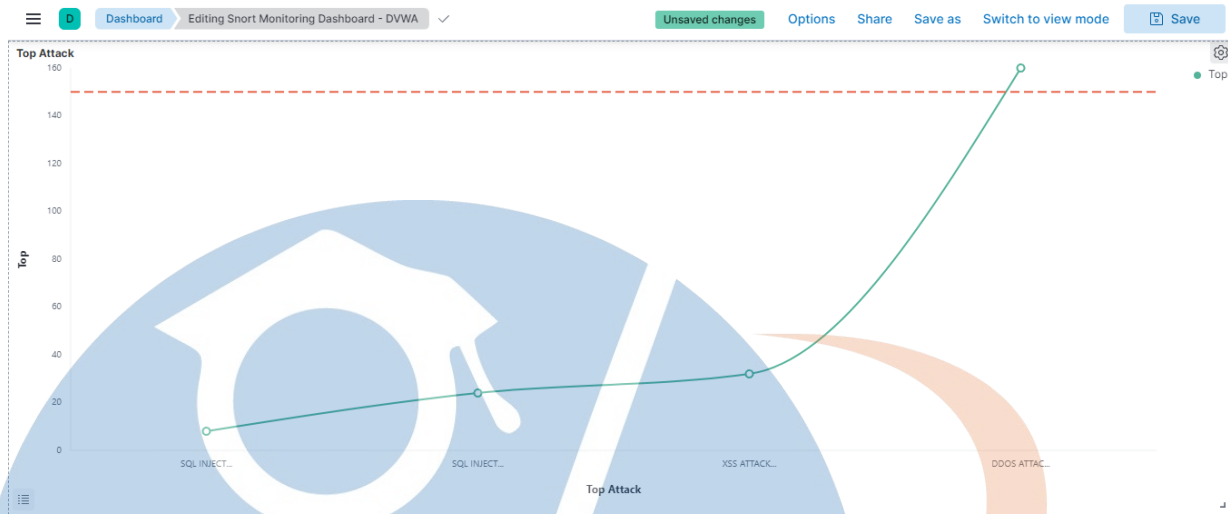
- Hasil Visualisasi negara asal penyerang menggunakan diagram Table

The screenshot shows a dashboard titled 'Editing Snort Monitoring Da...'. The main content is a table titled 'Country of Origin Attacks'. The table has three columns: 'Country of Origin Attacks', 'Total', and 'Total percentages'. The data is as follows:

Country of Origin Attacks	Total	Total percentages
Indonesia	192	100%
	192	

Gambar 5. 24 Negara Penyerang

- Hasil Visualisasi Top serangan menggunakan diagram Vertical bar



Gambar 5. 25 Top Serangan

- Hasil Visualisasi kota asal penyerang menggunakan diagram Table

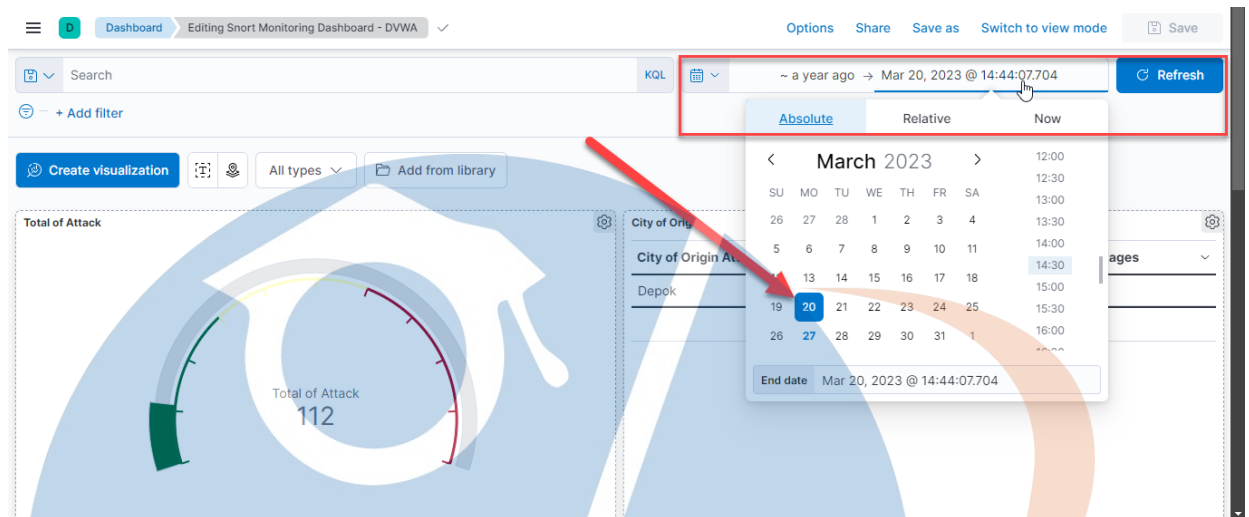
The screenshot shows a dashboard titled 'Snort Monitoring Dashboard - DVWA'. A table titled 'City of Origin' displays the total number of attacks and their percentages by city.

