

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Tinjauan Pustaka**

Penelitian ini menggunakan studi literatur sebagai sarana untuk kelengkapan data penelitian sekaligus untuk mempertajam masalah yang peneliti kaji. Peneliti telah mengkaji jurnal penelitian terdahulu. Jurnal tersebut dipilih berdasarkan tema dan topik permasalahan yang sesuai penelitian ini. Berikut adalah penjelasan dari jurnal yang dipilih.

Penelitian yang dilakukan oleh hristianto Paradise, dkk [9] yang berjudul “*Analysis of Distributed Denial of Service Attacks Using Support Vector Machine and Fuzzy Tsukamoto*”. Penelitian ini bertujuan untuk mendeteksi serangan DDoS menggunakan *Machine Learning SVM (Suport Vector Machine)* dan *Tsukamoto Fuzzy* dikarenakan dapat memudahkan dalam membedakan lalu lintas jaringan computer ketika mendeteksi serangan DDoS.

Penelitian yang dilakukan oleh Ilham Maulana dan Alamsyah [3] yang berjudul “*Optimalisasi Deteksi Serangan DDoS Menggunakan Algoritma Random Forest, SVM, KNN dan MLP pada Jaringan Komputer*”. Penelitian ini dilakukan karena melanjutkan penelitian terdahulunya, perbedaannya terdapat pada mencari parameter pada penelitian ini menggunakan GridsearchCV dalam mencari sebuah parameter terbaik dan hasil akurasi yang diberikan lebih tinggi dari penelitian sebelumnya serta *Random Forest* memiliki akurasi paling tinggi dibandingkan dengan *Machine Learning* yang lainnya.

Penelitian yang dilakukan oleh Muhammad Misbahul ,dkk [6] yang berjudul “*Analisa Sistem Identifikasi DDoS Menggunakan KNN Pada Jaringan Software Defined Network (SDN)*”. Penelitian ini dilakukan untuk memeriksa Bahasa Pemrograman Python dapat digunakan dengan baik atau tidak dalam mengklasifikasi paket serangan DDoS. Hasil yang didapatkan

Bahasa Pemrograman Python dapat mengklasifikasi dan *Machine Learning* yang digunakan adalah KNN (*K-Nearest Network*) sehingga penelitian ini mengatakan bahwa KNN cocok digunakan dalam mendeteksi serangan DDoS.

Penelitian yang dilakukan oleh Muhammad Aamir dan Syed Mustafa Ali Zaidi [10] berjudul “*Clustering Based Semi-Supervised Machine Learning for DDoS Attack Classification*”. Penelitian ini membahas mendeteksi serangan DDoS dan DoS menggunakan *Random Forest*, KNN, dan SVM, hasil akurasi yang di dapatkan dengan *Random Forest* 96.66%, KNN 95%, dan SVM 92%.

Penelitian yang dilakukan oleh Fadil Febriansyah, dkk [11] berjudul “Deteksi Serangan *Low Rate DDoS* pada Jaringan Tradisional menggunakan Algoritma *Decision Tree*”. Penelitian ini bertujuan membandingkan algoritma Entropy dengan Algoritma Gini Index sebagai menguji sebuah kelayakan dengan parameter dari hasil akurasi yang didapatkan. Metode yang digunakan pada penelitian ini adalah metode *Decision Tree* dalam pengujian nya. Hasil dari penelitian ini mengungkapkan bahwa algoritma *Gini Index* mendapat nilai akurasi yang tinggi sebesar 99.740% dan algoritma *Entropy* memiliki nilai akurasi 99.739%. Berdasarkan dari penelitian tersebut maka Algoritma *Gini Index* lebih sesuai untuk mengukur sebuah akurasi.

