

## BAB II TINJAUAN PUSTAKA

### 2.1. Artificial Intelligence

*Artificial Intelligence* (AI) adalah sebuah cabang ilmu komputer yang berfokus pada pengembangan algoritma dan teknik-teknik yang memungkinkan mesin untuk melakukan tugas-tugas yang memerlukan kecerdasan manusia [16]. Dalam beberapa tahun terakhir, teknologi AI telah mengalami perkembangan yang sangat pesat dan menjadi topik yang semakin menarik dalam dunia akademik dan industri.

Perkembangan teknologi AI yang pesat memberikan banyak manfaat, hal ini juga membuka potensi risiko keamanan siber yang semakin meningkat. Semakin banyak bot jahat yang menggunakan teknologi AI untuk tujuan merugikan, seperti serangan DDoS yang bisa mencegah layanan internet. Oleh karena itu, keamanan siber menjadi semakin penting dalam pengembangan teknologi AI.

### 2.2. Machine Learning

*Machine learning* adalah konsep penting dalam pengembangan teknologi kecerdasan buatan. Dalam *machine learning*, suatu sistem komputer dapat belajar sendiri dari data yang diberikan sehingga tidak perlu diprogram secara eksplisit oleh manusia [17]. Dengan menggunakan algoritma yang telah diprogram, sistem komputer dapat memproses data secara mandiri dan membuat prediksi atau tindakan berdasarkan data tersebut, yang nantinya dapat digunakan untuk berbagai macam aplikasi, seperti di bidang kesehatan, keuangan, dan otomotif.

Terdapat beberapa jenis *machine learning* yang digunakan dalam pengembangan teknologi AI, seperti *supervised learning*, *unsupervised learning*, dan *reinforcement learning*, yang masing-masing memiliki kegunaan dan aplikasi yang berbeda [18]. *Supervised learning* digunakan untuk memprediksi nilai atau kelas dari suatu data yang sudah diketahui, *semi-supervised learning* digunakan untuk memanfaatkan data yang tidak dilabeli untuk meningkatkan performa model pada data yang dilabeli, sementara *unsupervised learning* digunakan untuk menemukan pola atau struktur dalam data yang belum diketahui, dan *reinforcement*

learning digunakan untuk membuat keputusan berdasarkan tindakan dan pengalaman sebelumnya [17], [19].

### 2.3. *Deep Learning*

*Deep learning* adalah salah satu bidang dalam kecerdasan buatan. *Deep learning* menggunakan arsitektur jaringan syaraf tiruan (neural network) yang lebih besar dan lebih dalam dari jaringan syaraf tiruan biasa. Dalam *deep learning*, jaringan syaraf tiruan yang terdiri dari banyak lapisan dapat belajar secara otomatis dan membuat prediksi atau klasifikasi berdasarkan data yang diberikan [20]. Salah satu keunggulan dari *deep learning* adalah kemampuannya untuk mengolah data yang kompleks, seperti gambar atau suara sehingga dapat digunakan dalam berbagai bidang, seperti pengenalan wajah, deteksi objek, atau bahkan anomali jaringan.

*Neural network* pada *deep learning* terdiri dari beberapa lapisan, seperti lapisan *input*, lapisan *hidden layer*, dan lapisan *output*. Setiap lapisan terdiri dari beberapa neuron yang saling terhubung dengan bobot dan bias tertentu [21]. Proses pelatihan pada *neural network* dilakukan dengan menggunakan *training* data yang telah dilabeli. Pada setiap iterasi pelatihan, *neural network* akan memperbaiki bobot dan bias pada setiap lapisan berdasarkan nilai error pada *output* yang dihasilkan.

Cara kerja *deep Learning* adalah dengan menggunakan beberapa lapisan unit pemrosesan nonlinier untuk mengekstraksi dan mentransformasi fitur. Lapisan yang dekat dengan *input* data mempelajari fitur-fitur sederhana, sedangkan lapisan yang lebih tinggi mempelajari fitur-fitur yang lebih kompleks yang berasal dari fitur-fitur lapisan bawah. Arsitektur membentuk representasi fitur yang hierarkis dan kuat. Berdasarkan cara kerja tersebut, *deep learning* cocok untuk menganalisis dan mengekstraksi pengetahuan yang berguna dari data dalam jumlah besar dan data berdimensi tinggi [22], [23].

### 2.4. **Matriks Evaluasi pada *artificial intelligence* Model**

Evaluasi model *artificial intelligence* memiliki berbagai macam matriks untuk mengukur performa dan efektivitas model. Berikut adalah beberapa matriks evaluasi yang umum digunakan dalam bidang kecerdasan buatan, terutama untuk model pembelajaran mesin dan *deep learning* [24].

### 2.4.1 Matriks Konfusi

Metrik konfusi digunakan untuk mengevaluasi kinerja model prediksi yang membantu dalam melihat performa dari algoritma klasifikasi. Matriks evaluasi terdiri dari empat jenis, yaitu *True Positive* (TP) yang merupakan jumlah data yang diklasifikasikan sebagai positif dan benar, *True Negative* (TN) yang merupakan jumlah data yang diklasifikasikan sebagai negatif dan benar, *False Positive* (FP) yang merupakan jumlah data yang diklasifikasikan sebagai positif namun sebenarnya negatif, dan *False Negative* (FN) yang merupakan jumlah data yang diklasifikasikan sebagai negatif namun sebenarnya positif.

