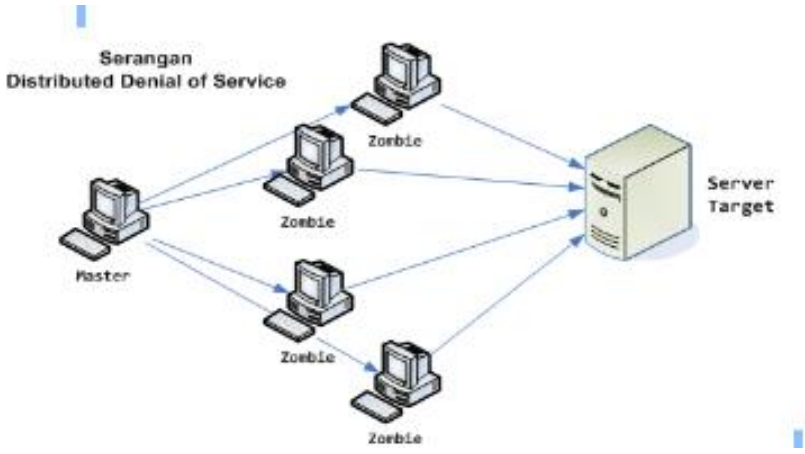
BAB II
TINJAUAN PUSTAKA

2.1 Landasaan Teori

2.1.1. Serangan Distributed Denial of Service (DDoS)

Serangan ini bertujuan untuk menghalangi pengguna mengakses semua sumber daya yang awalnya berasal dari komputer yang menjadi korban. Serangan DDoS adalah jenis serangan terkoordinasi berskala besar, biasanya dilakukan secara tidak langsung dengan memanfaatkan komputer lain yang terhubung ke internet. Ada berbagai jenis serangan DDoS. Serangan DDoS terutama dikategorikan ke dalam dua kelas utama: serangan menghabiskan bandwidth dan serangan menghabiskan sumber daya. Selama serangan bandwidth, jaringan korban akan dibanjiri lalu lintas data, sehingga memblokir lalu lintas yang sah untuk mencapai komputer korban. Sementara itu, serangan sumber daya menargetkan sumber daya dengan berfokus pada menghabiskan sumber daya komputer korban dengan mengirimkan banyak paket data. Serangan DDoS pada lapisan aplikasi terutama dilakukan untuk tujuan tertentu, termasuk mengganggu transaksi dan akses ke database. Serangan jenis ini biasanya membutuhkan sumber daya yang lebih sedikit dan sering dilakukan bersamaan dengan serangan ke lapisan jaringan. Gambaran serangan Distributed Denial of Service dapat dilihat pada Gambar 2.1[30].



Gambar 2. 1 Ilustrasi serangan *Distributed Denial of Service* (DDoS)[30]

Sejumlah penelitian telah mengusulkan *taksonomi* yang berkaitan dengan serangan DDoS [3]. Meskipun semua telah melakukan pekerjaan yang terpuji dalam mengusulkan *taksonomi* baru, cakupan serangan sejauh ini terbatas. Ada kebutuhan untuk mengidentifikasi serangan baru dan menghasilkan *taksonomi*

baru. Oleh karena itu, kami telah menganalisis serangan baru yang dapat dilakukan menggunakan protokol berbasis TCP / UDP di lapisan aplikasi dan mengusulkan taksonomi baru. Sisa dari sub-bagian ini telah dijelaskan *taksonomi* terperinci dari serangan DDoS dan diilustrasikan dalam Gambar 2.3, dalam hal serangan berbasis refleksi dan eksploitasi.

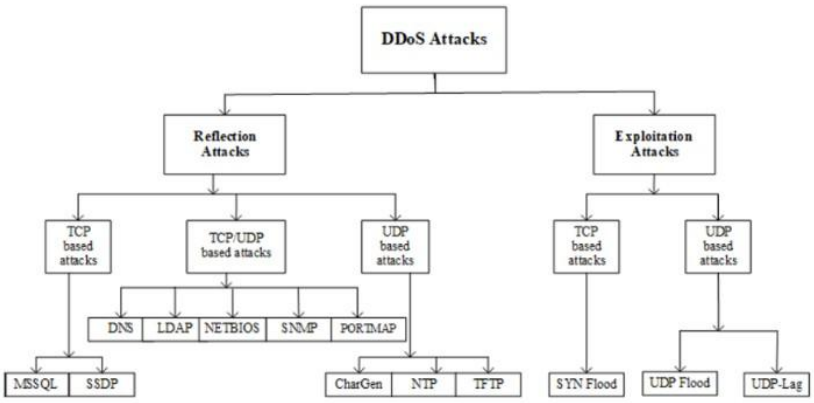
2.1.1.1 Serangan DDoS Berbasis Refleksi

Beberapa jenis serangan memanfaatkan komponen pihak ketiga yang terpercaya untuk menyembunyikan identitas penyerang, dengan cara mengirimkan paket-paket data ke server reflektor menggunakan alamat IP palsu yang disamarkan sebagai alamat IP milik korban. Teknik ini bertujuan untuk membanjiri sistem target dengan respons dari server reflektor, sehingga menyebabkan gangguan layanan. Serangan semacam ini sering dikenal sebagai reflected attacks, dan dapat dilakukan melalui protokol pada lapisan aplikasi serta lapisan transport, seperti *Transmission Control Protocol* (TCP) dan *User Datagram Protocol* (UDP), atau kombinasi keduanya. Berdasarkan klasifikasi yang ditampilkan pada ilustrasi sebelumnya, serangan berbasis TCP mencakup MSSQL dan SSDP, sedangkan UDP digunakan dalam serangan seperti CharGen, NTP, dan TFTP. Selain itu, terdapat pula protokol yang dapat dimanfaatkan melalui TCP maupun UDP, seperti DNS, LDAP, NETBIOS, dan SNMP.

