BAB V
IMPLEMENTASI

Pada bab ini penulis akan membahas implementasi yang akan dilakukan dari rancangan yang telah dibuat beserta pengujian dan hasil dari implementasi ELK sebagai alat visualisasi log Elena. Pada tahapan implementasi ini akan di bahas secara mendetail mengenai proses dari instalasi ELK, konfigurasi, visualisasi log dan pengujian log yang nantinya akan menghasilkan kesimpulan dari penelitian sebagai laporan tugas akhir penulis.

5.1 Implementasi Sistem

Pada proses implementasi visualisasi log Elena berbasis ELK ini, penulis membutuhkan beberapa hal yang harus dipersiapkan guna menunjang kemudahan saat menginstall ELK adapun persiapan nya akan dijelaskan dibawah ini.

5.1.1 Persiapan

Pada tahap persiapan ini, peneliti akan mempersiapkan kebutuhan atau alat yang diperlukan untuk instalasi ELK Stack adalah sebagai berikut :

5.1.1.1 Update Repositori Ubuntu

Sebelum melakukan instalasi ELK Stack di komputer lokal di perlukan *update repository* sistem operasi terlebih dahulu. Proses *update repository* dilakukan agar pada saat instalasi sudah lengkap *utility* dan *library* yang dibutukan untuk instalasi server ELK Stack. *Upgrade resporitory* dapat dilakukan dengan mengetikkan perintah di bawah ini :

```
$ sudo apt-get update
```

5.1.1.2 Instalasi Oracle Java JDK

a. Mengunduh Open JDK

Langkah pertama untuk instalasi ELK Stack adalah dengan terlebih dahuluinstalasi Open JDK versi 11. Hal ini karena Elasticsearch, yang merupakan bagian dari ELK Stack (Elasticsearch, Logstash, Kibana), membutuhkan Open JDK sebagai *runtime* Java yang diperlukan untuk menjalankan aplikasi. Berikut cara mengunduh Open JDK dengan mengetikkan perintah berikut :

```
$ sudo add-apt-repository ppa:openjdk-r/ppa
$ sudo apt-get update
```

b. Instalasi Open JDK

Setelah mengunduh Open JDK selajutnya mengetikkan perintah di bawahini untuk instalasi :

```
$ sudo apt-get install open-jdk-11-jdk
$ java -version
```

Berikut *output* yang muncul setelah Open JDK terinstall :

```
openjdk version "11.0.17" 2022-10-18

OpenJDK Runtime Environment (build 11.0.17+8-post-
Ubuntu-1ubuntu220.04)

OpenJDK 64-Bit Server VM (build 11.0.17+8-post-Ubuntu-
1ubuntu220.04, mixed mode, sharing)
```

5.1.1.3 Instalasi dan Konfigurasi Elasticsearch

a. Instalasi Elastic APT *Repository*

Tahapan selanjutnya yaitu menginstall *package* APT Elasticsearch dari repositori elastic.co lalu lakukan *update* dengan mengetikkan perintah dibawah :

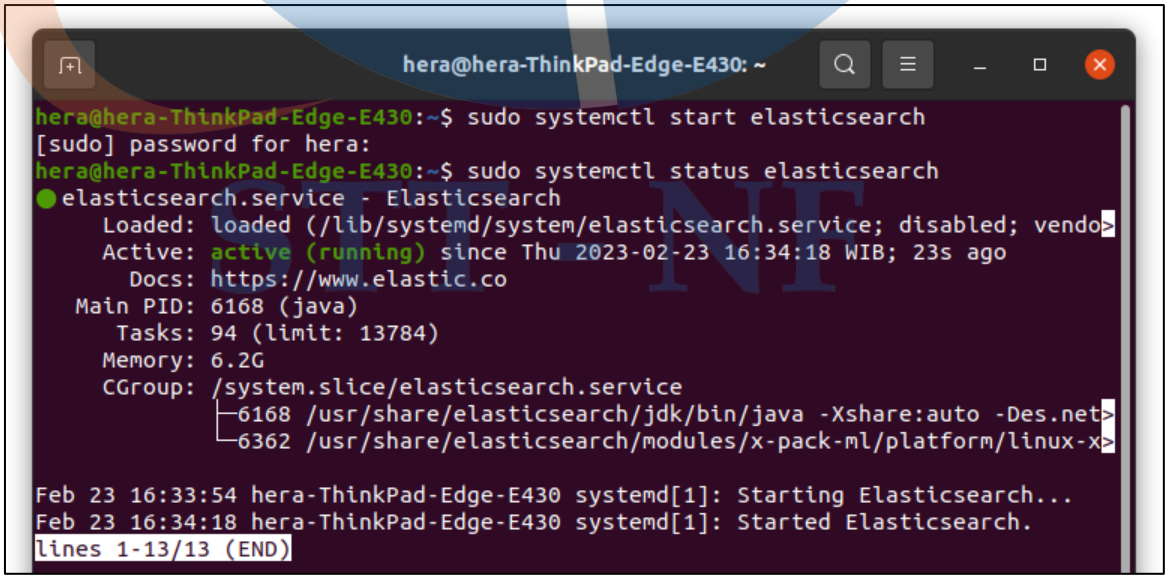
```
$ wget -qO - https://artifacts.elastic.co/GPG-KEY-
elasticsearch | sudo apt-key add -

$ echo" deb https://artifacts.elastic.co/packages/7.x/apt
stablemain" | sudo tee -a /etc/apt/sources.list.d/elastic-
7.x.list

$ sudo apt-get update
```

b. Instalasi Elasticsearch

Tahapan selanjutnya yaitu menginstalasi Elasticsearch dengan mengetikkan perintah dibawah ini :

```
$ sudo apt-get install elasticsearch
```

Setelah menginstall elasticsearch tahapan selanjutnya yaitu menyalakan *service* elasticsearch dengan mengetikkan perintah di bawah ini :



Gambar 5.1 Perintah Menghidupkan Elasticsearch

Setelah *service* di nyalakan, selanjutnya memastikan API elasticsearch sudah muncul dengan mengetikkan perintah :

```
curl localhost:9200
```

Berikut adalah *output* yang akan dikeluarkan dari perintah diatas yang berisi tentang informasi umum seperti versi elasticsearch yang telah diinstal di komputer.

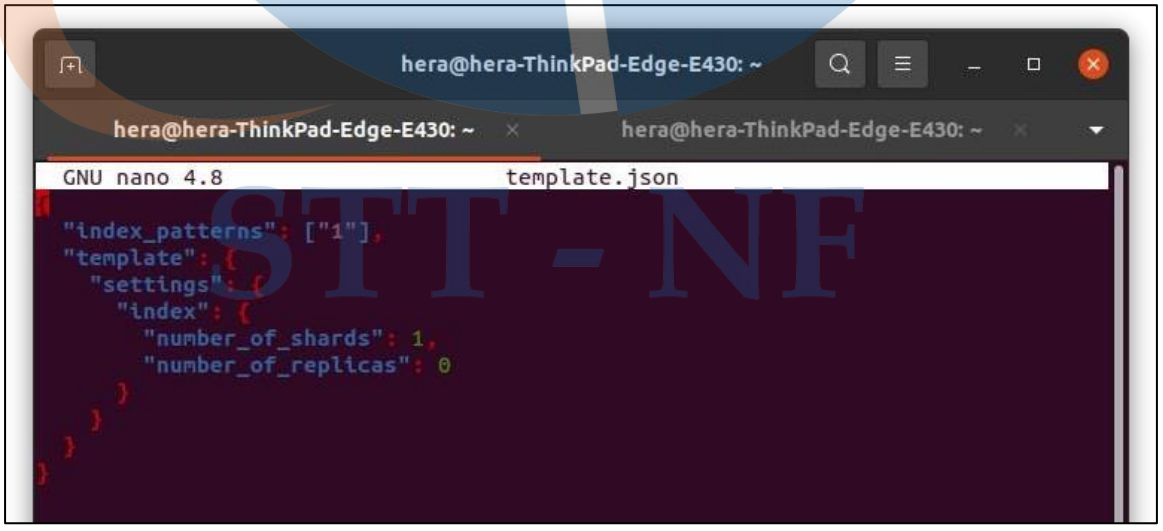
```
{
     "name" : "hera-ThinkPad-Edge-E430", "cluster_name"
     : "elasticsearch", "cluster_uuid" :
     "FegfnlrIS4uRrYtyq-duaA","version" : {
     "number" : "7.17.8", "build_flavor"
     : "default","build_type" : "deb",
     "build_hash" :
"120eabe1c8a0cb2ae87cffc109a5b65d213e9df1",
 "build_date" : "2022-12-02T17:33:09.72707285Z",
     "build_snapshot" : false, "lucene_version" :
     "8.11.1",
     "minimum_wire_compatibility_version" :"6.8.0",
```

c. Konfigurasi Elasticsearch

Tahapan selanjutnya yaitu membuat *file* JSON sementara di *index template* Elasticsearch untuk membuat satu replika pada satu server *index*. Tujuan *template* ini agar Elasticsearch dapat mengubah julmlah pecahan *index* menjadi satu dan jumlah replika menjadi nol. Mengubah *index* dengan mengganti karakter (*) pada *file* sebagai berikut :

```
{
 "index_patterns": ["*"],
 "template": {
   "settings": {
     "index": {
       "number_of_shards": 1,
       "number_of_replicas": 0
     }
   }
 }
}
```

Pada penelitian ini penulis menamai *index* dengan mengganti karakter (*)menjadi (1) pada *file* yang dinamain *template.json* sebagai berikut :



Gambar 5.2 Konfigurasi File Template Json pada Elasticsearch

Membuat *template index* dengan mengetikkan perintah di bawah :

```
$ curl – XPUT – H’Content-type:application/json’
http://localhost:9200/_index_template/defaults  -d
@template.json
```

Output yang harus dikeluarkan dari perintah diatas adalah :

```
{“acknowledged”:true}
```

5.1.1.4 Instalasi dan Konfigurasi Logstash

a. Instalasi Elastic APT *Repository*

Tahapan selanjutnya yaitu menginstall *package* logstash :

```
$ sudo apt-get install logstash
```

b. Konfigurasi Logstash

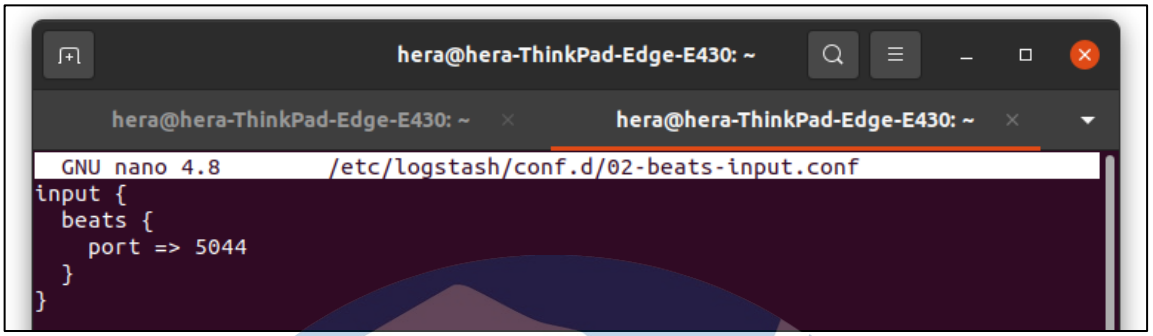
Setelah instalasi logstash, dilanjutkan dengan mengedit *file* konfigurasi yang terdapat pada `02-beats-input.conf` dengan mengetikkan perintah dibawah ini :

```
$ sudo nano /etc/logstash/conf.d/02-beats-input.conf
```

Selanjutnya salin konfigurasi dibawah ini pada *file* `02-beats-input.conf` tersebut dengan tujuan untuk mengkonfigurasi input pada TCP *port* 5044.

```
input { beats
  {
    port => 5044
  }
}
```

Setelah menyalin *file* konfigurasi sepeti dibawah ini selanjutnya simpan.



