# BAB II

# LANDASAN TEORI

## 2.1 Tinjauan Pustaka

### 2.1.1 DDoS dan Jenisnya

Menurut (Gupta & Dahiya, 2021) Serangan DDoS adalah serangan yang besar, terdistribusi, disengaja, dan terkoordinasi menggunakan beberapa mesin yang sudah diretas oleh hacker untuk menghambat layanan atau server tertentu. Penyerang mencoba menyerang layanan atau server jumlah data palsu yang besar untuk membuat target kekurangan sumber daya.

Serangan DDoS merupakan varian serangan DoS. Di mana perbedaannya terletak pada penggunaan sumber daya untuk menyerang target. DoS melibatkan penggunaan satu sumber daya serangan untuk menyerang target, sedangkan DDoS melibatkan penggunaan banyak sumber daya serangan yang terdistribusi secara luas untuk menyerang target.

Penyerang biasanya menciptakan jaringan mesin yang sudah diretas, yaitu botnet, dengan menyelipkan skrip jahat ke dalamnya secara diam-diam. Setelah mengambil sistem lain, penyerang mengirim spam, menyebarkan malware, dan cenderung menyerang sistem lain dengan memanfaatkan mesin yang telah diretas. Serangan DDoS merupakan serangan yang berdampak untuk bisnis online, beberapa menit saja mengalami down akan mengakibatkan kerugian finansial dan kehilangan reputasi.

Terdapat 3 jenis serangan DDoS, yaitu *voluminous attack* atau *flooding attack*, *protocol-based attack*, dan *application layer*.

#### 2.1.1.1 Voluminous Attack

*Voluminous attack* atau serangan berbasis volume merupakan serangan yang bertujuan untuk menghabiskan *bandwidth* target dengan mengirim data palsu dengan jumlah besar menggunakan komputer yang terdistribusi.

#### 2.1.1.2 Protocol-based

*Protocol-based attack* atau serangan berbasis protokol merupakan serangan yang memanfaatkan kerentanan pada protokol lapisan ke-3 dan 4. Contohnya adalah *TCP SYN* dan *ping of death attack*.

#### 2.1.1.3 Application Layer

Yang terakhir adalah *application layer* atau serangan pada lapisan OSI ketujuh atau lapisan aplikasi. Serangan ini dirancang untuk menghancurkan aplikasi web dan merupakan serangan yang lebih kompleks karena menguras jaringan dan sumber daya server secara bersamaan.

### 2.1.2 Metode Mendeteksi Serangan DDoS

Menurut (Kumar et al., 2016) terdapat 2 cara deteksi untuk serangan DDoS, yaitu dengan menggunakan *misuse detection* dan *anomaly-based detection*.

#### 2.1.2.1 Misuse Detection

*Misuse detection* menggunakan pendekatan mendefinisikan perilaku sistem abnormal dan perilaku lainnya sebagai normal. Metode ini bergantung pada pola serangan atau tanda-tanda yang dikenal untuk dideteksi.

##### 2.1.2.1.1 *Signatured-Based DDoS Detection*

*Signatured-based* atau skema deteksi berbasis tanda tangan meyimpan urutan pola dan *signature* serangan di dalam basis data. Ketika penyerang melakukan serangan atau terjadinya peretasan, IDS berusaha mengidentifikasi dengan mencocokkan dengan sebuah set tanda tangan serangan yang telah disimpan sebelumnya dalam basis data.

##### 2.1.2.1.2 *Rule-Based Detection*

Deteksi berbasis peraturan dibuat menggunakan sejumlah peraturan *if-else* yang dibuat dengan menganalisa serangan untuk mendeteksi penyalahgunaan dengan membandingkannya dengan data yang dipantau (biasanya *log*).

##### 2.1.2.1.3 *State-Transition Technique*

*State-Transition Technique* menggambarkan serangan sebagai sebuah urutan aktivitas yang menyebabkan peralihan dari sistem yang dipantau, menjadi keadaan peringatan jika serangan terdeteksi.

#### 2.1.2.2 Anomaly-Based Detection

Teknik deteksi berdasarkan anomali pertama membuat *normal behaviour* atau perilaku normal dari suatu subject, biasanya pengguna atau sistem. Jika

perilakunya meyimpang dari perilaku atau pola normal, ini dikenal sebagai anomali atau serangan.

#### 2.1.2.2.1 *Statistical Techniques*

Pendekatan secara statistik diawali dengan menetapkan perilaku normal dari pengguna berdasarkan apa yang diterima oleh penggunaan kebijakan sistem. Jika perilaku yang dipantau ditemukan menyimpang secara signifikan dari ambang batas perilaku normal yang sudah ditetapkan, hal ini dianggap aktivitas anomali atau sebuah serangan. Kebanyakan metode yang dirancang untuk mendeteksi anomali dari jaringan menggunakan berbagai macam perhitungan statistik seperti deviasi, jumlah kumulatif, korelasi, entropi, *mutual information*, dan kovarians.

