LAPORAN PROYEK AKHIR SEMESTER
ANALISIS ANOMALI TRAFIK WEB SERVER MENGGUNAKAN EKOSISTEM BIG DATA (ELK STACK) DAN MACHINE LEARNING

Pengelolaan Big Data
Dosen Pengampu**:** M. Syendi Apriko, S.Pd., M.Kom.

Disusun Oleh:

Nama	:	Khalifa Alhasan
NIM	:	23041450105
Kelas	:	2383G

PROGRAM STUDI SISTEM INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UIN RADEN FATAH PALEMBANG
2025

DAFTAR ISI

DAFTAR GAMBAR

BAB I
PENDAHULUAN

1.1 Latar Belakang

Di era digital saat ini, log server merupakan salah satu sumber data terbesar (*Big Data*) yang dihasilkan oleh infrastruktur teknologi informasi. Setiap detik, server web mencatat ribuan hingga jutaan interaksi pengguna, menghasilkan data semi-terstruktur yang memiliki karakteristik *Volume*, *Velocity*, dan *Variety*. Pengelolaan data log secara tradisional menggunakan *database* relasional seringkali mengalami kendala performa ketika volume data mencapai ukuran Gigabyte atau Terabyte.

Oleh karena itu, diperlukan pendekatan teknologi *Big Data* menggunakan platform seperti Elasticsearch, Logstash, dan Kibana (ELK Stack) untuk melakukan proses *ingestion*, penyimpanan, dan visualisasi secara efisien. Selain itu, sekadar menyimpan data log tidaklah cukup. Diperlukan analisis lanjutan untuk mendeteksi pola tidak wajar (*anomali*) yang mengindikasikan adanya serangan siber seperti *Distributed Denial of Service* (DDoS) atau upaya pembobolan sistem.

Proyek ini bertujuan untuk mensimulasikan pemrosesan Big Data secara *end-to-end*, mulai dari pengumpulan dataset log server berskala besar (>1GB), proses ETL (*Extract, Transform, Load*), hingga penerapan algoritma *Machine Learning* (Isolation Forest) untuk mendeteksi anomali trafik secara otomatis.

1.2 Rumusan Masalah

1 Bagaimana membangun *pipeline* Big Data untuk mengolah file log server berukuran besar (>1GB) menggunakan ELK Stack?
2 Bagaimana memvisualisasikan pola trafik server untuk mendapatkan wawasan (*insight*) operasional?
3 Bagaimana menerapkan algoritma *Machine Learning* untuk mendeteksi anomali pada trafik web server?

1.3 Tujuan Pembelajaran

1 Mempraktikkan teknik ETL pada data skala besar.
2 Menggunakan platform Big Data (Elasticsearch) untuk penyimpanan dan pencarian data.
3 Melakukan analisis data menggunakan metode statistik dan *Machine Learning* sederhana.
4 Menyajikan visualisasi data untuk mendukung pengambilan keputusan.

BAB II
METODOLOGI DAN ARSITEKTUR SISTEM

2.1 Deskripsi Dataset

Dataset yang digunakan dalam proyek ini adalah NASA HTTP Server Log. Dataset ini berisi catatan semua request HTTP ke server WWW NASA Kennedy Space Center di Florida.

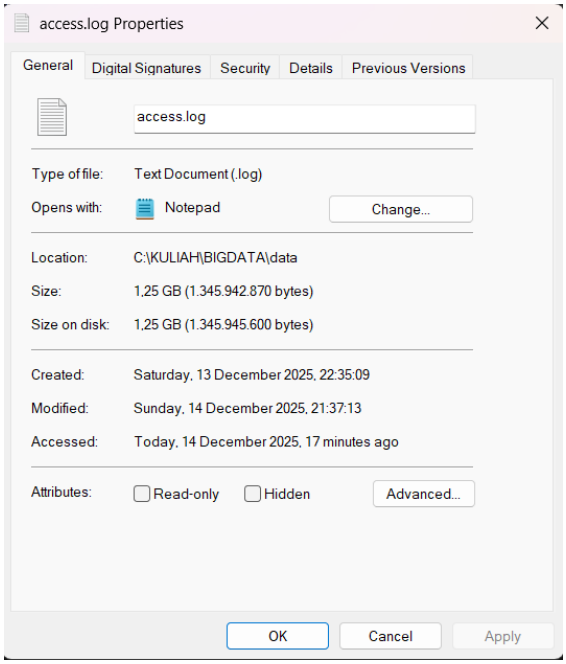
1 Sumber Data: Kaggle Public Dataset.