Gambar 5.3 Konfigurasi File 02-beats-input.conf

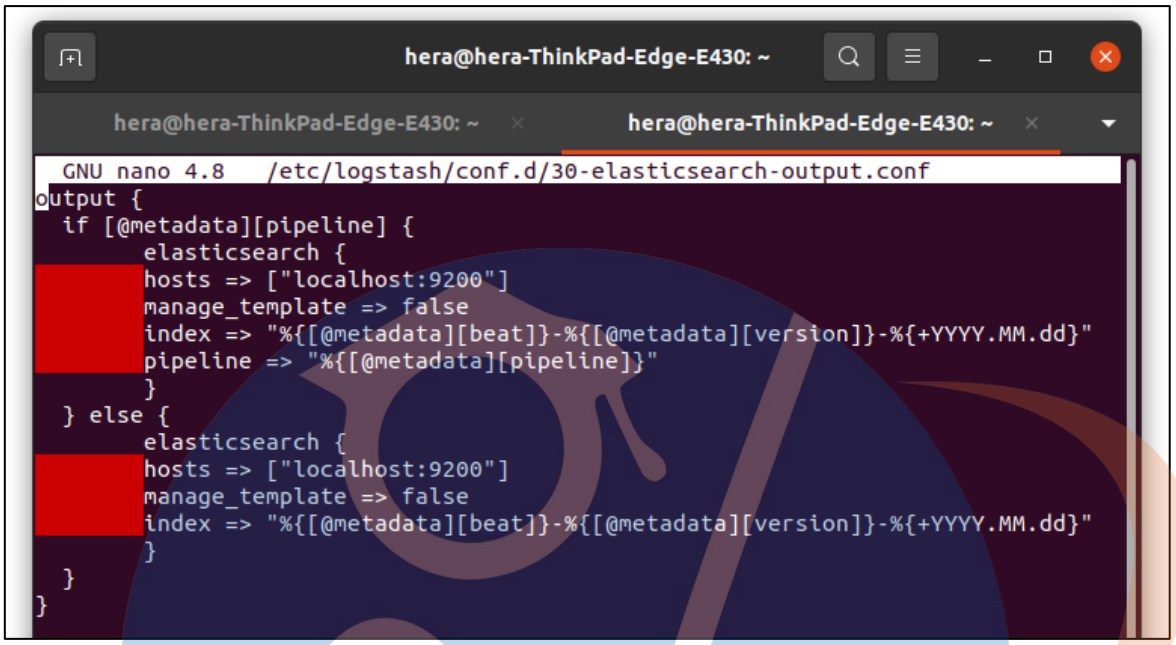
Selanjutnya mengatur *file* konfigurasi yang pada `30 elasticsearch-output.conf` dengan mengetikkan perintah berikut :

```
$ sudo nano /etc/logstash/conf.d/30-elasticsearch-output.conf
```

Selanjutnya salin *file* berikut dan masukkan konfigurasi ini pada *file* `elasticsearch-output.conf` diatas. *File* konfigurasi *output* logsatsh ini berguna untuk menyimpan data di elasticsearch yang berjalan pada `localhost:9200`.

```
output {
  if [@metadata][pipeline] {
     elasticsearch {
     hosts => ["localhost:9200"]

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

Setelah menyalin file konfigurasi sepeti dibawah ini selanjutnya simpan.

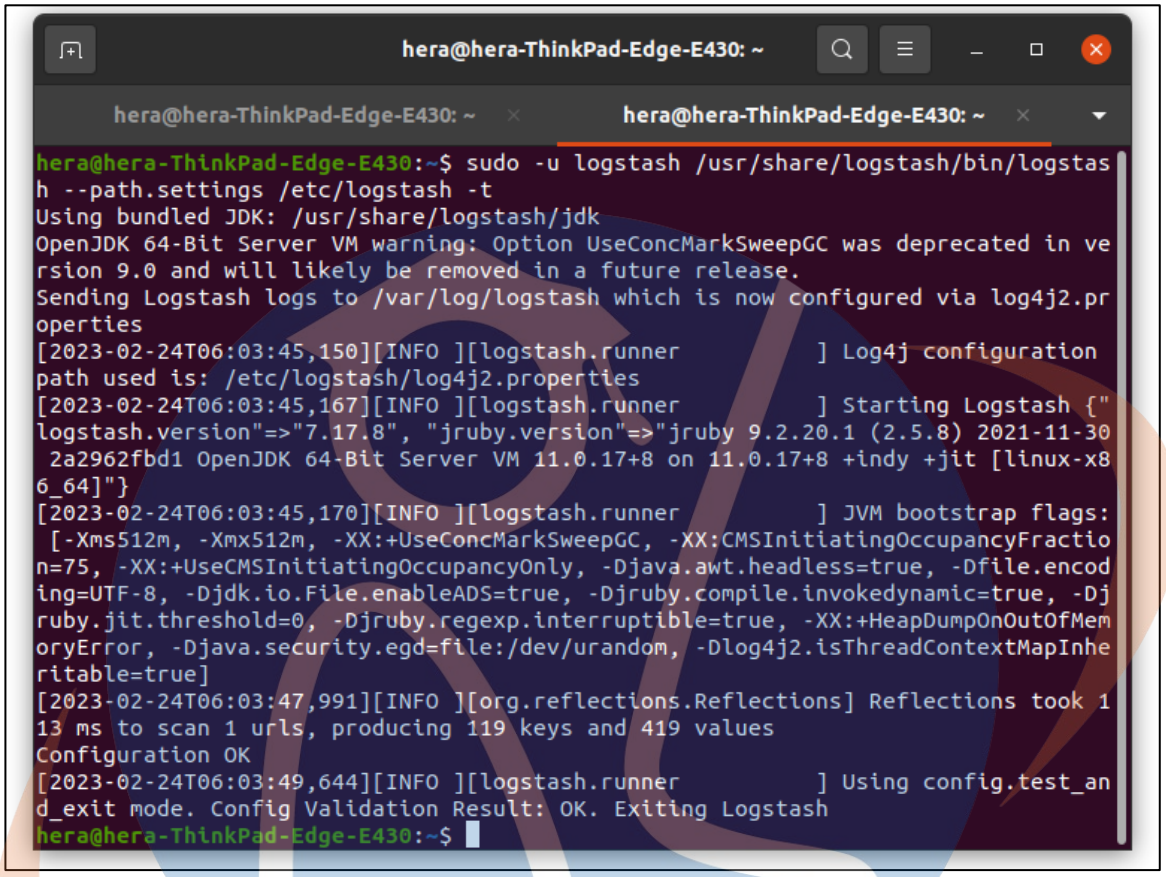


Gambar 5.4 Konfigurasi File 30-elasticsearch-output.conf

Selanjutnya jalankan perintah dibawah ini untuk memastikan bahwa konfigurasi berjalan dengan baik. Apabila tidak ada *syntax error* maka *output* yang dihasilkan adalah `Config Validation Result : OK. Exiting Logstash.`

```
$ sudo -u logstash
/usr/share/logstash/bin/logstash --path.settings
/etc/logstash -t
```

Apabila *output* pada komputer berupa *warning* dari *Open JDK* sepeti di bawah ini, maka logstash masih bisa berjalan dengan baik dan pesan tersebut dapat diabaikan.



Gambar 5.5 Validasi Config pada Logstash

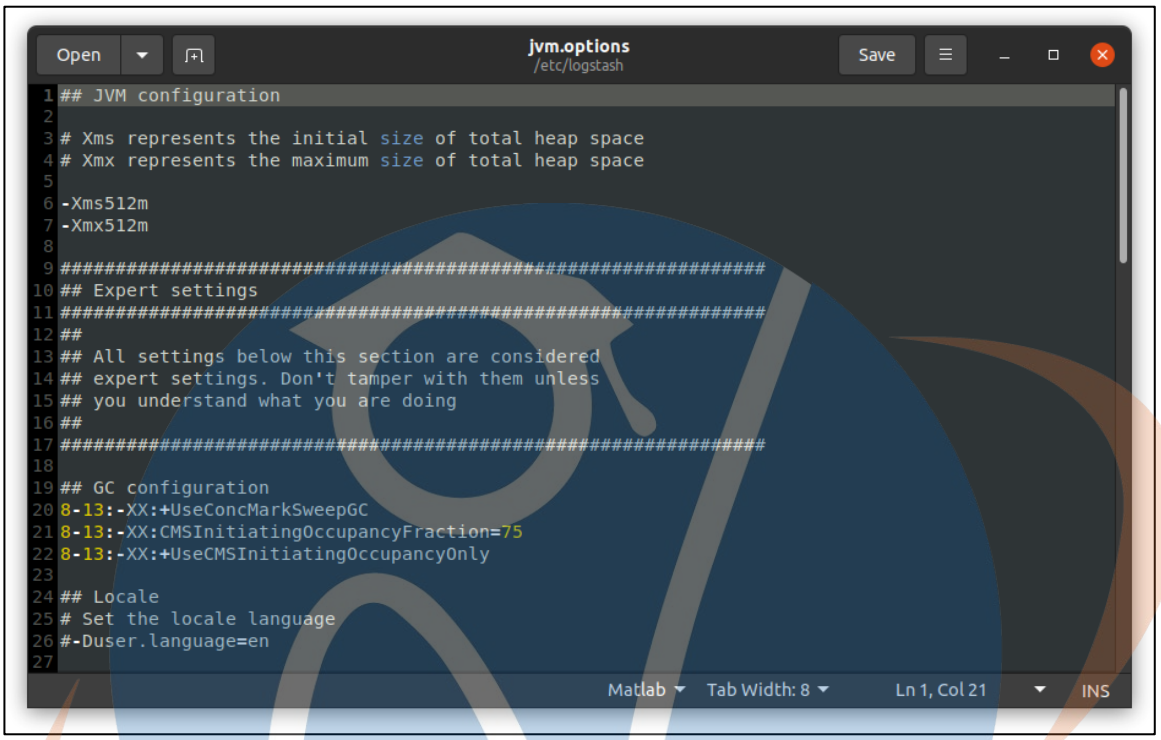
Selanjutnya mengubah nilai `Xms` dan `Xmx` menjadi 512m untuk mengatur server *memory* pada logstash dengan mengganti nilai sebagai berikut :

```
-Xms512m
-Xmx512m
```

Selanjutnya membuka file `jvm.options` dengan mengetikkan perintah di bawah ini :

```
$ sudo gedit /etc/logstash/jvm.options
```

Setelah itu mengganti nilai `Xms dan Xmx` menjadi 512m seperti dibawah ini lalu simpan *file*.



Gambar 5.6 Konfigurasi File JVM Options Logstash

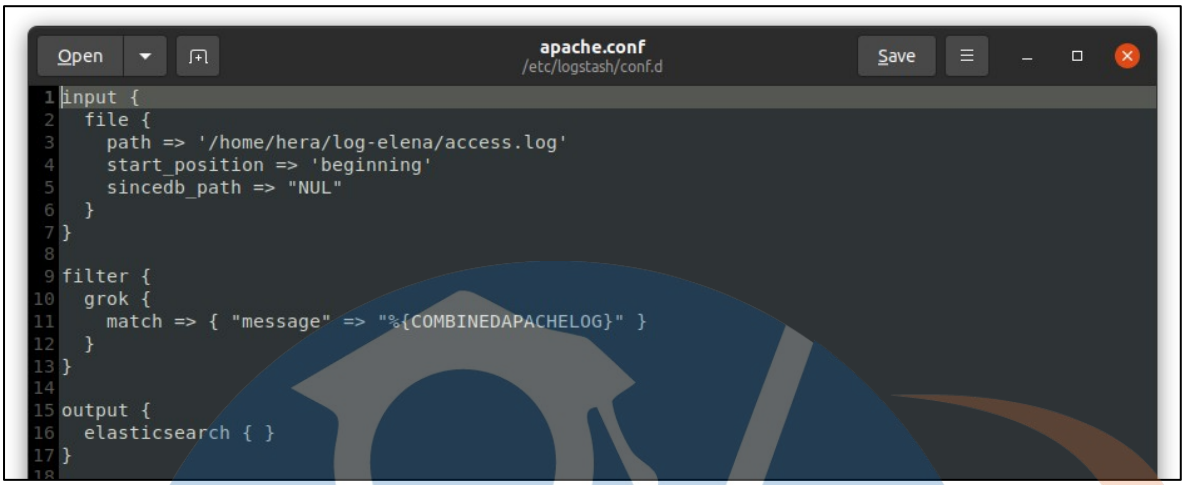
Selanjutnya membuat konfigurasi logstash pada `apache.conf` dengan mengetikkan perintah di bawah ini :

