

## **BAB 2**

### **DASAR TEORI**

#### **2.1 KAJIAN PUSTAKA**

Klasifikasi dengan metode CNN sudah banyak di implementasikan di berbagai bidang. Referensi dan rujukan hasil penelitian sebelumnya yang berhubungan dengan penelitian yang akan dilakukan merupakan tinjauan dari Pustaka ini.

Penelitian yang dilakukan pada tahun 2018 yang meneliti tentang klasifikasi tanaman pada citra resolusi tinggi dengan metode CNN. Penelitian ini bertujuan untuk klasifikasi semantik jenis tanaman pada citra resolusi tinggi menggunakan dataset dari lima jenis tanaman yaitu : kelapa, pisang, cabai, bawang merah, dan padi yang diambil menggunakan teknologi UAV dengan data latih yang didapat sebanyak 350 data, data validasi sebanyak 100 data dan pengujian data sebanyak 50 data citra. Proses *learning* jaringan menghasilkan akurasi 100% terhadap data *training*. Pengujian terhadap data validasi menghasilkan akurasi 93% dan akurasi terhadap data tes 82% [8].

Kemudian penelitian yang dilakukan pada tahun 2019 yang meneliti tentang jenis kendaraan menggunakan metode CNN. Penelitian ini bertujuan untuk mengurangi masalah kemacetan dan melakukan pengawasan penggunaan jalur lalu lintas, menggunakan 3 jenis kendaraan yaitu: mobil, motor dan sepeda yang diambil menggunakan aplikasi *Fatkun Batch Downloader* dengan data latih sebanyak 90 citra, data uji sebanyak 30 citra. Berdasarkan hasil pengujian evaluasi model menunjukkan tingkat akurasi sebesar 94,4% pada data pelatihan dan pada data pengujian tingkat akurasi mencapai 73,3% [9].

Penelitian lainnya yang dilakukan pada tahun 2020 yang meneliti tentang perbedaan batik Riau dan bukan batik Riau dengan metode CNN. Penelitian ini bertujuan untuk mengklasifikasikan perbedaan antara batik riau dengan batik bukan riau, kelas yang digunakan dalam penelitian ini yaitu batik riau dan juga batik bukan riau dengan jumlah data sebanyak 168 gambar. Berdasarkan hasil uji didapatkan akurasi sebesar 65% [10].

Penelitian yang dilakukan pada tahun 2020 yang meneliti tentang klasifikasi sidik jari dengan metode CNN menggunakan Resnet-50. Penelitian ini menggunakan *Convolutional Neural Network* (CNN) dengan model arsitektur Residual Network-50 (ResNet-50), Dataset yang digunakan diperoleh dari website National Institute of Standards and Technology (NIST) berupa citra sidik jari *grayscale* 8-bit, Total dataset yang digunakan sejumlah 2100 citra dengan 429 citra berpola *arch*, 403 citra berpola *tented arch*, 402 citra berpola *left loop*, 410 citra berpola *right loop*, dan 456 citra berpola *whorl*, berdasarkan dataset diatas mendapatkan hasil pengujian yang menunjukkan proses awal *Contrast Limited Adaptive Histogram Equalization* (CLAHE) dalam model CNN dapat meningkatkan performa akurasi dari sistem klasifikasi sidik jari mencapai 11,79%. Pada citra tanpa CLAHE diperoleh akurasi validasi 83,26%, sedangkan citra dengan CLAHE diperoleh akurasi validasi 95,05% [11].