### 2.4.2 Akurasi

Akurasi menunjukkan seberapa benar pengklasifikasi memprediksi titik data, seperti yang ditunjukkan pada Persamaan (3.1).

$$Akurasi = \frac{TP}{TP+TN+FP+FN} \quad (2.1)$$

Akurasi yang baik adalah akurasi yang cukup untuk mencapai tujuan. Mengejar akurasi yang tinggi mungkin tidak akan mendapatkan hasil yang diinginkan. Sebaliknya, fokus harus diarahkan pada memastikan bahwa akurasi model cukup untuk tujuan yang dimaksudkan [25]. Model yang lebih kompleks cenderung memiliki kemampuan yang lebih tinggi untuk mencapai akurasi yang lebih tinggi. Namun, peningkatan kompleksitas sering kali diiringi dengan risiko overfitting, di mana model bekerja sangat baik pada data pelatihan tetapi tidak mampu menggeneralisasi dengan baik pada data baru. Dalam banyak kasus, model yang lebih sederhana dengan akurasi yang cukup mungkin lebih diinginkan karena dapat memberikan keseimbangan antara performa dan generalisasi [26].

### 2.4.3 Presisi

Presisi memberikan kemungkinan seberapa benar pengklasifikasi memprediksi kelas positif. Presisi dihitung dengan Persamaan (3.2)

$$Presisi = \frac{TP}{TP+FN} \quad (2.2)$$

#### 2.4.4 Recall

*Recall* menunjukkan kemungkinan seberapa benar pengklasifikasi dapat mendeteksi kelas positif. *Recall* dihitung dengan Persamaan (3.3)

$$Recall = \frac{TP}{TP+FN} \quad (2.3)$$

#### 2.4.5 F1 Score

Skor F1 adalah keseimbangan rata-rata antara presisi dan *recall*, seperti yang ditunjukkan pada Persamaan (3.4)

$$F_1 \text{ Score} = 2x \frac{Presisi \times Recall}{Presisi + Recall} \quad (2.4)$$

#### 2.4.6 Loss Function

Fungsi kerugian adalah ukuran kesalahan model. Misalnya, dalam kasus klasifikasi biner, *Binary Cross-Entropy Loss* sering digunakan. Untuk regresi, *Mean Squared Error* (MSE) adalah fungsi kerugian yang umum digunakan.

#### 2.4.7 Receiver Operating Characteristic Curve (ROC Curve)

*ROC curve* adalah grafik yang menunjukkan performa model klasifikasi pada semua level *threshold* klasifikasi. Kurva ini memplot *True Positive Rate* (TPR) versus *False Positive Rate* (FPR).

### 2.5. Multilayer Perceptron (MLP)

*Multilayer Perceptron* (MLP) adalah contoh fundamental dari deep neural network. Arsitektur MLP terdiri dari beberapa hidden layer untuk menangkap dan memodelkan hubungan yang lebih kompleks antara *input* dan *output* dalam *dataset* latihan. Setiap neuron dalam hidden layer terhubung dengan neuron-neuron di lapisan sebelumnya dan lapisan berikutnya. Fungsi utamanya melibatkan pemodelan dan prediksi untuk memahami hubungan kompleks antara *input* dan *output* [27]. Dengan menerapkan metode pembelajaran yang tepat, *multilayer perceptron* mampu mengenali pola dan melakukan klasifikasi data dengan tingkat akurasi yang tinggi.

Selama proses pembelajaran, *multilayer perceptron* menggunakan algoritma backpropagation untuk mengoptimalkan bobot pada setiap koneksi antar

neuron. Proses di setiap neuron melibatkan penjumlahan *input* yang diterima, kemudian diproses melalui fungsi aktivasi. Fungsi aktivasi ini memungkinkan neuron menghasilkan *output* non-linear, memungkinkan *multilayer perceptron* untuk memodelkan hubungan yang kompleks antara *input* dan *output* [27]. Oleh karena itu *multilayer perceptron* mampu untuk memahami dan mengenali pola-pola kompleks sehingga memungkinkannya untuk memodelkan dan memprediksi masalah-masalah yang sulit.

Performa *multilayer perceptron* sangat dipengaruhi oleh jumlah dan ukuran lapisan neuron, serta jumlah data latih yang tersedia. Pemilihan fungsi aktivasi juga memainkan peran penting dalam performa jaringan sehingga dapat mengurangi kesalahan prediksi akibat *overfitting* dan meningkatkan akurasi klasifikasi [28]. Pemilihan jumlah neuron dalam lapisan tersembunyi dan jumlah lapisan tersembunyi biasanya melibatkan pendekatan uji coba dan kesalahan heuristik. Ini juga merupakan kasus penerapan penyetelan hyperparameter untuk meningkatkan kinerja jaringan. Oleh karena itu, penentuan parameter-parameter yang tepat menjadi kunci dalam penerapan *multilayer perceptron* pada bidang teknik.

