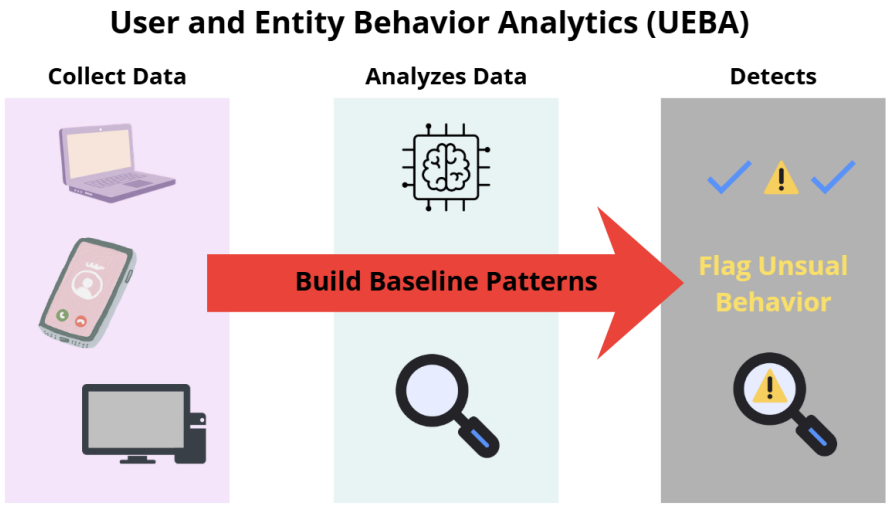
BAB 2
TINJAUAN PUSTAKA

2.1 Landasan Teori

Dasar teori yang mendukung implementasi metode User and Entity Behavior Analytics (UEBA) melibatkan konsep – konsep dasar seperti konsep keamanan jaringan, konsep dan metode deteksi anomali menggunakan UEBA, serta cara kerja dan penerapan Apache Spot dalam menganalisis perilaku jaringan. Berikut adalah landasan teori yang mendukung, antara lain:

2.1.1 User and Entity Behavior Analytics (UEBA)

User and Entity Behavior Analytics (UEBA) adalah pendekatan keamanan siber yang memfokuskan analisis pada perilaku individu (user) dan entitas (seperti perangkat, aplikasi, atau layanan) di dalam jaringan. Dengan memanfaatkan data log dan aktivitas yang terjadi, UEBA membangun profil perilaku normal yang dinamis untuk tiap pengguna dan entitas. Sistem ini kemudian memantau penyimpangan dari profil tersebut sebagai indikasi potensi ancaman. UEBA efektif dalam mendeteksi berbagai jenis serangan siber seperti insider threat, akses tidak sah, dan aktivitas malware, yang seringkali tidak terdeteksi oleh metode keamanan tradisional yang hanya mengandalkan signature-based detection (Desetty, 2024).



Gambar 2.1.1 User and Entity Behaviour (UEBA)

2.1.2 Machine Learning dalam Mendeteksi Anomali

Anomali pada lalu lintas jaringan merupakan indikasi adanya aktivitas yang tidak biasa atau berbahaya yang berbeda dari pola normal. Jenis-jenis anomali termasuk scanning port yang dilakukan oleh peretas untuk mencari celah keamanan, serangan Distributed Denial of Service (DDoS) yang membanjiri jaringan dengan trafik berlebihan, aktivitas malware yang mencoba menyebar, serta kebocoran data yang tidak sah. Deteksi dini anomali sangat penting untuk mencegah kerusakan dan menjaga ketersediaan, integritas, serta kerahasiaan data dalam jaringan (Galtsev & Sukhov, 2011).