Penelitian yang dilakukan pada tahun 2020 yang meneliti tentang sistem pengenalan rambu lalu lintas dengan metode HOG dan klasifikasi KNN berbasis Raspberry Pi. Penelitian ini bertujuan untuk memudahkan pengemudi mendapatkan informasi bahwa telah berada di area rambu pembatas kecepatan secara otomatis, Penelitian menerapkan metode *Histogram Of Oriented Gradients* (HOG) untuk mendapatkan ekstraksi fitur karakteristik dari rambu, kemudian mengklasifikasi rambu menggunakan metode *K-Nearest Neighbor* (KNN). Hasil akurasi pengenalan rambu lalu lintas menggunakan metode HOG berdasarkan jarak hasil deteksi mencapai rata-rata sebesar 95.83%. Hasil dari pengujian klasifikasi menggunakan metode K-NN yang terdiri dari 650 data *training* dan 48 data *testing* untuk k=1 dan k=2 mendapatkan akurasi sebesar 97.91% sedangkan untuk k=3, k=4, dan k=5 mendapatkan akurasi sebesar 95.83%, dan untuk rata-rata waktu komputasi dapat mengenali objek yaitu 897 milisecond. Dapat disimpulkan bahwa hasil akurasi penelitian ini berjalan dengan sangat baik dengan tingkat akurasi yang sangat tinggi [5].

## 2.2 DASAR TEORI

### 2.2.1 *Artificial Intelligence*

Kecerdasan Buatan adalah cabang ilmu pengetahuan dan teknologi yang menciptakan mesin cerdas dan program komputer untuk melakukan berbagai tugas yang membutuhkan kecerdasan manusia. Ini adalah sistem yang meniru berbagai fungsi yang dapat dilakukan manusia. AI menggunakan data *eksternal* seperti data besar untuk mencapai kinerja yang sangat baik untuk tugas yang diberikan. Dulu, AI hanyalah sebuah konsep yang terlihat dalam fiksi ilmiah dan debat yang membahas tentang pengaruh teknologi di dunia *modern*. Tapi sekarang, itu telah menjadi bagian dari kita dalam kehidupan kita sehari-hari. Dia telah menjadi fungsi kunci dari banyak teknis dan berbagai sektor lainnya. Kecerdasan Buatan membawa dampak signifikan pada industri seperti manufaktur, perawatan kesehatan, rantai pasokan, dll. Kemampuan AI untuk melakukan hal-hal yang tidak dapat dilakukan manusia, membawa banyak aplikasi yang menghasilkan peningkatan kinerja dan produktivitas [12].

### 2.2.2 *Machine Learning*

*Machine learning* merupakan bagian dari *artificial intelligence* atau kecerdasan buatan yang dapat bekerja membuat sebuah sistem menjadi lebih cerdas dengan mempelajari data-data yang tersedia dan algoritma atau bentuk pemrograman tersebut tidak di definisikan secara eksplisit. Dalam implementasi *machine learning* untuk dapat menyelesaikan sebuah permasalahan membutuhkan beberapa prasyarat yang jika prasyarat itu tidak dapat dipenuhi maka akan mempengaruhi nilai akurasi dari hasil atau solusi permasalahan. *Machine learning* membutuhkan ruang penyimpanan yang besar dikarenakan data yang dianalisis dan kompleksitas algoritma yang digunakan juga besar. Proses *machine learning* juga membutuhkan sebuah pelabelan yang memakan waktu cukup lama dan data yang dihasilkan bersifat bias [13].

#### 1. Konsep Dasar *Machine Learning*

Konsep ini meliputi kemampuan suatu mesin dapat meningkatkan kecerdasan dalam belajar mandiri yang akan meningkatkan produktivitas manusia..

##### a) Manfaat pembelajaran mesin dalam memprediksi

Inti dari pembelajaran mesin ini adalah memprediksi sesuatu hal berdasarkan pola maupun faktor lain yang telah dilakukan *training*. Pengenalan ini dilakukan secara mudah tanpa menghabiskan banyak waktu.

