

## BAB II

### KAJIAN PUSTAKA DAN DASAR TEORI

#### 2.1 Kajian Pustaka

Pada penelitian sebelumnya yang telah melakukan penelitian tentang "Klasifikasi Jenis Kelamin Pada Citra Wajah Menggunakan Metode *Naive Bayes*". Penelitian ini menggunakan pendekatan *Naive Bayes*, yang menghasilkan tingkat kecocokan akurasi 80% [8].

Penelitian sebelumnya tentang " Identifikasi Citra Wajah Menggunakan *Probabilistic Neural Network* dengan Ekstraksi Ciri Berbasis *Wavelet* " diterbitkan pada tahun 2019. Menurut penelitian ini, akurasi pada dekomposisi gelombang level 5 dilaporkan sebesar 97,25%, sedangkan akurasi pada dekomposisi *wave level* 6 ditemukan sebesar 95,37%, penurunan sebesar 1,88% [9].

Penelitian sebelumnya dengan judul "Ekstraksi Ciri untuk Klasifikasi Jenis Kelamin berbasis Citra Wajah menggunakan Metode *Compass Local Binary Patterns*". Sebagai pendekatan untuk karakteristik karakterisasi, karya ini menggunakan *Compass Local Binary Patterns* (CoLBP), dan sebagai alat klasifikasi, *Random Forest*. Hasil tes menunjukkan akurasi terbaik rata-rata 91,8% [10].

Sebelumnya dengan judul "Pengenalan Ekspresi Wajah Dari *Cross Dataset* Menggunakan *Convolutional Neural Network* (CNN)". Penelitian ini menghasilkan akurasi data uji sebesar 91,29% [11].

Penelitian dengan judul "Kombinasi Jaringan *Learning Vector Quantization* Dan *Normalized Cross Correlation* Pada 6 Pengenalan Wajah" pada tahun 2020. Untuk menentukan bobot awal untuk pengenalan wajah, penelitian ini mengintegrasikan jaringan LVQ dengan NCC. Menurut temuan tes, kedua pendekatan bersama-sama menghasilkan akurasi 94% untuk pengenalan wajah [7].

Sebelumnya telah dilakukan penelitian dengan judul "Klasifikasi *Gender* berbasis Wajah menggunakan Metode *Local Binary Pattern* dan *Random KNN*". Dalam penelitian ini, proses klasifikasi menggunakan metode random KNN. Hasil pengujian menunjukkan bahwa akurasi yang diperoleh adalah sebesar 75,5% [12].

Penelitian sebelumnya berjudul "Komparasi Metode Klasifikasi Untuk Deteksi Ekspresi Wajah Dengan Fitur *Facial Landmark*". ELM, SVM, dan K-NN adalah model klasifikasi yang digunakan dalam penyelidikan ini. Menurut temuan tes, nilai akurasi adalah 76% dan Skor F1 adalah 76% [13].

Dalam penelitian yang dilakukan dengan judul "Ekstraksi Ciri Untuk Klasifikasi *Gender* Berbasis Citra Wajah Menggunakan Metode *Histogram of Oriented Gradients*". Pada penelitian ini, telah diuji model klasifikasi menggunakan fitur HOG dari dataset *Color FERET*. Hasil pengujian menunjukkan bahwa model ini memiliki kemampuan untuk mengklasifikasikan jenis kelamin pria dan wanita dengan tingkat kebenaran sebesar 97,83% dan 95,92% masing-masing [14].

Penelitian terkait dengan judul "Klasifikasi Usia Berdasarkan Citra Wajah menggunakan Algoritma *Artificial Neural Network* dan *Gabor Filter*". Dalam penelitian tersebut, dilakukan pengujian menggunakan *confusion matrix* yang menghasilkan tingkat akurasi sebesar 83% [15].

Dalam penelitian sebelumnya dengan judul "Metode *Convolutional Neural Network* untuk Klasifikasi Wajah Bermasker", telah dilakukan deteksi wajah yang sedang mengenakan masker menggunakan 3 *layer* dan 5 *layer* dalam proses pelatihan dan validasi dengan jumlah *epoch* sebanyak 20 kali. Penggunaan 3 *layer* menghasilkan tingkat akurasi sebesar 99,20% untuk data *training* dan 70,59% untuk data validasi. Sementara itu, penggunaan 5 *layer* menghasilkan tingkat akurasi sebesar 98,20% untuk data *training* dan 82,35% untuk data validasi [16].