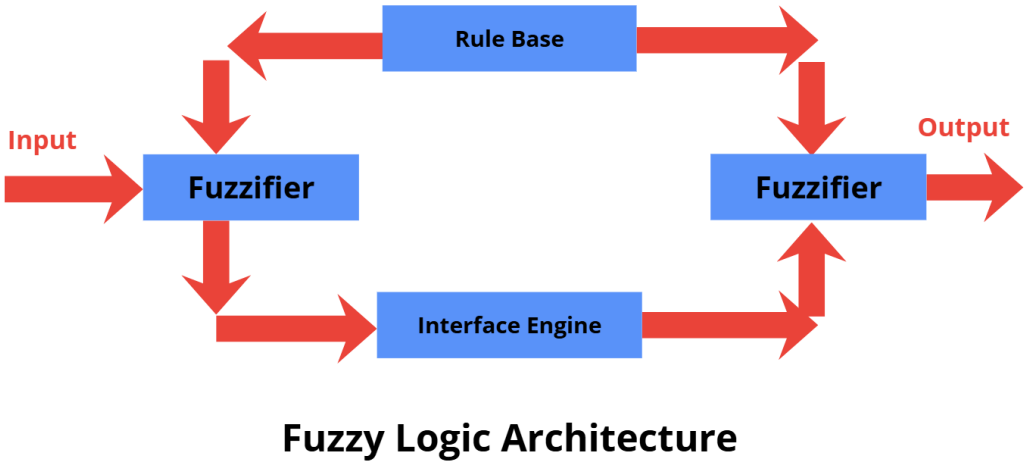
Machine learning dalam deteksi anomali memungkinkan sistem untuk belajar pola perilaku normal dari data yang ada dan mengenali penyimpangan yang mungkin menandakan ancaman. Pendekatan unsupervised learning sangat dominan karena data anomali jarang tersedia dalam bentuk berlabel. Metode seperti clustering, Principal Component Analysis (PCA), dan Latent Dirichlet Allocation (LDA) membantu mengelompokkan data dan mendeteksi outlier. Penggunaan machine learning meningkatkan kemampuan adaptasi sistem dalam menghadapi jenis ancaman baru yang belum dikenal sebelumnya (Ahmed et al., 2016).

Data aktivitas jaringan dan perilaku pengguna bersifat temporal, dimana waktu kejadian sangat penting untuk analisis. Time series analysis membantu memetakan pola perilaku sepanjang waktu, mendeteksi tren, musiman, atau lonjakan aktivitas yang tidak biasa. Analisis ini memungkinkan sistem UEBA untuk mengidentifikasi anomali yang bersifat temporal, seperti lonjakan trafik tiba-tiba atau perubahan pola akses yang terjadi dalam periode tertentu, yang seringkali merupakan tanda awal serangan atau kebocoran data (Ahmed et al., 2016).

2.1.3 Fuzzy Logic

Fuzzy Logic merupakan suatu bentuk logika bernilai banyak (many-valued logic) di mana nilai asli suatu variabel dapat berupa bilangan real antara 0 dan 1. Berbeda dengan logika Boolean klasik yang hanya berpatokan pada nilai mutlak benar (1) atau salah (0), logika fuzzy mampu menangani konsep kebenaran parsial (Zadeh, 1965). Konsep ini diimplementasikan untuk merepresentasikan ketidakpastian yang sering ditemui dalam sistem di dunia nyata. Dalam sistem

pakar dan Machine Learning, arsitektur Fuzzy Logic umumnya terdiri dari tiga tahapan utama: fuzifikasi (fuzzification), mesin inferensi (inference engine) yang menerapkan aturan If-Then, serta defuzifikasi (defuzzification). Pada konteks User and Entity Behavior Analytics (UEBA), logika ini sangat efektif untuk mengklasifikasikan tingkat ancaman jaringan yang bersifat ambigu menjadi kategori yang lebih terstruktur (misalnya: ancaman rendah, menengah, atau kritis) berdasarkan aturan kepakaran keamanan jaringan (Kaur & Vashisht, 2013).



Gambar 2.2.3 Diagram Fuzzy Logic

2.1.4 Apache Spot

Apache Spot adalah proyek open-source yang menyediakan platform analitik keamanan siber berbasis big data. Platform ini menggunakan ekosistem teknologi Apache seperti Hadoop untuk penyimpanan data besar, Apache Spark untuk pemrosesan cepat, dan Elasticsearch untuk indeksasi dan pencarian data. Apache Spot mengimplementasikan algoritma machine learning, terutama unsupervised learning, untuk menganalisis data lalu lintas jaringan dan mendeteksi anomali tanpa membutuhkan data berlabel. Dengan pipeline yang komprehensif mulai dari pengumpulan data, preprocessing, analitik hingga visualisasi, Apache Spot memungkinkan organisasi untuk mengidentifikasi ancaman secara proaktif dan responsif (Towle, n.d.)