b) Pembelajaran mesin membutuhkan pelatihan

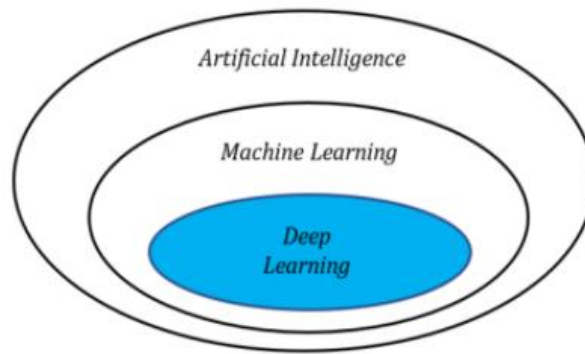
Untuk membuat suatu model maka diperlukan pemberian model pembelajaran mesin seperti apa hasil prediksi yang diinginkan.

c) Ketepatan >80% dianggap sukses

Teknologi ini tidak dapat mengetahui dimana platform dapat mencapai akurasi 100% dengan mengidentifikasi suatu gambar. Aturan yang dikatakan ini adalah bahwa model dengan tingkat akurasi 80% adalah sebuah kesuksesan. Sebagaimana jika memikirkan betapa bergunanya untuk mengidentifikasi suatu gambar yang berjumlah 800.000 dengan benar, sementara “mungkin tidak” mendapatkan 200.000 gambar yang benar, maka masih menyimpan 80% dari waktu yang dimiliki. Hal itu merupakan perspektif nilai yang sangat besar [14]

### **2.2.3 Deep Learning**

*Deep Learning* merupakan bagian dari *machine learning* sebagai cabang ilmu *Artificial Intelligence* yang menggunakan *Deep Neural Network* untuk menuntaskan permasalahan pada domain *machine learning*. *Deep Neural Network* merupakan jaringan syaraf yang mempunyai jumlah *hidden layer* lebih dari satu. Salah satu permasalahan yang sulit pada keberhasilan implementasi *deep learning* yaitu pengaturan nilai untuk berbagai *hyperparameter*, salah satu adalah topologi jaringannya sangat berkaitan erat jumlah *hidden neuron* dan jumlah *hidden layer*. Penentuan jumlah *hidden layer* dan jumlah *neuron* sangat penting dan mempunyai pengaruh yang besar terhadap kinerja *deep learning* [15].



Gambar 2.1 *Deep learning* sebagai bagian *Artificial intelligence* [16].

#### 2.2.4 Python

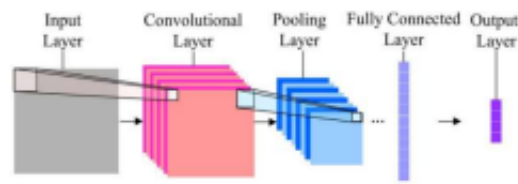
Python merupakan bahasa interpretif yang mudah dilaksanakan serta menggunakan kode-kode pemrograman yang sangat jelas, lengkap, serta mudah dipahami. Python juga merupakan pemrograman berorientasi objek, pemrograman fungsional dan pemrograman imperative atau biasa disebut juga sebagai pemrograman multi-paradigma. Python dapat digunakan untuk berbagai kebutuhan sistem perangkat lunak dan dapat dijalankan di berbagai macam platform dan sistem operasi [17].



Gambar 2.2 Python Programming [18].

#### 2.2.5 Convolutional Neural Network

*Convolutional Neural Network* (CNN) adalah metode perkembangan dari *Multilayer Perceptron* (MLP) yang digunakan untuk mengolah data dua dimensi seperti suara dan citra. *Convolutional neural network* merupakan bagian dari jenis *deep neural network* karena memiliki kedalaman jaringan yang tinggi dan kebanyakan di implementasikan untuk data citra. Pada klasifikasi citra, MLP dirasa tidak sesuai karena tidak dapat menyimpan informasi spasial dari data citra dan menganggap setiap piksel merupakan bagian fitur yang independent sehingga menghasilkan hasil yang kurang baik [19].

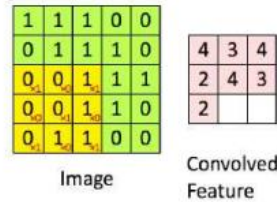


Gambar 2.3 Model Lapisan CNN [19].