Tabel 2. 1 Ringkasan Penelitian Sebelumnya

No	Judul	Tahun	Penulis	Machine Learning	Kelebihan	Kekurangan	Perbedaan
1	<i>Analysis of Distributed Denial of Service Attacks Using Support Vector Machine and Fuzzy Tsukamoto.</i>	2023	Paradise , Wahyu Adi Prabowo , Teguh Rijanandi	SVM dan Tsukamoto Fuzzy	Memiliki tingkat akurasi yang sempurna yaitu 100%.	Memiliki kekurangan variabel yang diteliti jumlahnya masih sedikit yaitu hanya 3 variabel saja.	Menggunakan <i>Dataset</i> yang berbeda dan memiliki jumlah variabel yang lebih banyak
2	Optimalisasi Deteksi Serangan DDoS Menggunakan Algoritma <i>Random Forest</i> , SVM, KNN dan MLP pada	2023	Ilham Maulana , Alamsyah	<i>Random Forest</i> , SVM ( <i>Support Virtual Machine</i> ), KNN ( <i>K-nearest Neighbor</i> ) dan MLP ( <i>Multiple Layer Perceptron</i> )	Memiliki akurasi yang tinggi pada algoritma <i>Random Forrest</i> dengan akurasi 0,9941, jurnal ini memiliki tingkat akurasi lebih tinggi dibanding dengan jurnal terdahulu dikarenakan menggunakan metode	Memakan waktu yang lebih lama dalam memproses pelatihan model dan juga harus menggunakan split data agar data dapat diproses.	Proses <i>pre-processing dataset</i> tidak menggunakan metode GridsearchCV dan menggunakan

No	Judul	Tahun	Penulis	Machine Learning	Kelebihan	Kekurangan	Perbedaan
	Jaringan Komputer.				GridsearchCV pada saat mencari parameter.		<i>dataset</i> yang berbeda.
3	Analisa Sistem Identifikasi DDoS Menggunakan KNN Pada Jaringan <i>Software Defined Network</i> (SDN)	2020	Muhammad Misbahul Azis, Yufis Azhar , Saifuddin	KNN ( <i>K-Nearest Neighbor</i> )	Memiliki akurasi yang sempurna yaitu 1 pada akurasi f-1 score mitigasi yang digunakan pada penelitian ini yaitu menggabungkan algoritma KNN dengan controller untuk memprediksi serangan yang datang sebagai acuan untuk mengaktifkan flowmod.	Ada beberapa paket yang tidak terdapat label dikarenakan pada penelitian ini tidak melalui proses clustering.	Memiliki perbandingan dengan <i>Machine Learning</i> yang lainnya dan menggunakan <i>dataset</i> yang berbeda dengan tahun yang lebih muda.
4	<i>Clustering Based Semi-Supervised Machine Learning for</i>	2019	Muhammad Aamir dan Shahzad, Ghalib A. Shah	Random Forest, KNN, dan SVM	Hasil akurasi yang di dapatkan dengan <i>Random Forest</i> 96.66%, KNN 95%, dan SVM 92%.	Pada jurnal ini kurang dalam penjelasan atau langkah menangani data	Penelitian ini memiliki penanganan dalam menyeimbangkan

No	Judul	Tahun	Penulis	Machine Learning	Kelebihan	Kekurangan	Perbedaan
	<i>DDoS Attack Classification</i>					yang tidak seimbang.	data dan dataset yang digunakan pada penelitian ini tahun 2019 sedangkan dataset pada referensi penelitian adalah tahun 2017
5	Deteksi Serangan <i>Low Rate DDoS</i> pada Jaringan Tradisional menggunakan Algoritma <i>Decision Tree</i>	2023	Fadil Febriansyah , Zian Asti Dwiyanti , Diash Firdaus	Decision Tree	Metode ini membandingkan sebuah algoritma yang digunakan dengan nilai paling besar akurasi ada pada algoritma Gini Index tetapi untuk Recall algoritma Entropy lebih unggul.	Menurut peneliti jurnal ini memiliki kekurangan pada jumlah data dan juga dapat membandingkan dengan algoritma yang lain nya.	Memiliki jumlah data yang besar dan variabel yang banyak.

## 2.2 Landasan Teori

### 2.2.1 *Transmission Control Protokol (TCP)*

*Transmission Control Protokol (TCP)* merupakan salah satu jenis protocol yang memungkinkan kumpulan komputer agar dapat berkomunikasi dan bertukar data di suatu *network* atau jaringan. TCP adalah protocol yang berada di lapisan transpor (baik itu dalam tujuh lapis model referensi OSI atau model DARPA) yang berorientasi sambungan (*connection-oriented*) dan dapat diandalkan (*reliable*). TCP dipakai untuk aplikasi-aplikasi yang membutuhkan keandalan data [12].