City of Origin Attack	Total	Total percentages
Depok	192	100%
	192	

Gambar 5. 26 Kota Penyerang

STT - NF

- Hasil Visualisasi Filter untuk menampilkan data sesuai dengan waktu yang diinginkan



Gambar 5. 28 Filter Dashboard

### 5.5.7 Hasil Uji Efektivitas Visualisasi

Tahapan ini akan memaparkan dari Uji Efektivitas Visualisasi serangan aplikasi web yang divisualisasikan oleh kibana. Peneliti akan melakukan survey dan pengamatan terhadap beberapa orang yang bekerja khususnya dibidang IT yang nantinya akan mencoba melakukan pengoprasian dashboard monitoring serangan aplikasi web ini, berikut ini merupakan kumpulan data dari hasil pengujiannya :

Tabel 5. 7 Hasil Uji Efektivitas Penggunaan

No	Skenario Uji Efektivitas Penggunaan	SS	S	M	SM
1	Membuka halaman dashboard monitoring	0	0	0	10
2	Menampilkan visualisasi data dari hasil serangan	0	0	0	10
3	Melakukan filter serangan berdasarkan waktu	0	0	2	8
4	Melakukan filter serangan berdasarkan nama serangan	0	0	1	9

Tabel 5. 8 Hasil Uji Efektivitas Kelayakan

No	Skenario Uji Efektivitas Kelayakan	STS	TS	S	SS
1	Tampilan dashboard yang disajikan untuk sebuah aktivitas monitoring nyaman untuk digunakan	0	0	1	9
2	Diagram - diagram pada dashboard tersebut masuk akal dan sudah tepat digunakan	0	0	0	10
3	Fitur filter pada dashboard membantu proses penyaringan data serangan	0	0	1	9
4	Mudah untuk menyimpulkan serangan yang terjadi	0	0	3	7

Tabel 5. 9 Deskripsi dan Bobot nilai kuesioner/survey

Keterangan	Deskripsi	Kriteria	Nilai
SS & STS	Sangat Sulit & Sangat Tidak Setuju	Kesulitan dalam penggunaannya dalam serta harus berkali – kali mempelajarinya dan sangat tidak setuju dengan visulaisasi yang disajikan disarankan untuk mengganti jenis visualisasi	1
S & TS	Sulit & Tidak Setuju	Kesulitan dalam penggunaan pertama tanpa harus mempelajari berulang kali dan tidak setuju terhadap visualisasi serta fitur yang disajikan namun masih bisa ditoleransi tanpa harus mengganti fitur / visualsasi	2

Keterangan	Deskripsi	Kriteria	Nilai
M & S	Mudah & Setuju	Mudah dalam penggunaan pertama namun harus mempelajari beberapa hal dan setuju terhadap visualisasi dan fitur yang sudah disajikan	3
SM & SS	Sangat Mudah & Sangat Setuju	Mudah dalam penggunaan pertama serta dapat mengenal dan mengetahui langsung fitur serta visualisasi yang disajikan dan sangat setuju terhadap fitur dan visualisasi yang telah disajikan	4

Setelah itu akan dilakukan perhitungan presentase terhadap data dari survey yang sudah dilakukan dan dikumpulkan ke 2 skenario yaitu skenario uji efektivitas penggunaan dan uji efektivitas visualisasi, untuk perhitungan maka ditentukan dengan rumus sebagai berikut :

$$Persentase = \frac{\text{Nilai Total}}{\text{Nilai Maksimal}} \times 100\%$$

Nantinya nilai total akan didapatkan dari hasil penjumlahan dari setiap perkalian jumlah responden dengan bobot nilai yang sudah ditentukan. Nilai maksimal merupakan nilai tertinggi yang didapatkan dari total masing masing pertanyaan skenario setelah itu nantinya akan didapatkan dari tabel hasil dari kedua skenario adalah sebagai berikut

Tabel 5. 10 Persentase Hasil Uji Efektivitas Penggunaan

No	Skenario Uji Efektivitas Penggunaan	SS	S	M	SM	Total	Psentase
1	Membuka halaman dashboard monitoring	0	0	0	40	40	100%

No	Skenario Uji Efektivitas Penggunaan	SS	S	M	SM	Total	Psentase
2	Menampilkan visualisasi data dari hasil serangan	0	0	0	40	40	100%
3	Melakukan filter serangan berdasarkan waktu	0	0	6	32	38	95%
4	Melakukan filter serangan berdasarkan nama serangan	0	0	3	36	39	97,5%

Tabel 5. 11 Persentase Hasil Uji Efektivitas Kelayakan

No	Skenario Uji Efektivitas Kelayakan	STS	TS	S	SS	Total	Persentase
1	Tampilan dashboard yang disajikan untuk sebuah aktivitas monitoring nyaman untuk digunakan	0	0	3	36	39	95 %
2	Diagram - diagram pada dashboard tersebut masuk akal dan sudah tepat digunakan	0	0	0	40	40	100 %
3	Fitur filter pada dashboard membantu proses penyaringan data serangan	0	0	3	36	39	95 %
4	Mudah untuk menyimpulkan serangan yang terjadi	0	0	9	28	37	92,5 %

Selanjutnya kedua skenario ini akan ditotalkan dan dicari hasil rata ratanya untuk mengukur sejauh mana efektivitas dari penggunaan dan efektivitas dari kelayakan dashboard monitoring ini dengan perhitungan sebagai berikut

$$\text{Rata - rata} = \frac{100 + 100 + 100 + 97,5 + 95 + 95 + 95 + 92,5}{8}$$

$$\text{Rata - rata} = 96,87 \%$$

Setela itu akan didapatkan tingkat efektivitas yang didapatkan dari perhitungan kedua skenario tersebut berdasarkan dengan tabel berikut :

*Tabel 5. 12 Interpretasi Penilaian Efektivitas*

No	Persentase	Interprestasi
1	0% - 25%	Sangat Tidak Layak
2	26% - 50%	Tidak Layak
3	51% - 75%	Cukup Layak
4	76% - 100%	Layak

Berdasarkan hasil kuisisioner/survey dari skenario yang telah dilakukan, maka didapatkan hasil yaitu sebesar 96,87%. Oleh karena itu dapat disimpulkan bahwa dashboard monitoring serangan aplikasi web menggunakan ELK Stack ini yang datanya didapatkan dari hasil pendeteksian snort terhadap serangan suatu website layak untuk digunakan.