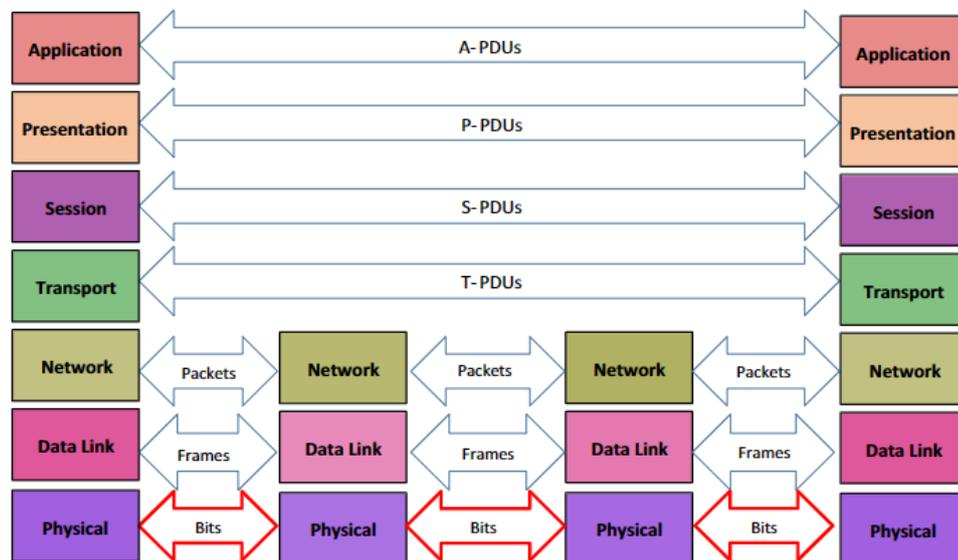
## 2.6. Jaringan Komputer

Jaringan komputer merupakan kumpulan beberapa komputer atau perangkat elektronik yang saling terhubung dan dapat berkomunikasi satu sama lain dengan tujuan untuk berbagi sumber daya dan informasi. Jaringan komputer dapat mencakup perangkat keras, perangkat lunak, dan protokol komunikasi yang digunakan untuk memfasilitasi pertukaran data antar perangkat. Jaringan komputer dimana pengguna dapat mengakses sumber daya yang tersedia dari jarak jauh, dan memungkinkan terjadinya kolaborasi dan pertukaran informasi antara pengguna di seluruh dunia disebut dengan internet [29].

Protokol jaringan komputer sangat penting dalam menjalankan jaringan komputer. Protokol jaringan komputer adalah seperangkat aturan yang mengatur bagaimana data ditransmisikan melalui jaringan. Beberapa protokol yang umum digunakan dalam jaringan komputer adalah *Transmission Control Protocol/Internet Protocol* (TCP/IP), *User Datagram Protocol* (UDP), dan *Hypertext Transfer Protocol* (HTTP) [30]. TCP/IP adalah protokol standar yang

digunakan dalam internet dan jaringan komputer modern. Protokol ini bertanggung jawab untuk mengatur pengiriman dan penerimaan data antar perangkat yang terhubung [8]. UDP merupakan protokol yang lebih sederhana dan lebih cepat dari TCP/IP, namun kurang dapat diandalkan karena tidak memiliki mekanisme pengontrolan kesalahan [8]. Sedangkan HTTP adalah protokol yang digunakan untuk mengakses halaman web di internet [31].

Proses komunikasi data melalui jaringan memiliki kerangka logika terstruktur yang di sebut model OSI (*Open Sistem Interconnection*). Model Osi menerapkan teknik penataan yang di sebut *layering*. Osi *layer* terdiri dari 7 lapisan dengan masing-masing mempunyai peran dan fungsi yang berbeda [8].



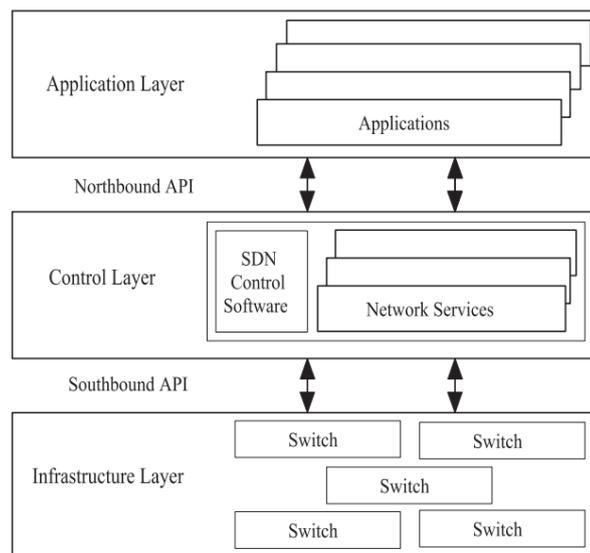
Gambar 2.1 Arsitektur jaringan berdasarkan model OSI [30].

Berikut ini merupakan penjelasan untuk gambar 2.3. Model OSI terdiri dari tujuh lapisan yang mengatur komunikasi jaringan. Lapisan Aplikasi (7) menyediakan layanan langsung ke aplikasi pengguna. Lapisan Presentasi (6) mengonversi data ke format yang dipahami aplikasi. Lapisan Sesi (5) mengelola sesi komunikasi. Lapisan *Transport* (4) memastikan pengiriman data yang andal. Lapisan Jaringan (3) merutekan paket data di jaringan. Lapisan Tautan Data (2) memproses transfer data antar perangkat. Lapisan Fisik (1) menangani transmisi sinyal fisik data. Setiap lapisan mempunyai fungsi khusus untuk mendukung operasi jaringan yang efektif. [30], [31].

## 2.7. *Software-Defined Networks*

*Software Defined Network* (SDN) merupakan suatu konsep dalam bidang jaringan komputer yang memisahkan lapisan kontrol (*control plane*) dari lapisan penerusan (*data plane*) dalam infrastruktur jaringan. Dalam konteks ini, kontrol dan manajemen jaringan diatur secara terpusat dan terpisah dari perangkat keras jaringan fisik. Penerapan SDN memungkinkan administrator jaringan untuk mengelola dan mengontrol jaringan dengan lebih fleksibel dan efisien [32]. Konsep ini memungkinkan penggunaan perangkat lunak untuk mengatur dan mengendalikan alur lalu lintas jaringan, memfasilitasi penyesuaian dan pengaturan yang lebih cepat dan mudah.