### 2.2.2 *Use Datagram Protocol (UDP)*

*Use Datagram Protocol (UDP)* adalah protocol transfer yang mengirimkan data pada kecepatan bit tetap yang dicirikan dengan data yang dikirim dalam paket dengan ukuran tetap dengan interval tetap antara setiap paket. Sumber aliran UDP tidak berusaha untuk mendeteksi jikatujuan penerima datanya ada dan UDP tidak ada fase pembentukan koneksi dan trafik., hanya mengalir dari sumber ke tujuan tanpa adanya umpan balik dari tujuan atau dari node perantara [12].

### 2.2.3 *Cyberattack*

*Cyberattack* merupakan semua jenis tindakan yang sengaja dilakukan untuk mengganggu kerahasiaan (*confidentiality*), integritas (*integrity*), dan ketersediaan (*availability*) informasi. Tindakan ini bisa ditujukan untuk mengganggu secara fisik maupun dari alur logis sistem informasi [13].

### 2.2.4 *Distributed Denial of Service (DDoS)*

*Distributed Denial of Service* atau lebih dikenal dengan singkatan DDoS merupakan aktifitas pengiriman paket dalam jaringan dengan jumlah besar yang ditujukan untuk membanjiri jaringan dengan data sehingga suatu *host* menjadi tidak dapat diakses oleh pengguna yang sebenarnya [14]. DDoS merupakan sebuah percobaan penyerangan dari beberapa sistem komputer yang menargetkan sebuah *server* agar jumlah *traffic* menjadi terlalu tinggi sampai

*server* tidak bisa mengatur requestnya, penyerangan DDoS juga menggunakan beberapa sistem computer [15].

#### 2.2.5 *Server*

*Server* merupakan seperangkat komputer yang berisi program - program yang mampu menghasilkan informasi dan informasi tersebut didistribusikan kepada komputer pengguna yang mengaksesnya. *Server* sederhana dapat berupa satu komputer untuk beberapa layanan aplikasi, atau jika jaringannya lebih kompleks dan rumit, maka *server* dapat disetting hanya untuk memberikan satu atau beberapa layanan saja, sementara layanan yang lain diserahkan kepada *server* yang lain, jadi disini terjadi kolaborasi dan kerjasama dari beberapa *server* untuk memberikan layanan dan informasi kepada beberapa client [16].

#### 2.2.6 *Machine Learning*

*Machine Learning* merupakan salah satu cabang ilmu komputer yang luas dan memiliki tujuan untuk memungkinkan komputer “belajar” tanpa diprogram secara langsung [17]. *Machine Learning* bisa didefinisikan sebagai algoritma yang bertujuan menemukan dan mengaplikasikan pola-pola di dalam data. Algoritma pada *Machine Learning* menggunakan teknik - teknik statistik untuk menemukan pola-pola tersebut. Seringkali data yang dicari polanya berukuran besar. Data yang diolah juga tidak hanya teks, tetapi dapat juga berupa gambar, audio, video, atau aktivitas - aktivitas pengguna selama berselancar atau mengakses internet [18].

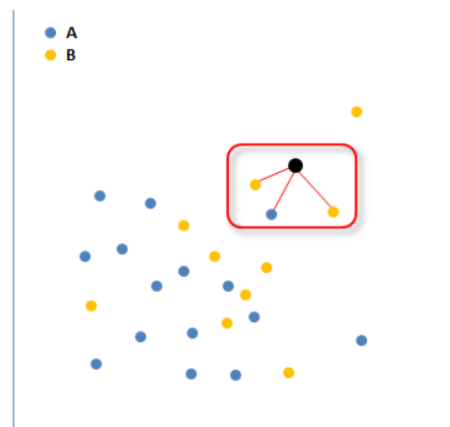
#### 2.2.7 *Deep Learning*

*Deep Learning* (DL) adalah salah satu cabang dari *Machine Learning* dengan konsep yang lebih kompleks untuk pembentukan arsitekturnya. Persamaan yang mendasar pada ML dan DL adalah dengan adanya pengembangan algoritma berdasarkan data, termasuk data yang terstruktur ataupun data yang tidak terstruktur agar bisa digunakan untuk proses training model. DL juga mempunyai model yang lebih rumit dibandingkan dengan model ML dikarenakan adanya layer-layer untuk pembelajaran. Berbeda dengan model *Machine Learning*, DL membutuhkan data yang lebih banyak

yang lebih banyak karena agar DL bisa bekerja dengan baik dan menghasilkan 22 model yang lebih teroptimisasi [19].