Perlu dicatat bahwa proyek Apache Spot telah memasuki status retired sejak tahun 2019 dan tidak lagi dikembangkan secara aktif oleh Apache Software

Foundation. Namun, prinsip-prinsip arsitektur yang diperkenalkan oleh Apache Spot yaitu integrasi antara penyimpanan terdistribusi (HDFS), pemrosesan paralel (Spark), dan visualisasi interaktif (Kibana) untuk analisis keamanan jaringan tetap menjadi referensi yang relevan dan banyak diadopsi dalam implementasi sistem keamanan modern. Penelitian ini mengadopsi konsep arsitektur Apache Spot sebagai blueprint, bukan menggunakan perangkat lunaknya secara langsung.

2.1.5 Python

Python adalah bahasa pemrograman tingkat tinggi yang serbaguna, ditafsirkan (interpreted), dan bersifat open-source. Diciptakan oleh Guido van Rossum pada tahun 1991, Python didesain dengan filosofi yang menekankan pada keterbacaan kode melalui penggunaan spasi putih (whitespace) yang signifikan. Sintaksnya yang bersih dan sederhana membuatnya mudah dipelajari, bahkan oleh pemula (Cutting & Stephen, 2021)

2.1.6 Docker

Docker adalah platform perangkat lunak open-source yang digunakan untuk mengemas aplikasi beserta seluruh dependensinya ke dalam sebuah unit standar yang disebut kontainer. Analogi terbaiknya adalah kontainer pengiriman barang, di mana semua barang yang dibutuhkan sudah dimasukkan ke dalam satu kotak yang dapat diangkut oleh kapal, kereta, atau truk apa pun tanpa perlu khawatir isinya rusak atau tidak kompatibel (Boettiger, 2015).

2.1.7 Dataset CESNET-timeseries2024

CESNET-timeseries2024 merupakan sebuah dataset publik berskala besar yang berisi data lalu lintas jaringan dunia nyata yang dikumpulkan selama 40 minggu pada tahun 2024. Dataset ini dirilis oleh CESNET atau bisa disebut juga Jaringan Riset dan Edukasi Nasional Ceko. Dataset ini dirancang khusus untuk penelitian di bidang keamanan siber, terutama penelitian di bidang keamanan siber. Dataset ini awalnya masih mentah yang mencakup 66 miliar alur data (IP flows) dan 4 triliun paket data dengan total volume sekitar 3.7 Petabyte. Namun, setelah di agregasi berdasarkan waktu (contoh per 10 menit, per jam, dan per hari), data ini mencakup sekitar 275.000 alamat IP yang aktif dalam waktu kurun 40 minggu dengan volume 41.5 Gigabyte, dataset inilah yang akan digunakan pada penelitian

ini.Secara sederhana, ini adalah rekaman aktivitas jaringan yang sangat besar dan detail yang sengaja dibuat untuk membantu para peneliti menguji seberapa baik algoritma mereka dalam menemukan hal-hal aneh pada jaringan untuk mendeteksi anomali dan peramalan (forecasting) lalu lintas jaringan (Koumar et al., 2025).

2.1.8 Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) merupakan algoritma machine learning yang digunakan untuk mencari hal-hal yang tersembunyi dari sekumpulan data. Latent Dirichlet Allocation (LDA) bekerja dengan cara mengelompokan data secara otomatis berdasarkan pola atau fitur secara bersamaan tanpa harus memahami konteks dari data tersebut. Hal ini membuat LDA termasuk ke dalam kategori Unsupervised Learning. Unsupervised Learning berarti model tidak menggunakan label melainkan belajar langsung dari data untuk menemukan pola-pola yang dominan. Setelah ditemukan pola-pola atau fitur dari data, anomali atau outlier kemudian diidentifikasi dengan pola yang paling tidak sesuai atau paling tidak dominan dibandingkan dengan pola yang normal. Hal ini membuat algoritma LDA sangat cocok dengan tujuan penelitian ini yaitu mencari anomali dari sekumpulan log jaringan yang bertumpuk tanpa diketahui apa aktivitasnya (Campbell et al., 2015). Secara matematis, LDA mengasumsikan bahwa setiap dokumen dihasilkan melalui proses generatif berikut:

1. Untuk setiap topik k, distribusi kata ϕk diambil dari distribusi Dirichlet dengan parameter β: $\phi k \sim Dir(\beta)$
2. Untuk setiap dokumen d, distribusi topik θd diambil dari distribusi Dirichlet dengan parameter α: $\theta d \sim Dir(\alpha)$
3. Untuk setiap kata ke-n dalam dokumen d:
    - Topik Zd,n diambil dari distribusi multinomial: $Zd,n \sim Multinomial(\theta d)$
    - Topik Wd,n diambil dari distribusi multinomial: $Wd,n \sim Multinomial(\phi Zd,n)$

Dalam konteks deteksi anomali jaringan, "dokumen" adalah profil perilaku satu entitas, "kata" adalah kombinasi fitur diskrit, dan "topik" adalah pola perilaku laten. Setelah model dilatih, setiap data baru dievaluasi berdasarkan distribusi topiknya. Data yang tidak sesuai dengan topik manapun (probabilitas tersebar

merata) diidentifikasi sebagai anomali (Blei, 2012).

2.1.9 Quantile Discretizer

Quantile Discretizer adalah teknik pra-pemrosesan data yang mengubah fitur kontinu menjadi fitur diskrit (kategorikal) berdasarkan distribusi kuantil data. Berbeda dengan metode diskritisasi berbasis interval sama (equal-width binning) yang rentan terhadap outlier, diskritisasi berbasis kuantil (equal-frequency binning) membagi data ke dalam sejumlah bucket di mana setiap bucket mengandung jumlah data yang kurang lebih sama.

Dalam implementasi Apache Spark MLlib, Quantile Discretizer menerima parameter numBuckets (jumlah bucket) dan relativeError (toleransi kesalahan dalam perhitungan kuantil). Nilai relativeError yang lebih kecil menghasilkan batas kuantil yang lebih presisi namun membutuhkan waktu komputasi yang lebih lama.

Diskritisasi dipilih dalam penelitian ini karena algoritma LDA memerlukan input dalam bentuk token kategorikal, bukan nilai numerik kontinu. Dengan mengubah fitur jaringan menjadi level diskrit (level_1 hingga level_10), setiap kombinasi fitur dapat direpresentasikan sebagai "kata" dalam pendekatan Bag-of-Words (Zaharia et al., 2016).

2.1.10 Bag-of-Words dan Feature Crossing

Bag-of-Words (BoW) adalah representasi teks yang berasal dari bidang Natural Language Processing (NLP), di mana sebuah dokumen direpresentasikan sebagai kumpulan kata (bag) tanpa memperhatikan urutan atau tata bahasa, melainkan hanya frekuensi kemunculan setiap kata (Zhang et al., n.d.).

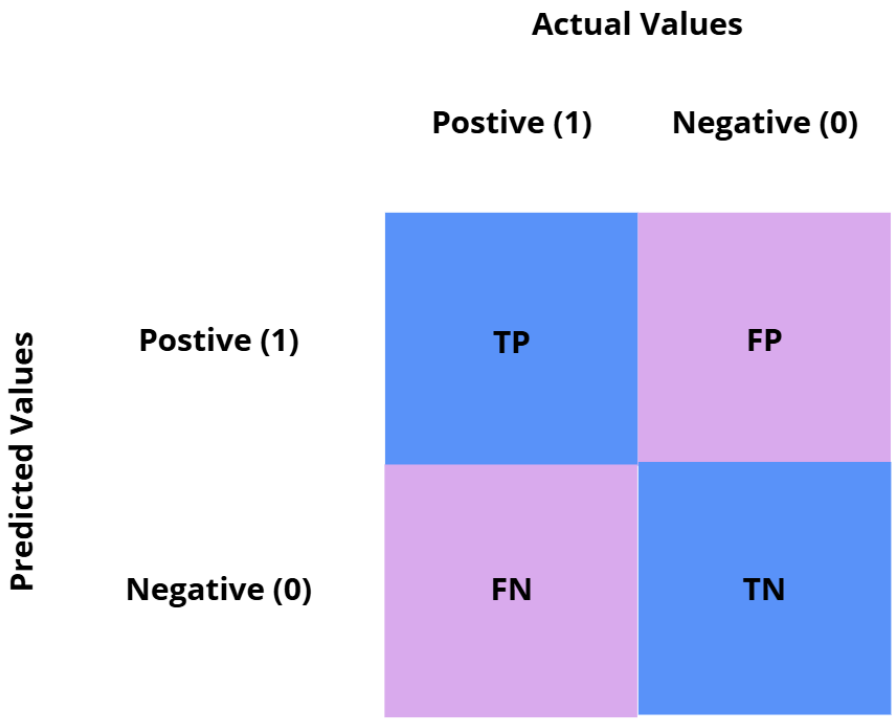
Dalam konteks penelitian ini, pendekatan BoW diadaptasi untuk merepresentasikan profil perilaku jaringan. Setiap baris data jaringan dianggap sebagai satu "dokumen", dan kombinasi fitur diskrit (hasil diskritisasi) dianggap sebagai "kata". Sebagai contoh, kombinasi “level_4||level_7||level_2||level_5||level_3” merepresentasikan satu profil perilaku spesifik dari suatu entitas jaringan.