Keuntungan dari SDN melibatkan beberapa aspek krusial, seperti fleksibilitas, efisiensi, dan skalabilitas [33]. Pertama, dari segi fleksibilitas, SDN memberikan kebebasan kepada administrator untuk mengubah dan mengkonfigurasi jaringan dengan cepat dan mudah melalui perangkat lunak. Kedua, dalam hal efisiensi, SDN menyajikan sentralisasi kontrol yang mengoptimalkan penggunaan sumber daya jaringan. Oleh karena itu, SDN dapat menghindari overprovisioning dan meningkatkan efisiensi penggunaan jaringan secara menyeluruh. Terakhir, skalabilitas menjadi keunggulan utama. SDN memungkinkan jaringan untuk diperluas dengan lebih mudah dan efisien karena kontrol dan manajemen jaringan terpusat.



Gambar 2.2 Jenis serangan siber [33]

Gambar 2.2 menunjukkan arsitektur Software-Defined Network (SDN), yang terdiri dari 5 komponen utama yaitu, *application layer*, *control layer*, *infrastructure layer*, *southbound interface*, dan *northbound interface*. Lapisan Aplikasi berfungsi sebagai ruang untuk aplikasi dan layanan jaringan, yang mencakup segala sesuatu mulai dari manajemen *bandwidth* hingga optimalisasi lalu lintas. Lapisan Kontrol memegang peran sentral dalam mengatur pengaturan dan keputusan jaringan, berkomunikasi dengan perangkat keras jaringan melalui *southbound interface* seperti *OpenFlow*. Lapisan Infrastruktur, merupakan perangkat keras fisik seperti *switch* dan router, yang mengirimkan lalu lintas data sesuai dengan instruksi dari lapisan kontrol. Dengan *southbound interface*, dan *northbound interface*, SDN memungkinkan komunikasi yang efisien antara lapisan kontrol, aplikasi, dan infrastruktur. Sebagai hasilnya, SDN meningkatkan efisiensi manajemen, fleksibilitas yang ditingkatkan, dan adaptabilitas dinamis jaringan sesuai kebutuhan aplikasi.

Jaringan berbasis Software-Defined Network (SDN) memiliki dua komponen utama dalam pengembangan dan pengujian yaitu Mininet dan Ryu *controller*. Dengan menggunakan Mininet sebagai emulator dan Ryu sebagai *controller*, peneliti dapat menguji aplikasi SDN dalam lingkungan terkendali sebelum mengimplementasikannya dalam jaringan fisik yang sebenarnya. Hal ini memudahkan pengembangan, pengujian, dan validasi solusi SDN tanpa memerlukan infrastruktur jaringan yang rumit.

### 2.7.2 Mininet

Mininet adalah emulator jaringan SDN yang memungkinkan pengguna membuat jaringan virtual yang terdiri dari *host*, *switch*, dan *controller* SDN. Dengan Mininet, pengguna dapat membuat topologi jaringan yang terisolasi dalam lingkungan virtual, memungkinkan pengembang dan peneliti untuk menguji dan mengembangkan aplikasi SDN tanpa memerlukan infrastruktur fisik yang sebenarnya. Mininet menyediakan berbagai opsi konfigurasi untuk *host*, *switch*, dan *controller* sehingga pengguna dapat mensimulasikan berbagai skenario jaringan.

### 2.7.3 Ryu controller

Ryu adalah sebuah platform pengembangan *controller* SDN yang bersifat open-source. Ryu menyediakan kerangka kerja yang kuat dan fleksibel untuk membangun aplikasi pengontrol jaringan SDN. *Controller* ini mendukung protokol *OpenFlow*, yang memungkinkan pengontrol untuk mengambil keputusan terkait arus lalu lintas dan mengonfigurasi perangkat keras jaringan sesuai kebutuhan. Ryu juga dapat digunakan untuk mengimplementasikan berbagai kebijakan jaringan, seperti deteksi intrusi, manajemen *bandwidth*, dan lainnya.

## 2.8. Quality of Service

*Quality of Service* (QoS) adalah metode untuk mengukur dan memastikan kualitas jaringan komputer berdasarkan atribut kinerja yang terhubung dengan layanan tertentu [34]. QoS memiliki kemampuan untuk mendefinisikan atribut layanan jaringan yang tersedia, memastikan kehandalalan dalam penyampaian data, serta mendefinisikan tingkat kecepatan dalam sistem komunikasi. Pengukuran QoS memiliki standar yang digunakan untuk mengevaluasi kinerja jaringan telekomunikasi dan internet yaitu standar TIPHON (*Telecommunications and Internet Protocol Harmonization Over Networks*) [35]. Standar ini menetapkan kategori kualitas untuk 4 parameter QoS utama, yaitu *throughput*, *delay*, *jitter*, dan *packet loss*. Berikut adalah tabel yang menunjukkan kategori standar TIPHON untuk parameter-parameter tersebut.