### 2.2.8 *K-Nearest Neighbor* (KNN)

Algoritma K-Nearest Neighbor atau (KNN) merupakan metode untuk melakukan klasifikasi terhadap objek yang berdasarkan dari data latih yang jaraknya paling dekat dengan objek tersebut. KNN adalah algoritma *supervised learning* maksudnya hasil dari *query instance* yang baru diklasifikasi berdasarkan mayoritas dari kategori pada algoritma KNN. Dimana kelas yang paling banyak muncul yang nantinya akan menjadi kelas hasil dari klasifikasi [20].



Gambar 2. 1 Arsitektur K-Nearest Neighbor [21]

Gambar 2.1 menunjukkan bahwa algoritma K-Nearest Neighbors (KNN) mencari data terdekat. Data A berwarna biru dan B berwarna kuning, masing-masing merupakan dua kelas yang ditunjukkan pada gambar. Data baru (berwarna hitam) dibandingkan dengan data yang sudah ada. Dalam kasus ini, data baru sangat dekat dengan dua warna kuning dan satu warna biru, seperti yang ditunjukkan dalam kotak merah. Oleh karena itu, dapat disimpulkan bahwa data yang baru diperbarui (hitam) termasuk ke dalam kelas data B (kuning) [21]. Pada saat pembangunan model tentukan parameter yang biasa digunakan dengan mencari label dari K terdekatnya yang



nanti akan menjadi penentuan label ( $n\_neighbor$ ), metode penentuan bobot tetangga ( $weight$ ), dan parameter jarak [20].

#### 2.2.9.1 Menentukan Nilai K

Langkah kritis dan memiliki dampak signifikan terhadap kinerja model. K mengacu pada jumlah tetangga terdekat yang akan digunakan untuk membuat prediksi terhadap suatu titik data baru. Jika Nilai K terlalu kecil  $K=1$  maka dapat menyebabkan *overfitting*, jika nilai K terlalu besar maka dapat mengabaikan perbedaan data [22].

#### 2.2.9.2 Weight

*Weight* digunakan untuk mengambil keputusan, *weight* dibagi menjadi 2 yaitu *uniform* dan *distance*. *Uniform* adalah semua titik antar tetangga memiliki beban dan pengambilan keputusan yang sama. *Distance* adalah beban disesuaikan dengan jarak setiap titik [23].

#### 2.2.9.3 Parameter Jarak

Parameter jarak untuk menghitung jarak antara dua titik untuk parameter jarak yang digunakan adalah *Euclidean*. Rumus *Euclidean* pada persamaan (1) [24]:

$$d(a, b) = \sum_{i=0}^n (X_i - Y_i)^2 \quad (1)$$

Keterangan :