```
$ sudo /etc/logstash/conf.d/apache.conf
```

Salin *file* konfigurasi di bawah ini kemudian simpan dalam file tersebut.

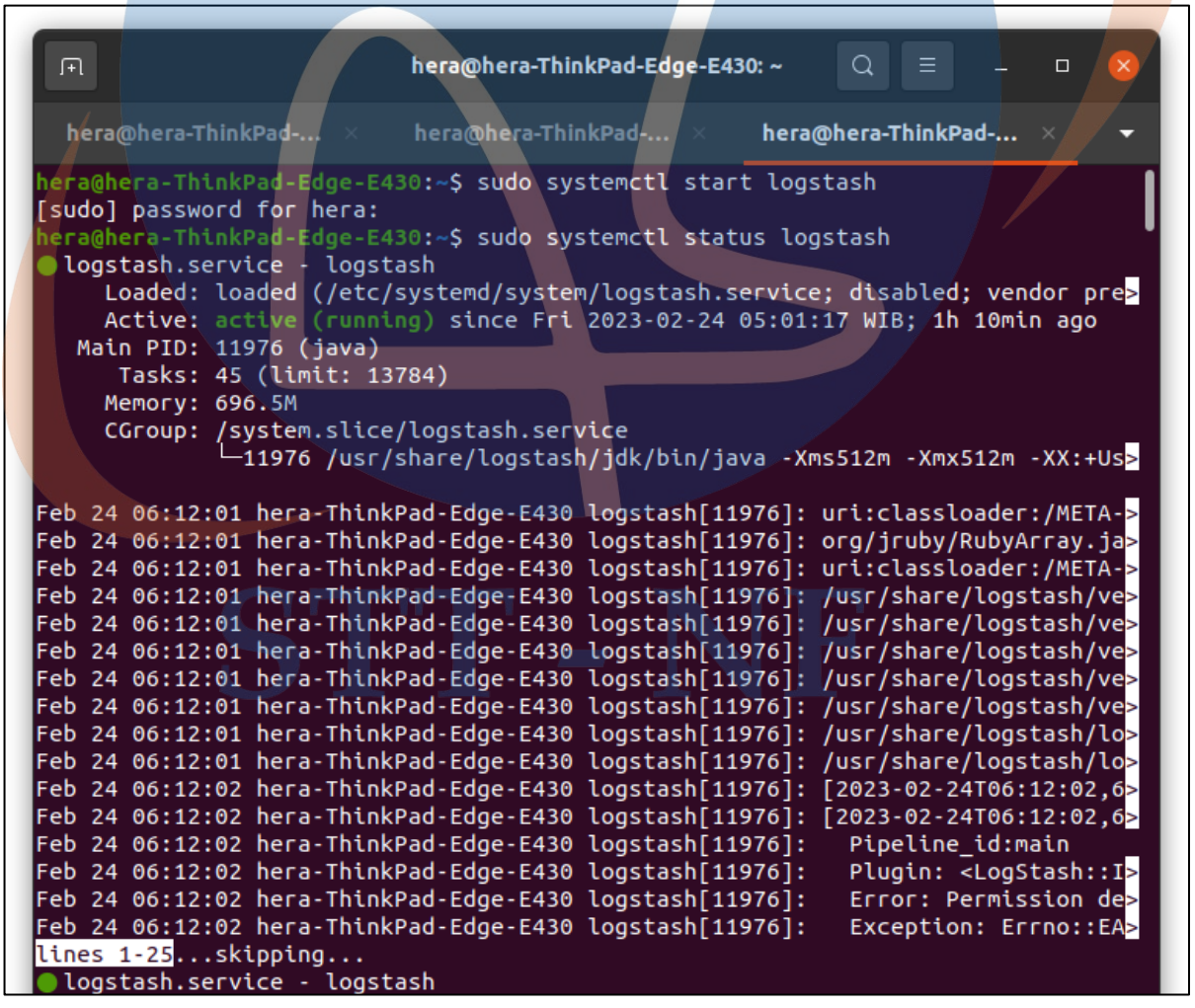
```
input  {
    file {

      path => T/var/www/*/logs/access.logTstart_position

      => TbeginningT

    }

}


filter {grok

   {

    match  =>  {  "message"  =>"{COMBINEDAPACHELOG}"  }

   }

}

output  {

     elasticsearch  {  }

}
```

Pada penelitian ini penulis mengatur *path* konfigurasi seperti di bawah ini :



Gambar 5.7 Konfigurasi Apache.conf Logstash

Tahapan selanjutnya yaitu menyalakan ulang logstash pada komputer :



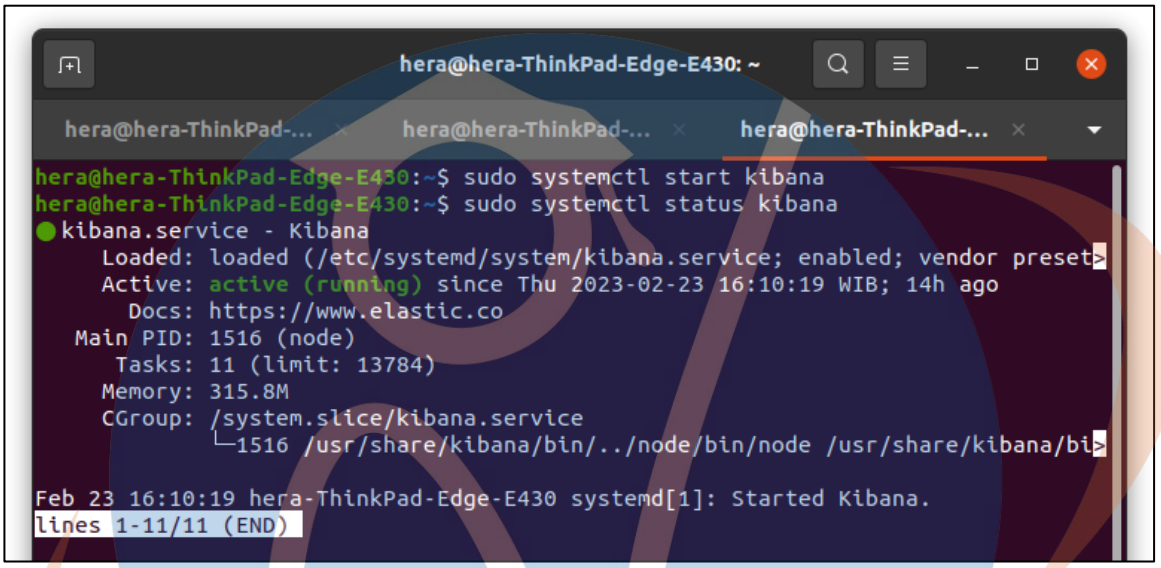
Gambar 5.8 Menghidupkan Logstash

5.1.1.5 Instalasi Kibana

Instalasi kibana dapat dilakukan dengan mengetikkan perintah berikut :

```
$ sudo apt-get install kibana
```

Selanjutnya jalankan kibana dengan mengetikkan perintah di bawah ini :



Gambar 5.9 Menghidupkan Kibana

5.1.2 Implementasi Rsync dan SSHpass

Tahapan selanjutnya yaitu sinkronisasi dan mengunduh access log elena yang terdapat pada *cloud server*. Berikut adalah informasi server yang menyimpan akses log elena :

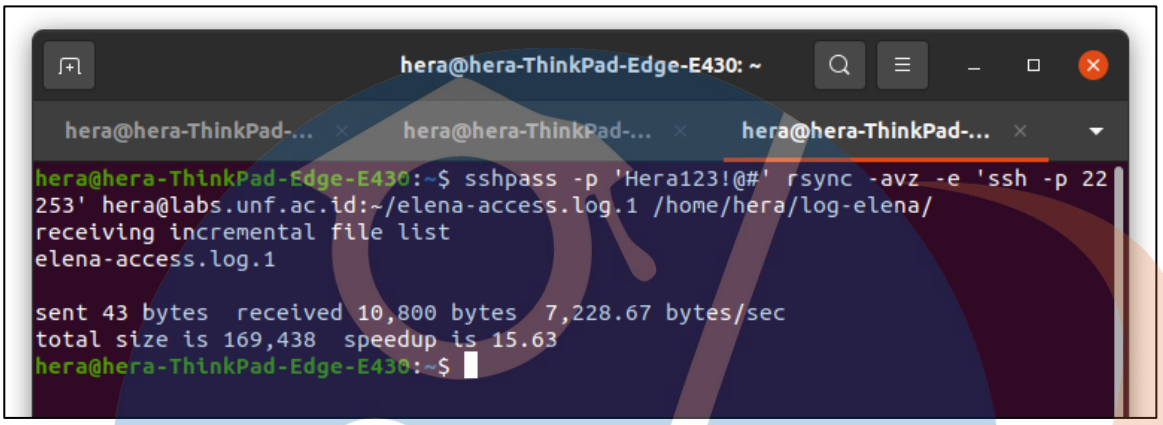
Table 5.1 Informasi Server Akses Log Elena

Informasi Server Akses Log Elena	
Host	labs.unf.ac.id
Port SSH	22542

Selanjutnya mengunduh *sshpass* untuk autentikasi ke server secara otomatis dengan mengetikkan perintah berikut :

```
$ sudo install sshpass
$ sshpass -p THera123!@#T rsync -avz -e Tssh -p
22253T   hera@labs.unf.ac.id:~/elena- access.log.1
/home/hera/log-elena/
```

Pada penelitian ini penulis menggunakan perintah *rsync* dan *SSHpass* untuk sinkronisasi data log secara *remote* tanpa memasukkan autentikasi *password* untuk masuk ke server elena buatan dengan nama `labs.unf.ac.id`. Perintah tersebut berguna untuk mengunduh data log dengan nama `elena-access.log.1` dan kemudian file log disimpan di komputer lokal pada direktori `log-elena`.

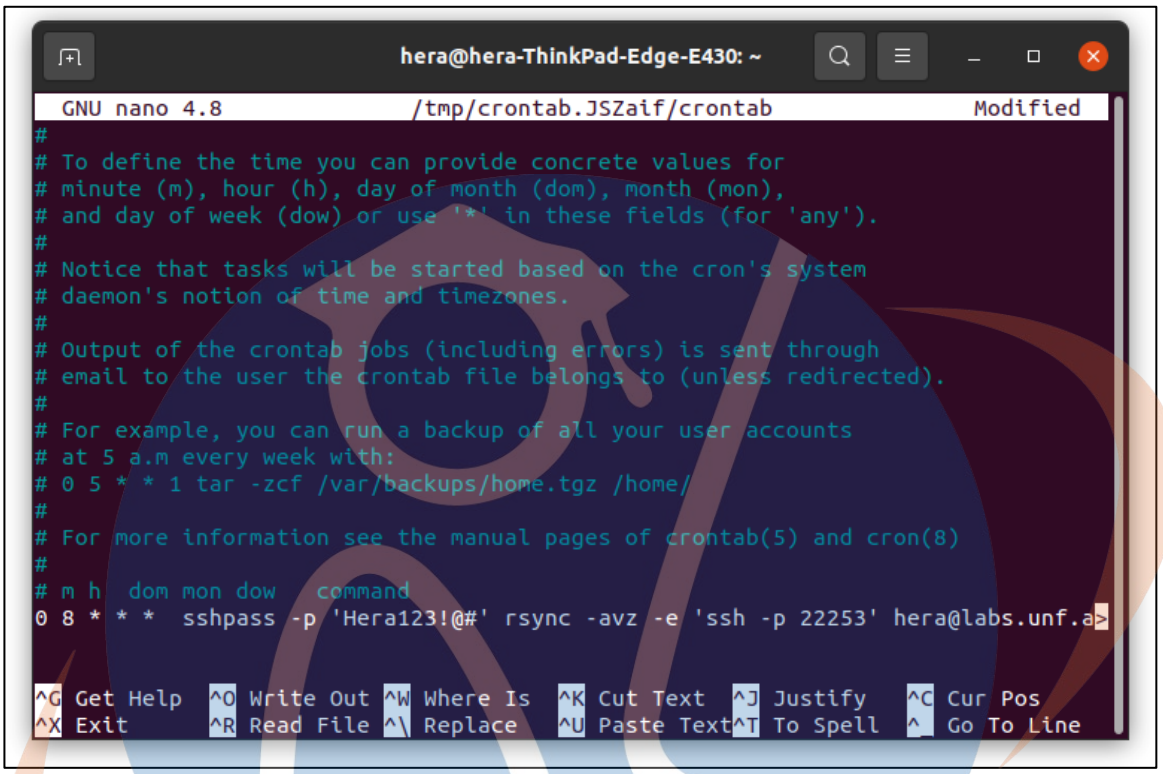


Gambar 5.10 Implementasi Rsync dan SSHpass

Selain itu untuk menjadwalkan *rsync* pada waktu tertentu dapat memasukkan perintah SSHpass diatas pada file konfigurasi crontab dengan mengetikkan perintah di bawah ini :