Tabel 2.1 Standar TIPHON [35]

Parameter QoS	Sangat Bagus	Bagus	Sedang	Buruk
<i>Throughput</i>	> 2,1 Mbps	1200 kbps – 2,1 Mbps	700 - 1200 kbps	338 - 700 kbps
<i>Delay</i>	< 150 ms	150-300 ms	300 - 450 ms	> 450 ms
<i>Jitter</i>	0 ms	0 - 75 ms	75 - 125 ms	125 - 225 ms
<i>Packet Loss</i>	0 - 2%	3 - 14%	12 - 24%	> 25%

Tabel di atas menunjukkan standar TIPHON untuk berbagai kategori kualitas parameter QoS. Dengan menggunakan standar ini, kinerja jaringan dapat di bandingkan dan dinilai secara lebih terstruktur dan sistematis. Kategori-kategori

ini membantu dalam memahami seberapa baik jaringan berfungsi dan area mana yang perlu ditingkatkan. Implementasi QoS sesuai dengan standar TIPHON memastikan bahwa jaringan dapat memberikan layanan yang optimal dan dapat diandalkan kepada penggunanya. Berikut merupakan penjelasan dari masing-masing parameter.

### 2.8.2 *Throughput*

*Throughput* merupakan jumlah total data yang melewati *bandwidth* pada waktu tertentu. *Throughput* didapatkan dari jumlah paket yang diterima dibagi dengan waktu pengiriman [35]. *Throughput* pada umumnya bersifat dinamis, tergantung pada kondisi lalu lintas jaringan. Oleh karena itu *throughput* dapat mencerminkan kapabilitas sebenarnya dari suatu jaringan dalam mengirimkan suatu data. Semakin tinggi *throughput*, semakin baik jaringan dalam menangani volume data. Nilai *throughput* dapat dikatakan baik atau buruk berdasarkan standar TIPHON yang dapat dilihat pada Tabel 2.1.

### 2.8.3 *Delay*

*Delay* adalah waktu yang diperlukan paket data untuk bergerak dari pengirim ke penerima [34], biasanya diukur dalam milidetik (ms). *Delay* mencerminkan seberapa cepat data dapat berpindah dalam jaringan. *Delay* rendah menunjukkan jaringan yang responsif dan efisien. Namun, nilai *delay* dapat meningkat karena kemacetan jaringan, jarak fisik, atau kecepatan pengolahan. Menurut standar TIPHON, nilai *delay* dapat dikategorikan sebagai baik atau buruk, yang bisa dilihat pada Tabel 2.1.

### 2.8.4 *Jitter*

*Jitter* dapat dikatakan sebagai variasi waktu kedatangan paket data yang diterima [34]. Variasi ini dapat mengganggu aplikasi *real-time* seperti panggilan suara atau video streaming. *Jitter* diukur dalam milidetik (ms) dan nilai yang lebih rendah menunjukkan jaringan yang stabil. *Jitter* yang tinggi dapat menyebabkan kualitas komunikasi yang buruk. Standar TIPHON juga menyediakan kategori untuk menilai kualitas *jitter* pada jaringan, yang dapat dilihat pada Tabel 2.1.

### 2.8.5 Packet loss

*Packet Loss* terjadi ketika satu atau lebih paket data gagal mencapai tujuannya dalam transmisi jaringan. *Packet loss* dapat disebabkan oleh kemacetan jaringan, atau kesalahan perangkat keras [34]. *Packet loss* yang rendah menunjukkan jaringan yang andal dan efisien, sementara *packet loss* yang tinggi dapat menyebabkan masalah seperti keterlambatan pengiriman data, atau hilangnya informasi penting. Standar TIPHON menetapkan ambang batas untuk menilai kualitas *packet loss*, yang juga dapat dilihat pada Tabel 2.1.

### 2.9. Cyber Security

*Cyber Security* atau keamanan siber adalah teknik yang di rancang untuk melindungi sistem dari serangan, pencurian, kerusakan, modifikasi, atau akses yang tidak sah [24]. Tujuan utama *Cyber Security* adalah untuk memastikan perlindungan data. Oleh karena itu, Sistem komputer yang aman dan stabil harus memastikan kerahasiaan, ketersediaan, dan integritas informasi. Prinsip ini disebut triad CIA (*confidentiality, integrity, dan availability*) [8], [36].



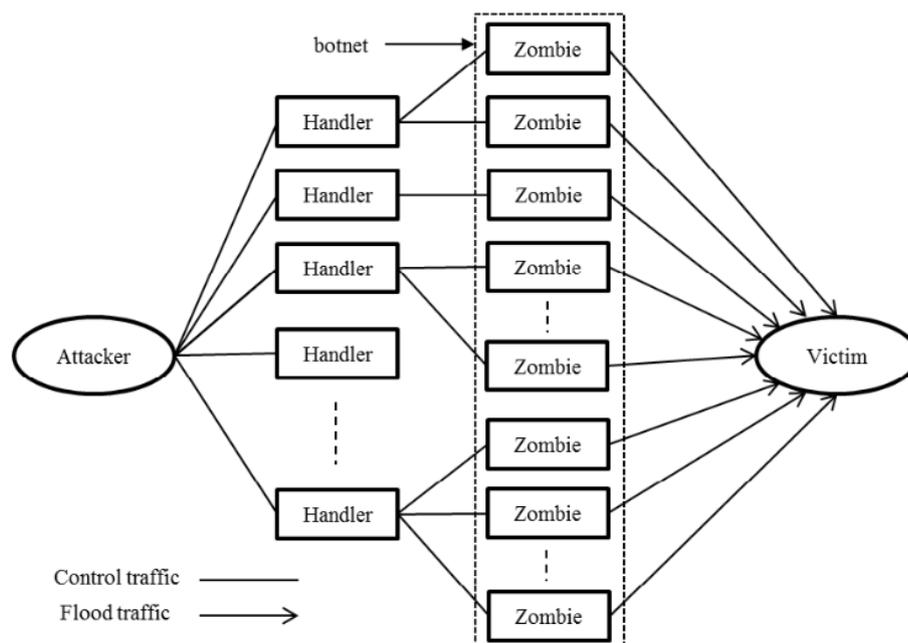
Gambar 2.3 Jenis serangan siber [37]

*Confidentiality, Integrity, dan Availability (CIA)* adalah tiga prinsip utama yang harus dipertimbangkan pada jaringan komputer. *Confidentiality* mengacu pada perlindungan data dari akses yang tidak sah atau tidak diizinkan [38]. *Integrity* berarti memastikan bahwa data yang tersimpan tidak dapat dimodifikasi oleh pihak yang tidak berwenang dan sesuai dengan kebijakan dan prosedur organisasi [39]. *Availability* berhubungan dengan ketersediaan sumber daya atau informasi yang diinginkan [36].