**Tabel 2.1 Perbandingan Penelitian Sebelumnya**

No	Jurnal	Metode	Akurasi Test (%)
1	Klasifikasi Jenis Kelamin Pada Citra Wajah Menggunakan Metode <i>Naive Bayes</i>	<i>Naive Bayes</i>	80%
2	Identifikasi Citra Wajah Menggunakan <i>Probabilistic Neural Network</i> dengan Ekstraksi Ciri Berbasis <i>Wavelet</i>	<i>Probabilistic Neural Network (PNN)</i>	97,25%
3	Ekstraksi Ciri untuk Klasifikasi Jenis Kelamin berbasis Citra Wajah	<i>Compass Local Binary Patterns (CoLBP)</i>	91,8%

No	Jurnal	Metode	Akurasi Test (%)
	menggunakan Metode <i>Compass Local Binary Patterns</i>		
4	Pengenalan Ekspresi Wajah Dari <i>Cross Dataset</i> Menggunakan <i>Convolutional Neural Network (CNN)</i>	<i>Convolutional Neural Network</i>	91,29%
5	Kombinasi Jaringan <i>Learning Vector Quantization</i> Dan <i>Normalized Cross Correlation</i> Pada 6 Pengenalan Wajah	<i>Learning Vector Quantization</i> Dan <i>Normalized Cross Correlation</i>	94%
6	Klasifikasi <i>Gender</i> berbasis Wajah menggunakan Metode <i>Local Binary Pattern</i> dan <i>Random KNN</i>	<i>Local Binary Pattern</i> dan <i>Random KNN</i>	75,5%
7	Komparasi Metode Klasifikasi Untuk Deteksi Ekspresi Wajah Dengan Fitur <i>Facial Landmark</i>	ELM, SVM, dan K-NN	76%
8	Ekstraksi Ciri Untuk Klasifikasi <i>Gender</i> Berbasis Citra Wajah Menggunakan Metode <i>Histogram of Oriented Gradients</i>	<i>Histogram of Oriented Gradients</i>	96%
9	Klasifikasi Usia Berdasarkan Citra Wajah menggunakan Algoritma <i>Artificial Neural Network</i> dan <i>Gabor Filter</i>	<i>Artificial Neural Network (ANN)</i>	83%
10	Metode <i>Convolutional Neural Network</i> untuk Klasifikasi Wajah Bermasker	<i>Convolutional Neural Network</i>	98,20%

## 2.2 Dasar Teori

### 2.2.1 Usia

Wajah manusia dapat mengungkapkan detail terkait usia, manusia dapat menyimpulkan usia melalui indera penglihatan dengan melihat perubahan pola

pada wajah seseorang. Wajah manusia berubah secara signifikan seiring bertambahnya usia, termasuk peningkatan keriput, penyesuaian struktur tulang pipi, dan perubahan jarak antara fitur wajah yang menonjol seperti mata, hidung, dan mulut.

Usia manusia dapat dibagi menjadi berbagai rentang atau kategori, yang sesuai dengan tahap perkembangan individu [15].

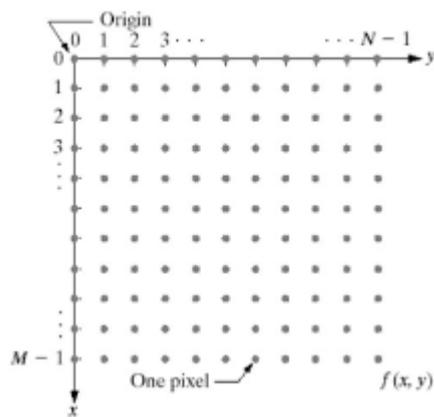
**Tabel 2.2 Kategori Kelompok Umur [15]**