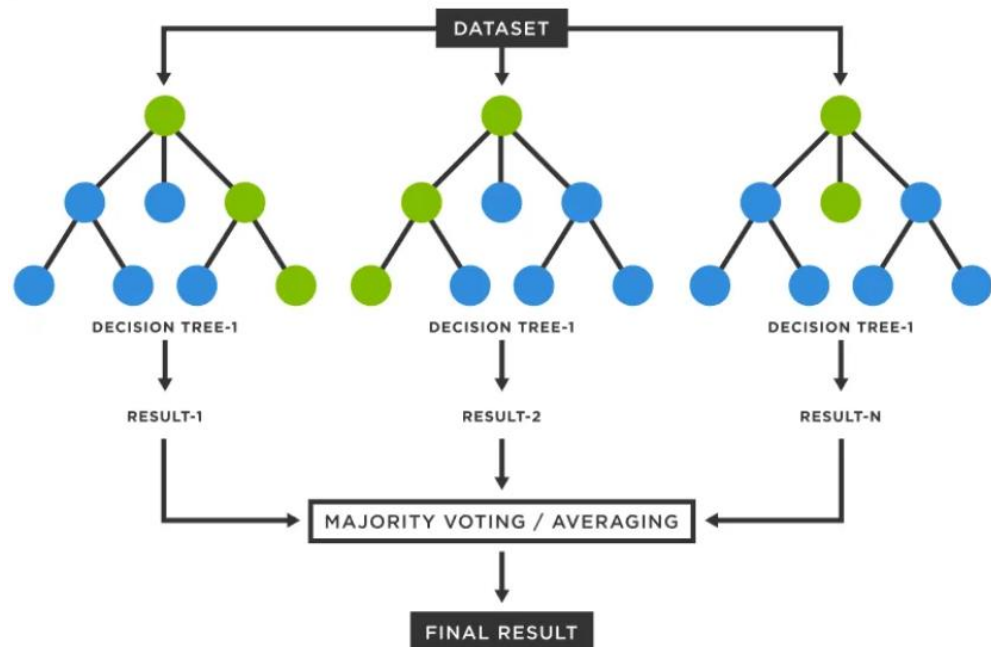
$d(a, b)$  = jarak *Euclidean*

$X, Y$  = data 1, data 2

$i$  = Fitur ke –

$n$  = Jumlah fitur

### 2.2.9 Random Forest



Gambar 2. 2 Arsitektur *Random Forest* [25]

Gambar 2.6 Arsitektur *Random Forest* pertama-tama melibatkan pembuatan sejumlah pohon keputusan ( $n$ ) melalui penggunaan *dataset*. Setiap pohon dalam kelompok menghasilkan keputusan yang berbeda. Selanjutnya, proses voting dilakukan, di mana prediksi dari setiap pohon dihitung dan hasil akhir ditentukan berdasarkan mayoritas suara. Hasil akhir ini adalah prediksi yang paling umum atau paling sering dilakukan di antara prediksi dari seluruh pohon di *Random Forest* [25]. Majority Voting memiliki rumus yang dapat dilihat pada persamaan (4) [26]:

$$f(x) = \operatorname{argmax}_y \sum_{i=1}^K 1(C_i = k) \quad (4)$$

Keterangan :

$K$  = jumlah kelas

$C_i$  = prediksi model ke –  $i$

$1(C_i=k)$  = fungsi infikator jika 1 benar dan 0 salah

*Random Forest* (Pohon Acak) merupakan salah satu jenis metode bootstrap *aggregating* (*bagging*) yang memiliki cara kerja dengan membangkitkan

sejumlah pohon dari data sampel di mana pembuatan satu pohon pada saat proses *training* tidak bergantung terhadap pohon sebelumnya kemudian dalam pengambilan keputusannya diambil berdasarkan voting terbanyak [27]. Parameter yang biasa dilakukan dalam Pembangunan model adalah *n\_estimator*, *max\_depth*, *max\_features*, dan *bootstraps*. *n\_estimator* digunakan untuk menentukan jumlah pohon, *max\_depth* seberapa dalam maksimum dari sebuah pohon, *max\_features* berfungsi untuk maksimal fitur yang digunakan dalam setiap split, dan *bootstraps* difungsikan untuk menanggapi dari *overfitting* [27].

#### 2.2.10 Scikit-Learn

*Scikit-learn* adalah sebuah pustaka analisis data open source, dan standar emas untuk *Machine Learning* (ML) dalam ekosistem Python. Pustaka ini meliputi sejumlah metode algoritma *data mining*, termasuk: klasifikasi, regresi dan *clustering* [28].

#### 2.2.11 Random Under-Sampling

Random under sampling melakukan pemilihan acak data dari kelas mayoritas untuk dihapus dari kumpulan data latih. Dengan melakukan random under sampling, jumlah data latih dari kelas mayoritas akan berkurang. Proses under sampling ini dapat diulang hingga distribusi kelas yang diinginkan tercapai pada data latih. Pendekatan ini cocok untuk digunakan pada kumpulan data dengan kelas yang tidak seimbang, asalkan kelas minoritas cukup untuk pembuatan model. Kekurangan dari undersampling adalah data acak yang dihapus dari kelas mayoritas bisa jadi merupakan data yang penting atau sangat berguna dalam pembangunan model klasifikasi yang baik. Pada undersampling, memungkinkan untuk membuat jumlah data yang sama dari kelas mayoritas dan minoritas atau hanya mengurangi data mayoritas hingga jumlah tertentu [29].