Pendekatan ini memungkinkan algoritma LDA — yang awalnya dirancang untuk analisis topik pada dokumen teks — diterapkan pada data jaringan dengan memperlakukan setiap profil perilaku sebagai "topik" laten yang dapat dipelajari

secara unsupervised. Representasi BoW kemudian dikonversi menjadi vektor numerik menggunakan CountVectorizer, yang memetakan setiap token unik ke indeks dalam vektor sparse berdasarkan frekuensi kemunculannya.

2.1.11 Confusion Matrix

Matriks Konfusi (Confusion Matrix) merupakan instrumen evaluasi fundamental yang digunakan untuk mengukur dan memvisualisasikan kinerja suatu model klasifikasi. Matriks ini menyajikan perbandingan langsung antara nilai kebenaran aktual (Ground Truth) dengan hasil prediksi yang dihasilkan oleh algoritma (Han et al., 2011). Pada skenario klasifikasi biner, matriks konfusi terdiri dari empat metrik utama: True Positive (TP) yang merepresentasikan deteksi anomali yang benar, True Negative (TN) untuk deteksi trafik normal yang tepat, False Positive (FP) yang mengindikasikan alarm palsu, serta False Negative (FN) yang terjadi ketika sistem gagal mendeteksi serangan aktual. Berdasarkan keempat metrik dasar ini, berbagai parameter evaluasi tingkat lanjut seperti Akurasi (Accuracy), Presisi (Precision), dan Recall (Sensitivitas) dapat dihitung untuk mengukur efektivitas sistem deteksi intrusi secara komprehensif (Fawcett, 2006).



Gambar 2.1.9 Tabel Confusion Matrix

2.1.12 Isolation Forest

Isolation Forest (iForest) adalah algoritma unsupervised learning yang dirancang secara spesifik untuk mendeteksi anomali. Diperkenalkan oleh Liu, Ting, dan Zhou (2008), algoritma ini mendobrak pendekatan konvensional yang memprofilkan titik data normal terlebih dahulu. Sebaliknya, Isolation Forest mengisolasi anomali secara langsung berdasarkan dua sifat alaminya: jumlahnya yang minoritas dan atributnya yang sangat menyimpang dari data normal. Algoritma ini bekerja dengan membangun sekumpulan pohon keputusan acak (random decision trees). Karena anomali memiliki nilai yang ekstrem, data ini memerlukan proses pembagian (splitting) yang jauh lebih sedikit untuk dapat dipisahkan dari sekumpulan data lainnya. Oleh karena itu, titik data yang memiliki jarak lintasan (path length) sangat pendek dari akar (root) hingga ke daun terminal (leaf node) akan diidentifikasi sebagai ancaman atau anomali (Liu et al., 2012).

2.1.13 Apache Kafka

Apache Kafka adalah platform distributed event streaming yang dirancang untuk menangani aliran data secara real-time dengan throughput tinggi dan latensi rendah. Awalnya dikembangkan oleh LinkedIn dan kemudian menjadi proyek open-source di bawah Apache Software Foundation, Kafka telah menjadi standar industri untuk arsitektur event-driven dan stream processing (Fragkoulis et al., 2024).