#### 2.1.2.2.2 *Machine Learning dan Data Mining Techniques*

*Machine learning* dan *data mining* bermain peran yang penting dalam pengembangan mekanisme deteksi yang efisien untuk melindungi sumber daya dari serangan karena kemampuannya yang membantu sistem belajar tanpa secara jelas diprogram. Perkembangan cabang kecerdasan ini memungkin sistem menggunakan dua pendekatan yang berbeda, yaitu *supervised learning*, di mana pembelajaran algoritma menggunakan label data untuk memprediksi label kelas dari kejadian yang tidak diketahui. Akurasinya akan tinggi jika dilatih dengan data yang tepat namun akan sulit mendeteksi serangan yang tidak diketahui dan *unsupervised learning*, dapat mendeteksi kelompok kejadian serupa tanpa sepengetahuan sebelumnya. Dapat mendeteksi serangan yang tidak diketahui tetapi dapat mengakibatkan *false positive*. Ada pun pendekatan *hybrid* yang menggabungkan

*supervised* dan *unsupervised* untuk meningkatkan akurasi deteksi untuk serangan yang sudah diketahui maupun belum diketahui.

#### 2.1.2.2.3 *Soft Computing Techniques*

Prinsip dasar yang menggambarkan *soft computing* adalah mencapai solusi yang murah dan tangguh dengan memanfaatkan toleransi terhadap ketidaktepatan, ketidakpastian, ketidakjelasan, dan kebenaran parsial. Lima pendekatan utama dalam komputasi lunak adalah logika fuzzy, komputasi neural, komputasi evolusioner, pembelajaran mesin, dan penalaran probabilistik. Kemampuan metode *soft computing* telah terbukti dalam menyelesaikan masalah pencocokan pola kompleks dan kecerdasan mesin dalam banyak domain penerapan kehidupan nyata.

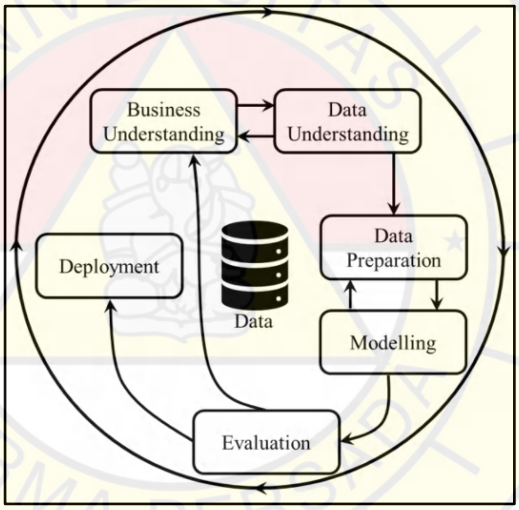
#### 2.1.2.2.4 *Knowledge-Based Techniques*

Dalam deteksi DDoS, pendekatan *knowledge-based* dapat mengidentifikasi kelas serangan DDoS yang sudah diketahui. Dalam pendekatan ini, menggunakan pengetahuan sebelumnya yang diperoleh dari riwayat serangan DDoS sebelumnya saat mengembangkan solusi pertahanan. membuat aturan atau *signature* untuk setiap jenis serangan yang diketahui, dan selama deteksi, keadaan lalu lintas jaringan dicocokkan dengan aturan atau *signature* yang telah ditentukan sebelumnya. Jika ada kecocokan, sistem akan memberikan peringatan, jika tidak, dianggap normal.

### 2.1.3 *Cross-Industry Standard Process for Data Mining* (CRISP-DM)

Untuk membangun model mendeteksi serangan DDoS berbasis *machine learning* tentu membutuhkan metode. Metode *data mining* yang digunakan untuk penelitian ini adalah *cross-industry standard process for data mining* (CRISP-DM).

Di tahun 2000, *cross-industry standard process for data mining* (CRISP-DM) 1.0 *data mining* diterbitkan oleh Chapman et al. (2000). Sudah lebih dari 20 tahun, namun versi terbaru dari CRISP-DM yang diharapkan belum juga diterbitkan. Namun, CRISP-DM masih menjadi panduan yang paling banyak digunakan untuk proyek *data mining*. CRISP-DM juga bisa dilihat sebagai panduan untuk melakukan proyek *data mining*. Terdiri dari enam fase: *business understanding*, *data understanding*, *data preparation*, *modelling*, *evaluation* dan yang terakhir adalah *deployment* (Garn, 2024). Di bawah ini pada gambar 2.1 merupakan gambaran umum CRISP-DM.