#### 2.2.12 Normalisasi Min Max Scalling

Teknik normalisasi min-max mengubah skala data asli agar semua nilainya berada dalam rentang 0 dan 1. Proses ini membantu memastikan bahwa semua

fitur data memiliki skala yang sama, sehingga tidak ada satu fitur yang mendominasi analisis. Normalisasi min-max dirumuskan pada persamaan (5) [30].

$$W_{norm} = \left( \frac{W_i - W_{min}}{W_{max} - W_{min}} \right) \quad (5)$$

Keterangan persamaan (1), sebagai berikut:

- $W_{norm}$  : Nilai data setelah dinormalisasi
- $W_i$  : Nilai data asli
- $W_{min}$  : Nilai minimum dalam kumpulan data
- $W_{max}$  : Nilai maksimum dalam kumpulan data

### 2.2.13 Matplotlib

Matplotlib adalah pustaka Python yang digunakan untuk membuat berbagai visualisasi data, seperti grafik, diagram, dan plot (Kimm et al., 2021). Pustaka ini menyediakan beragam jenis grafik dan plot, termasuk bar chart, line chart, scatter plot, dan heatmap. Selain itu, Matplotlib juga mendukung berbagai pengaturan tampilan grafik, seperti penambahan label, legenda, dan skala. Dengan kemampuan ini, Matplotlib menjadi alat yang sangat berguna untuk menyajikan data secara visual, sehingga memudahkan dalam menganalisis dan memahami data [31].

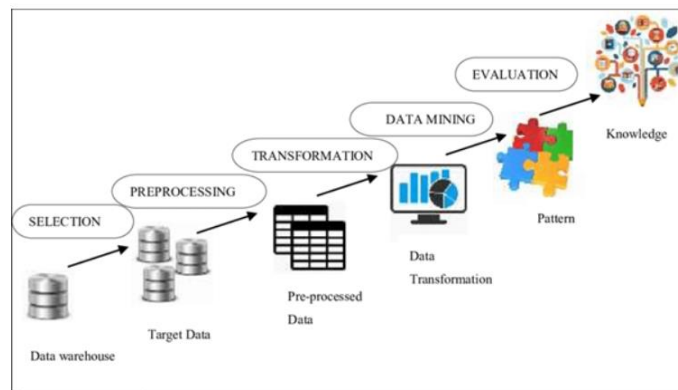
### 2.2.14 Numpy

*NumPy* adalah pustaka fungsi yang memungkinkan Anda melakukan banyak tugas manipulasi data umum dengan Python. Banyak interaksi antara *array NumPy* dan Python sangat mirip dengan apa yang akan Anda lakukan dengan variabel Python biasa. Untuk tugas yang lebih lanjut, *NumPy* juga memiliki fungsi yang bekerja dengan aljabar linier, transformasi *Fourier*, dan matriks[32].

### 2.2.15 Knowledge Discovery in Database (KDD)

*Knowledge Discovery in Database (KDD)* merupakan metode untuk memperoleh pengetahuan dalam basis data yang dibentuk. Dalam basis data tersebut terdapat data-data yang berhubungan. Dari hasil tersebut dapat digunakan menjadi sebagai basis pengetahuan (*Knowledge Base*) untuk

keperluan pengambilan keputusan [33]. Berikut Gambar 2.7 adalah ilustrasi proses dari KDD.



Gambar 2. 3 Ilustrasi Proses Knowledge Discovery in Database [34]

#### 2.2.15.1 Selection

Pada langkah ini, seluruh data dikumpulkan dan diintegrasikan ke dalam satu *dataset*, yang menjadi dasar pemodelan *deep learning*. Data bisa berasal dari dokumentasi, hasil pengambilan data dari internet atau menggunakan *dataset* yang sudah tersedia di *website* seperti Kaggle [35].

#### 2.2.15.2 Preprocessing

Pada proses ini, dilakukan langkah-langkah seperti load data, menghilangkan noise, cleaning data, dan keseimbangan data untuk meningkatkan kualitas data untuk diproses ke proses selanjutnya [35].

#### 2.2.15.3 Transformation

Data transformation merupakan tahapan di saat data ditransformasikan dan disesuaikan untuk kebutuhan data mining. Pada langkah ini, dilakukan transformasi data seperti augmentasi data dengan *resizing* ataupun *scaling* untuk membantu pembelajaran model *Deep Learning* yang lebih baik [35].

#### 2.2.15.4 Data Mining

*Data mining* adalah proses yang menggunakan teknik *statistic*, matematika, kecerdasan buatan dan *Machine Learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai *database* yang besar [36].