2.1.1.2 Serangan DDoS berbasis Eksploitasi

Serangan semacam itu terjadi saat identitas penyerang disamarkan dengan memanfaatkan komponen pihak ketiga yang valid. Penyerang mengirimkan paket ke server reflektor dengan alamat IP sumber yang diubah menjadi alamat IP target korban, bertujuan untuk membanjiri korban dengan paket balasan. Serangan ini juga bisa dilakukan melalui protokol lapisan aplikasi yang memanfaatkan protokol lapisan transport, seperti TCP dan UDP. Serangan eksploitasi yang menggunakan TCP meliputi banjir SYN, sedangkan serangan yang berbasis UDP mencakup banjir UDP dan UDP-Lag. Serangan banjir UDP dimulai pada host jarak jauh dengan mengirimkan sejumlah besar paket UDP ke port acak di mesin target dengan kecepatan yang sangat tinggi.

Akibatnya, bandwidth jaringan yang ada terpakai habis, menyebabkan sistem mengalami kemacetan dan penurunan kinerja. Sementara itu, banjir SYN juga menggunakan sumber daya server dengan memanfaatkan proses jabat tangan tiga arah TCP. Serangan ini diawali dengan mengirimkan paket SYN secara terus-menerus ke mesin target hingga server mengalami kerusakan atau gangguan. Serangan UDP-Lag adalah tipe serangan yang menginterupsi hubungan antara klien dan server. Serangan ini kerap diterapkan dalam permainan online di mana pemain berusaha menghambat atau mengacaukan gerakan pemain lainnya untuk meraih keuntungan. Serangan ini dapat dilakukan dengan dua cara: menggunakan perangkat keras yang disebut sakelar jeda, atau melalui perangkat lunak yang beroperasi di jaringan dan mengisi bandwidth pengguna lainnya.



Gambar 2. 2 Serangan berbasis Refleksi dan Exploitasi

2.1.1.3 Serangan Pada DDoS

Dalam serangan DDoS, berbagai jenis serangan akan dijelaskan di bawah ini::

a. ***HTTP Flood***. Dalam tipe serangan ini, pelaku memanfaatkan permintaan HTTP GET atau POST untuk menyerang server atau aplikasi. HTTP Flood menggunakan lebih sedikit bandwidth dan paling efektif ketika permintaan paket dapat mendorong target untuk mengembalikan sumber daya maksimum yang mungkin [3].

b. ***UDP Flood***. Penyerang mengirimkan banyak paket *User Datagram Protocol* (UDP) ke target dari port dan host acak. Ini memaksa korban untuk terus memeriksa aplikasi yang mendengarkan di port itu, namun

karena tidak ada aplikasi yang terdeteksi, ia merespons dengan paket *'Destination Unreachable'* yang mengakibatkan habisnya sumber daya [3].

c. ***ICMP Flood*** *(Ping).* ICMP Flood bertujuan untuk menggenangi target dengan permintaan paket ICMP (ping) tanpa menunggu respons [3].
d. ***SYN Flood***. Serangan SYN Flood memanfaatkan protokol handshake tiga langkah dari koneksi TCP. Dalam jabat tangan tiga arah, permintaan SYN dijawab dengan SYN-ACK dari host dan akhirnya ACK dari pemohon. Penyerang secara konstan mengirimkan permintaan SYN tanpa merespons SYN-ACK dari korban atau dengan memanfaatkan alamat IP palsu untuk mengirimkan permintaan SYN. Meskipun begitu, jabat tangan tetap belum memadai dan pada akhirnya menguras lebih banyak sumber daya dari korban.
e. ***Ping of Death.*** Dalam serangan dini kematian, protokol IP diubah untuk mengirimkan malware paket ke sasaran. Ping kematian terkenal dua puluh tahun yang lalu tetapi sekarang tidak efektif sebagai serangan lainnya.
f. ***Slowloris***. Serangan *Slowloris* diarahkan ke server web di mana penyerang memanfaatkan sumber daya minimal untuk menyerang sistem dengan meminta koneksi ke target dan segera setelah koneksi terjalin, penyerang berusaha mempertahankan koneksi terbuka selama mungkin dan mengirimkan paket HTTP palsu untuk melemahkan server web.
g. ***NTP Amplification***. Dalam serangan Amplifikasi NTP, penyerang memanfaatkan paket UDP untuk menyerang server Network Time Protocol publik, yaitu protokol yang digunakan untuk menyelaraskan jam komputer. Ini merupakan serangan amplifikasi karena rasio query terhadap respon adalah serangan semacam itu dapat berkisar antara 1:20 - 1:200 atau lebih.
h. ***Zero-day DDoS attacks***. "Zero-day" merujuk pada istilah untuk segala bentuk serangan yang belum dikenal atau yang baru saja muncul. Serangan-serangan ini memanfaatkan kelemahan yang belum dilindungi oleh mekanisme pertahanan.

2.1.2. Stratified Sampling

Stratified Sampling adalah teknik statistik yang melibatkan pembagian populasi menjadi subkelompok atau strata berdasarkan karakteristik tertentu, kemudian memilih sampel acak dari setiap strata. Metode ini umumnya digunakan dalam studi penelitian untuk memastikan bahwa sampel tersebut representatif terhadap populasi dan untuk meningkatkan presisi estimasi. Dengan melakukan stratifikasi populasi, variabilitas dalam setiap strata berkurang, yang dapat menghasilkan hasil yang lebih akurat. Secara keseluruhan, pengambilan sampel acak berstrata merupakan alat yang berharga bagi peneliti yang ingin mendapatkan sampel representatif dari populasi yang lebih besar[31]. Tujuan utama dari teknik ini adalah untuk meningkatkan representasi sampel, memastikan semua subkelompok penting terwakili, dan meningkatkan presisi serta akurasi estimasi parameter populasi.