Arsitektur Kafka terdiri dari tiga komponen utama:

1. Producer

   Aplikasi yang mengirim pesan (message) ke Kafka. Dalam konteks penelitian ini, Kafka Producer membaca file CSV data testing dan mengirim setiap baris sebagai pesan JSON ke Kafka Topic.
2. Broker

   Server Kafka yang menyimpan dan mendistribusikan pesan. Pesan disimpan dalam struktur yang disebut Topic, yang merupakan kategori logis untuk mengelompokkan pesan. Setiap Topic dapat memiliki satu atau lebih Partition untuk meningkatkan paralelisme.
3. Consumer

Aplikasi yang membaca pesan dari Kafka Topic. Kafka menggunakan mekanisme Consumer Group dan Offset Tracking untuk menjamin bahwa setiap pesan diproses tepat sekali (exactly-once) atau setidaknya sekali (at-least-once), bahkan jika consumer mengalami kegagalan.

Keunggulan utama Kafka dibandingkan message broker tradisional (seperti RabbitMQ) adalah kemampuannya untuk menyimpan pesan dalam jangka waktu tertentu (retention period), sehingga pesan tidak hilang meskipun consumer sedang tidak aktif. Dalam konteks sistem UEBA, properti ini sangat penting untuk menjamin tidak ada data log jaringan yang hilang selama proses recovery atau downtime sistem.

2.1.14 Elasticsearch

Elasticsearch adalah search engine dan analytics engine terdistribusi berbasis Apache Lucene yang dirancang untuk menyimpan, mencari, dan menganalisis data dalam skala besar secara near real-time. Elasticsearch menggunakan struktur data inverted index yang memungkinkan pencarian teks sangat cepat, serta mendukung penyimpanan data dalam format JSON document yang fleksibel dan schema-free (Elastic N.V., 2024).

Dalam konteks penelitian ini, Elasticsearch berfungsi sebagai indexed storage yang menyimpan hasil deteksi anomali dari Apache Spark. Setiap kejadian anomali disimpan sebagai satu dokumen JSON dalam sebuah Index (analogi dari tabel dalam database relasional). Data yang tersimpan di Elasticsearch kemudian divisualisasikan melalui Kibana.

2.1.15 Apache Hadoop

Apache Hadoop adalah sebuah kerangka kerja open-source yang menjadi fondasi ekosistem big data, dirancang untuk memungkinkan penyimpanan dan pemrosesan data dalam skala masif secara terdistribusi di atas klaster komputer. Hadoop memiliki dua komponen inti: HDFS (Hadoop Distributed File System), yaitu sistem file yang andal dan toleran terhadap kegagalan (fault-tolerant) untuk menyimpan data berukuran besar, dan MapReduce, yaitu model pemrograman untuk memproses data tersebut secara paralel. Dalam konteks arsitektur penelitian ini, HDFS memegang peranan krusial sebagai fondasi penyimpanan untuk

menampung dataset CESNET-timeseries2024. Sementara itu, peran pemrosesan oleh MapReduce akan digantikan oleh Apache Spark yang lebih cepat dan fleksibel, yang akan membaca data langsung dari HDFS untuk dianalisis (Shvachko et al., 2010).

2.1.16 Synthetic Anomaly Injection

Synthetic Anomaly Injection (Injeksi Anomali Buatan) adalah sebuah metodologi evaluasi yang sangat krusial dalam eksperimen keamanan siber, khususnya ketika dataset otentik yang digunakan bersifat unsupervised dan tidak memiliki label Ground Truth (Shiravi et al., 2012). Dalam analisis log lalu lintas jaringan yang masif, melacak lokasi pasti dari suatu serangan tanpa proses audit manual hampir mustahil dilakukan. Oleh karena itu, evaluasi model anomali dilakukan dengan menyisipkan baris-baris data yang telah dimodifikasi secara sengaja untuk merepresentasikan skenario serangan siber. Melalui pendekatan ini, posisi, waktu, dan jenis serangan diketahui secara mutlak oleh penguji. Keberhasilan model deteksi kemudian diukur dari kemampuannya untuk mengidentifikasi titik data buatan tersebut di tengah variasi trafik jaringan normal. Metodologi ini memastikan bahwa evaluasi algoritma berjalan secara objektif dan matematis (García et al., 2014).