**Gambar 2. 1 Gambaran umum fase CRISP-DM (Garn, 2024)**

### 2.1.3.1 Understanding

#### 2.1.3.1.1 *Business Understanding*

Pada fase awal ini berfokus pada tujuan dan kebutuhan dari proyek dalam perspektif bisnis. Di tahap pertama ini terdapat 4 hal penting. Yang pertama adalah menentukan tujuan bisnis, yaitu memberikan latar belakang dan menentukan

kriteria keberhasilannya. Yang kedua adalah evaluasi terhadap ketersediaan sumber daya, persyaratan, hipotesis, dan kendala yang relevan dalam proyek. Selain itu, risiko yang mungkin timbul perlu diidentifikasi, bersama dengan langkah-langkah darurat yang akan diambil. Selain itu, penting untuk menyusun terminologi yang tepat, memperkirakan biaya, dan menganalisis manfaat. Yang ketiga fokus kepada tantangan-tantangan dalam data mining dan menentukan tujuan dari data mining dan kriteria keberhasilannya. Yang terakhir adalah membuat rencana proyek, yaitu beruapa alat-alat dan tekniknya.

#### 2.1.3.1.2 *Data Understanding*

Pada fase ini mencangkup langkah-langkah, 1.) pengumpulan data, mengumpulkan data yang relevan; 2.) mendeskripsikan data, jenis data yang ada, struktur data, serta atribut-atribut yang dimilikinya; 3.) melakukan eksplorasi data, menemukan pola atau insight yang mungkin tersembunyi di dalamnya; dan 4.) memvalidasi kualitas dari data, memastikan kualitas data yang terkumpul.

### 2.1.3.2 Preparation and Modeling

#### 2.1.3.2.1 *Data Preparation*

Fase ini bertujuan untuk menghasilkan dataset dengan deskripsi yang jelas. Ini melibatkan pemilihan data yang relevan, Proses pembersihan data harus didokumentasikan, dan pembentukan dataset yang sesuai. Selain itu, tahap ini juga mencakup integrasi data dari berbagai sumber, seperti penggabungan data atau restrukturisasi format data jika diperlukan.

#### 2.1.3.2.2 *Modeling*

Empat hal penting dalam fase ini adalah 1.) memilih teknik permodelan, pemilihan teknik dan memperjelas tentang hipotesis yang dibuat; 2.) membuat desain uji coba, bisa mencakup deskripsi dari eksperimen (misalnya, uji coba sistematis) atau bahkan desain eksperimental; 3.) membuat model, pembuatan model biasanya melibatkan eksplorasi beberapa metode, yang setiap metode dijalankan dengan konfigurasi parameter yang berbeda; 4.) penilaian model, menilai dari kualitas model yang sudah dibuat.

### 2.1.3.3 Evaluation and Deployment

#### 2.1.3.3.1 Evaluation

Setelah model dibuat, perlu dilakukan evaluasi untuk memastikan bahwa model yang telah dibuat sesuai dengan tujuan bisnis. Ini membutuhkan persetujuan model dari para pemangku kepentingan. Tahapan ini juga mencakup peninjauan proses dan dokumentasi langkah-langkah berikutnya (keputusan dan tindakan).

#### 2.1.3.3.2 *Deployment*

Setelah proyek telah menyelesaikan fase *evaluation* dengan sukses, perlu disiapkan untuk tahap implementasi. Hal ini membutuhkan pembuatan rencana untuk implementasi. Selain itu, rencana pemantauan dan pemeliharaan akan memastikan kesuksesan proyek berlangsung. Selanjutnya, proyek juga perlu menyusun laporan akhir (presentasi) dan melakukan peninjauan proyek tersebut.

### 2.1.4 *Unified Modeling Language* (UML)

Unified Modeling Language (UML) merupakan standar yang menyatukan tiga filosofi desain yang berbeda pada tahun 1980-an. *Artchitect* dan pengembang

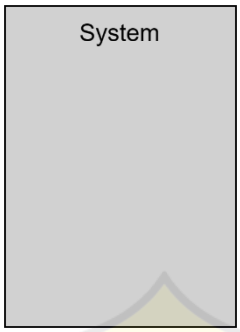
menggunakan diagram kelas dan urutan UML untuk menggambarkan struktur dan alur kerja, tetapi sebagian besar jenis diagram UML lainnya tidak digunakan (R, Mark & Ford, N, 2020).

UML merupakan alat standar berbentuk diagram untuk perancangan. Sebagai bahasa grafis, digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan artefak dari sistem perangkat lunak. Hal ini membantu dalam meningkatkan pemahaman tentang perangkat lunak, sistem, atau produk yang akan dikembangkan di antara pengembang. Diagram UML mencakup jenis diagram penting seperti Diagram *Usecase*, *activity*, *class*, *sequence*, *colloboration* atau *communication*, *state machine*, *component*, dan *deployment* (Sundaramoorthy, 2022).