Ada beberapa komponen utama yang ada dalam *convolutional neural network* yaitu diantaranya adalah :

1. *Convolutional Layer*

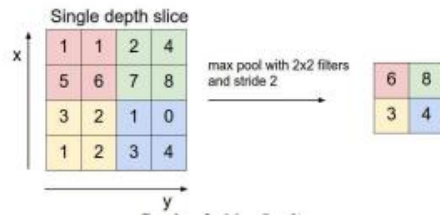
*Convolution layer* merupakan bagian lapisan yang penting dalam metode *convolutional neural network* [18]. *Output* ataupun hasil dari bagian lapisan ini adalah sebuah citra baru yang menunjukkan fitur dari citra *input*, *convolutional layer* memiliki beberapa *neuron* yang tersusun sangat rapat sehingga membentuk filter Panjang dan tinggi(pixel). Contoh dari operasi konvolusi ditunjukkan pada gambar 2.4 [19].



Gambar 2.4 Operasi Konvolusi [19].

2. *Pooling Layer*

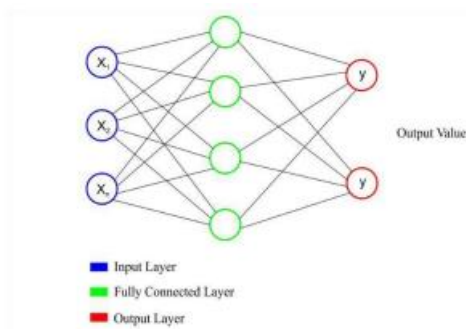
*Pooling layer* merupakan bagian lapisan dari *convolutional neural network* yang berfungsi untuk *feature maps* sebagai masukan dan mengolahnya dengan operasi statistic dari nilai piksel yang terdekat. Dalam model *convolutional neural network* lapisan ini biasanya disisipkan secara teratur setelah lapisan konvolusi. Salah satu hal yang sangat penting dalam pembentukan model *convolutional neural network* yaitu dengan memilih lapisan *pooling* yang dapat memberikan keuntungan kinerja model sehingga membuat model dari *convolutional neural network* menjadi lebih baik [19].



Gambar 2. 5 pooling layer [20]

### 3. Fully Connected Layer

*Fully Connected Layer* merupakan *feature map* yang dihasilkan dari *feature extraction* berbentuk *multidimensional array*. Sehingga perlu melakukan proses “*flatten*” menjadi sebuah bentuk vector agar dapat digunakan sebagai input dari *fully-connected layer*. Lapisan ini biasanya digunakan pada metode mullti lapisan *perceptron* yang bertujuan untuk mengolah data agar dapat di klasifikasikan [21].



Gambar 2. 6 Fully Connected Layer [21].

### 4. Dropout

Lapisan *Dropout* merupakan suatu Teknik *regularisasi* jaringan syaraf yang melakukan pemilihan *neuron* secara acak dan tidak akan digunakan selama proses *learning*. Fungsi lain dari *dropout* yaitu mencegah terjadinya *overfitting* dan mempercepat proses literasi data. Acuan pada *droupout* yaitu pada hilangnya *neuron* yang berupa *hidden layer* maupun layer yang tidak terlihat didalam jaringan syaraf [21].

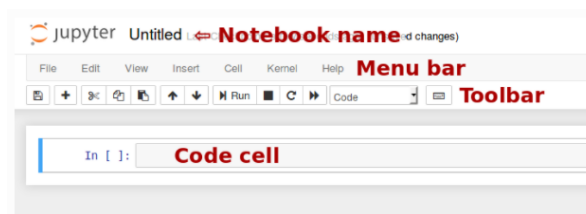
### 2.2.6 Jupyter

*Jupyter notebook* merupakan aplikasi yang berbasis *web* bersifat *open source* yang dipergunakan dalam data *cleaning* dan transformasi data, simulasi

angka, data visual, model statistic, *machine learning*. Ada dua komopnen yang saling terhubung dalam *jupyter notebook* yaitu:

- Aplikasi *web*: aplikasi berbasis *browser* untuk proses pembuatan dokumen secara saling terhubung yang merupakan penggabungan dari teks penjelasan, matematika komputasi dan keluaran media.
- Dokumen buku catatan: mewakili semua konten yang terlihat di aplikasi *web*, termasuk *input* dan *output* dari komputasi, penjelasan teks, matematika, gambar objek, dan mewakili objek di media kaya.

*Jupyter Notebook* dapat diaplikasikan kedalam berbagai format statis, termasuk HTML( contoh untuk posting blog), *reStructuredText*, LaTeX, PDF, dan *slide show* melalui perintah `nbconvert`.



Gambar 2. 7 Tampilan *Jupyter Notebook*

Pada *software Jupyter Notebook* mempunyai beberapa menu yang memiliki fungsi yang berbeda-beda. Beberapa bagian *user interfcace* pada *software Jupyter Notebook* sebagai berikut :

1. Menu bar: Menu ini menyediakan beberapa pilihan yang berbeda yang dapat memodifikasi cara fungsi *notebook*..
2. Toolbar : menu ini memberikan solusi cepat untuk melakukan operasi yang sering digunakan dalam notebook, hanya dengan mengklik ikon..
3. Code cell : Sel kode memungkinkan untuk mengedit dan menulis kode baru, dengan penyorotan sintaksis penuh dan penyelesaian tab. Bahasa pemrograman yang digunakan bergantung pada kernel, dan kernel default (IPython) menjalankan kode Python [22].



### 2.2.7 Subsampling

*Subsampling* merupakan proses reduksi *size* sebuah data dalam citra. Tujuan dari *subsampling* yaitu untuk meningkatkan inyarisasi posisi dari fitur. Dalam pemodelan *convolutional neural network*, metode *subsampling* yang digunakan adalah *maxpooling* yang membagikan *ouput* dari *convolutional neural network* menjadi beberapa bagian *grid* kecil, kemudian mengambil nilai maksimal yang digunakan untuk menyusun matriks citra yang telah mengalami reduksi [20].

### 2.2.8 Model Sequential

Model Sequential umumnya merupakan komposisi linear Keras Layer. Model ini cukup mudah, ringan serta memiliki kemampuan untuk mewakili semua yang tersedia di jaringan saraf. Model sequential menampilkan kelas model untuk membuat model yang disesuaikan kebutuhan penggunaannya. Pengguna bisa menggunakan konsep sub-kelas untuk membuat model kompleks secara individu.

### 2.2.9 Confusion Matrix

*Confusion Matrix* merupakan tabel yang memiliki kombinasi berbeda antara nilai prediksi dan juga nilai actual yang digunakan untuk mengukur performa dari model klasifikasi *machine learning*. Ada empat istilah yang ada pada *confusion matrix* berfungsi menjadi representasi hasil dari proses klasifikasi yaitu *True Positive*(TP), *False Negative*(FN), *True Negative*(TN), *False Positive* (FP).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (2.1)$$

$$Precision = \frac{TP}{FP+TP} \times 100\% \quad (2.2)$$

$$Recall = \frac{TP}{FN+TP} \times 100\% \quad (2.3)$$

Ket : TP = *True Positif*

TN = *True Negatif*

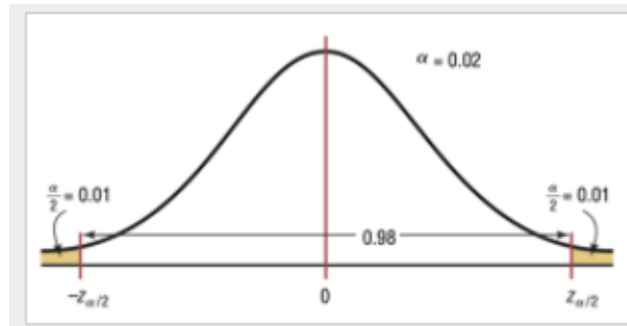
FP = *False Positive*

FN = *False Negative*

Dari hasil *confusion matrix* dapat digunakan untuk menghitung nilai *accuracy, precision, recall* [23].