2.1.17 Big Data dan Apache Spark

Big data merujuk pada volume data yang sangat besar, bervariasi, dan terus bertambah dengan cepat, seperti data lalu lintas jaringan. Apache Spark adalah platform pemrosesan big data yang dirancang untuk melakukan analisis secara cepat dan efisien dengan memanfaatkan kemampuan distributed computing. Dalam Apache Spot, Spark digunakan untuk menjalankan pipeline analitik yang kompleks, mulai dari preprocessing data, ekstraksi fitur, hingga penerapan algoritma machine learning secara paralel, sehingga mampu menangani data dalam jumlah besar dengan performa tinggi (Zaharia et al., 2010).

2.1.18 Streamlit

Streamlit adalah framework Python open-source yang dirancang untuk membangun aplikasi web interaktif dan dashboard data dengan cepat, tanpa memerlukan pengetahuan mendalam tentang pengembangan web frontend seperti

HTML, CSS, atau JavaScript. Streamlit memungkinkan pengembang untuk mengubah skrip Python menjadi aplikasi web yang dapat dijalankan di browser hanya dengan beberapa baris kode, sehingga sangat cocok untuk keperluan prototyping, visualisasi data, dan aplikasi machine learning (Streamlit Inc., 2024).

Dalam konteks penelitian ini, Streamlit dipilih sebagai Lapisan Antarmuka (Interface Layer) karena beberapa alasan. Pertama, seluruh pipeline pemrosesan data dalam penelitian ini dibangun menggunakan Python (PySpark, kafka-python, dan scikit-learn), sehingga Streamlit memungkinkan integrasi langsung antara antarmuka pengguna dengan logika pemrosesan tanpa perlu membangun API terpisah. Kedua, Streamlit menyediakan komponen interaktif bawaan seperti file uploader, tombol, dan iframe yang dibutuhkan untuk mengimplementasikan tiga modul utama sistem. Ketiga, Streamlit mendukung penyematan (embedding) dashboard eksternal melalui iframe, yang dimanfaatkan untuk menampilkan dashboard Kibana secara langsung di dalam antarmuka sistem.

Antarmuka sistem UEBA yang dibangun dengan Streamlit terdiri dari tiga tab utama, yaitu: (1) Tab Training Model untuk mengunggah data dan melatih model LDA, (2) Tab Inference untuk menjalankan proses inferensi real-time melalui Kafka, dan (3) Tab Dashboard untuk menampilkan visualisasi hasil deteksi anomali dari Kibana.

2.1.19 Kibana untuk Visualisasi Data

Kibana adalah alat visualisasi data open-source yang digunakan bersama Elasticsearch untuk menyajikan data dalam bentuk dashboard yang interaktif dan mudah dipahami. Dalam implementasi Apache Spot, Kibana memungkinkan analis keamanan untuk melihat data yang telah diproses secara real-time, memvisualisasikan pola dan anomali, serta melakukan eksplorasi data menggunakan berbagai grafik, diagram, dan peta panas. Visualisasi ini sangat membantu dalam mempercepat identifikasi ancaman serta mengambil keputusan keamanan yang tepat (Subhadra, 2020).

2.2 Penelitian Terdahulu

Penelitian oleh Komisarek et al. (2021) dalam studi mereka yang berjudul Machine Learning Based Approach to Anomaly and Cyberattack Detection in

Streamed Network Traffic Data mengembangkan pendekatan deteksi anomali dan serangan siber secara real-time dengan memanfaatkan pipeline berbasis big data. Dalam penelitian ini, data lalu lintas jaringan diproses menggunakan kombinasi teknologi seperti Apache Kafka untuk streaming, Elasticsearch untuk indexing data, serta Kibana untuk visualisasi. Sistem ini kemudian dikombinasikan dengan model pembelajaran mesin (menggunakan TensorFlow) untuk mengidentifikasi pola anomali dalam data yang masuk secara langsung. Meskipun tidak secara eksplisit menggunakan kerangka kerja Apache Spot, pendekatan yang mereka ajukan sangat relevan dengan prinsip kerja User and Entity Behavior Analytics (UEBA), terutama dari sisi arsitektur data pipeline dan integrasi visualisasi dengan Kibana untuk mendukung proses analitik perilaku secara efisien.