Terdapat dua jenis utama dalam pengambilan sampel berstrata yang penerapannya didasarkan pada tujuan penelitian[31]:

1. *Proportionate Stratified Random Sampling* (Pengambilan Sampel Berstrata Proporsional): Jumlah sampel yang diambil dari setiap strata sebanding dengan ukuran strata tersebut terhadap total populasi. Artinya, strata yang lebih besar akan memiliki perwakilan sampel yang lebih banyak. Metode ini digunakan ketika tujuan utamanya adalah untuk menghasilkan sampel yang sangat representatif terhadap populasi secara keseluruhan.
2. *Disproportionate Stratified Random Sampling* (Pengambilan Sampel Berstrata Tidak Proporsional): Jumlah sampel yang diambil dari setiap strata tidak sebanding dengan ukuran strata dalam populasi. Teknik ini digunakan ketika peneliti memiliki alasan khusus untuk memberikan bobot lebih pada strata tertentu, misalnya karena strata tersebut sangat kecil namun penting untuk dianalisis secara mendalam, atau karena variabilitas dalam strata tersebut lebih tinggi.

2.1.3. SMOTE *(Synthetic Minority Over-sampling Technique)*

SMOTE (*Synthetic Minority Over-sampling Technique*) merupakan salah satu metode *oversampling* yang digunakan untuk menangani isu ketidakseimbangan kelas (*class imbalance*) dalam dataset. Ketidakseimbangan kelas muncul saat

jumlah data dalam satu kelas (umumnya kelas mayoritas) jauh lebih tinggi dibandingkan dengan kelas lainnya (kelas minoritas), sehingga algoritma pembelajaran mesin cenderung berpihak pada kelas [32]. Tidak seperti metode *oversampling* konvensional yang sekadar menduplikasi data minoritas, SMOTE beroperasi dengan menciptakan data sintetis. SMOTE melakukan interpolasi antara sampel minoritas dan tetangga terdekatnya, sehingga menciptakan titik data baru di sepanjang jalur antara kedua titik tersebut. Melalui metode ini, distribusi kelas menjadi lebih seimbang tanpa membuat model overfit karena penggandaan data yang identik

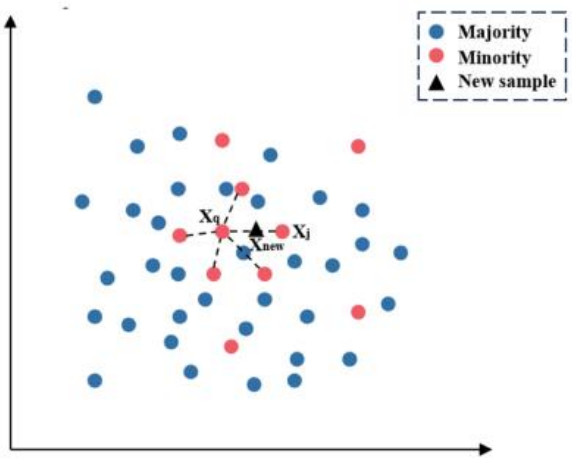
Langkah-langkah algoritmanya adalah sebagai berikut.

Tentukan jumlah sampel kelas minoritas yang akan disintesis, dilambangkan sebagai n_{sempel}.

1) Untuk setiap sampel kelas minoritas yang dipilih X_q, hitung jarak Euclidean ke semua sampel kelas minoritas lainnya. Pilih k tetangga terdekat dari X_q, di mana k = 5 secara default.
2) Pilih secara acak sampel X_j dari k sampel tetangga terdekat dan sintesis sampel kelas minoritas baru X_{new} dengan X_q menggunakan (1).

$$X_{new} = X_q + \alpha(X_q + X_j) \tag{2.1}$$

   Di mana α (0,1) dihasilkan secara acak selama proses operasi.
3) Jika jumlah sampel yang disintesis mencapai pemasangan yang diperlukan, algoritme akan berhenti. Jika tidak, ulangi dari langkah 2.



Gambar 2. 3 Diagram skematik dari proses pembuatan sampel sintetis algoritme SMOTE[32].

2.1.4. *Fiture Selection*

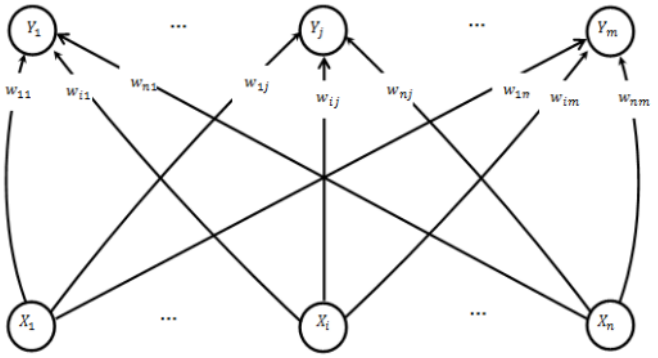
Seleksi fitur adalah teknik untuk memilih subset fitur yang paling relevan dari sejumlah fitur dalam dataset. Proses ini bertujuan untuk mengecilkan dimensi data dengan menghapus fitur yang tidak relevan, redundan, atau bising. Dalam menyederhanakan fitur yang akan digunakan, pemilihan subset fitur memiliki beberapa alasan mendasar [22]:

a. Menyederhanakan data atau model agar lebih mudah dianalisis.
b. Mengurangi waktu training (mengurangi kompleksitas model dan inferensi).
c. Menghindari *curse of dimensionality*.
d. Menghapus fitur yang tidak informatif.
e. Meningkatkan generalisasi dengan mengurangi *overfitting.*

Salah satu metode seleksi fitur adalah menghilangkan atribut yang memiliki nilai variance sama dengan 0. Menurut teori informasi atau entropi, fitur ini memiliki nilai informasi yang rendah. Dengan kata lain, atribut yang tidak dapat membedakan antara satu kelas dan yang lain adalah tidak informatif

2.1.2.1 Learning Vector Quantization (LVQ)

LVQ adalah tipe jaringan saraf buatan yang berfungsi dengan cara mempelajari representasi dari data masukan dan menggolongkannya ke dalam kategori-kategori tertentu. LVQ tergolong dalam kategori pembelajaran terawasi, artinya algoritma ini membutuhkan data yang telah diberi label untuk proses pelatihannya. Sebuah lapisan kompetitif akan secara otomatis mempelajari cara melakukan klasifikasi vektor-vektor masukan. Kelas-kelas yang didapat sebagai hasil dari lapisan kompetitif ini hanya bergantung pada jarak antara vektor-vektor input. Jika dua vektor input hampir serupa, maka lapisan kompetitif akan menempatkan kedua vektor input itu dalam kategori yang sama.