## 2.10. Serangan DDoS

Serangan DDoS (*Distributed Denial of Service*) merupakan jenis serangan siber yang bertujuan untuk menyerang infrastruktur komputer dengan cara membanjiri server dengan lalu lintas internet yang berlebihan [40]. Serangan DDoS dapat dilakukan dengan menggunakan banyak komputer yang terinfeksi oleh malware atau dengan menggunakan jaringan botnet yang terdiri dari ribuan komputer sehingga server menjadi overload dan tidak dapat memproses permintaan dari pengguna yang sah [41]. Dalam serangan DDoS, lalu lintas yang berlebihan dan palsu dapat berasal dari ribuan atau bahkan jutaan sumber yang terdistribusi di seluruh dunia sehingga sulit untuk mengidentifikasi dan memblokir serangan.

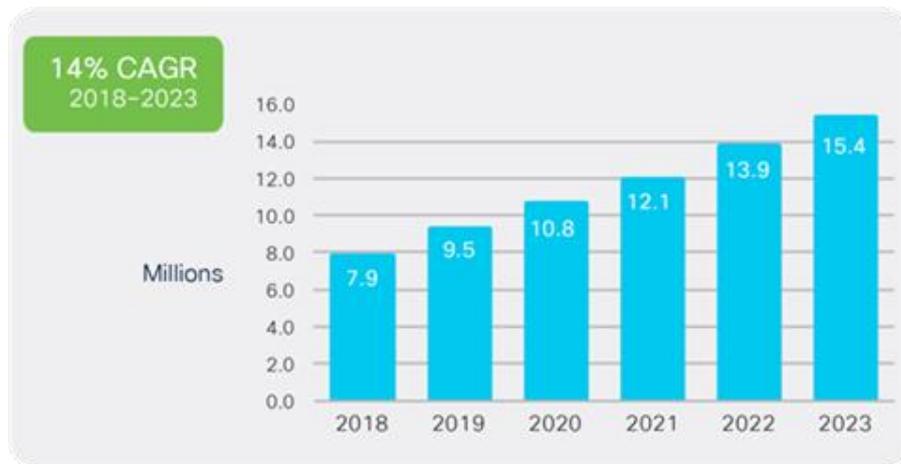
Serangan DDoS menjadi semakin canggih dalam beberapa tahun terakhir. Saat ini, alat serangan DDoS telah menjadi otomatis dan canggih sehingga memungkinkan penyerang untuk mengeksekusi serangan secara otomatis dengan sedikit campur tangan manusia. Pada dasarnya setiap penyerang menggunakan teknik perentasan yang berbeda untuk mencari kelemahan dalam sistem komputer yang terhubung ke internet. Skema serangan DDoS yaitu penyerang akan mencoba mengontrol mesin komputer sebanyak mungkin hingga membentuk zombie botnet seperti yang di tunjukan pada gambar 2.5. Botnet adalah jaringan komputer atau perangkat yang terhubung ke Internet yang berkomunikasi satu sama lain [42].



Gambar 2.1 Arsitektur serangan DDoS [41]

Satu zombie hanya menyediakan sedikit data, tetapi lalu lintas kumulatif dari banyak zombie yang muncul di sistem target akan menjadi sangat besar dan menghabiskan sumber daya. Ukuran botnet menentukan tingkat dan besarnya intensitas serangan. Botnet dalam jumlah besar bisa melakukan serangan yang menghancurkan dan parah. Sulit bagi sistem pertahanan untuk membedakan antara lalu lintas abnormal *flash crowds* dari serangan DDoS karena mereka hanya berbeda dalam beberapa parameter. *Flash Crowds* (FC) merupakan jenis lalu lintas jaringan yang mirip dengan lalu lintas DDoS, tetapi berasal dari pengguna yang sah. FC hampir serupa dengan serangan DDoS karena pengguna menggunakan akses ke sistem secara bersamaan dan tiba-tiba [41].

Dalam laporan Internet tahunan Cisco pada gambar 2.6, menunjukkan tren lonjakan serangan DDoS dari 2018 hingga 2023.