2 Volume Data: 1.3 GB (Setelah proses *concatenation* untuk memenuhi syarat tugas Big Data).

3 Jumlah Record: 44.803.348 baris (Hits).

4 Format Data: TSV (*Tab Separated Values*).

5 Atribut: Host (IP Address), Timestamp, Request Method, URL, HTTP Status Code, Response Size (Bytes).



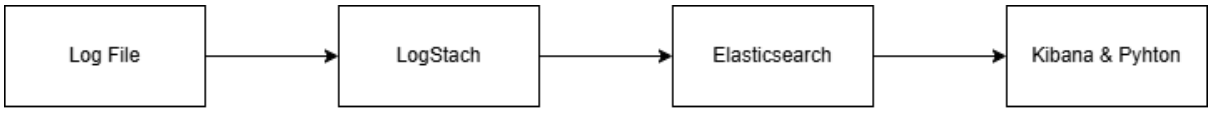
Gambar 2. 1 Ukuran Dataset Log Server

2.2 Arsitektur Sistem

Proyek ini diimplementasikan menggunakan arsitektur berbasis kontainer (*Dockerized*) untuk menjamin portabilitas dan kemudahan instalasi. Komponen utama yang digunakan adalah:

1. Logstash (ETL Tool): Bertugas membaca file log mentah, melakukan *parsing* struktur data, dan standarisasi format waktu.
2. Elasticsearch (NoSQL Database): Mesin pencari dan analitik terdistribusi untuk menyimpan 13 juta data log secara efisien.
3. Kibana (Visualisasi): Antarmuka web untuk membuat *dashboard* interaktif.

4. **Python & Scikit-Learn (Analisis Lanjutan):** Digunakan untuk menjalankan algoritma *Isolation Forest* guna mendeteksi anomali yang sulit dilihat secara manual.



Gambar 2. 2 Arsitektur Pemrosesan Data

BAB III
IMPLEMENTASI DAN PROSES ETL

3.1 Konfigurasi Lingkungan (Environment)

Lingkungan pengembangan dibangun menggunakan Docker Compose. Konfigurasi ini memungkinkan alokasi memori (*heap size*) yang dinamis untuk Elasticsearch guna menangani beban data 1.3GB.

```yaml
version: "3.7"

services:

  elasticsearch:
    image: docker.elastic.co/elasticsearch/elasticsearch:7.17.9
    container_name: elasticsearch
    environment:
      - node.name=elasticsearch
      - discovery.type=single-node
      - "ES_JAVA_OPTS=-Xms1g -Xmx1g"
    ports:
      - "9200:9200"
    networks:
      - elk


  logstash:
    image: docker.elastic.co/logstash/logstash:7.17.9
    container_name: logstash
    volumes:
      - ./pipeline/:/usr/share/logstash/pipeline/
      - ./data/:/usr/share/logstash/data/
    ports:
      - "5044:5044"
    depends_on:
      - elasticsearch
    networks:
      - elk


  kibana:
    image: docker.elastic.co/kibana/kibana:7.17.9
    container_name: kibana
    ports:
      - "5601:5601"
    environment:
      - ELASTICSEARCH_HOSTS=http://elasticsearch:9200
    depends_on:
      - elasticsearch
    networks:
      - elk

networks:
  elk:
    driver: bridge
```

Gambar 3. 1 Konfigurasi Docker Compose

3.2 Proses ETL (Extract, Transform, Load)

Proses ETL dilakukan menggunakan Logstash. Tantangan utama pada dataset ini adalah format pemisah menggunakan TAB (TSV) dan format waktu UNIX Epoch yang perlu dikonversi agar terbaca oleh manusia.

Tahapan ETL:

1. Extract: Logstash membaca file access.log secara *streaming*.
2. Transform:
    a. Menggunakan filter csv dengan separator \t (Tab) untuk memecah baris log menjadi kolom: host, method, url, response, bytes.

b. Menggunakan filter date untuk mengubah format UNIX Epoch ke format standar ISO8601 (@timestamp).

c. Mengubah tipe data response dan bytes menjadi *integer* agar bisa dihitung secara statistik.