```
$ crontab -e
```

Output yang akan dikeluarkan oleh sistem berupa *file* konfigurasi crontab, selanjutnya masukkan baris perintah *SSHpass* diatas pada *file* konfigurasi crontab tersebut lalu atur penjadwalan setiap jam 8 menjadi seperti dibawah ini :



Gambar 11 Penjadwalan Sinkronisasi Data pada Cronjob

5.1.3 Visualisasi Akses Log Elena

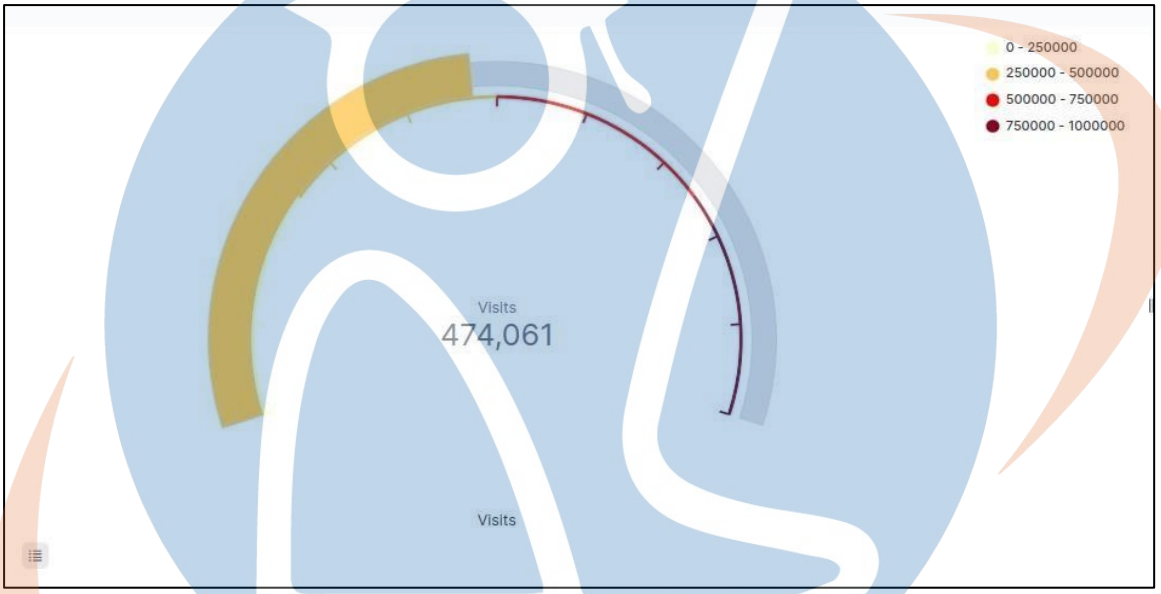
Data akses log yang digunakan dalam implementasi visualisasi pada penelitian ini adalah data log elena dari bulan oktober 2021 dengan jumlah data log sebesar 474061. Data log yang sudah otomatis diunduh selanjutnya dikirim oleh logstash dan terindeks di elasticsearch untuk selanjutnya akan divisualisasikan pada kibana. Visualisasi akan dilakukan berdasarkan perancangan visualisasi yang telat dibuat. Berikut adalah dashboard visualisasi data akses log elena yang sudah di buat di perancangan sebelumnya.

Gambar 5.12 Dashboard Visualisasi Akses Log Elena

5.1.3.1 Visualisasi *Visits*

Visualisasi *visits* pada penelitian ini adalah gambaran dari jumlah keseluruhan aktifitas pengunjung web elena yang tercatat pada log. Berikut langkah langkah dalam membuat visualisasi *visits* :

1. Menentukan diagram visualisasi, penulis menggunakan diagram *goal*
2. Kemudian diperlukan mengubah *metrics aggregation* menjadi '*count*' dan memasukkan *custom label* menjadi '*visits*'
3. Selanjutnya mengatur skala dan warna skala pada diagram *goal*.



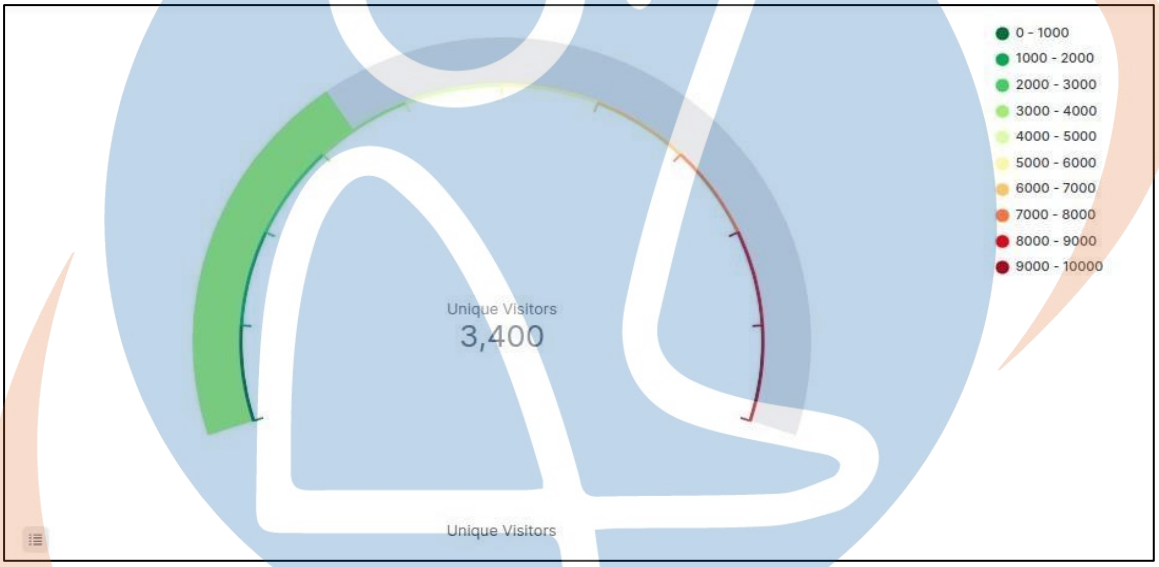
Gambar 5.13 Visualisasi Visits Log Elena

Berdasarkan visualisasi data tersebut informasi yang didaptkan yakni terdapat total 474,061 jumlah pengunjung web elena selama bulan oktober 2021. Data *visits* disajikan dalam bentuk *goal* karena bentuk visualisasi tersebut dapat merepresentasikan nilai tunggal padametrik, sehingga akan mudah untuk di pahami pembaca. Kegunaan dari visualisasi data visits ini adalah untuk memberikan informasi tentang jumlah pengunjung yang mengunjungi website elena sehingga dapat digunakan untuk memahami perilaku pengunjung website elena.

5.1.3.2 Visualisasi *Unique Visitor*

Visualisasi *unique visitor* pada penelitian ini adalah gambaran dari jumlah pengunjung unik berdasarkan *IP address* pengunjung web elena yang tercatat pada log. Berikut langkah langkah dalam membuat visualisasi *unique visitor* :

1. Menentukan diagram visualisasi, pada penelitian ini penulis memilih diagram *gauge.*
2. Kemudian mengubah *metrics aggregation* menjadi '*count*' dan memilih *field 'clientp.keywords'*.Selanjutnya masukkan *custom label* dan perlu diubah menjadi '*visits*'.
3. Kemudian mengatur skala dan warna skala untuk diagram *gauge* yang dibuat.



Gambar 5.14 Visualisasi Unique Visitor Log Elena

Berdasarkan visualisasi data tersebut terdapat total 3400 jumlah *IP address* pengunjung web elena selama bulan oktober 2021. Data *visits* disajikan dalam bentuk *gauge* karena bentuk visualisasi *gauge* tersebut dapat merepresentasikan perubahan terhadap target jumlah pengunjung unik dalam periode waktu tertentu. Selain itu visualisasi *unique visitors* berguna dan untuk mengetahui jumlah pengunjung yang mengunjungi situs web elena dan seberapa sering mereka kembali. Informasi ini dapat digunakan untuk menilai kinerja situs dan mengembangkan strategi pemasaran *online* yang lebih baik.

5.1.3.3 Visualisasi *Total Bytes*

Visualisasi *Total Bytes* pada penelitian ini adalah gambaran jumlah keseluruhan total data dan total *bytes* yang didapatkan dari halaman web elena yang di akses dan tercatat pada log. Berikut langkah langkah dalam membuat visualisasi *total bytes* :

1. Diagram visualisasi dibuat dengan memilih diagram *table*.
2. Kemudian mengubah *metrics aggregation* menjadi '*count*' dan masukkan *custom label* '*Total Data*'.
3. Kemudian tambahkan 2 *buckets* yaitu *'split rows'*
4. Kedua *agregartion* tersebut diubah menjadi '*terms*' dan masing-masing *fieldnya* menjadi *request keyword* dan *bytes.keyword.*

URL	Total Bytes	Total Data
/theme/image.php/lambda/theme/1632635432/bg/block-divider	118	27,361
/theme/image.php/lambda/theme/1632635432/glyphicons-halflings	4338	25,658
/theme/image.php/lambda/theme/1632635432/bg/icon-search	632	23,849
/theme/image.php/lambda/core/1632635432/t/collapsed	393	9,339
/theme/image.php/lambda/core/1632635432/t/expanded	392	9,199
/mod/quiz/processattempt.php?cmid=21609	1555	6,097
/lib/ajax/service-nologin.php?info=core_get_string&cachekey=1632635434&args=%5B%7B%22i...	232	5,471
/login/index.php	1527	3,900
/	0	3,501
/mod/quiz/processattempt.php?cmid=21609	1556	3,237

1 2 3 4 5 6

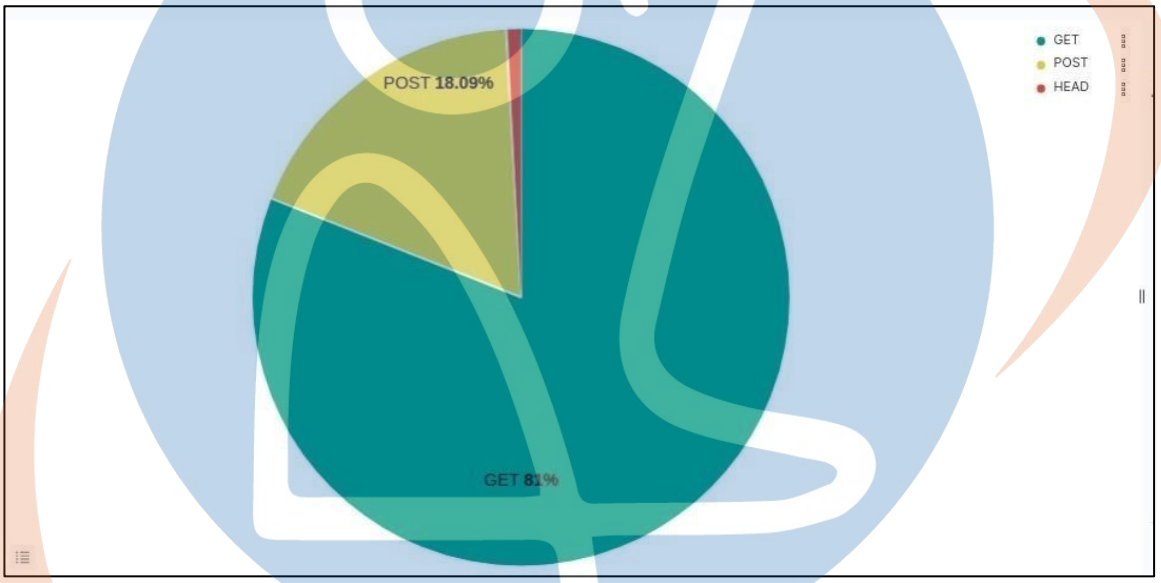
Gambar 5.15 Visualisasi Total Bytes Log Elena

Berdasarkan visualisasi data tersebut disajikan data 10 *url* web elena dan data *total bytes* yang tercatat pada log elena selama bulan oktober 2021. Data *total bytes* disajikan dalam bentuk *table* karena bentuk visualisasi tersebut dapat merepresentasikan data secara mendetail dan dapat membandingkan isi data dalam beberapa kategori. Selain itu visualisasi *total bytes* berguna dan untuk mengetahui ukuran *total bytes* serta mengoptimalkan ukuran file dan gambar pada situs, untuk mengurangi beban jaringan serta mempercepat waktu memuat situs web elena.

5.1.3.4 Visualisasi *Method Access*

Visualisasi *Method Access* pada penelitian ini adalah gambaran jumlah keseluruhan permintaan *(request)* ke web server elena dan tercatat pada log. Berikut langkah langkah dalam membuat visualisasi *method access* :

1. Visualisasi akan dibuat menggunakan diagram *pie chart*
2. Mengubah *metrics aggregation* menjadi '*count*' dan memasukkan *custom label* '*Method Access*'.
3. Kemudian menambahkan *buckets* dan *agregartion* yang diubah menjadi '*terms*'.
4. Selanjutnya perlu mengatur *field* menjadi *verb.keyword.*



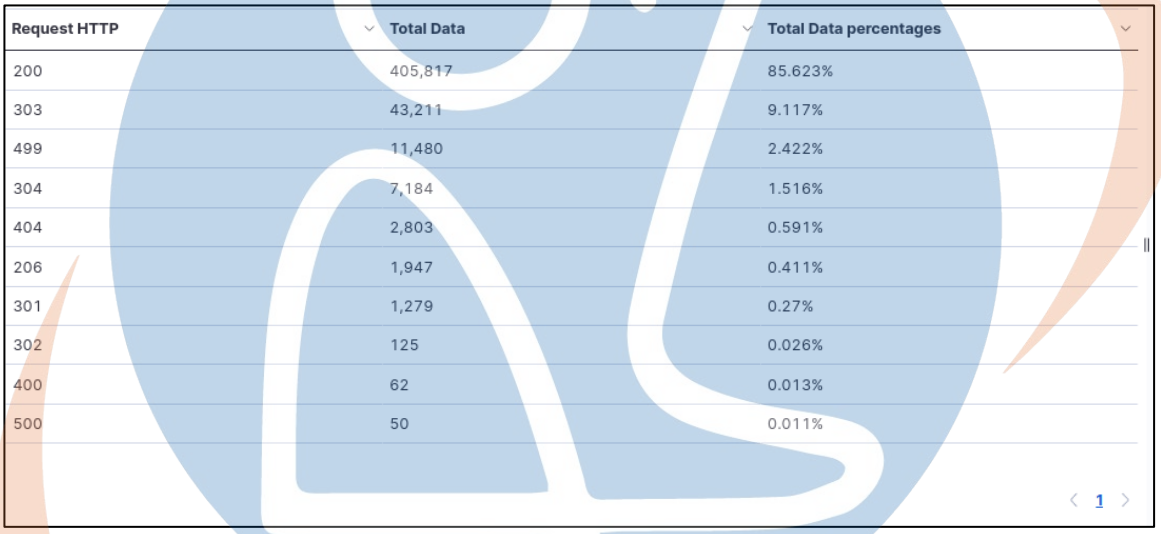
Gambar 5.16 Visualisasi Method Access Log Elena