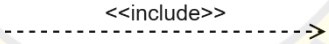
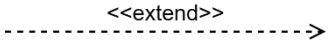
#### 2.1.4.1 Use Case Diagram

Ini berkaitan dengan proses mengidentifikasi kebutuhan fungsional dari sistem yang sedang dipertimbangkan (Sundaramoorthy, 2022). Mencakup pemahaman mendalam tentang apa yang seharusnya dilakukan oleh sistem dalam hal fungsi dan kinerja yang diinginkan.

**Tabel 2. 1 Elemen-elemen Usecase Diagram**

| No. | Elemen | Simbol | Fungsi |
|---|---|---|---|
| 1. | *System Boundary* |  | *System Boundary* adalah gambaran cakupan sistem yang menunjukkan fungsi lengkap yang dimiliki sistem. Ini membantu dalam memahami interaksi antara pengguna dan sistem serta mengidentifikasi kebutuhan fungsional. |
| 2. | *Actors* |  | Aktor dalam sistem adalah entitas yang berinteraksi dengan aplikasi atau sistem yang sedang dikembangkan, seperti individu, organisasi, atau sistem luar. Dalam diagram use case, aktor |

| | | | |
|---|---|---|---|
| | | | mewakili pengguna dalam skenario atau kasus penggunaan yang menjelaskan interaksi mereka dengan sistem. Pemilihan aktor yang tepat penting karena memengaruhi fungsionalitas yang dimodelkan dalam sistem. |
| 3. | *Usecases* |  | Use Case adalah representasi visual dari berbagai fungsionalitas bisnis dalam sistem, memastikan bahwa proses bisnis bersifat diskrit dan dapat diidentifikasi dengan jelas. Identifikasi kasus penggunaan |

| | | | |
|---|---|---|---|
| | | | merupakan langkah awal dalam proses ini. |
| 4. | *Association* |  | Asosiasi menggambarkan koneksi antara aktor dan use case. |
| 5. | *Include* |   | Merepresentasikan situasi di mana satu usecase menyertakan fungsionalitas dari satu usecase lainnya. |
| 6. | *Extend* |   | Extend antara dua use case menunjukkan hubungan di mana use case yang meng- |

| | | | |
|---|---|---|---|
| | | | extend menambahkan perilaku tambahan terhadap fungsionalitas yang ada dari use case dasar. |
| 7. | *Generalization* |  | *Generalization* adalah hubungan di antara use case induk dan satu atau lebih use case anak. |

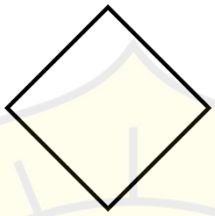
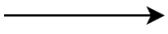
#### 2.1.4.2 Activity Diagram

Ini mencakup analisis mendalam tentang proses-proses yang terlibat dalam memenuhi kebutuhan fungsional tersebut, termasuk tahapan-tahapan yang harus dilalui secara berurutan serta kegiatan yang dapat dilakukan secara bersamaan (Sundaramoorthy, 2022). Dengan memahami urutan dan paralelisme aktivitas, pengembang dapat merancang sistem yang efisien dan sesuai dengan kebutuhan pengguna.

**Tabel 2. 2 Elemen-elemen Activity Diagram**

| No. | Elemen | Simbol | Fungsi |
|---|---|---|---|
| 1. | *Initial state* |  | Ini adalah titik awal dari proses atau aliran kerja yang akan dijalankan dalam sistem tersebut. |
| 2. | *Final State* |  | Ini menunjukkan titik di mana semua aktivitas atau proses dalam sistem telah selesai atau berhenti. |
| 3. | *Swimlanes* |  | Sebuah swimlane terdiri dari dua partisi, yaitu partisi atas yang mewakili entitas seperti aktor, use case, kelas, dll., dan partisi kedua yang berfokus pada serangkaian aktivitas yang terlibat. Swimlane memiliki dua tipe, yaitu vertikal dan horizontal. Dalam vertikal swimlane, partisi-partisi tersebut disusun secara vertikal, sementara dalam horizontal swimlane, |

| | | | |
|---|---|---|---|
| | | | penyusunan dilakukan secara horizontal. |
| 4. | *Action state* |  | Sebuah *action state* menggambarkan langkah konkret atau proses yang terjadi dalam sistem. Ini dapat mencakup operasi yang dilakukan oleh sistem, aktivitas bisnis yang terjadi dalam suatu proses, atau langkah-langkah tertentu dalam alur kerja. |
| 5. | *Synchronization* |  | *Synchronization* menggambarkan dua atau lebih aktivitas yang terjadi pada waktu atau tingkat yang sama. Ada dua jenis *Synchronization*, yaitu *Fork* yang membagi alur aktivitas tunggal menjadi dua atau lebih aktivitas konkuren, dan *Join* yang menggabungkan dua |