No	Kategori Kelompok Umur	Umur
1	Balita	0 - 5 Tahun
2	Anak-Anak	5 - 11 Tahun
3	Remaja Awal	12 - 16 Tahun
4	Remaja Akhir	17 - 25 Tahun
5	Dewasa Awal	26 - 35 Tahun
6	Dewasa Akhir	36 - 45 Tahun
7	Lansia (Lanjut Usia) Awal	46 - 55 Tahun
8	Lansia (Lanjut Usia) Akhir	56 - 65 Tahun

Pada tabel 2.2 kelompok usia yang diambil diantaranya anak-anak dengan umur 5 sampai 11 tahun, remaja awal dengan umur 12 sampai 16 tahun, remaja akhir dengan umur 17 sampai 25 tahun, dewasa awal dengan umur 26 sampai 35 tahun, dewasa akhir dengan umur 36 sampai 45 tahun, lansia awal dengan umur 46 sampai 55 tahun dan lansia akhir dengan umur 56 sampai 65 tahun.

### 2.2.2 Definisi Citra

Citra dapat dijelaskan sebagai hasil pengambilan gambar suatu objek yang memiliki nilai koordinat khusus, serta memiliki kecerahan warna pada objek tersebut. Citra terdiri dari sejumlah piksel yang diatur dalam bentuk *array* dua dimensi yang menyimpan nilai-nilai aktual dan direpresentasikan dalam urutan *bit* tertentu. Proses sampling mengubah citra *analog* menjadi sejumlah baris (M) dan kolom (N), dengan x dan y sebagai koordinat khusus, dan f mewakili intensitas atau tingkat keabuan pada titik koordinat (x, y) dalam citra [1].



**Gambar 2.1 Koordinat Citra [1]**

Pada gambar 2.1 dapat dilihat citra bisa memiliki makna yang lebih luas tergantung pada konteksnya. Di bidang ilmu komputer, pengolahan citra melibatkan teknik-teknik untuk memanipulasi dan menganalisis citra dengan tujuan tertentu, seperti penyempurnaan kualitas gambar, deteksi objek, segmentasi dan pengenalan pola.

### 2.2.3 Pengolahan Citra

Pengolahan citra merupakan prosedur yang digunakan untuk secara khusus mengubah piksel dalam foto *digital*. Kualitas gambar, pengenalan pola, dan aspek lain dari pemrosesan gambar semuanya ditingkatkan oleh perkembangan mengidentifikasi manusia (biometrik), mendapatkan citra gambar, foto, atau *video* (*content-based image and video retrieval*), mengedit *video*, dan lain sebagainya. Pengolahan citra adalah proses mengubah gambar fisik menjadi gambar *digital* dan menerapkan berbagai operasi di atasnya untuk mengekstrak informasi yang dapat digunakan. Sistem pengolahan citra umumnya memperlakukan semua citra sebagai sinyal 2D ketika menerapkan metode pengolahan sinyal tertentu yang telah ditentukan sebelumnya [1].

### 2.2.4 Python

*Python* merupakan bahasa pemrograman yang tersedia secara *open source* atau *freeware*, sehingga dapat dikembangkan oleh siapa saja. *Python* sering digunakan untuk membuat berbagai jenis program, termasuk program berbasis *Command Line Interface* (CLI), antarmuka grafis (GUI), aplikasi *mobile*, *web*, dan

*Internet of Things*. Bahasa *Python* digunakan untuk mengembangkan aplikasi, menulis perintah komputer, dan melakukan analisis data. Sebagai bahasa pemrograman yang serbaguna, *Python* dapat digunakan untuk menulis program dalam berbagai bidang dan memecahkan berbagai masalah. *Python* juga dikenal sebagai bahasa pemrograman tingkat tinggi. Dalam dunia profesional, *Python* digunakan oleh berbagai profesi, mulai dari pengembang *back-end*, profesional IT, hingga ilmuwan data [17]. *Python* memiliki beberapa pustaka yang sering digunakan diantaranya:

1. *Matplotlib*

Untuk membuat visual dua dan tiga dimensi (2D dan 3D) yang sangat baik, pengembang menggunakan *Matplotlib*. Banyak aplikasi ilmiah menggunakan *Matplotlib*. Data dapat direpresentasikan secara grafis menggunakan berbagai format grafik, termasuk grafik batang dan grafik garis. Selain itu, dapat memplot beberapa grafik secara simultan. Keuntungan menggunakan *Matplotlib* adalah portabilitasnya yang dapat dijalankan di berbagai *platform* [17].