Berdasarkan visualisasi data tersebut disajikan data 3 *method access* yang tercatat di log elena yaitu *GET* sebanyak 81%, *POST* sebanyak 18,9% dan *HEAD* sebanyak 1.1% dari total yang tercatat pada log elena selama bulan oktober 2021. Data tersebut disajikan dalam bentuk *pie charts* karena bentuk visualisasi ini mempermudah mengidentifikasi *request* atau respon yang tidak sesuai pada visualisasi *method access* tersebut. Informasi tentang *method access* pada log web elena dapat digunakan untuk memantau dan menganalisis permintaan klien ke server. Hal tersebut dapat membantu administrator sistem web elena dalam memperbaiki kinerja server, memperbaiki *bug*, dan meningkatkan pengalaman pengguna di situs web.

5.1.3.5 Visualisasi *Request HTTP*

Visualisasi *Request HTTP* pada penelitian ini adalah gambaran jumlah data yang merupakan bagian dari informasi dan disimpan oleh server setiap kali ada *request*. Berikut langkah langkah dalam membuat visualisasi *request http*:

1. Visualisasi dibuat dengan memilih diagram *table.*
2. Selanjutnya mengubah *metrics aggregation* menjadi '*count*' dan masukkan *custom label* '*Request HTTP*'.
3. Selanjutnya tambahkan *buckets 'split rows'.*
4. Lalu pilih *agregartion* dan diubah menjadi '*terms*' serta pilih *field 'response.keyword'.*



Request HTTP	Total Data	Total Data percentages
200	405,817	85.623%
303	43,211	9.117%
499	11,480	2.422%
304	7,184	1.516%
404	2,803	0.591%
206	1,947	0.411%
301	1,279	0.27%
302	125	0.026%
400	62	0.013%
500	50	0.011%

1

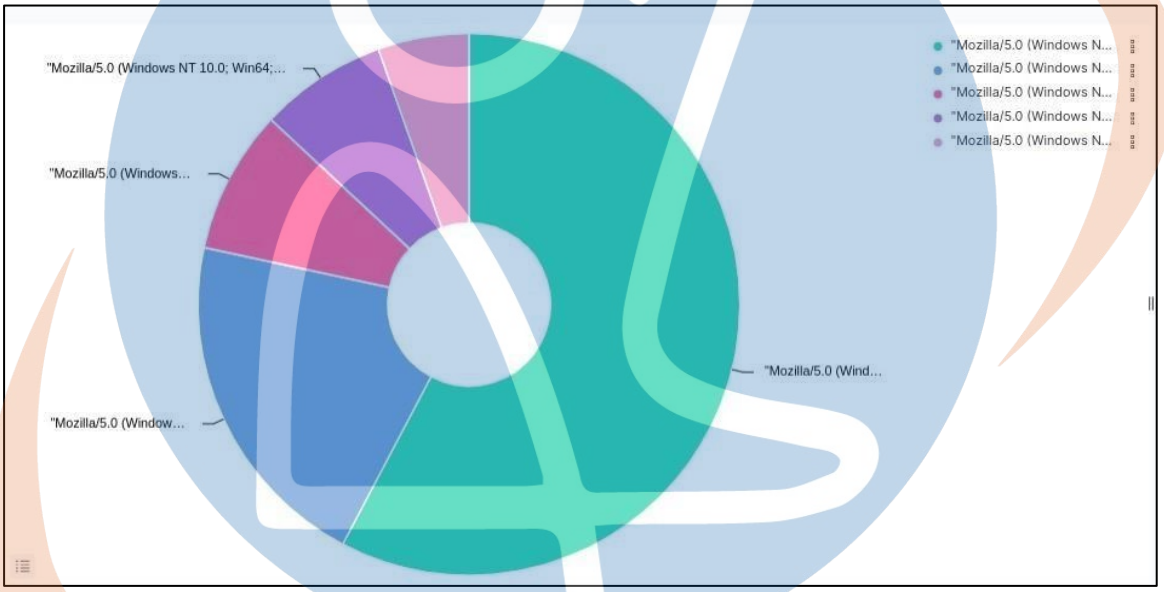
Gambar 5.17 Visualisasi Request HTTP Log Elena

Berdasarkan visualisasi data tersebut disajikan 10 data *request HTTP* yangtercatat, beserta total data dan persentase data tersebut dari keseluruhan data log elena selama bulan oktober 2021. Selain itu, *request HTTP* 200 menandakan sebagian besar permintaan klien berhasil diproses oleh server. Sedangkan untuk yang lain seperti 404 *"Not Found"* akan muncul apabila *file* yang diminta tidak ditemukan di server atau 500 *"Internal Server Error"* jika terjadi masalah pada sisi server. Data disajikan dalam bentuk *table* karena bentuk visualisasi ini berguna untuk membantu administrator sistem elena untuk mengetahui status dari *request dan respon* dan mempermudah dalam memperbaiki kesalahan.

5.1.3.6 Visualisasi *Most Used Browser and OS*

Visualisasi *Most Used Browser and OS* pada penelitian ini adalah gambaran jumlah data *browser* dan *OS* yang digunakan pengunjung dalam mengakses web elena. Berikut langkah langkah dalam membuat visualisasi *most used browser and OS*:

1. Visualisasi dibuat denganmemilih diagram '*donut chart'*.
2. Mengubah *metrics aggregation* menjadi '*count*' dan memasukkan *custom label* '*Most Used Browser and OS*'.
3. Kemudian menambahkan *buckets 'split rows'*.
4. Selanjutnya *agregartion* diubah menjadi '*terms*' dan *fieldnya* diubah menjadi *'agent.keyword'*.



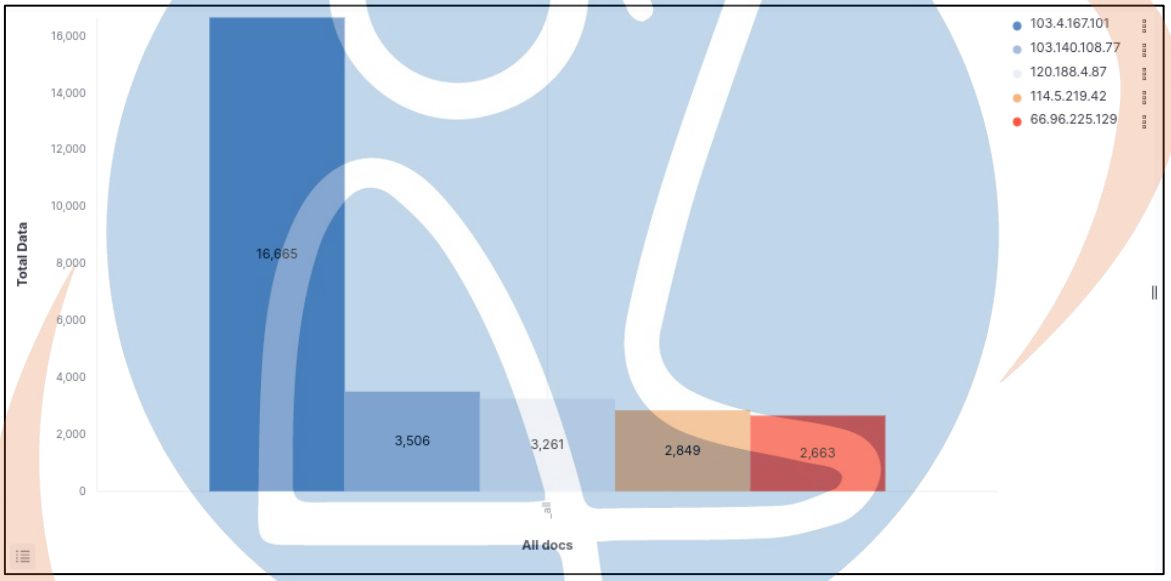
Gambar 5.18 Visualisasi Most Used Browser and OS Log Elena

Berdasarkan visualisasi data tersebut disajikan 5 data *most used browser and OS* yang tercatat dapat diketahui bahwa sistem operasi paling banyak digunakan adalah *windows* dan *browser* yang paling banyak digunakan adalah *google chrome* dari keseluruhan data log elena selama bulan oktober 2021. Data disajikan dalam bentuk *donut charts* karena bentuk visualisasi ini berguna untuk membantu mengetahui *browser* dan sistem operasi mana yang banyak digunakan untuk akses web elena.

5.1.3.7 Visualisasi *Top 10 IP Address*

Visualisasi *Top 10 IP Address* pada penelitian ini adalah gambaran jumlah data *browser* dan *OS* yang digunakan pengunjung dalam mengakses web elena. Berikut langkah langkah dalam membuat visualisasi *top 10 ip address*:

1. Visualisasi dibuat dengan memilih diagram '*bar chart*
2. Selanjutnya mengubah *metrics aggregation* menjadi '*count*' dan memasukkan *custom label menjadi* '*total data*'.
3. Kemudian menambahkan *buckets 'split rows'* serta mengubah *agregartion* menjadi '*terms*'
4. Selanjutnya mengubah *field* menjadi *'clientip.keyword'*.



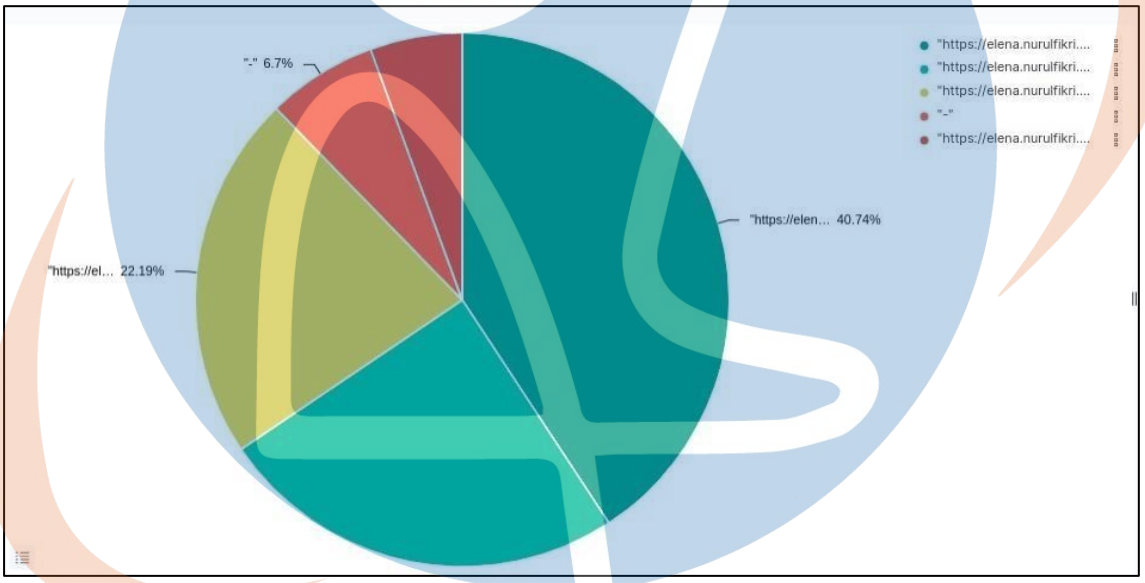
Gambar 5.19 Visualisasi Top 10 Ip Address Log Elena

Berdasarkan visualisasi data tersebut disajikan 5 data dari *10 IP Adress* yang paling banyak mengakses elena selama bulan oktober 2021. Data disajikan dalam bentuk *bar charts* karena bentuk visualisasi ini berguna untuk mengetahui perbandingan jumlah data 10 Ip terbanyak yang mengakses web elena dan dari data tersebur dapat memberikan pemahaman tentang lokasi geografis pengunjung web berdasarkan *IP adress.* Selain itu, jika sejumlah besar permintaan berasal dari satu *IP address*, maka dapat menunjukkan adanya bot atau *script* otomatis yang melakukan akses web.

5.1.3.8 Visualisasi *Top 5 Url*

Visualisasi *Top 5 Url* pada penelitian ini adalah gambaran jumlah data 5 *url* yang terbanyak dikunjungi pengunjung web elena: Berikut langkah langkah dalam membuat visualisasi *top 5 urls*:

1. Visualisasi dibuat dengan memilih diagram '*pie chart'*.
2. Mengubah *metrics aggregation* menjadi '*count*' dan memasukkan *custom label* '*Top 5 Urls*'.
3. Kemudian menambahkan *buckets 'split rows'* lalu *agregartion* diubah menjadi '*terms*'
4. Selanjutnya kolom *field* diatur menjadi *'referrer.keyword'*.



Gambar 5.20 Visualisasi Top 5 Url Log Elena

Berdasarkan visualisasi data tersebut disajikan 5 data *Url* web elena yang paling banyak diakses pengunjung web elena selama bulan oktober 2021 dengan hasil *Url* paling banyak diakses adalah link *Url* untuk memuat *file* CSS dari tema yang digunakan pada web elena, yang kedua adalah dashboard *course* pada web elena dan lainnya. Data disajikan dalam bentuk *pie charts* karena bentuk visualisasi ini berguna untuk membantu mengetahui jumlah pesentase dan informasi *Url* apa saja yang ada diwebsite elena dan paling banyak pengunjungnya.

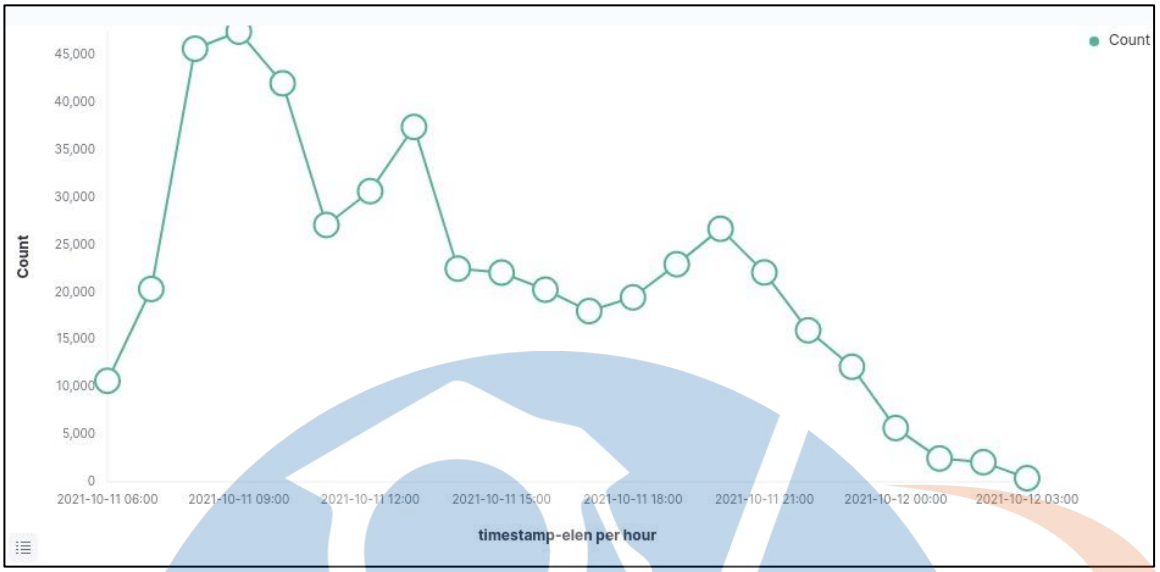
5.1.3.9 Visualisasi Data *Timeseries* (perjam)

Visualisasi Data *Timeseries* pada penelitian ini adalah gambaran data informasi aktifitas pengguna pada periode waktu tertentu dan lebih rinci dan biasanya digunakan untuk melihat kinerja situs web. Dalam membuat visualisasi ini diperlukan *field* yang dicustomisasi agar tipe datanya menjadi *date* dari sumber data berasal dari *field 'request.keyword'*. Berikut adalah *script* untuk merubah tipe data menjadi date di kibana.