| | | | |
|---|---|---|---|
| | | | atau lebih aktivitas konkuren menjadi satu alur tunggal dengan memastikan hanya satu aktivitas yang berjalan pada satu waktu. |
| 6. | *Decision* |  | *Decision* merupakan elemen penting yang memiliki satu input dan dua atau lebih output, tergantung pada kondisi yang telah ditetapkan. Setiap output memiliki kondisi yang terkait dengannya, yang menentukan jalur yang akan diambil oleh aliran data. Jika suatu kondisi terpenuhi, sistem akan mengarahkan aliran data sesuai dengan output yang sesuai. Selain itu, terdapat opsi "*else"* yang dapat diaktifkan jika tidak ada kondisi yang terpenuhi |
| 7. | *Transition* |  | *Transition* adalah panah yang menggambarkan perpindahan |

| | | | |
|---|---|---|---|
| | | | dari keadaan aktivitas sumber ke keadaan aktivitas target yang dipicu oleh penyelesaian aktivitas dari keadaan aktivitas sumber. |
| 8. | *Final Flow* |  | *Final Flow* menunjukkan terminasi dari aliran proses atau jalur dalam aktivitas yang sedang dilakukan. Ini menandakan bahwa aktivitas telah selesai atau berhenti tanpa ada output tambahan yang dihasilkan |

### 2.1.5 Pengenalan *Machine Learning* dan *Data Mining*

#### 2.1.5.1 Konsep Machine Learning

*Machine learning* artinya adalah pembelajaran mesin. Seperti manusia yang belajar untuk hal yang tidak diketahui. Seperti halnya anak kecil yang bertanya ke ibunya mengenai suatu benda dan ibunya pun memberitahu informasi mengenai benda tersebut. Ketika ada anak kecil bermain sendirian dan menemukan benda yang mirip dengan benda yang ia tanyakan sebelumnya, maka anak kecil tersebut dapat menyebutkan nama benda tersebut. *Machine learning* meniru bagaimana

manusia belajar sehingga menghasilkan pengetahuan dan bukan sekedar informasi (Saputra & Kristiyanti, 2022).

#### 2.1.5.1.1 Data, Informasi, dan Pengetahuan

Data yang dihasilkan setiap hari sangat banyak. Untuk perlu diketahui bagaimana mengolah data menjadi informasi dan setelah itu menjadi pengetahuan. Contohnya adalah ada dua orang yang berada di suatu ruangan dingin dengan suhu 5 derajat celsius. Kedua orang tersebut merasa kedinginan. Dan setelah itu menaikkan suhu agar tidak dingin. Jika dilihat dari contoh, kita bisa ketahui yang merupakan data adalah suhu di ruangan, yang dalam hal ini adalah 5 derajat Celsius. Informasi yang dapat diperoleh dari data adalah bahwa suhu tersebut membuat kedua orang merasa kedinginan. Tindakan untuk menaikkan suhu agar tidak dingin merupakan implementasi dari pengetahuan yang diperoleh dari informasi tersebut (Saputra, 2023).

### 2.1.5.2 Konsep Data Mining

Data merupakan himpunan informasi mentah yang tepat. Sedangkan *mining* adalah penambangan atau penggalian. Proses penambangan data mirip dengan menambang emas di mana data awalnya belum terlalu berharga karena belum mengandung banyak informasi. Namun, melalui proses pengolahan data, informasi berharga diekstraksi, yang kemudian dapat digunakan sebagai pengetahuan yang berguna untuk membuat keputusan bagi banyak orang (Saputra, 2023).

### 2.1.5.3 Algoritma Klasifikasi untuk Deteksi Serangan

Klasifikasi artinya adalah pengelompokan. Syarat klasifikasi adalah atribut yang ingin digunakan dan target yang ingin diklasifikasikan yang berbentuk data diskrit, binari, atau kategori (Saputra, 2023).

#### 2.1.5.3.1 *Naïve Bayes*

*Naïve Bayes* merupakan salah satu algoritme yang digunakan untuk melakukan klasifikasi berdasarkan kemungkinan sesuai dengan teorema *bayes*. Teorema *Bayes* atau Hukum *Bayes* adalah suatu prinsip yang memungkinkan kita untuk menghitung kemungkinan suatu kejadian di masa depan berdasarkan informasi atau pengalaman sebelumnya tentang kondisi yang berkaitan dengan kejadian tersebut. Dengan demikian, prinsip ini berguna untuk membantu dalam proses pengambilan keputusan. Algoritme *Naïve Bayes* disebut "naif" karena mengasumsikan dengan sangat sederhana bahwa kemunculan suatu fitur tidak tergantung pada kemunculan fitur lainnya (kelas) (Saputra, 2023). Secara matematis, Teorema *Bayes* dinyatakan dalam bentuk rumus berikut :

$$P(D) = \frac{P(h)P(h)}{P(D)} \quad (1)$$

Penjelasan :