2. *Pandas*

Pengguna dapat menjalankan operasi pada data deret waktu dan data terstruktur, seperti tabel dan deret, menggunakan struktur data *Pandas* yang efisien dan mudah beradaptasi. *Pandas* dapat digunakan, misalnya, untuk membaca, menulis, menggabungkan, memfilter, dan mengelompokkan data. [17].

3. *NumPy*

*NumPy* merupakan sebuah pustaka yang banyak digunakan oleh para *developer* untuk mempermudah pembuatan dan pengelolaan deret, manipulasi bentuk logis, serta melakukan operasi aljabar linier. *NumPy* juga mendukung integrasi dengan berbagai bahasa seperti C dan C++ [17].

4. *Keras*

*Keras* merupakan sebuah pustaka dalam bahasa pemrograman *Python* yang digunakan untuk jaringan saraf dengan dukungan yang luar biasa dalam pemrosesan data, visualisasi, dan berbagai fitur lainnya. Pustaka *Keras*

mendukung berbagai jenis jaringan saraf dan memiliki struktur modular yang memungkinkan fleksibilitas dalam menulis aplikasi inovatif [17].

### 2.2.5 *Tensorflow*

*TensorFlow* merupakan sebuah perangkat lunak yang dapat digunakan untuk melakukan perhitungan numerik menggunakan grafik aliran data secara fleksibel dan terukur. Pengguna dapat secara efektif membangun dan melatih jaringan saraf dan model pembelajaran mesin lainnya menggunakan perpustakaan ini dan banyak alat terkaitnya, serta menggunakannya dalam pengaturan dunia nyata. Algoritma inti *TensorFlow* ditulis dalam bahasa C++ dan menggunakan CUDA (*Compute Unified Device Architecture*) yang optimal untuk komputasi paralel, sebuah *platform* dan API (*Application Programming Interface*) yang dikembangkan oleh NVIDIA. *TensorFlow* juga memiliki API yang tersedia dalam beberapa bahasa. API *Python* merupakan yang paling lengkap dan stabil, dan bahasa lainnya yang secara resmi didukung termasuk *JavaScript*, *C++*, *Java*, *Go*, dan *Swift*. [18].

### 2.2.6 *Convolutional neural network (CNN)*

*Convolutional Neural Network* salah satu pendekatan pembelajaran mendalam yang digunakan di bidang pemrosesan gambar dan visi komputer adalah jaringan saraf konvolusional (CNN). CNN digunakan untuk aplikasi seperti segmentasi wilayah, deteksi objek, dan klasifikasi gambar. Jaringan saraf manusia berfungsi sebagai inspirasi untuk desain CNN, yang terdiri dari lapisan dengan dimensi tiga dimensi (3D) (lebar, tinggi, dan kedalaman). Sementara dimensi kedalaman berkaitan dengan jumlah lapisan di CNN, dimensi lebar dan tinggi sesuai dengan ukuran lapisan [16].

*Feature extraction layer* terdiri dari beberapa bagian, diantaranya:

#### 1. *Convolutional Layer*

*Convolutional layer* merupakan salah satu komponen utama dalam arsitektur *Convolutional neural network (CNN)*. *Layer* ini bertanggung jawab untuk mengekstraksi fitur-fitur penting dari *input* gambar atau citra. Pada setiap *convolutional layer*, terdapat sejumlah filter atau *kernel* yang berukuran kecil,

biasanya berbentuk matriks dengan bobot yang harus dipelajari oleh jaringan. Filter ini digeser secara berulang-ulang pada seluruh *input* gambar dengan langkah tertentu (*stride*), dan dilakukan operasi konvolusi antara filter dan bagian *input* yang sedang diperiksa. Operasi konvolusi ini melibatkan perkalian elemen-elemen matriks filter dengan elemen-elemen yang sesuai dari *input*, dan hasilnya dijumlahkan untuk menghasilkan *output* dari filter tersebut [11].