```
if (doc['timestamp.keyword'].size() > 0) {
  def epochMilli =
Instant.from(DateTimeFormatter.ofPattern('dd/MMM/yyyy:HH:mm
:ss
Z').parse(doc['timestamp.keyword'].value)).toEpochMilli();
  if (epochMilli != 0) {
    emit(epochMilli);
  }
}
```

Setelah *filed* customisasi dibuat, maka dilanjutkan untuk membuat data timeseries. Berikut langkah langkah dalam membuat visualisasi data *timeseries*:

1. Visualisasi dibuat dengan memilih diagram '*line chart'*.
2. Mengubah *metrics aggregation* menjadi '*count*' dan memasukkan *custom label* '*Tmeseries*'.
3. Kemudian menambahkan *buckets 'split rows'* lalu *agregartion* diubah menjadi '*Date Histogram*'.
4. Selanjutnya kolom *field* diatur menjadi *'timestamp.elen'*.
5. Selanjutnya atur interval menjadi '*hour'*.



Gambar 5.21 Visualisasi Data Timeseries Log Elena

Berdasarkan visualisasi data tersebut disajikan data informasi aktifitas pengguna pada periode waktu tertentu dan lebih rinci yang dilakukan pengunjung web elena selama bulan oktober 2021. Hasilnya dapat diketahui jam paling sibuk pada visualisasi tersebut yakni sekitar jam 9 pagi, jam 12 siang dan jam 7 malam. Data disajikan dalam bentuk *line charts* karena bentuk visualisasi ini berguna untuk memperlihatkan tren atau pola perubahan data dari waktu ke waktu. Line chart menampilkan data dalam bentuk garis yang berhubungan dengan waktu atau dimensi waktu lainnya seperti bulan, tahun, atau periode lainnya. Selain itu visualisasi ini berguna untuk memperlihatkan bagaimana volume kunjungan pengguna di situs web apakah meningkat atau menurun selama beberapa bulan dan juga dapat membantu dalam mengambil keputusan untuk mengembangkan sistem elena berdasarkan tren dan pola data pengunjung yang terlihat.

5.1.3.10 Visualisasi *Traffic Log* (perbulan)

Visualisasi Data *Traffic Log* pada penelitian ini adalah gambaran data informasi aktifitas pengguna pada periode waktu tertentu perbulan yang digunakan untuk melihat kinerja situs web dan juga memberikan gambaran umum tentang pola kunjungan ke web elena. Dalam membuat visualisasi ini memakai *field* yang sudah dicustomisasi seperti pada visualisasi *timeseries*. Berikut langkah - langkah dalam membuat visualisasi data *traffic log*:

1. Visualisasi dibuat dengan memilih diagram '*line chart'*.
2. Mengubah *metrics aggregation* menjadi '*count*' dan memasukkan *custom label* '*Tmeseries*'.
3. Kemudian menambahkan *buckets 'split rows'* lalu *agregartion* diubah menjadi '*Date Histogram*'.
4. Selanjutnya kolom *field* diatur menjadi *'timestamp.elen'.*
5. Selanjutnya atur interval menjadi '*month'.*

Gambar 5.22 Visualisasi Data Traffic Log Perbulan Log Elena

Berdasarkan visualisasi data tersebut disajikan data informasi aktifitas pengguna pada periode waktu tertentu pengunjung web elena selama bulan oktober 2021. Hasilnya akses kunjungan hanya terdapat pada bulan oktober 2021 sebanyak 474061 kali Karena data yg digunakan hanya data bulan oktober. Data disajikan dalam bentuk *line charts* karena bentuk visualisasi ini berguna untuk menunjukkan jumlah total kunjungan ke website elena selama setiap bulan tertentu. Diagram *line chart* juga dapat membantu untuk melihat tren dan pola kunjungan pada bulan tertentu.

5.14 Pengujian

Pada penelitian ini, penulis akan menjelaskan pengujian yang dilakukan setelah mengimplementasian visualisasi log elena berbasis ELK Stack. Pengujian yang diimplementasikan adalah pengujian efektivitas yaitu menguji kesesuaian data parsing pada logstash dengan indeks yang tersimpan di elasticsearch dan juga pengujian efektifitas pada visualisasi yang sudah dibuat. Perhitungan pengujian efektfitifitas dengan cara berikut :