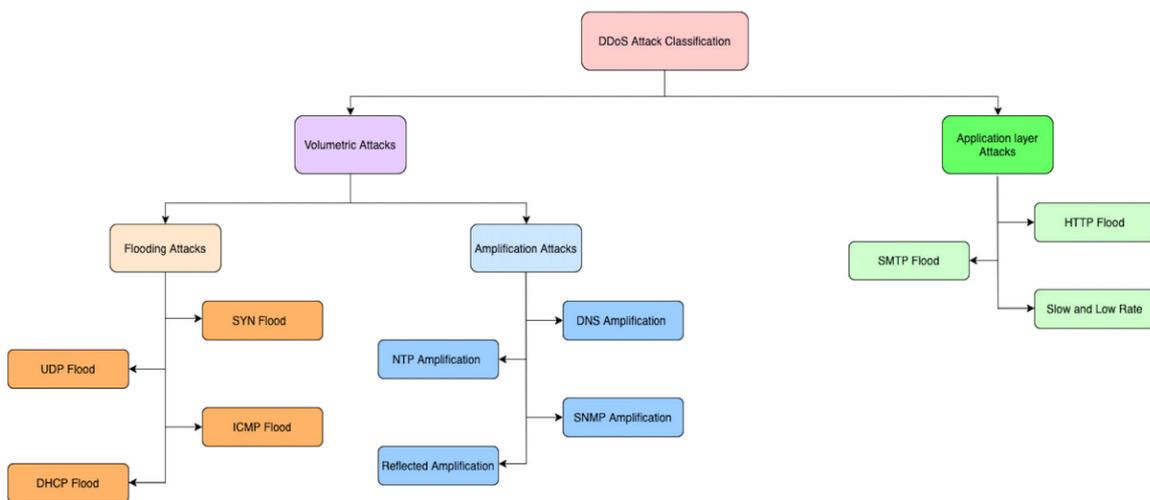
Gambar 2.2 Tren global serangan DDoS [4]

Dapat diamati bahwa pada tiap tahun terjadi peningkatan serangan DDoS. Mulai dari tahun 2018 terdapat serangan DDoS sebanyak 7,9 juta, hingga pada tahun 2023 terdapat 15,4 juta serangan [4].

Laporan Kaspersky menyebutkan bahwa terdapat peningkatan serangan DDoS pada kuartal 1 tahun 2022 yaitu sebanyak 1.5 kali lipat atau sebesar 46% dibanding tahun sebelumnya [43]. Serangan DDoS terbesar terjadi pada tahun 2016 dengan menggunakan botnet Mirai baru, dan serangan tersebut ditargetkan pada server Dyn yang bergerak di sebagian besar infrastruktur *Domain Name Sistem* (DNS) [41]. Mirai adalah sumber utama serangan pada perangkat IoT. Perkiraan

skala serangan pada server Dyn yaitu sebesar 1,2 terabit (1.200 Gbps) dengan sekitar 100.000 agen penyerang [42].

Serangan DDoS dapat di klasifikasikan berdasarkan kerentanan terhadap jaringan dan bot yang tersedia. Pengklasifikasian serangan DDoS dijelaskan pada gambar 2.7.



Gambar 2.3 Taksonomi serangan DDoS [17].

- *HTTP flood attacks*: Serangan banjir HTTP dimaksudkan untuk membanjiri sumber daya server web menggunakan sejumlah besar permintaan HTTP yang dihasilkan oleh botnet [17].
- *Slow- and low-rate attacks*: Dalam kasus ini penyerang menghasilkan lalu lintas ke server web korban dan kemudian menjaga koneksi tetap terbuka (atau aktif) tanpa balasan sampai sumber daya server dikonsumsi [44].
- *Session Initiation Protocol (SIP)*: adalah protokol yang paling umum untuk mengelola pensinyalan di antara pihak-pihak komunikasi untuk menyediakan fungsionalitas yang diperlukan untuk mendaftarkan pengguna, memeriksa status, dan mengelola sesi [45].
- *Reflector attack*: Tujuan utama serangan reflektor adalah untuk menutupi identitas penyerang sebenarnya melalui penggunaan reflektor pihak ketiga dan kemudian memanfaatkan sumber daya mereka [46].
- *DNS amplification attack*: Dalam kasus serangan DDoS jenis ini, penyerang bertujuan untuk mengeksploitasi kerentanan dalam domain name sistem (DNS) untuk mengubah (yaitu, memperkuat) pesan kecil yang awalnya dikirim oleh penyerang menjadi pesan yang jauh lebih besar [42].

- *SYN flooding attack*: Dalam kasus serangan banjir SYN, server korban menerima sejumlah besar paket SYN tetapi tidak pernah menerima ACK terakhir yang diperlukan untuk menyelesaikan metode jabat tangan tiga arah (*three-way handshake*) melalui protokol TCP [17]. Akibatnya, antrian server kewalahan, yang menyebabkan semua permintaan yang masuk dari klien yang sah dibatalkan
- *User Datagram Protocol (UDP) flooding attack*: adalah jenis serangan DDoS di mana penyerang menargetkan dan membanjiri *port* acak pada server yang ditargetkan dengan paket IP termasuk paket UDP [47].
- *ICMP flooding attack*: juga dikenal sebagai serangan *ping*, bertujuan untuk menargetkan server korban dengan sejumlah besar permintaan gema. Server korban yang ditargetkan harus mengirimkan paket respons untuk setiap permintaan yang diterima dari pengirim [17].

### 2.11. Python

Python adalah bahasa pemrograman tingkat tinggi yang sangat populer di kalangan ilmuwan data dan pengembang *machine learning*. Hubungan Python dengan *deep learning* sangat erat karena banyak library dan framework yang tersedia untuk membangun *deep learning* seperti NumPy, Pandas, SciPy, dan TensorFlow [48], [49]. Dalam pengolahan gambar, Python memiliki library OpenCV dan Pillow yang dapat digunakan sebagai pengkonversi data ke dalam bentuk gambar dan mengolah gambar secara lebih kompleks seperti deteksi objek.