## 2. *Stride*

*Stride* merupakan sebuah parameter yang mengatur jumlah pergeseran filter pada saat operasi konvolusi. Jika nilai *stride* adalah 1, maka filter konvolusi akan digeser sebanyak 1 piksel secara *horizontal*, dan kemudian juga secara *vertikal*. [16].

## 3. *Padding*

*Zero Padding* merupakan suatu parameter yang menentukan jumlah piksel (yang memiliki nilai 0) yang akan ditambahkan di setiap sisi *input*. Penggunaan *Zero Padding* bertujuan untuk memanipulasi dimensi *output* dari lapisan konvolusi (*Feature Map*). Dengan menambahkan piksel nol di sekitar *input*, ukuran *output* dapat dipertahankan atau diperbesar, yang memungkinkan pemrosesan lebih lanjut dengan menjaga informasi spasial dari gambar atau data masukan [11].

Untuk menghitung dimensi dari *feature map*, dapat digunakan rumus sebagai berikut [11]:

$$output = \frac{W-N+2P}{S} + 1 \quad (2.1)$$

W : Panjang / tinggi *input*

N : panjang / tinggi filter

P : *Zero Padding*

S : *Stride*

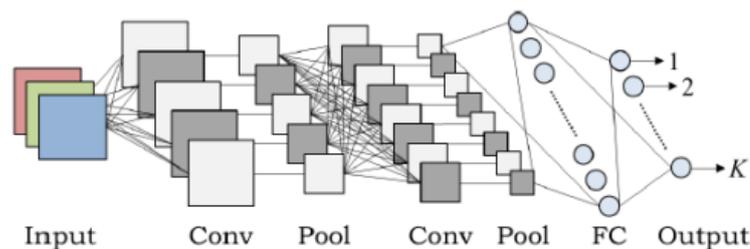
## 4. *Pooling Layer*

*Pooling layer* merupakan salah satu komponen yang sering digunakan dalam arsitektur *Convolutional neural network* (CNN) setelah *convolutional layer*. Tujuannya untuk mengurangi dimensi spasial (*spatial dimension*) dari fitur yang diperoleh dari *convolutional layer*.

Pada setiap *pooling layer*, dilakukan operasi *downsampling* atau pengurangan ukuran spasial pada setiap fitur *map* yang dihasilkan dari *convolutional layer*. Operasi ini dilakukan dengan membagi fitur *map* menjadi daerah-daerah (*region*) yang disebut *pooling windows* atau *kernel*, kemudian memilih nilai yang representatif dari setiap *region* tersebut. Ada beberapa jenis *pooling layer* yang umum digunakan, seperti *Max Pooling* dan *Average Pooling*. *Max Pooling* memilih nilai maksimum dari setiap *region*, sementara *Average Pooling* mengambil nilai rata-rata. Kedua jenis *pooling* tersebut dapat diterapkan dengan ukuran dan langkah (*stride*) yang dapat disesuaikan [16].

#### 5. *Fully-Connected Layer* (FC Layer)

*Fully-Connected Layer*, juga dikenal sebagai *Dense Layer*, terletak di akhir CNN dan bertanggung jawab untuk melakukan klasifikasi atau regresi berdasarkan fitur-fitur yang diperoleh dari *layer* sebelumnya yang terlihat pada gambar 2.2. Setiap *neuron* dalam *Fully-Connected Layer* terhubung dengan semua *neuron* di *layer* sebelumnya, dan menghitung produk titik antara *input* dan bobot yang terkait. Setelah melalui fungsi aktivasi, hasilnya digunakan untuk menghasilkan *output*. *Fully-Connected Layer* membantu menggabungkan fitur-fitur menjadi representasi yang lebih abstrak untuk membuat keputusan akhir [11].