### 2.2.10 *Confidence Level*

*Confidence level* merupakan gagasan yang berasal dari Teorema Batas Central yang dinotasikan  $100(1 - \alpha)\%$  sebagai gagasan pokok yang berasal dari teorema tersebut. Teorema ini ialah apabila suatu kondisi ditarik sampel secara berulang ulang, maka nilai atribut rata-rata yang diperoleh dari sampel tersebut sejajar dengan nilai real populasinya. Bentuk nilai tersebut akan menjadi nilai-nilai sampel yang menandakan lebih tinggi atau lebih rendah jika dibandingkan dengan nilai populasinya. Dalam suatu bentuk distribusi normal, sekitar 95 % nilai sampel berada dakan dua simpangan baku (standart deviation) dari nilai populasi sebenarnya. Jika tingkat kepercayaan mencapai 95% dipilih, maka nilai 95 dari 100 sampel akan mempunyai nilai populasi yang sebenarnya dalam jangkauan ketetapan yang sudah dispesifikasi sebelumnya.



Gambar 2. 8 *Confidence level*

Berdasarkan grafik tersebut, hubungan antara  $\alpha$  dan *confidence level* adalah bahwa tingkat kepercayaan yang dinyatakan adalah bentuk persentase yang sesuai antara nilai *decimal* dari bentuk  $1-\alpha$ , dan sebaliknya. Secara umum, semakin tinggi tingkat koefisiennya, maka semakin akurat hasil pengujiannya. Tabel 2.1 berikut mencantumkan koefisien kepercayaan dan tingkat kepercayaan [24].

Tabel 2. 1 *Confidence coefficient and confidence level*

Confidence Coefficient (1- $\alpha$ )	Confidence level 100(1- $\alpha$ )%
0,90	90%
0,95	95%
0,95	99%

### 2.2.11 TensorFlow

TensorFlow adalah platform machine learning open source berbasis Python, yang dikembangkan terutama oleh Google. Sama seperti NumPy, tujuan utama TensorFlow adalah untuk memungkinkan para insinyur dan peneliti untuk memanipulasi ekspresi matematika di atas tensor numerik. Tetapi TensorFlow jauh melampaui cakupan NumPy dengan cara berikut:

- Dapat menghitung gradien dari ekspresi terdiferensiasi apapun secara otomatis, sehingga sangat cocok untuk pembelajaran mesin.
- Dapat berjalan tidak hanya pada CPU, tetapi juga pada GPU dan TPU, akselerator perangkat keras yang sangat paralel.
- Komputasi dapat lebih mudah didistribusikan ke banyak mesin.

Penting untuk diingat bahwa TensorFlow lebih dari sekadar pustaka tunggal. Nya benar-benar sebuah platform, rumah bagi ekosistem komponen yang luas, beberapa dikembangkan oleh Google dan beberapa dikembangkan oleh pihak ketiga. Misalnya, ada TF-Agents untuk penelitian pembelajaran penguatan, TFX untuk manajemen alur kerja pembelajaran mesin kekuatan industri, TensorFlow Melayani untuk penyebaran produksi, dan ada TensorFlow Repositori hub dari model yang telah dilatih sebelumnya. Bersama-sama, komponen ini mencakup area yang sangat luas berbagai kasus penggunaan, dari penelitian mutakhir hingga aplikasi produksi skala besar [25].