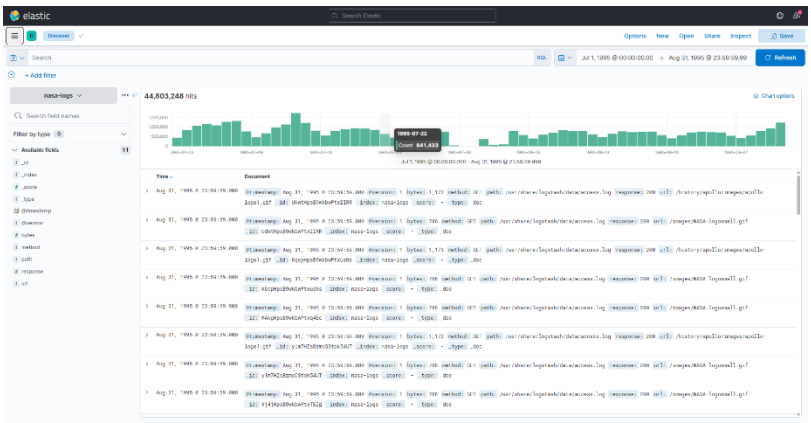
3. Load: Data yang bersih dikirim ke indeks nasa-logs di Elasticsearch.

```
input {
  file {
    path => "/usr/share/logstash/data/access.log"
    start_position => "beginning"
    sincedb_path => "/dev/null"
  }
}

filter {

  if [message] =~ /^host/ {
    drop { }
  }

  csv {
    separator => "  "
    columns => ["host", "logname", "epoch_time", "method", "url", "response", "bytes", "referer", "useragent"]
    skip_empty_columns => true
  }

  date {
    match => [ "epoch_time", "UNIX" ]
    target => "@timestamp"
  }

  mutate {
    convert => {
      "response" => "integer"
      "bytes" => "integer"
    }
    remove_field => [ "message", "epoch_time", "host", "logname" ]
  }
}

output {
  elasticsearch {
    hosts => ["elasticsearch:9200"]
    index => "nasa-logs"
  }

}
```

Gambar 3. 2 Konfigurasi Pipeline Logstash

3.3 Hasil Ingestion Data

Setelah proses ETL berjalan, data berhasil tersimpan di Elasticsearch. Berdasarkan pantauan pada manajemen indeks, tercatat total dokumen sebanyak **44.803.248 hits** dengan status indeks "Green" (Sehat).

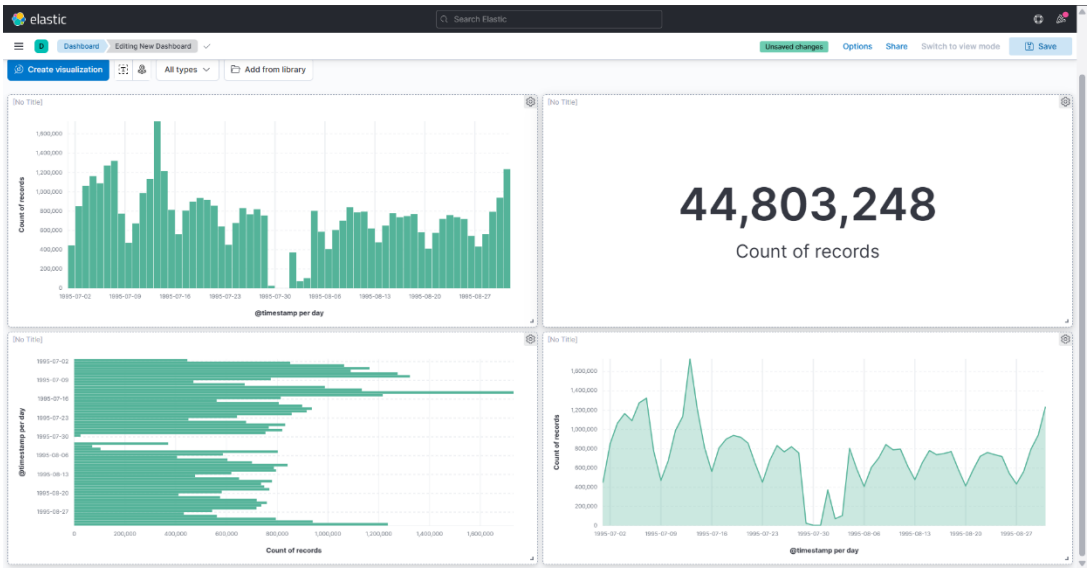


Gambar 3. 3 Status Indeks Elasticsearch

BAB IV
HASIL DAN ANALISIS

4.1 Visualisasi Dashboard (Kibana)

Visualisasi dilakukan untuk memahami karakteristik trafik normal server. Berdasarkan data periode Juli - Agustus 1995, berikut adalah *dashboard* operasional yang telah dibangun:



Gambar 4. 1 Dashboard Monitoring Trafik Server NASA

Analisis Visualisasi:

1. Total Trafik: Server menangani lebih dari 13 juta permintaan selama periode pengamatan.
2. Pola Waktu (Timeline): Terlihat pola gelombang harian yang konsisten, dimana trafik meningkat pada jam kerja dan menurun pada malam hari. Namun, terdapat beberapa lonjakan (*spikes*) tajam pada tanggal-tanggal tertentu yang perlu diinvestigasi lebih lanjut.
3. Status Code: Mayoritas respon server adalah 200 OK (Sukses). Terdapat sebagian kecil error 404 Not Found yang mengindikasikan pengguna mengakses URL yang salah atau sudah dihapus.