Gambar 2. 4 Asitektur Model Jaringan (LVQ)

Algoritma ini mencari unit output terdekat dengan vektor input. Jika x dan w merupakan kelas yang sama, maka pindahkan bobot terhadap vektor input baru. Jika x dan w merupakan kelas yang berbeda, maka pindahkan bobot dari iput vektor.

a. Euclidian distance

Penentuan jarak yang di gunakan dalam algoritma LVQ yang paling sering adalah euclidian. Euclidian itu sendiri ferfungsi untuk mengkajij ukuran yang dapat yang dapat digunakan sebagai jarak kedekatan antara 2 objek dengan setiap vektor input x, hitung jarak antara x dan setiap vektor bobot w_j. Berikut persaamaan euclidian:

$$\boldsymbol{d(x, w_{ji}) = \sqrt{\sum_{i=0}^{n}(x_i - w_{ji})^2}} \qquad (2.2)$$

Keterangan:

d = jarak data x_i *ke* w_j

x_i = Komponen ke-i dari vektor input

w_{ji} = Komponen ke-i dari vektor bobot

n = Jumlah fitur

b. Pembaruan bobot

Pemrosesan yang terjadi pada setiap neuron adalah mencari jarak antar suatu vektor input ke *weight*/bobot yang bersangkutan yaitu w_1 dan w_2. w_1 merupakan *vektor weight*/bobot yang menghubungkan setiap neuron pada lapisan input ke neuron

pertama pada lapisan *output*, sedangkan w_2 merupakan *vektor weight*/bobot yang menghubungkan setiap neuron pada lapisan input ke neuron kedua lapisan output.

Jika vektor input diklasifikasikan benar (kelas vektor input sama dengan kelas bobot pemenang), maka bobot didekatkan ke vektor input:

$$w_j(t+1) = w_j(t) + a(t) \cdot (x(t) - w_j(t)) \quad (2.3)$$

Jika vektor input diklasifikasikan salah (kelas vektor input berbeda dengan kelas bobot pemenang), maka bobot dijauhkan dari vektor input:

$$w_j(t+1) = w_j(t) - a(t) \cdot (x(t) - w_j(t)) \quad (2.4)$$

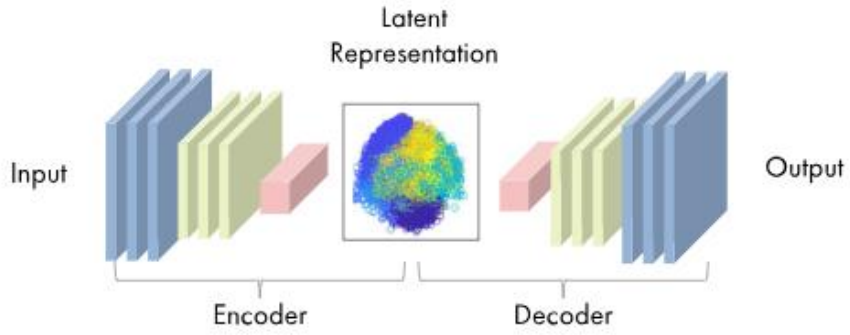
Keterangan:

α(t) = laju pembelajaran yang menurun seiring waktu (0<α<10)

t = iterasi saat ini

2.1.2.2 Autoencoder

Autoencoder (AE) merupakan suatu teknologi pembelajaran mesin yang dapat dimanfaatkan untuk menekan kompleksitas data melalui proses perhitungan yang melibatkan Jaringan Saraf (NN). *Autoencoder* berfungsi dengan mereduksi data masukan menjadi representasi ruang laten atau *compressed representation*, dan kemudian membangun kembali data input dari representasi yang sudah dipadatkan itu. *Autoencoder* memanfaatkan jaringan neural yang terdiri dari dua komponen, yaitu *encoder dan decoder*. Encoder berfungsi untuk menerima data input dan mengubahnya menjadi representasi yang lebih ringkas (*encoding*), sedangkan decoder berperan untuk mengubah representasi itu kembali menjadi data yang serupa dengan data asli (*decoding*). Melalui metode ini, *Autoencoder* mampu memahami data dengan cara yang lebih optimal, sehingga berkontribusi dalam mengurangi kebutuhan ruang penyimpanan dan mempercepat durasi pemrosesan.



Gambar 2. 5 *Autoencoder*

Secara matematis, dapat di tuliskan autoencoder sebagi persamaan berikut:

$$f(\boldsymbol{d}, \theta) = dec\big(enc(\boldsymbol{X})\big) \tag{2.5}$$

Keterangan

$\boldsymbol{dec}$ = decoder

$\boldsymbol{enc}$ = encoder

$\boldsymbol{X}$ = input

Encoder merubah input kedalam bentuk dimensi yang lebih kecil, melalui persamaan

$$C = enc(\boldsymbol{X}) = g(x, \boldsymbol{U}, a) \tag{2.6}$$

Keterangan

$\boldsymbol{C}$ = Encoder

$\boldsymbol{X}$ = Input

$\boldsymbol{U}$ = weight matrik

$\boldsymbol{a}$ = bias

Yang selanjudnya nilai encoder di rekontruksi kembali seperti inputan pertama, yang bisa di sebut dengan decoder.

$$f(\boldsymbol{d}, \theta) = dec(\boldsymbol{C}) = h(\boldsymbol{C}, \boldsymbol{W}, \beta) \tag{2.7}$$