**Gambar 2.2** Arsitektur CNN [16]

#### 6. *Rectified Linear Unit* (ReLU)

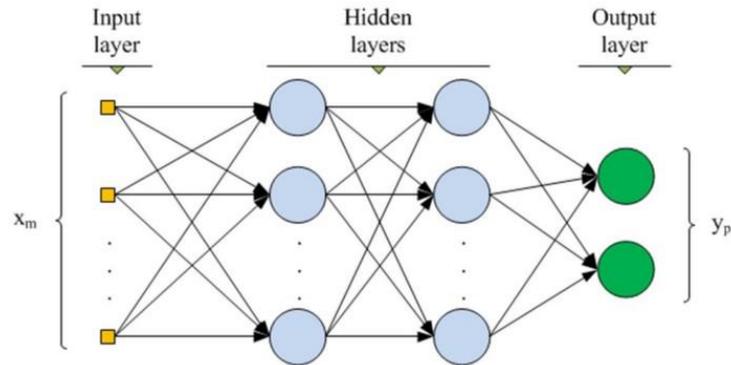
ReLU merupakan suatu fungsi yang menghasilkan keluaran *neuron* 0 jika nilainya negatif, sedangkan jika nilainya positif, keluaran *neuron* akan sama dengan nilai *inputnya*. ReLU merupakan salah satu jenis fungsi aktivasi. Penggunaan ReLU memiliki keunggulan dalam pemrosesan data yang besar dengan kecepatan yang tinggi [11]. Berikut bentuk persamaan fungsinya [11]:

$$f = \max(0, x) \quad (2.2)$$

f : fungsi ReLU

$x$  : nilai batas ReLu [11]

Pada umumnya, *deep learning* melibatkan penggunaan lebih dari tiga lapisan (*layers*) dan dalam konteks pembelajarannya, sering disebut *Convolutional neural network* (CNN) [16].



**Gambar 2.3** Klasifikasi *Convolutional neural network* [16]

Pada gambar 2.3 merupakan tiga bagian dari lapisan *Convolutional neural network* (CNN).

1. *Input layer*

*Input Layer* adalah lapisan pertama dalam jaringan saraf dan bertanggung jawab untuk menerima *input* dari luar. *Input* dapat berupa fitur atau data mentah dari masukan yang ingin diproses oleh jaringan. Setiap *neuron* dalam *Input Layer* mewakili nilai *input* yang sesuai. Jumlah *neuron* dalam *Input Layer* ditentukan oleh dimensi *input* yang diberikan. Misalnya, dalam pengenalan gambar, setiap *neuron* dapat mewakili intensitas piksel pada lokasi tertentu dalam citra [16].

2. *Hidden layer*

*Hidden Layer* merupakan lapisan-lapisan di antara *Input Layer* dan *Output Layer*. Jumlah dan ukuran *Hidden Layer* dapat bervariasi tergantung pada kompleksitas tugas dan arsitektur jaringan yang digunakan. Setiap *neuron* dalam *Hidden Layer* mengambil *input* dari lapisan sebelumnya (baik itu *Input Layer* atau *Hidden Layer* sebelumnya) dan menghasilkan *output* yang akan menjadi *input* untuk lapisan berikutnya. Proses ini melibatkan komputasi bobot dan bias yang dipelajari selama pelatihan jaringan. *Hidden Layer* berfungsi

untuk mengambil fitur-fitur yang lebih abstrak dari *input* dan mempelajari representasi yang lebih kompleks [16].

Lapisan ini melakukan perhitungan dengan mengalikan nilai dari lapisan *input* dengan bobot yang telah diinisialisasi, kemudian menambahkan nilai bias ke hasil perkalian tersebut [16]. Dapat menggunakan rumus sebagai berikut [16]:

$$z_{in_i} = \sum_{j=1}^n X_j * V_{j,i} + V_{o,i} \quad (2.3)$$

$z_{in_i}$  : *input node hidden layer* ke-i dengan jumlah *node* n

$X_j$  : *node X* ke-j

$V_{j,i}$  : bobot V untuk  $X_j$  dengan *node*  $Z_i$

$V_{o,i}$  : bias V untuk  $z_{in_i}$

Selanjutnya dengan memasukkan fungsi aktivasi ReLU pada seluruh hasil perhitungan sehingga didapatkan nilai *output* Z .Kemudian digunakan pada proses perhitungan di *output layer* [16].