Penelitian oleh GÖRMEZ et al. (2023)dalam studi mereka yang berjudul A User and Entity Behavior Analysis for SIEM Systems: Preprocessing of The CERT Dataset membahas pentingnya tahap pra-pemrosesan data log untuk meningkatkan performa sistem UEBA dalam lingkungan SIEM. Mereka menggunakan data dari CERT dan menekankan pada teknik ekstraksi fitur untuk memaksimalkan deteksi perilaku mencurigakan pengguna secara akurat. Hasil penelitian ini menunjukkan bahwa proses pra-pemrosesan yang baik dapat meningkatkan efektivitas model deteksi dalam sistem keamanan berbasis analitik perilaku pengguna.

Penelitian oleh Tuyishime et al. (2023) melalui studi yang berjudul Enhancing Cloud Security—Proactive Threat Monitoring and Detection Using a SIEM-Based Approach menyajikan pendekatan penguatan keamanan siber di lingkungan cloud menggunakan arsitektur SIEM. Studi ini mengintegrasikan komponen seperti Web Application Firewall (WAF) dan Microsoft Defender, serta memanfaatkan visualisasi data keamanan untuk membantu proses monitoring ancaman secara proaktif. Meskipun tidak secara eksplisit menggunakan UEBA, pendekatan yang mereka ajukan memiliki relevansi erat terhadap integrasi analitik perilaku dalam sistem keamanan.

Penelitian oleh Rahmawati et al. (2024) dalam studi mereka yang berjudul Enhancing Network Security Through Real-Time Threat Detection with Intrusion Prevention System and ELK Stack mengusulkan integrasi antara Intrusion Prevention System (IPS) Suricata dan platform ELK Stack (Filebeat, Logstash,

Elasticsearch, dan Kibana). Sistem ini mampu mendeteksi ancaman secara real-time dan menyediakan visualisasi data serangan yang komprehensif melalui Kibana, sehingga sangat membantu analis dalam mengambil tindakan dengan cepat.

Penelitian saya (2025) berjudul Implementasi Sistem User and Entity Behavior Analytics (UEBA) menggunakan Apache Spot untuk mendeteksi anomali pada data lalu lintas jaringan dengan Dataset CESNET-timeseries2024 bertujuan merancang dan mengimplementasikan sistem UEBA berbasis Apache Spot yang mampu mendeteksi berbagai perilaku anomali pada jaringan menggunakan algoritma Latent Dirichlet Allocation (LDA) yang merupakan algoritma bawaan dari Apache Spot itu sendiri. Evaluasi sistem dilakukan secara fungsional dan kuantitatif menggunakan metrik performa serta pengujian visualisasi dashboard menggunakan Apache Kibana untuk memastikan efektivitas deteksi anomali.

Tabel 2.2.1 Perbandingan variabel penelitian

Penelitian (Tahun)	Fokus Utama	Metodologi	Teknologi	Dataset	Output
Komisarek et al. (2021)	Deteksi anomali & serangan siber secara *real-time*.	Big data streaming pipeline dengan machine learning.	Kafka, Elasticsearch, TensorFlow	Data lalu lintas jaringan streamed.	Kibana
GÖRMEZ et al. (2023)	Pentingnya pra-pemrosesan data untuk sistem UEBA.	Ekstraksi fitur dari data log untuk meningkatkan deteksi.	Sistem SIEM	CERT Dataset	Implisit di dalam dashboard SIEM.
Tuyishime et al. (2023)	Penguatan keamanan di lingkungan cloud.	Integrasi berbagai alat keamanan berbasis SIEM.	WAF, Microsoft Defender	Log dari lingkungan cloud.	Visualisasi data (alat tidak spesifik).

Rahmawati et al. (2024)	Deteksi ancaman real-time dengan integrasi IDS.	Menggabungkan IDS (Suricata) dengan platform logging.	Suricata, ELK Stack	Log lalu lintas jaringan.	Kibana
Penelitian Saat Ini (2025)	Implementasi sistem UEBA berbasis arsitektur open-source.	Mereplikasi arsitektur Apache Spot dengan unsupervised learning.	Apache Spark, HDFS, LDA, Apache Kibana, Elasticsearch, Apache Kafka	CESNET-timeseries2024	Kibana