Keterangan

W = Weight matrik

β = Bias

2.1.5. Cyber Scurity

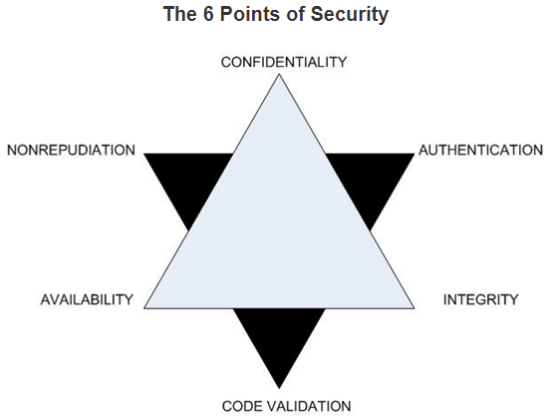
Keamanan jaringan merupakan aktivitas yang ditujukan untuk menjaga fungsi dan keutuhan jaringan serta data Anda. Ini meliputi teknologi perangkat keras serta perangkat lunak. Keamanan jaringan yang optimal mengatur akses ke jaringan dan mengidentifikasi berbagai ancaman serta menghentikannya saat memasuki atau menyebar di dalam jaringan Anda. [33] Sebuah jaringan dapat dianggap aman jika memiliki 3 elemen yang dikenal dengan CIA (*Confidentiality, Integrity, Availability*). Aspek kerahasiaan atau *confidentiality* merupakan elemen dalam keamanan jaringan yang mengatur akses terhadap informasi, di mana hanya individu yang telah memperoleh izin yang dapat mengakses informasi tertentu. Selanjutnya, aspek integritas atau keutuhan mengacu pada tingkat kepercayaan terhadap informasi tertentu, di mana kepercayaan ini mencakup akurasi serta konsistensi dari informasi yang tersedia. Oleh karena itu, diperlukan perlindungan terhadap informasi dari perubahan oleh pihak-pihak yang tidak berwenang. Aspek terakhir yang perlu diperhatikan adalah aspek ketersediaan. Konsep ketersediaan informasi berarti bahwa informasi itu selalu dapat diakses saat diperlukan oleh individu yang memiliki hak atas informasi tersebut.

Gambar 2.5

Gambar 2. 6 Ilustrasi *triad cyber security* [34]

Maka saat diperlukan oleh pengguna, data/informasi bisa segera diakses dan dimanfaatkan. Di samping itu, terdapat tiga konsep utama yang perlu dipahami oleh

para praktisi dan profesional keamanan untuk menerapkan prinsip-prinsip CIA dengan tepat yaitu mencakup tiga aspek penting: *Autentikasi, Otorisasi, dan Nonrepudiasi*. Konsep CIA dapat dilihat pada gambar 2.6.



Gambar 2. 7 Konsep CIA [35]

Autentikasi, otorisasi, dan nonpenyangkalan merupakan tiga konsep utama dalam keamanan informasi. *Autentikasi* berfungsi sebagai cara untuk memverifikasi identitas seseorang atau sumber pesan guna memastikan keasliannya, menjadi langkah awal dalam memastikan bahwa entitas tersebut adalah seperti yang mereka klaim. Setelah melakukan verifikasi identitas, Otorisasi menetapkan hak akses atau izin yang diberikan kepada pengguna, program, atau proses, mengatur apa yang diizinkan bagi pengguna dalam sistem. Sementara itu, *Nonrepudiation* menjamin bahwa pengirim data tidak dapat membantah telah mengirimkan pesan, serta penerima tidak dapat membantah telah menerimanya. Ide ini umumnya diterapkan dengan *kriptografi asimetris*, di mana kunci privat dipakai untuk menandatangani informasi dan kunci publik digunakan untuk verifikasi. Sistem ini menyajikan bukti yang tak terbantahkan mengenai sumber dan penerimaan data, sehingga memperkuat keamanan dan kepercayaan dalam pertukaran informasi. *Kriptografi asimetris* berfungsi penting karena menjamin bahwa hanya pemilik kunci privat yang dapat menandatangani informasi, sedangkan kunci publik dapat digunakan untuk verifikasi oleh siapa pun[34].

Salah satu serangan yang paling terkenal terhadap ketersediaan suatu informasi adalah *Distributed Denial of Service* (DDoS). Tujuan utama dari serangan DDOS adalah untuk mengisi sumber daya yang tersedia bagi pengguna,

sehingga pengguna tidak dapat mengakses informasi yang seharusnya bisa diakses. Jika tiga aspek keamanan jaringan ini terlindungi, diperlukan algoritma klasifikasi untuk mengidentifikasi paket yang masuk ke dalam jaringan komputer. Dalam Tugas Akhir ini, penulis menerapkan algoritma klasifikasi yang berfokus pada faktor internal dan eksternal untuk mengidentifikasi paket yang masuk, sehingga bisa mendeteksi serangan yang terjadi pada jaringan komputer.

2.1.6. *Machine Learning*

Pembelajaran mesin merupakan bagian dari kecerdasan buatan yang memanfaatkan berbagai operasi statistik untuk mengekstrak dan menganalisis data yang diperlukan, mengenali fitur-fitur baru, serta mendukung proses pengambilan keputusan [36]. Tujuan utama dari pembelajaran mesin adalah agar komputer dapat belajar dari data yang diberikan oleh para ahli [37]. Teknik pembelajaran mesin juga mencakup berbagai aturan dan cara yang mengidentifikasi atau memperkirakan pola atau praktik data yang baru. Metode-metode ini bisa digunakan dalam keamanan siber dan dapat dikategorikan ke dalam teknik yang terawasi dan tidak terawasi. Meskipun telah terjadi peningkatan yang besar dalam penerapan pembelajaran mesin di bidang keamanan siber, alat-alat ini masih belum sempurna karena memerlukan pengawasan manusia yang tinggi, dan algoritmanya perlu terus dilatih ulang karena data tidak bisa diotomatisasi dengan cukup baik[38], [39]. *Machine learning* dapat belajar dari data yang diberikan, sehingga memungkinkan penggunaan data historis sebagai masukan untuk meramalkan data di masa mendatang. Keunggulan dari penggunaan metode machine learning adalah karena kesederhanaannya dalam menjalankan proses pembelajaran[40].