### 3. *Output layer*

*Output Layer* merupakan lapisan terakhir dalam jaringan saraf dan bertanggung jawab untuk menghasilkan *output* akhir berdasarkan representasi yang dikembangkan oleh lapisan sebelumnya. Jumlah neuron dalam *Output Layer* bergantung pada jenis tugas yang ingin diselesaikan. Misalnya, untuk klasifikasi biner, *Output Layer* biasanya memiliki satu neuron yang mengeluarkan nilai probabilitas untuk kelas positif. Untuk klasifikasi multikelas, jumlah neuron di *Output Layer* akan sesuai dengan jumlah kelas yang ada. *Output Layer* juga dapat menggunakan fungsi aktivasi khusus, seperti *sigmoid* untuk klasifikasi biner atau *softmax* untuk klasifikasi multikelas, untuk menghasilkan *output* yang sesuai dengan tugas yang diberikan [16].

Pada tahap ini, nilai hasil dari lapisan tersembunyi dikalikan dengan bobot yang telah diinisialisasi, dan kemudian nilai tersebut ditambahkan dengan nilai bias [16]. Dapat menggunakan rumus sebagai berikut [16]:

$$y_{in_i} = \sum_{j=1}^m Z_j * W + W_{o,i} \quad (2.4)$$

$y_{in_i}$  : masukan *node hidden layer* Z ke-i dengan jumlah *node* m

$W_{j,i}$  : bobot W untuk  $Z_j$  dengan *node*  $Y_i$

$W_{o,i}$  : bias W untuk  $y_{in_i}$

$Z_j$  : *node Z ke-j*

Dalam penelitian ini, fungsi aktivasi *softmax* digunakan untuk menghasilkan nilai *output*  $Y$  dari semua perhitungan yang dilakukan [16]. Dapat digunakan rumus sebagai berikut [16]:

$$Y_i = \frac{e^{y\_in_i}}{\sum_{i=1}^m e^m} \quad (2.5)$$

$M$  : semua masukan untuk *output layer* sejumlah  $m$

$y\_in_i$  : imput *node layer* ke- $i$

$Y_i$  : *output* untuk *output layer* ke- $i$  [16]

### 2.2.7 Confusion matrix

Dapat dilihat pada tabel 2.3, *confusion matrix* merupakan teknik evaluasi yang berharga untuk mengukur efisiensi dan akurasi dalam proses kategorisasi. Tabel dengan empat kombinasi nilai yang diharapkan dan nilai aktual adalah *Confusion matrix*. *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) adalah empat istilah yang digunakan dalam *Confusion matrix*. Istilah-istilah ini masing-masing merupakan hasil dari prosedur klasifikasi yang telah dilakukan. Berikut beberapa metrik evaluasi, seperti akurasi, *precision*, dan *recall*, menggunakan *Confusion matrix*. [19].

**Tabel 2.3 Confusion matrix [19]**

		Prediksi	
		TRUE	FALSE
Aktual	TRUE	True Positive	False Positive
	FALSE	False Negative	True Negative

*True Positive* : Jumlah data aktual yang sebenarnya benar diprediksi benar [20].

*True Negative* : Jumlah data aktual yang sebenarnya salah diprediksi salah [20].

*False Positive* : jumlah data aktual yang sebenarnya benar diprediksi salah [20].

*False Negative*: jumlah data aktual yang sebenarnya salah diprediksi benar [20].

Berikut merupakan rumus untuk mengukur seberapa tingkat kebenaran suatu proses klasifikasi dari sistem yang telah dibuat.

#### 1. Accuracy

Akurasi merupakan persentase dari total data yang di prediksi dan dinilai benar [20].

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} 100\% \quad (2.6)$$

TP : *True Positive*

TN : *True Negative*

FP : *False Positive*

FN : *False Negative*

## 2. *Precision*

*Precision* merupakan data yang diambil berdasarkan informasi yang kurang tepat, tidak tepat dan salah [20].

$$Precision = \frac{TP}{(TP+FP)} 100\% \quad (2.7)$$

TP : *True Positive*

FP : *False Positive*

## 3. *Recall*

*Recall* merupakan data yang tidak di prediksi dengan benar [20].

$$Recall = \frac{TP}{(TP+FN)} 100\% \quad (2.8)$$

TP : *True Positive*

FN : *False Negative*