- *h* : Hipotesis data dengan suatu *class* tertentu
- *D* : Data yang belum memiliki *class*
- *P(h)* : Probabilitas hipotesis (*prior probability*)
- *P(D)* : Probilitas *D*
- *P(h|D)* : Probabilitas *h* berdasarkan kondisi *D* (*Posterior Probability*)

- *P(D|h)* : Probabilitas *D* berdasarkan kondisi pada hipotesis *h*

#### 2.1.5.3.2 *K-Nearest Neighbors* (KNN)

*Nearest Neighbors* merupakan algoritme klasifikasi yang menggunakan data terdekat untuk melakukan prediksi pada data baru. Algoritme ini sistem mencari sejumlah tetangga terdekat dari data uji dan menentukan kelas data uji berdasarkan mayoritas kelas dari tetangga terdekat yang ditemukan. Algoritme ini efektif untuk dataset dengan jumlah kecil sampai sedang, karena semakin besar jumlah datanya maka semakin lama waktu yang dibutuhkan untuk melakukan klasifikasi (Saputra, 2023)

#### 2.1.5.3.3 *Random Forest*

Algoritme *random forest* merupakan jenis algoritme *ensemble learning* yang digunakan untuk melakukan klasifikasi, regresi, dan pengelompokan data. *Ensemble learning* adalah algoritme *machine learning* yang menggabungkan beberapa model atau algoritme *machine learning* untuk meningkatkan akurasi prediksi. *Random Forest* bekerja dengan menggabungkan banyak pohon keputusan dibangun dengan acak. Setiap pohon keputusan dihasilkan dari subset acak dari data latih.

### 2.1.5.4 *Confusion Matrix*

*Confusion matrix* memberikan pemahaman yang jelas mengenai kinerja model klasifikasi yang diberikan. Suatu nilai evaluasi memiliki makna atau tidak dapat diketahui dengan menggunakan *confusion matrix* (Hossain, 2023). Istilah-istilah pada *confusion matrix* dijelaskan sebagai berikut:

1. True Positive (TP): Ini adalah kasus positif, di mana model dengan benar memprediksi mereka sebagai kasus positif.
2. False Positive (FP): Ini adalah kasus yang sebenarnya bukan positif, tetapi model memprediksi mereka sebagai kasus positif. Kesalahan ini merupakan kesalahan tipe 1.
3. True Negative (TN): Ini adalah kasus negatif, di mana model dengan benar memprediksi mereka sebagai kasus negatif.
4. False Negative (FN): Ini adalah kasus yang sebenarnya positif, tetapi model secara tidak tepat memprediksi mereka sebagai kasus negatif. Kesalahan ini merupakan kesalahan tipe 2.

#### 2.1.5.4.1 *Accuracy*

Akurasi adalah metrik evaluasi yang paling sederhana. Ini dihitung dengan membagi jumlah prediksi yang benar dengan total jumlah pengamatan. Skor akurasi valid ketika dataset seimbang; setiap kelas dalam dataset memiliki jumlah objek data yang sama.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (2)$$

#### 2.1.5.4.2 *Precision*

Skor presisi digunakan untuk menentukan rasio kasus positif yang diprediksi dengan benar dari semua kasus positif yang diprediksi oleh model. Metrik ini sangat penting ketika sistem harus memiliki tingkat kesalahan positif yang rendah, yaitu meminimalkan jumlah kasus yang salah diklasifikasikan sebagai positif.

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

#### 2.1.5.4.3 *Recall*

Skor recall, juga disebut sensitivitas, digunakan untuk menghitung tingkat *true-positive*. Skor ini dihitung sebagai rasio kasus positif yang diprediksi dengan benar terhadap semua kasus dalam kelas positif aktual.

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

#### 2.1.5.4.4 *F1 Score*

Skor F1 adalah rata-rata harmonik dari skor presisi dan recall. Ini digunakan saat kedua metrik, presisi dan recall, perlu diperhatikan dalam penilaian sistem.

$$F1\ Score = 2 \times \frac{Precsion \times Recall}{Precsion + Recall} \quad (5)$$

### 2.1.6 Perangkat Lunak dan *Tools* Terkait

#### 2.1.6.1 Wireshark

Menurut (Jain, 2022) Wireshark merupakan penganalisa protokol jaringan sumber terbuka yang banyak digunakan. Wireshark merupakan alat yang hampir semua *network administrator* atau *network engineer* untuk menganalisa pola lalu lintas jaringan, menangani masalah protokol jaringan, dan melakukan analisa mendalam terhadap celah keamanan jaringan. *Wireshark* mempunyai banyak fitur, mendukung dari sekumpulan protokol dan enkapsulasi umum dan tidak umum, mendukung semua sistem operasi. Menyediakan GUI yang mudah digunakan dan dipahami dan kemampuan pencarian tingkat lanjut melalui jutaan paket yang

memungkinkan *network administrator* untuk menganalisa kejadian jaringan dengan cepat.