### 2.12. Kajian Pustaka

Bagian ini dirancang untuk mengumpulkan dan menganalisis hasil penelitian terdahulu yang telah dilakukan sebelumnya terkait topik klasifikasi serangan DDoS dan penggunaan CNN untuk metode pengklasifikasikan. Dalam kajian pustaka ini, akan dipaparkan metode, teknik yang telah digunakan dan kesimpulan dari penelitian sebelumnya. Tujuannya adalah untuk memberikan landasan teori yang kuat dan komprehensif sebagai dasar dalam melakukan penelitian lanjutan mengenai topik ini.

Penelitian [40] menunjukkan bahwa DDoS adalah serangan yang paling sering terjadi pada jaringan dan perangkat IoT. Penelitian ini menggunakan transfer learning ResNet18 sebagai metode pengklasifikasian serangan DDoS. Hasil penelitian tersebut dibagi menjadi 2, yaitu klasifikasi binary yang digunakan untuk mendeteksi serangan DDoS dan klasifikasi *multi-class* yang menentukan jenis serangan DDoS. Dalam penelitian tersebut, ResNet18 berhasil mengenali serangan DoS dan DDoS dengan persentase 99,9% dan mampu mengklasifikasikan serangan seperti SYN *flood*, UDP *flood*, dan NTP Amplification dengan akurasi sebesar 87% [40]. Hasil dari penelitian ini menunjukkan bahwa penggunaan teknik transfer learning dapat meningkatkan kemampuan sistem untuk mengenali serangan DDoS serta lebih andal dan efektif dibandingkan dengan teknik aturan statis yang telah ditentukan sebelumnya.

Penelitian selanjutnya membahas perbandingan kinerja dari mekanisme pengklasifikasian serangan DDoS pada *Software Defined Network* (SDN). Hasil penelitian terdahulu menunjukkan bahwa metode *machine learning* merupakan pendekatan yang paling efektif dalam mendeteksi serangan DDoS [50]. Namun, metode *machine learning* memerlukan jumlah data yang besar untuk dilatih dan seringkali memerlukan waktu komputasi yang lebih lama untuk diproses. Metode statistik dan metode *threshold*, di sisi lain, tidak memerlukan banyak data dan dapat bekerja dengan cepat [50]. Namun, metode ini seringkali kurang akurat dibandingkan dengan metode *machine learning*. Oleh karena itu, penggunaan metode *machine learning*, dalam mengklasifikasikan serangan DDoS dianggap lebih unggul dibandingkan dengan metode statistik dan *threshold*.

Penelitian yang dilakukan oleh [46] membahas tentang efisiensi penggunaan algoritma *deep learning* untuk mendeteksi aktivitas abnormal seperti DDoS. Penelitian ini menggunakan metode *convolutional neural network* (CNN). Selain itu, algoritma lain juga digunakan seperti *decision tree* (D-Tree), *support vector machine* (SVM), *K-nearest neighbors* (K-NN), dan *neural network* (NN) sebagai pembanding kinerja. Dengan menggunakan iterasi yang sama, semua algoritma di atas dibandingkan kinerjanya sebagai pengklasifikasi serangan DDoS menggunakan kumpulan data serangan DDoS standar seperti NSL-KDD dan hasil simulasi serangan. Hasil penelitian menunjukkan bahwa penggunaan algoritma

CNN bekerja dengan lebih baik daripada pengklasifikasi lainnya dengan akurasi sebesar 99% [46].

Dalam penelitian yang berjudul "*Deep Learning enabled Intrusion Detection and Prevention Sistem over SDN Networks*," yang dilakukan oleh [51], kajian ini mengeksplorasi penerapan teknik *deep learning* untuk meningkatkan keamanan jaringan *Software Defined Network* (SDN). Fokus utama dari penelitian ini adalah merancang dan mengimplementasikan sistem deteksi intrusi dan pencegahan berbasis *deep learning* guna melindungi jaringan SDN dari berbagai serangan siber, khususnya *secure shell* (SSH) *brute-force attacks*. Hasil penelitian menunjukkan bahwa *Multi-Layer Perceptron* (MLP) memiliki kemampuan yang lebih baik dalam menangani masalah non-linear dan kompleks dibandingkan dengan algoritma lain seperti *Convolutional Neural Network* (CNN), *Long Short-Term Memory* (LSTM), dan *Stacked Auto-encoder* (SAE). Dengan demikian, berdasarkan temuan ini, dapat disimpulkan bahwa MLP menjadi salah satu algoritma *deep learning* yang unggul dan memiliki aplikasi luas di berbagai bidang.

Hasil penelitian yang dilakukan oleh [52] mengenai mekanisme deteksi serangan DDoS berbasis *deep learning* dengan menggunakan sFlow dan *adaptive polling* sampling dalam *Software-Defined Networking* (SDN) menunjukkan bahwa penggunaan metode sampling berbasis sFlow dan adaptive polling di lapisan data dapat mengurangi beban jaringan dan memungkinkan *Intrusion Detection System* (IDS) seperti Snort untuk merekam aktivitas jahat secara efektif. Penelitian ini menggunakan Snort IDS dan model *deep learning Stacked Autoencoders* (SAE) untuk meningkatkan akurasi deteksi dengan data lalu lintas jaringan yang dikumpulkan secara *real-time*. Evaluasi menunjukkan bahwa pendekatan berbasis sFlow menghasilkan akurasi deteksi yang lebih tinggi dibandingkan dengan metode *adaptive polling*, dengan tingkat *True Positive* sebesar 95% dan tingkat *False Positive* kurang dari 4%. Hasil ini menunjukkan bahwa penggunaan teknik sampling yang tepat dalam lingkungan SDN dapat secara signifikan meningkatkan kemampuan deteksi serangan DDoS dengan efisiensi sumber daya yang lebih baik.