#### 2.2.15.5 Evaluation

Proses Evaluasi adalah proses pemeriksaan apakah pola yang ditemukan bertentangan dengan fakta atau hipotesa yang ada sebelumnya. Dalam tahap evaluasi, akan digunakan *Confusion Matrix* untuk mengevaluasi kinerja algoritma CNN dengan melihat hasil evaluasi berupa nilai accuracy [37].

#### 2.2.16 One-Hot Encoding

One-Hot Encoding adalah teknik yang digunakan untuk mengonversi variabel kategori atau label menjadi bentuk numerik. Teknik ini mengubah setiap nilai kategori menjadi vektor biner, di mana setiap nilai kategori memiliki kolom sendiri yang diwakili oleh nilai 0 atau 1. Oleh karena itu, penerapan One-Hot Encoding menjadi langkah penting dalam pengolahan data non-numerik untuk memastikan efektivitas model [38].

#### 2.2.17 Reshaping Data

Reshaping data, atau penataan ulang data, adalah proses mengubah susunan data dari bentuk awal menjadi bentuk yang lebih sesuai untuk keperluan analisis tanpa mengubah nilai aktual dalam data itu sendiri. Mirip seperti mengatur ulang furnitur di ruangan agar fungsinya lebih optimal, reshaping data bertujuan untuk membuat data lebih mudah dibaca, dimanipulasi, dan dianalisis. Data mentah seringkali tidak langsung siap untuk dianalisis; terkadang datanya terlalu melebar (banyak kolom) atau terlalu dalam (banyak baris) sehingga menyulitkan untuk melihat pola atau tren yang ada. Reshaping data membantu mengubah susunannya agar lebih sesuai dengan teknik analisis yang akan digunakan [39].

Reshaping data memiliki beberapa manfaat yaitu mempermudah visualisasi data, mempermudah analisis statistik, dan mempermudah pemahaman data. Data yang sudah di-reshape seringkali lebih mudah dibuat menjadi chart atau grafik yang informatif, lebih siap untuk analisis statistik yang memerlukan format tertentu, dan lebih mudah dibaca serta dipahami, sehingga memudahkan menemukan pola dan insight penting. Dengan memahami konsep dan teknik reshaping data, kita dapat mengubah susunan data menjadi bentuk

yang lebih optimal untuk keperluan visualisasi, analisis statistik, dan pemahaman data secara keseluruhan [39].

### 2.2.18 Confusion matrix

*Confusion matrix* adalah tabel yang menyatakan klasifikasi jumlah data uji yang benar dan jumlah data uji yang salah [40]. Pentingnya melakukan pengukuran ketepatan klasifikasi yaitu untuk melihat seberapa baik model pengklasifikasian yang dilakukan dengan penggunaan *confusion matrix* yang mengukur ketepatan dengan melihat nilai pada tiap kelas prediksi dan kelas actual [41]. Berikut Tabel 2.2 dari *confusion matrix*:

Tabel 2. 2 Confusion Matrix

		Kelas Aktual	
		Negatif	Positif
Prediksi	Negatif	TN	FN
	Positif	FP	TP

TP(*True Positif*):Data positif dan diprediksi benar positif [42]

TN(*True Negatif*):Data negatif dan diprediksi benar negatif [42]

FP(*False Positif*):Data yang diperkirakan positif tetapi diprediksi negatif [42]

FN(*False Negatif*):Data yang diperkirakan negatif tetapi diprediksi positif [42] pengukuran *confusion matrix* yang biasanya sering digunakan :

Accuracy menggambarkan seberapa tepat model dapat melakukan klasifikasi dengan benar [43].

$$Accuracy = \frac{TP+TN}{Total\ Data} \quad (6)$$

1. *Precision* memberikan tingkat ketepatan dari data yang diminta dengan hasil prediksi yang diberikan oleh model yang telah dibuat [43].

$$Precision = \frac{TP}{TP+FP} \quad (7)$$

2. *Recall* memberikan seberapa tepat dalam menemukan sebuah informasi [43].

$$Recall = \frac{TP}{TP+FN} \quad (8)$$

3. *F score* adalah rata-rata *harmonic* antara precision dan recall [44]. Skor F1 dapat dikatakan sebagai perbandingan rata-rata presisi dan recall yang dibobotkan. Skor F1 dapat dikatakan terbaik jika ada semacam keseimbangan antara presisi dan recall dalam sistem. Nilai dari skor F1 dapat dilihat pada persamaan (9) [45].

$$F \text{ Score} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \quad (9)$$