2.1.4.1 Tipe Algoritma *Machine Learning*

Algoritma machine learning dapat secara umum dibedakan menjadi Algoritma pembelajaran terawasi (*Supervised Learning*) dan Algoritma pembelajaran tidak terawasi (*Unsupervised Learning*)[14].

a. Algoritma Pembelajaran Terawasi

   Algoritma pembelajaran yang dikendalikan (*Supervised Learning*) umumnya diterapkan untuk mengatasi masalah klasifikasi dan regresi karena mempermudah proses deteksi atau pengambilan keputusan. Ini memanfaatkan data yang telah dipelajari sebelumnya untuk

meramalkan kejadian di masa yang akan datang. Data masukan yang digunakan untuk melatih algoritma pembelajaran adalah yang diberi label [14].

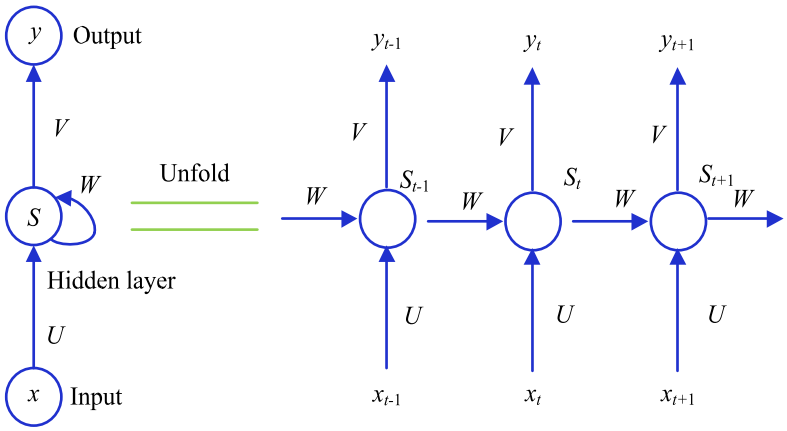
b. Algoritma Pembelajaran Tanpa Pengawasan

Algoritma pembelajaran tanpa pengawasan memanfaatkan data input yang tidak diberi label untuk melatih sistem. Dengan kata lain, data masukan tidak dilengkapi dengan label. Saya menemukan pola tersembunyi dalam data yang tidak berlabel, dan mengelompokkannya ke dalam kategori yang menunjukkan kesamaan. Kinerja awal dari algoritma pembelajaran ini kurang baik, namun sistem dapat beradaptasi untuk memperbaiki kinerjanya [14].

2.1.4.2 *Recurent Neural Network* (RNN)

Recurrent Neural Network, atau RNN, merupakan jaringan neural dalam yang dilatih menggunakan data berurutan atau deret waktu untuk membangun model *machine learning* yang mampu membuat prediksi atau kesimpulan secara berurutan berdasarkan input yang berurutan. RNN adalah evolusi dari Jaringan Saraf Tiruan (JST) dan strukturnya mirip dengan *Multilayer Perceptron* (MLP)[41].

Secara fundamental, *Recurrent Neural Network* (RNN) memiliki parameter yang serupa dengan *neural network* biasa, yang membedakan adalah ide dasar dari RNN itu sendiri. *Recurrent Neural Network* (RNN) menggunakan *hidden layer* sebelumnya sebagai input untuk proses berikutnya. Layer tersembunyi sebelumnya menyimpan informasi dari proses ekstraksi fitur di awal, sehingga ketika layer tersembunyi digunakan sebagai input untuk proses berikutnya, itu mempertahankan memori (ingatan) dari input yang lalu [28].



Gambar 2. 8 Standar RNN

Seperti yang di tunjukan pada gambar 2.7 *xt* mewakili input pada waktu *t*, *St* mewakili *output* dari lapisan tersembunyi pada waktu *t (* bernama memori pada waktu *t*), *yt* mewakili *output* dari lapisan keluar pada waktu *t*, dan [*W*, *U*, *V*] adalah parameter bersama (*W* menunjukkan bobot input, *U* menunjukkan bobot input pada keadaan saat ini, dan *V* menunjukkan bobot output). Di mana *f*(.) adalah fungsi aktivasi, dan *bh* dan *bo* masing-masing adalah vektor bias dari lapisan tersembunyi dan luar. Frekuensi pelatihan metode ini bergantung pada jumlah epoch. Epoch merupakan parameter yang menentukan seberapa sering sebuah algoritma atau metode dijalankan, terutama dalam konteks algoritma yang berkaitan dengan Deep Learning atau Neural Network [42].

Sehingga dapat di jelaskan proses model sebagai berikut:

Persamaan *status hidden layer*

$$S_t = f(Ux_t + WS_{t-1} + b_h) \tag{2.8}$$

Keterangan

s_t = vektor status tersembunyi pada waktu t

w = matriks bobot yang menghubungkan status tersembunyi sebelumnya s_{t-1} ke status saat ini

U = matriks bobot yang menghubungkan input x_t ke status tersembunyi

b_h = bias untuk status tersembunyi

f = fungsi aktivasi *ReLU*

Persamaan keluaran *(output)*

$$y_t = f(Vs_t + b_o) \tag{2.9}$$

di mana:

y_t = keluaran pada waktu ttt

V = matriks bobot keluaran

b_o = bias keluaran

f = fungsi aktivasi keluaran

2.1.4.3 Fungsi Asitektur Recurent Network (RNN)

- *Bidirectional recurrent neural networks* (BRRN)

  RNN searah hanya bisa menggunakan input sebelumnya untuk meramalkan kondisi saat ini, sedangkan RNN dua arah, atau BRNN, memanfaatkan data di masa depan untuk meningkatkan akurasinya.