Table 5.2 Perhitungan Pengujian Efektifitas

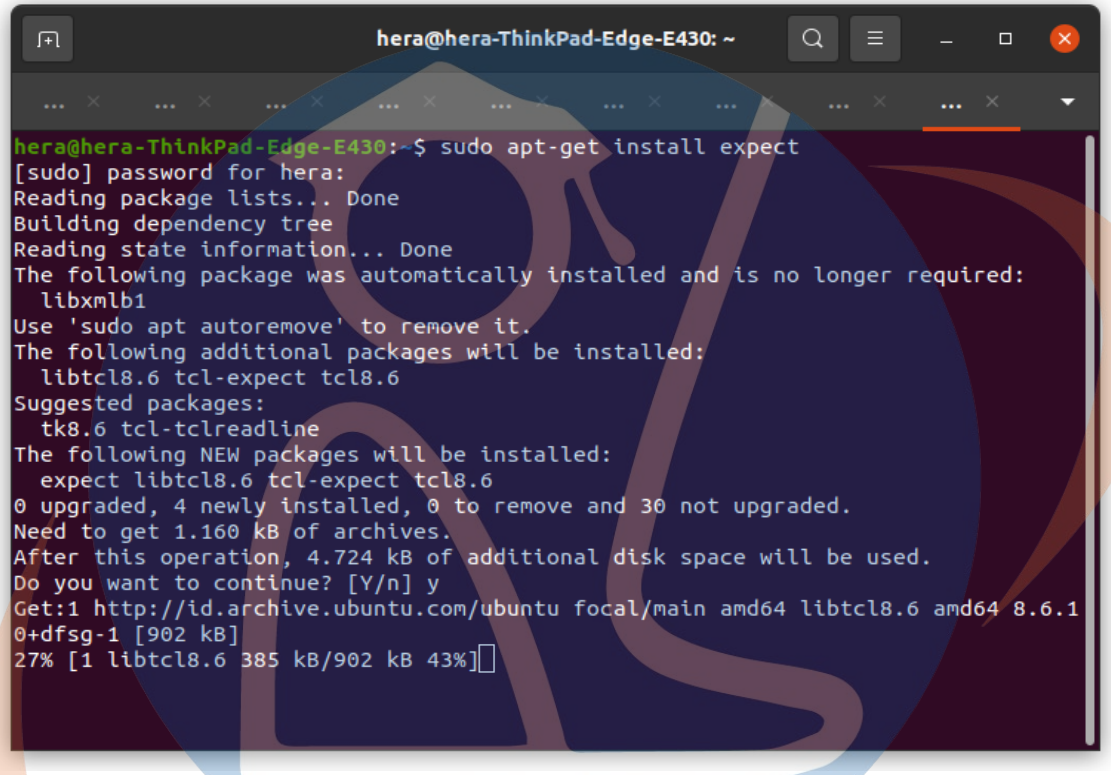
Perhitungan Pengujian Efektifitas
Efektifitas Pengujian : (Output/Input) * 100%
Rata-rata Efektifitas : (Total Efektifitas / Jumlah pengujian)

Tingkat Efektitifitas = Rata-rata Efektifitas

5.1.4.1 Pengujian Pengiriman Data Log (Logstash ke Elasticsearch)

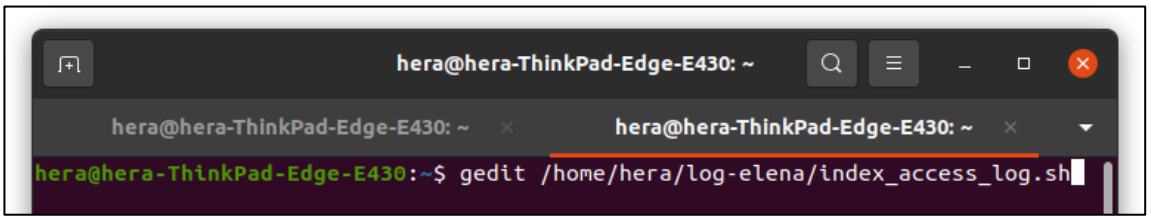
Pengujian ini digunakan untuk memeriksa pengiriman data log dari logstash ke elasticsearch untuk mengetahui apakah jumlah data yang di kirim oleh logstash dalam waktu tertentu dapat terindeks di elasticsearch. Proses pengujian dilakukan dengan cara membuat file *shell script* yang berisi kumpulan baris perintah untuk otomatis melakukan eksekusi mengirim data log dari logstash pada jumlah tertentu dan terindeks di elasticsearch. *Shell Script* yang telah dibuat kemudian akan dikirim pada rentang waktu tertentu dengan mengatur penjadwalan menggunakan cronjob. Setelah log dikirim akan diketahui jumlah logyang berhasil dikirim dan terindeks di elasticsearch.

Shell script / *script bash* pada pengujian ini berisi kumpulan baris perintah untuk melakukan pengiriman data log pada jumlah terntentu dan waktu tertentu yang dapat diatur sesuai dengan kebutuhan dari logstash ke elasticsearch. Sebelum membuat *file shell script* pastikan komputer sudah menginstall *expect* dengan mengetikkan perintah dibawah ini.



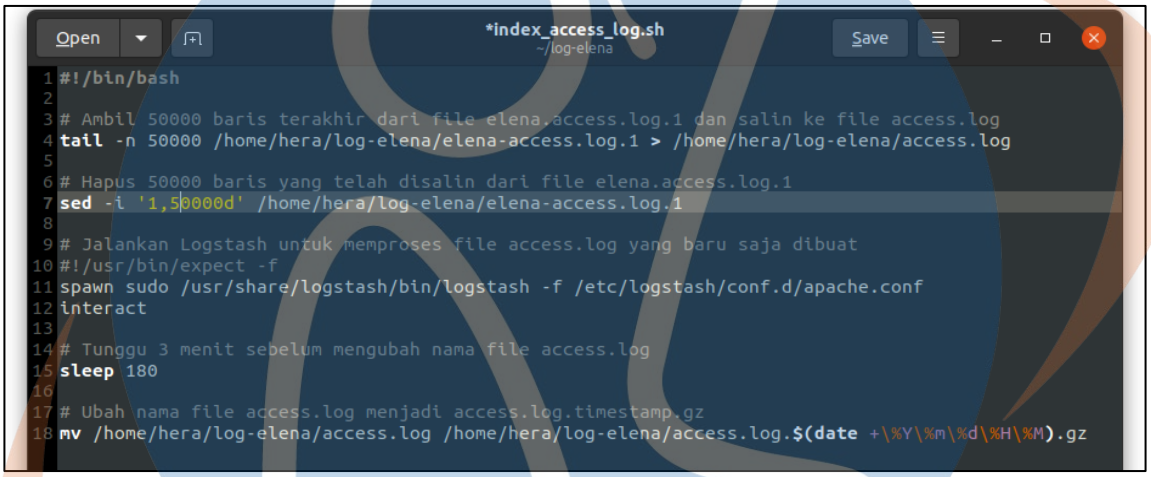
Gambar 5.23 Instalasi Expect

File script bash dapat dibuat dengan mengetikkan perintah sebagai berikut :



Gambar 5.24 Membuat File index_access_log_sh

Selanjutnya mengubah *file shell script* tersebut dan masukkan kumpulan baris perintah untuk pengujian log yang telah dirancang sebagai berikut :

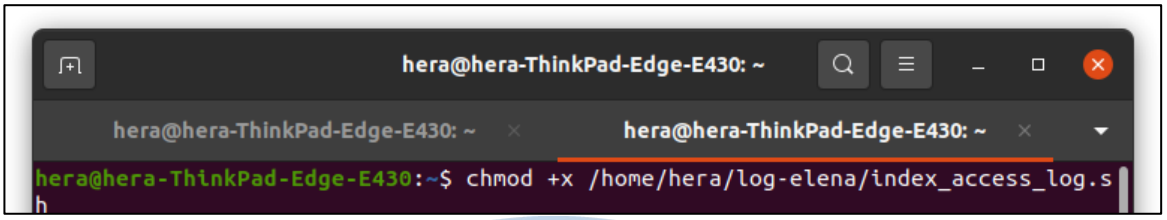


Gambar 5.25 Membuat File Shell Script

Pada *script* tersebut terdapat beberapa baris perintah yang penulis buat untuk melakukan kebutuhan pengujian pengiriman log, yakni sebagai berikut :

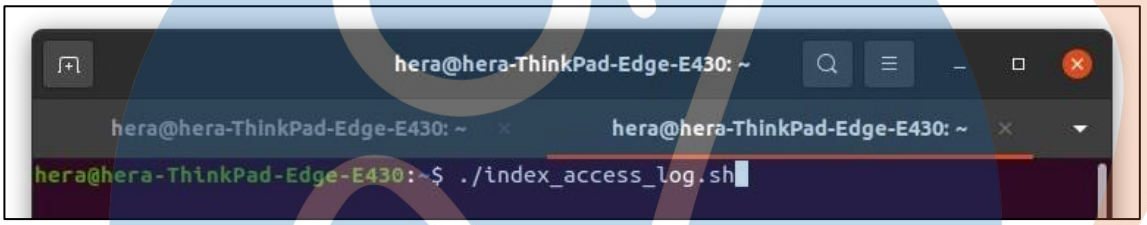
1. Perintah "`tail`" melakukan pembagian file data akses log menjadi 50000 log dari file `elena.access.log.1` kemudian log yg sudah di bagi disimpan ke file `access.log`.
2. Baris perintah "`sed`" digunakan untuk menghapus data akses log sebanyak 50000 log yang sudah disalin dari `elena-access-log.1` kefile `access.log`.
3. Selanjutnya perintah untuk mengindeks data log dari logstash ke elasticsearch dimasukkan pada *script* tersebut dengan menggunakan perintah "`spawn sudo`" agar perintah logstash dapat dieksekusi di *script* tanpa perlu memasukkan authentikasi dari user terminal linux.
4. Pengiriman file logstash dilakukan secara bergantian dengan nama file `access.log` Setelah data dikirim dan berhasil diindeks selanjutnya ekstensi file `access.log` akan berubah menjadi "`tar.gz`" agar ukuran file kecil.

Selanjutnya simpan file *script* tersebut dan selanjutnya mengetikkan perintah dibawah ini untuk memberikan izin ekseskusi perintah pada file *script* tersebut.



Gambar 5.26 Peintah Authentikasi File Script

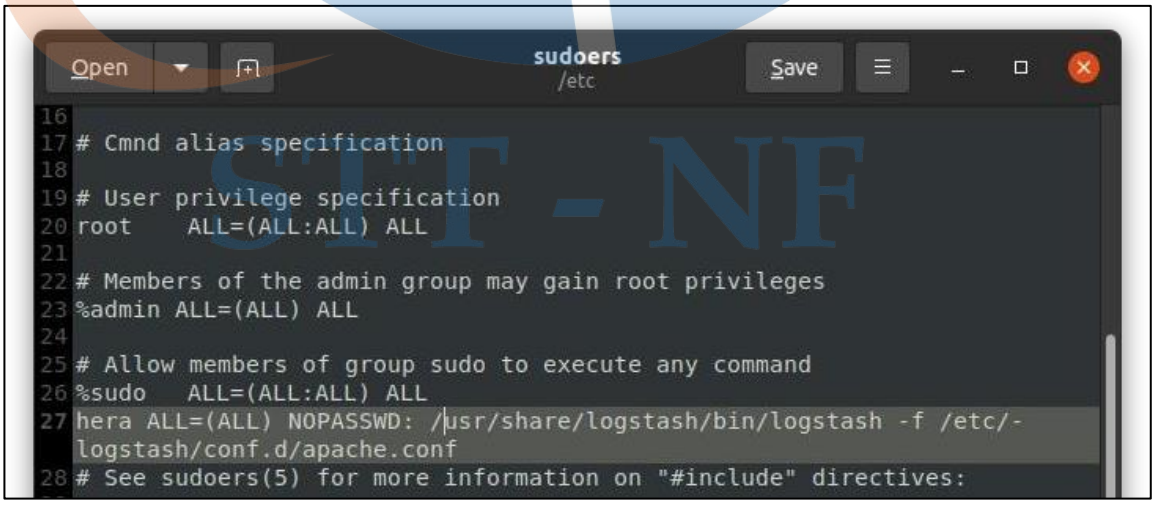
Selanjutnya menjalankan perintah untuk eksekusi file *script split_logs* dengan mengetikkan perintah di bawah ini :



Gambar 5.27 Perintah Menjalankan File Script

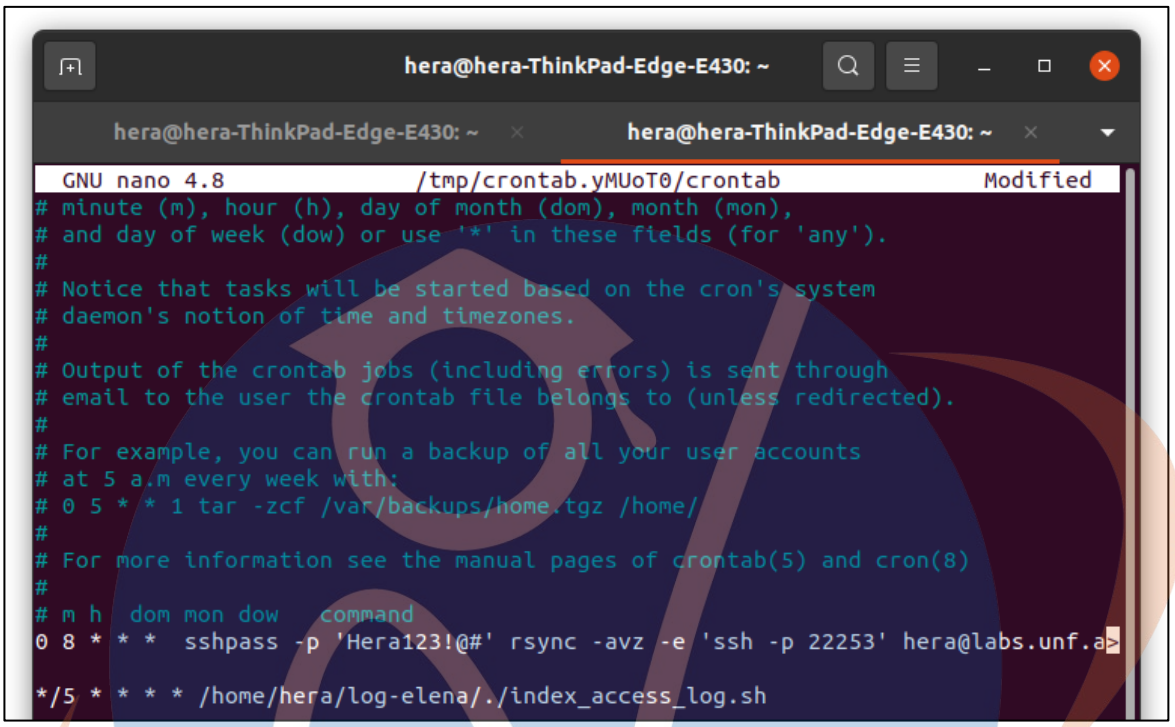
Kemudian memasukkan *permission* agar dapat mengeksekusi perintahtanpa "sudo" dengan memasukkan perintah seperti dibawah ini :