#### 2.1.6.2 Kali Linux

Kali Linux adalah distribusi khusus dari sistem operasi Linux yang berbasis pada Ubuntu Linux, yang berbasis pada Debian Linux. Tujuan Kali adalah untuk para praktisi keamanan yang ingin melakukan berbagai kegiatan, seperti pengujian keamanan, pengembangan eksploitasi atau *reverse engineering*, atau forensik digital. Yang membedakan distribusi Linux adalah lapisan perangkat lunak tambahan yang disertakan di atas inti Linux, menjadikannya unik. Kali Linux tidak hanya menyertakan utilitas penting, tetapi juga ratusan paket perangkat lunak yang khusus digunakan dalam pekerjaan keamanan (Messier, 2024).

#### 2.1.6.3 VMware Workstation

VMware Workstation adalah contoh dari hypervisor Tipe 2 yang memungkinkan virtualisasi pada sistem operasi desktop. Hypervisor ini memfasilitasi pembuatan dan pengelolaan mesin virtual (VM) yang berjalan secara simultan dengan sistem operasi asli. VMware Workstation memungkinkan konfigurasi sumber daya komputasi seperti CPU, memori, dan penyimpanan. Ini memberikan fleksibilitas dalam mengalokasikan sumber daya yang diperlukan untuk VM sesuai dengan kebutuhan pengguna. Dengan VMware Workstation, pengguna dapat mengonfigurasi sumber daya jaringan virtual, termasuk switch virtual, adapter jaringan, dan server DHCP virtual. Ini memungkinkan simulasi lingkungan jaringan kompleks dan pengujian berbagai skenario jaringan dalam lingkungan yang terisolasi (von Oven, 2023).

### 2.1.6.4 Python

Bagi banyak orang, bahasa pemrograman Python memiliki daya tarik yang kuat. Sejak pertama kali muncul pada tahun 1991, Python telah menjadi salah satu bahasa pemrograman terinterpretasi yang paling populer, bersama dengan Perl, Ruby, dan lainnya. Python dan Ruby menjadi sangat populer sejak sekitar tahun 2005 untuk membangun situs web menggunakan berbagai kerangka kerja web mereka, seperti Rails (Ruby) dan Django (Python). Bahasa-bahasa seperti itu sering disebut sebagai bahasa skrip, karena mereka dapat digunakan untuk dengan cepat menulis program kecil atau skrip untuk mengotomatisasi tugas-tugas lain. Di antara bahasa-bahasa terinterpretasi, karena berbagai alasan sejarah dan budaya, Python telah mengembangkan komunitas komputasi ilmiah dan analisis data yang besar dan aktif. Dalam 20 tahun terakhir, Python telah berkembang dari bahasa komputasi ilmiah yang paling mutakhir atau "pada risiko Anda sendiri" menjadi salah satu bahasa yang paling penting untuk ilmu data, pembelajaran mesin, dan pengembangan perangkat lunak secara umum di dunia akademis dan industri (McKinney, 2022).

### 2.1.6.5 Jupyter

*Jupyter* merupakan perangkat komputasi eksploratif yang membolehkan pengguna untuk menulis serta menjalankan kode dalam dokumen yang interaktif. Fungsinya adalah untuk menyederhanakan proses analisis dan visualisasi data. Sebagai bagian dari Jupyter. IPython Shell, yang merupakan bagian dari Jupyter, menawarkan lingkungan interaktif yang lengkap untuk mengeksekusi kode Python.

Ini menyediakan fitur-fitur seperti pelengkapan tab, introspeksi, dan beragam fungsi lain yang memperkaya pengalaman pengguna(McKinney, 2022).

#### 2.1.6.6 CICFlow Meter

*CICFlowMeter* adalah sebuah alat yang memungkinkan pengguna untuk menganalisis lalu lintas jaringan dengan efisien. Dengan alat ini, pengguna dapat menghasilkan arus dua arah dari paket jaringan, termasuk arah maju (dari sumber ke tujuan) dan arah mundur (dari tujuan ke sumber). Selain itu, CICFlowMeter juga dapat mengekstraksi lebih dari 80 fitur statistik lalu lintas jaringan, seperti durasi, jumlah paket, jumlah byte, dan panjang paket, yang dapat memberikan wawasan mendalam tentang karakteristik lalu lintas tersebut. Hasil analisis dapat disimpan dalam file dengan ekstensi .csv, memudahkan pengguna untuk mengolah dan menganalisis data lebih lanjut. Alat ini memiliki manfaat yang luas, baik untuk penelitian maupun aplikasi keamanan jaringan, terutama dalam mendeteksi serangan Distributed Denial-of-Service (DDoS). Dengan demikian, CICFlowMeter menjadi alat yang vital dalam analisis lalu lintas jaringan dan menjaga keamanan sistem (Zidane, 2022).