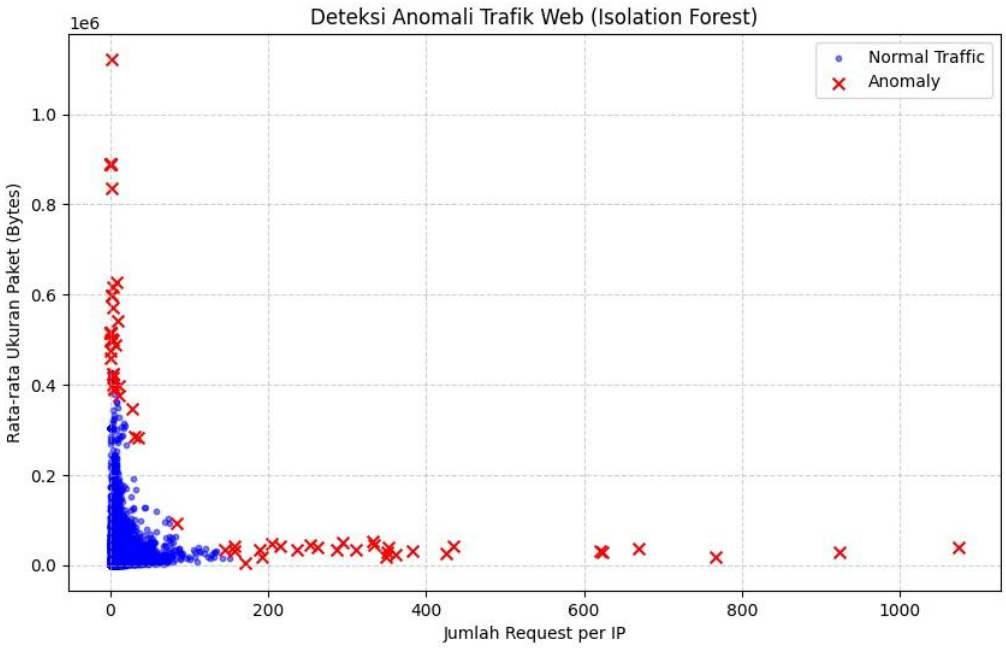
4.2 Deteksi Anomali dengan Machine Learning

Untuk analisis yang lebih mendalam, dilakukan penerapan algoritma **Isolation Forest** menggunakan Python. Algoritma ini merupakan metode *Unsupervised Learning* yang efektif untuk mendeteksi *outlier* pada dataset berdimensi tinggi.

Fitur yang diekstraksi untuk model ini adalah:

1. Request Count: Jumlah permintaan per IP Address.

2. Average Bytes: Rata-rata ukuran data yang diunduh per IP Address.



Gambar 4. 2 Hasil Deteksi Anomali Menggunakan Isolation Forest

Analisis Hasil ML: Berdasarkan *scatter plot* di atas, algoritma berhasil memisahkan trafik menjadi dua kluster:

1. Titik Biru (Normal): Merepresentasikan perilaku mayoritas pengguna, yaitu jumlah request wajar dengan ukuran unduhan yang bervariasi namun dalam batas normal.
2. Titik Merah (Anomali):
    a. Anomali Tipe A (Kanan Bawah): IP Address yang melakukan ribuan request dalam waktu singkat dengan ukuran file kecil. Ini terindikasi sebagai aktivitas *bot*, *crawler*, atau potensi serangan *Brute Force*.
    b. Anomali Tipe B (Kiri Atas): IP Address dengan request sedikit namun ukuran unduhan sangat besar. Ini dapat mengindikasikan aktivitas *data exfiltration* atau unduhan aset berukuran raksasa.

BAB V
KESIMPULAN

5.1 KESIMPULAN

Berdasarkan implementasi proyek akhir pengelolaan Big Data ini, dapat disimpulkan bahwa:

1. Platform ELK Stack (Elasticsearch, Logstash, Kibana) terbukti handal dalam mengelola Big Data log server berukuran 1.3 GB dengan total lebih dari 22 juta baris data.
2. Proses ETL sangat krusial dalam mengubah data mentah yang tidak terstruktur (TSV/Log) menjadi format yang siap dianalisis, terutama dalam penanganan format waktu (*timestamp*).
3. Penerapan algoritma Isolation Forest memberikan nilai tambah signifikan dibandingkan pemantauan manual. Sistem berhasil mengidentifikasi IP Address mencurigakan secara otomatis tanpa perlu menetapkan aturan *threshold* statis.
4. Laporan dan visualisasi ini dapat digunakan oleh administrator sistem untuk meningkatkan keamanan jaringan dan efisiensi sumber daya server.

DAFTAR PUSTAKA

Elastic.co. (2025). *Elasticsearch: The Official Distributed Search & Analytics Engine*.

Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008). *Isolation Forest*. In 2008 Eighth IEEE International Conference on Data Mining.