- *Memori jangka pendek yang panjang* (LSTM)

  LSTM merupakan jenis arsitektur RNN yang terkenal, dikembangkan oleh *Sepp Hochreiter dan Juergen Schmidhuber* sebagai jawaban atas masalah hilangnya gradien. Dalam penelitian mereka (tautan tersedia di luar ibm.com), mereka mencoba menyelesaikan masalah ketergantungan jangka panjang. Dengan kata lain, jika kondisi sebelumnya yang mempengaruhi prediksi saat ini tidak berasal dari masa lalu, model RNN mungkin tidak dapat memprediksi keadaan saat ini dengan akurat.

  LSTM memiliki "sel" dalam lapisan tersembunyi jaringan saraf, yang memiliki tiga gerbang: gerbang input, gerbang output, dan gerbang lupa. Gerbang ini mengatur aliran data yang diperlukan untuk meramalkan keluaran dalam jaringan. Contohnya, bila kata ganti yang mewakili jenis kelamin, seperti "dia", muncul beberapa kali dalam kalimat sebelumnya, Anda dapat menghilangkannya dari status sel.

- *Gated recurrent unit* (GNU)

  GRU mirip dengan LSTM karena GRU juga bertujuan untuk menyelesaikan masalah memori jangka pendek dalam model RNN. Daripada memakai "status sel" untuk mengatur informasi, ia memakai

status yang tidak terlihat, dan alih-alih menggunakan tiga gerbang, ia hanya memiliki dua gerbang, yaitu gerbang reset dan gerbang pembaruan. Seperti pada gerbang di LSTM, gerbang reset dan pembaruan mengatur seberapa banyak dan informasi apa yang perlu disimpan.

2.1.5 *Confusion Matrix*

Confusion matrix berfungsi sebagai alat evaluasi untuk menilai kinerja model klasifikasi dengan memperbandingkan hasil prediksi model dengan data yang sebenarnya. Matriks kebingungan menyajikan informasi yang bersifat prediktif dalam bentuk klasifikasi. Tabel ini mengategorikan hasil prediksi ke dalam empat kelompok utama, yaitu *true positive (TP), false positive (FP), false negative (FN), dan true negative (TN)*. Contoh matriks kebingungan adalah sebagai berikut [42].

Tabel 2.1 *Confusion Matrix*

		Predicted	
		Positive	***Negative***
Actual	***Positive***	TP	FN
	Negative	FP	TN

Keterangan:

a. *True Positive (TP):* Jumlah data aktual dengan nilai positif yang diprediksi sebagai positif.
b. *False Positive (FP):* Jumlah data aktual dengan nilai negatif yang diprediksi sebagai positif.
c. *False Negative (FN):* Jumlah data aktual dengan nilai positif yang diprediksi sebagai negatif.
d. *True Negative (TN):* Jumlah data aktual dengan nilai negatif yang diprediksi sebagai negatif.

Confusion matrix digunakan untuk menghitung beberapa *matrix* evaluasi seperti *accuracy, precision, recall, dan F1-Score*. Berikut persamaan dari masing-masing *matrix* tersebut.

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.10)$$

$$precision = \frac{TP}{TP+FN} \quad (2.11)$$

$$recall = \frac{TP}{TP+FN} \quad (2.12)$$

$$f1 - score = \frac{2*precision*recall}{precision+recall} \quad (2.13)$$

Hasil pengukuran dengan menggunakan confusion matrix di atas dimanfaatkan untuk menilai kinerja model, serta memberikan analisis yang lebih mendalam mengenai distribusi kesalahan prediksi antara kelas positif dan negatif [43]. Pemahaman yang mendalam tentang nilai TP, FP, FN, dan TN sangat penting, contohnya, peningkatan nilai TP dan penurunan FN akan menghasilkan recall yang lebih tinggi, yang sangat signifikan dalam aplikasi di mana kegagalan mendeteksi kasus positif dapat berdampak serius.

2.1.6 Google colab

Google Colaboratory digunakan untuk penelitian ini. Seperti namanya, Google Colaboratory adalah laboratorium online yang dapat diakses oleh semua orang yang menjalankan program Python yang dapat dieksekusi. Aplikasi online ini digunakan bersama dengan Google drive yang berisi dataset untuk eksperimen.[44]

Gambar 2. 9 Google Colab

2.1.7 Phyton

Python adalah bahasa pemrograman yang memungkinkan Anda bekerja dengan cepat dan mengintegrasikan sistem secara lebih efektif. *Python* dikembangkan di bawah *lisensi open source* yang disetujui OSI, membuatnya dapat digunakan dan didistribusikan secara bebas, bahkan untuk penggunaan komersial. *Lisensi Python* dikelola oleh *Python Software Foundation*. [45] Pada Tugas Akhir ini, bahasa pemrograman python akan digunakan sebagai bahasa pemrograman dalam

melakukan pengambilan data, dan juga untuk mengklasifikasi data. Logo Python ditunjukkan pada gambar 2.9.

Gambar 2. 10 Logo *Python*[45]

2.2 Penelitain terkait

Penelitian terkait merupakan upaya penulis atau peneliti dalam meninjau penelitian terdahulu, dimana penelitian-penelitian tersebut memiliki topik teori maupun konsep berkesinambungan dengan penelitian yang akan di lakukan. Beberapa penelitian yang terkait dengan deteksi DDoS attack menggunakan mechince learning sebagai media deteksi dalan serangan cyber security DDoS attack pada suatu jaringan.

Penelitian yang dilakukan Elsayed dkk [17] melilahat sistem deteksi jaringan tradisional mengalami beberapa kekurangan, dan kumpulan data yang digunakan tidak berisi pola serangan terbaru. Peneliti mengusulkan sistem deep learning dengan menggunakan model RNN dengan metode seleksi fitur menggunakan Autoencoder dalam sistem deteksi anomali Serangan DdosNet. Penelitian ini menujukan hasil yang baik degan akurasi 99% dapat mengatasi anomali yang terjadi di dalam jaringan SDN. Pemilihan fitur yang tepat sangat penting dalam meningkatkan akurasi dan efisiensi sistem deteksi serangan dengan menghilangkan fitur yang kurang relevan. Dengan seleksi fitur menggunakan Autoencoder, model dapat lebih fokus pada pola serangan yang signifikan, sehingga meningkatkan kinerja deteksi anomali secara lebih optimal.