```
$ sudo gedit/etc/sudoers
```



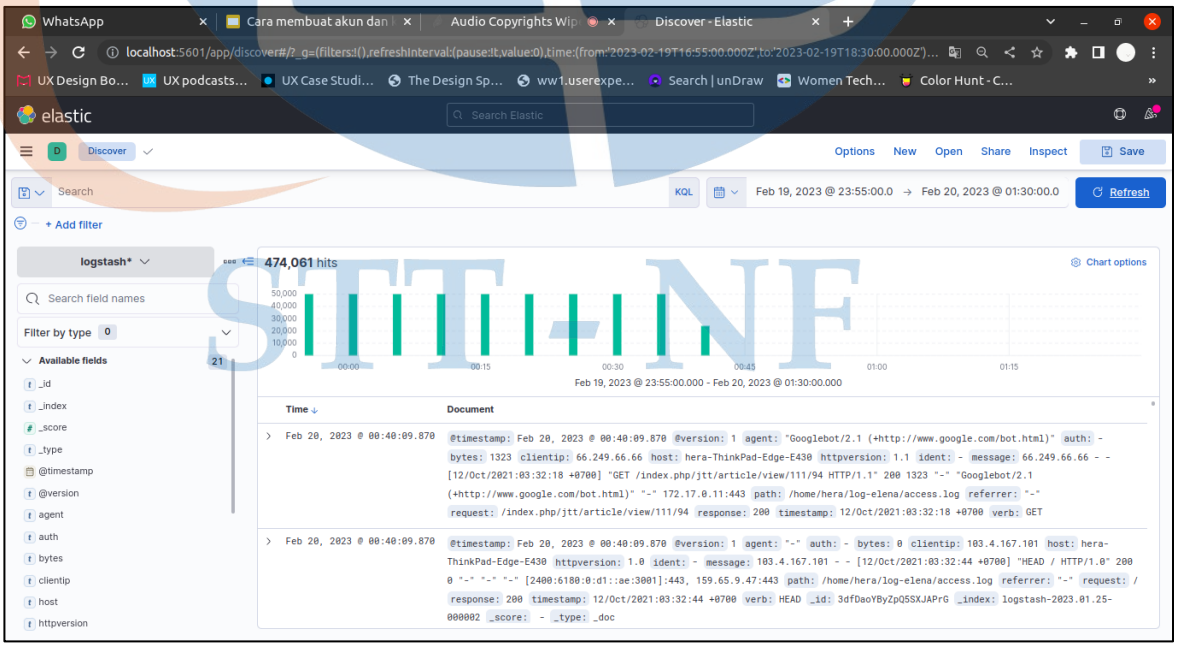
Gambar 5.28 Mengatur Permission Sumber

Selanjutnya memasukkan perintah *rsync* dan *script* pada *crontab* agar dapat dieksekusi secara teratur berdasarkan waktu yang dibuat. File akan diatur sebagai berikut:



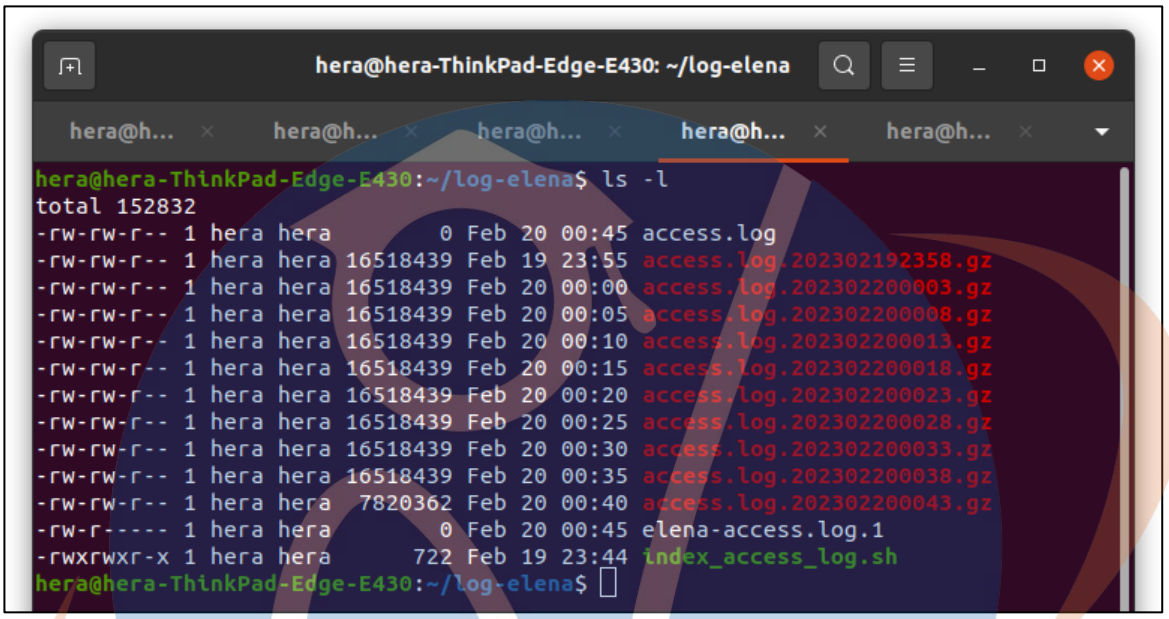
Gambar 5.29 Mengatur Penjadwalan Rsync dan Shell Script

Selanjutnya mengecek pada kibana apakah data log berhasil dikirim dengan menggunakan *script* yang sudah dijalankan.



Gambar 5.30 Discover Data Pegujian pada Kibana

Selanjutnya mengecek direktori log-elena dengan mengetikkan perintah sebagai berikut untuk mengetahui apakah *script* berjalan dengan baik yaitu dapat membagi log, mengirim log, menghapus log dan mengubah ekstensi data log setelah di indeks dengan mengetikkan perintah berikut :



Gambar5.31 Perubahan File Direktori log-elena

Pada kedua gambar tersebut terlihat bahwa file log `elena.access.log.1` berhasil membuat file `access.log` dan berisi file 50000 log dan berhasil di kirim ke logstash. Kemudian 50000 file log yg sudah disalin akan dihapus dari file log `elena.access.log.1` untuk selanjutnya file `access.log` akan diindeks di elasticsearch dan hasilnya terlihat di *discover* data kibana bahwa data dapat di indeks dan divisualkan dengan baik. Selain itu, data *script* berhasil mengubah ekstensi file mennjadi `tar.gz`. Proses ini diulang sebanyak 10 kali dalam pengujian pada penelitian ini.

5.1.4.2 Pengujian Visualisasi Data Log (Kibana)

Pengujian ini digunakan untuk memeriksa apakah data yang dikirim logstash sudah bisa diindeks di kibana dan berhasil divisualisasikan berdasarkan rancangan visualisasi yang sudah dibuat oleh penulis. Terdapat 8 rancangan visualisasi yang sudah dibuat dan selanjutnya akan dilakukan pengujian. Pengujian akan dilakukan dengan melihat apakah visualisasi secara efeketif ditampilkan oleh kibana dan juga melihat kesesuaian data log dengan data yg divisualisasi kan di dashboard kibana.

5.1.4.3 Hasil Pengujian Mengirim Data Log (Logstash ke Elasticsearch)

Table 5.5 Hasil Pengujian Indexing Akses Log

No.	Jumlah Log (input)	Jumlah Log (output)	Waktu Awal	Waktu Akhir	Selisih waktu /detik	Hasil Uji	Persentase
1.	50.000 log	50.000 log	00.32.31	00.33.00	29	Berhasil menyalin	100 %
2.	50.000 log	50.000 log	00.35.36	00.36.08	32	Berhasil menyalin	100 %
3.	50.000 log	50.000 log	00.38.17	00.39.50	33	Berhasil menyalin	100 %
4.	50.000 log	50.000 log	00.40.40	00.41.10	30	Berhasil menyalin	100 %
5.	50.000 log	50.000 log	00.42.53	00.43.25	32	Berhasil menyalin	100 %
6.	50.000 log	50.000 log	00.45.01	00.45.32	31	Berhasil menyalin	100 %
7.	50.000 log	50.000 log	00.47.11	00.47.45	34	Berhasil menyalin	100 %

Table 5.5 Hasil Pengujian Indexing Akses Log

8.	50.000 log	50.000 log	00.50.33	00.51.09	33	Berhasil menyalin	100 %
9.	50.000 log	50.000 log	00.56.04	00.56.35	31	Berhasil menyalin	100 %
10.	24061 log	24061 log	00.59.09	00.59.32	23	Berhasil menyalin	100 %
Rata- rata							100 %

Penjelasan pengujian indeks log:

1. Pada Pengujian ke 1, jumlah input data log = jumlah output 50000 log maka efektifitas data = (50000/50000) x 100% = 100%
2. Pada Pengujian ke 2, jumlah input data log = jumlah output 50000 log maka efektifitas data = (50000/50000) x 100% = 100%
3. Pada Pengujian ke 3, jumlah input data log = jumlah output 50000 log maka efektifitas data = (50000/50000) x 100% = 100%
4. Pada Pengujian ke 4, jumlah input data log = jumlah output 50000 log maka efektifitas data = (50000/50000) x 100% = 100%
5. Pada Pengujian ke 5, jumlah input data log = jumlah output 50000 log maka efektifitas data = (50000/50000) x 100% = 100%
6. Pada Pengujian ke 6, jumlah input data log = jumlah output 50000 log maka efektifitas data = (50000/50000) x 100% = 100%
7. Pada Pengujian ke 7, jumlah input data log = jumlah output 50000 log maka efektifitas data = (50000/50000) x 100% = 100%
8. Pada Pengujian ke 8, jumlah input data log = jumlah output 50000 log maka efektifitas data = (50000/50000) x 100% = 100%
9. Pada Pengujian ke 9, jumlah input data log = jumlah output 50000

log maka efektifitas data = (50000/50000) x 100% = 100%

10. Pada Pengujian ke 10, jumlah input data log = jumlah output 50000 log maka efektifitas data = (50000/50000) x 100% = 100%

Rata-rata efektifitas indeksing data log = (100% + 100% + 100% +100% +100% +100% +100% +100% +100% +100%) / 10 = 100%

Kesimpulannya : data pengujian dapat dikatakan memiliki **tingkat efektifitas 100%.**

5.1.4.4 Hasil Pengujian Visualisasi Log (Elasticsearch ke Kibana)

Table 5.4 Hasil Pengujian Visualisasi ELK Akses Log

No.	Pengujian	Deskripsi	Jenis Visualisasi	Hasil Uji	Persentase
1.	*Visits*	Menampilkan visualisasi *Visits* pada Kibana.	*Goal*	Berhasil menampilkan	100 %
2.	*Unique Visitors*	Menampilkan visualisasi *Visitors* pada Kibana.	*Gauge*	Berhasil menampilkan	100 %
3.	*Bytes Total*	Menampilkan visualisasi *Bytes Total* pada Kibana	*Table*	Berhasil menampilkan	100 %
4.	*Method Access*	Menampilkan visualisasi *Method Access* pada Kibana	*Donut Charts*	Berhasil menampilkan	100 %
5.	*Response HTTP*	Menampilkan visualisasi *Response HTTP* pada Kibana	*Table*	Berhasil menampilkan	100 %
9.	*Data Timeseries* (perjam)	Menampilkan visualisasi data *timeseries* perjam	*Line Chart*	Berhasil menampilkan	100 %
10.	*Traffic Log* (perbulan)	Menmpilkan visualisasi data jumlah *traffic* log perbulan	*Line Chart*	Berhasil menampilkan	100 %

Table 5.4 Hasil Pengujian Visualisasi ELK Akses Log

6.	*Most Used Browser and OS*	Menampilkan visualisasi *Bytes Total* pada Kibana	*Pie Charts*	Berhasil menampilkan	100 %
7.	*Top 10 Ip Address*	Menampilkan visualisasi *Top 10 Ip Address* pada Kibana	*Bar Char*	Berhasil menampilkan	100 %
8.	*Top 5 Url*	Menampilkan visualisasi *Top 5 Url* pada Kibana	*Pie Charts*	Berhasil menampilkan	100 %

Penjelasan pengujian visualisasi:

1. Pada Pengujian visualisasi ke 1 yaitu "*Visits"*, data log yang diinput berhasil di visualisasikan dalam bentuk *goal* maka efektifitas data = 100%.
2. Pada Pengujian visualisasi ke 2 yaitu "*Unique visitor"*, data log yang diinput berhasil di visualisasikan dalam bentuk *gauge* maka efektifitas data = 100%.
3. Pada Pengujian visualisasi ke 3 yaitu "*Bytes Total"*, data log yang diinput berhasil di visualisasikan dalam bentuk *table* maka efektifitas data = 100%.
4. Pada Pengujian visualisasi ke 4 yaitu "*Method Access"*, data log yang diinput berhasil di visualisasikan dalam bentuk *donut charts* maka efektifitas data = 100%.
5. Pada Pengujian visualisasi ke 5 yaitu "*Response HTTP"*, data log yang diinput berhasil di visualisasikan dalam bentuk *table* maka efektifitas data = 100%.
6. Pada Pengujian visualisasi ke 6 yaitu "*Most used browser and OS"*, data log yang diinput berhasil di visualisasikan dalam bentuk *pie chart* maka efektifitas data = 100%.

7. Pada Pengujian visualisasi ke 7 yaitu "*Top 10 Ip Address"*, data log yang diinput berhasil di visualisasikan dalam bentuk *bar chart* maka efektifitasdata = 100%.
8. Pada Pengujian visualisasi ke 8 yaitu *"Top 5 Url"*, data log yang diinput berhasil di visualisasikan dalam bentuk *pie chart* maka efektifitas data = 100%.

Rata-rata efektifitas visualisasi log = (100% + 100% + 100% +100% +100% +100% +100% +100% +100% +100% +100%) / 10 = 100%

Kesimpulannya: data pengujian dapat dikatakan memiliki **tingkat efektifitas 100%.**