## 2.2 Kajian Penelitian Terdahulu

Sangat krusial mengetahui penelitian terdahulu untuk mengetahui keterkaitannya dengan penelitian yang ditulis oleh penulis. Berikut ini adalah gambaran mengenai studi sebelumnya.

**Tabel 2. 3 Daftar Kajian Penelitian Terdahulu**

| No. | Penulis | Judul | Data | Metode | Publikasi |
|---|---|---|---|---|---|
| 1. | Fluorida Fibrianda & Bhawiyuga, 2018 | Analisis Perbandingan Akurasi Deteksi Serangan Pada Jaringan Komputer Dengan Metode Naïve Bayes Dan Support Vector Machine (SVM) | ISCX 2012 | • Naïve Bayes<br>• Support Vector Machine | Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer<br>Vol. 2, No. 9, September 2018, hlm. 3112-3123 |
| | **Hasil :**<br>• Naive Bayes: 85,055%<br>• SVM Linear: 99,995%<br>• SVM Polynomial: 99,999%<br>• SVM Sigmoid: 99,995% | | | | |
| | **Kelemahan Penelitian :** | | | | |

<table>
<tr><td></td><td colspan="5">Penelitian menggunakan dataset ISCX 2012 yang mungkin tidak mencakup semua jenis serangan terkini. Model yang dikembangkan mungkin tidak generalisasi dengan baik ke skenario dunia nyata yang berbeda. Beberapa metode, seperti SVM Linear, membutuhkan waktu komputasi yang signifikan, yang dapat mempengaruhi efisiensi dalam penerapan praktis.</td></tr>
<tr><td rowspan="3">2.</td><td>Fauzi et al., 2023</td><td>Penerapan Random Forest dan Adaboost untuk Klasifikasi Serangan DDoS</td><td>CICDDoS2019</td><td>● Random Forest<br>● Adaboost</td><td>Journal on Education<br>Volume 05, No. 03, Maret-April 2023, pp. 7925-7937</td></tr>
<tr><td colspan="5">Hasil :<br>● Random Forest: 99%<br>● Adaboost: 99%</td></tr>
<tr><td colspan="5">Kelemahan Penelitian :<br>Penelitian ini hanya menggunakan satu dataset, CICDDoS2019, yang mungkin membatasi validitas hasil karena tidak adanya perbandingan dengan dataset lain.</td></tr>
</table>

| | | | | | |
|---|---|---|---|---|---|
| 3. | Zidane, 2022 | Klasifikasi Serangan Distributed Denial of-Service (DDoS) menggunakan Metode Data Mining Naïve Bayes | CICIDS 2018 | Naïve Bayes | Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer Vol. 6, No. 1, Januari 2022, hlm. 172-180 |
| | **Hasil :**<br>Naive Bayes: 95% | | | | |
| | **Kelemahan Penelitian :**<br>Keterbatasan dataset yang digunakan, keterbatasan dalam menghadapi serangan DDoS yang lebih kompleks dan variatif | | | | |
| 4. | Nasution & Basuki, 2021 | Implementasi Algoritma C5.0 Untuk Klasifikasi | CICDDoS 2019 | C5.0 | Jurnal Pengembangan Teknologi Informasi dan |

| | | | | | |
|---|---|---|---|---|---|
| | | Serangan DDoS | | | Ilmu Komputer Vol. 5, No. 1, Januari 2021, hlm. 389-395 |
| | Hasil :<br>● Akurasi sebesar 98,38%<br>● presisi sebesar 98,39%<br>● recall sebesar 98,37% | | | | |
| | **Kelemahan Penelitian :**<br>Penelitian ini menghadapi tantangan imbalancing class, di mana terdapat perbedaan jumlah instance antara kelas. Meskipun algoritme C5.0 menunjukkan hasil yang akurat, waktu komputasi yang dibutuhkan relatif lebih lama dibandingkan algoritme C4.5 karena proses boosting yang dilakukan. Dataset yang digunakan adalah CICDDoS2019 yang mungkin tidak mencakup semua variasi serangan DDoS terbaru. Hanya algoritme C4.5 yang digunakan sebagai pembanding, sehingga belum tentu representatif terhadap semua metode klasifikasi yang ada. | | | | |