Penelitian yang dilakukan oleh Junhong Li [22] menunjukkan bahwa kombinasi jaringan saraf padat, autoencoder, dan Koefisien Korelasi Pearson dapat meningkatkan akurasi deteksi serangan DDoS secara signifikan. Hasil penelitian ini membuktikan bahwa autoencoder berperan penting sebagai kompresor fitur, yang memungkinkan peningkatan akurasi klasifikasi, seperti pada serangan SNMP dari 75,0% menjadi 88,2%, serta peningkatan deteksi serangan WebDDoS hingga 93,3%. Penggunaan autoencoder dalam seleksi dan kompresi fitur terbukti efektif

dalam meningkatkan kinerja sistem deteksi serangan DDoS. Dengan menghilangkan fitur yang kurang relevan dan mempertahankan informasi penting, model dapat lebih akurat dalam mengidentifikasi berbagai jenis serangan, sehingga meningkatkan efisiensi dan keandalan sistem keamanan jaringan.

Dalam penelitian lain yang dilakukan Sangeeta Devi [46] mengembangkan metode deteksi serangan DDoS berbasis pembelajaran transfer dengan memanfaatkan seleksi fitur menggunakan Learning Vector Quantization (LVQ) dan Principal Component Analysis (PCA). Setelah seleksi fitur, data diklasifikasikan menggunakan Decision Tree (DT), Naïve Bayes (NB), dan Support Vector Machine (SVM), dengan hasil menunjukkan bahwa mekanisme berbasis LVQ pada model DT memiliki akurasi tertinggi, yaitu 98,78%. Hasil penelitian ini menunjukkan bahwa metode LVQ lebih efektif dibandingkan PCA dalam mempertahankan informasi penting, yang pada akhirnya membantu model klasifikasi, terutama Decision Tree, dalam mendeteksi serangan secara lebih akurat dan efisien. Temuan dari penelitian ini menujukan pemafaatan teknik seleksi fitur memiliki dampak yang cukup besar dalam sisitem clasifikasi yang mempegahuri akurasinya.

Penelitian yang di lakukan Pravin R. Kshirsagar dkk[26] dilakukan terhadap serangan Cloud besbasis DDoS. Teknik penambangan data analis siber dapat membantu dalam deteksi intrusi. Teknik pemilihan atribut juga penting dalam menjaga dimensi kumpulan data tetap rendah. Dalam penelitian ini disediakan satu metode, dan dataset diambil dari dataset NSL-KDD. Pada strategi pertama, metode penyaringan yang disebut kuantisasi vektor pembelajaran (LVQ) digunakan, dan dalam strategi kedua, metode penyederhanaan dimensi yang disebut PCA. Setelah dilakukan pemilihan fitur selanjudnya dilakukan clasifikasi dengan model Naive bayes, decition tree, dan SVM. Dari hasil percoaan tersebut menunjukkan bahwa SVM berbasis LVQ berkinerja dengan tingkat akurasi 99,85%. Metode LVQ terbukti lebih efektif dibandingkan PCA dalam mempertahankan informasi kritis, sehingga memungkinkan model SVM mencapai performa deteksi yang lebih tinggi dan efisien dalam lingkungan cloud.

Penelitian yang dilakukan Reddy SaiSindhuTheja dkk[47]. Melakukan sistem deteksi serangan DoS yang efisien yang menggunakan Algoritma Pencarian Gagak Oposisi (OCSA), yang mengintegrasikan metode Algoritma Pencarian Gagak (CSA) dan Pembelajaran Berbasis Oposisi (OBL). Dimana OCSA digunakan sebagai teknik seleksi fitur dan RNN sebagai teknik clasifikasi sistem serangan DoS. Tingakt akurasi yang di

dapat dari sistem yang di usulkan dalm penelitan ini cukup baik yaitu mencapai 94.12%. penelitian ini memberikan gambaran teknik seleksi fitur dapat meningkatkan kemampuan suatu algoritma dalam meningkatkan tingkat akurasi dan teknin RNN yang dapat menyipan informasi sebelumnya dapat mengefesiensikan kinerja algoritma sehingga dapat melakukan deteksi secara cepat tanpa harus membacadata ulang kembali data sebelumnya.

Berdasarkan hasil penelitian terdahulu tersebut, dapat di simpulkan bahwa setiap metode pemilihan fitur memiliki kelebihan dan kekuranganya masing-masing. Berbagai metode seleksi fitur, seperti Autoencoder dan Learning Vector Quantization (LVQ), terbukti efektif dalam meningkatkan akurasi deteksi serangan dengan menghilangkan fitur yang kurang relevan dan mempertahankan informasi penting. Penelitian-penelitian tersebut menunjukkan bahwa kombinasi antara teknik seleksi fitur dan algoritma klasifikasi, seperti Recurrent Neural Network (RNN), Decision Tree, Naïve Bayes, dan Support Vector Machine (SVM), dapat menghasilkan tingkat akurasi yang tinggi dalam mendeteksi serangan DDoS. penelitian yang mengusulkan pendekatan hybrid antara RNN dan metode seleksi fitur LVQ-Autoencoder berpotensi untuk meningkatkan efisiensi dan akurasi deteksi serangan DDoS. RNN dapat menyimpan informasi temporal yang penting, sementara LVQ dan Autoencoder dapat membantu dalam mengidentifikasi fitur penting yang signifikan. Dengan menggunakan dataset DDoS evaluation dataset (CIC-DDoS2019) peneliti akan mengexplorasi pengaruh kombinasi antara klasifikasi dan pemilihan fitur terhadap kinerja deteksi serangan DDOS. Di harapkan kombinasi ini dapat menghasilkan model yang lebih efisien dalam sistem deteksi DDoS, sehingga dapat mendukung proses deteksi anomali yang lebih cepat, akurat, dan terjangkau.