### 2.2.12 Keras

Keras dirancang untuk menjadi API jaringan saraf tingkat tinggi yang dibangun di atas framework seperti TensorFlow, CNTK, dan Theano. Salah satu manfaat besar menggunakan Keras sebagai pengantar pembelajaran mendalam untuk pemula adalah sangat ramah pengguna, fungsi lanjutan seperti pengoptimal dan lapisan sudah ada di perpustakaan dan tidak harus ditulis dari awal. Inilah sebabnya mengapa Keras populer tidak hanya di kalangan pemula tetapi juga ahli berpengalaman. Juga, perpustakaan memungkinkan pembuatan prototipe cepat dari jaringan saraf, mendukung berbagai arsitektur jaringan, dan dapat dijalankan baik CPU maupun GPU. Keras digunakan untuk membuat dan melatih jaringan saraf dan tidak menawarkan banyak hal algoritme pembelajaran mesin lainnya, termasuk algoritme yang diawasi seperti dukungan mesin vektor dan algoritma tanpa pengawasan seperti k-means clustering. Keras memang menawarkan API yang dirancang dengan baik yang dapat digunakan untuk membuat dan melatih jaringan saraf, yang menghilangkan banyak upaya yang diperlukan untuk menerapkan linear aljabar dan kalkulus multivariat secara akurat. Modul khusus yang tersedia dari perpustakaan Keras, seperti neural lapisan, fungsi biaya, pengoptimal, skema inisialisasi, aktivasi fungsi, dan skema regularisasi, akan dijelaskan secara menyeluruh di seluruh buku ini. Semua modul ini memiliki fungsi yang relevan yang dapat digunakan untuk mengoptimalkan kinerja untuk melatih jaringan saraf untuk tugas-tugas tertentu [26].

### 2.2.13 Fungsi Aktivasi

Fungsi aktivasi merupakan operasi perhitungan yang dilakukan setelah proses konvolusi selesai dilakukan. Fungsi aktivasi yang sering digunakan pada *convolutional neural network* diantaranya ReLU (*Rectified Barisar Unit*), *softmax*. Fungsi aktivasi ReLU merupakan fungsi yang merubah nilai negatif pada matrix menjadi 0 dan nilai positif akan tetap. Persamaan fungsi ReLU dapat ditunjukkan dialam persamaan 2.4

$$f(x) = \max(0, x) \quad (2.4)$$

Keterangan :

$F(x)$  = Nilai dari ReLU *activation*

$X$  = Nilai matriks dari citra

Sedangkan untuk fungsi aktivasi *softmax* biasanya diterapkan pada lapisan terakhir jaringan syaraf . *Softmax* berguna untuk mengubah nilai *output* dari lapisan akhir dalam jaringan syaraf ke distribusi dasar kemungkinan.

Persamaan softmax ditunjukkan dalam persamaan 2.5 [27].

$$\hat{y} = \frac{\exp(x_i)}{\sum \exp(x_i)} \quad (2.5)$$

keterangan :

$\hat{y}$  = Hasil dari aktivasi fungsi *softmax*

$X_i$  = Kelas ke I (i=1,2..dst).

$j$  = nilai dari vektor

#### 2.2.14 *Training, Validation, Testing Set*

Dalam pembangunan model *machine learning* terdapat dua istilah penting yaitu *training* dan *testing*. *Training* adalah proses pelatihan untuk membentuk model dan *testing* adalah proses pengujian dari data yang telah dilakukan proses *learning*. *Dataset* adalah sebuah kumpulan dari data (sampel dalam statistik). Pada umumnya *dataset* dibagi menjadi tiga jenis yaitu

1. *Training set* adalah kumpulan himpunan data yang akan digunakan untuk proses pelatihan atau membangun model.
2. *Validation set* adalah kumpulan himpunan data yang akan digunakan untuk mengoptimasi pelatihan model. Proses ini sangat berguna untuk model dapat mengenali pola secara generic.
3. *Testing set* adalah himpunan data yang digunakan untuk menguji hasil dari pelatihan model yang telah selesai. *Testing tes* ini bersifat *unseen* data yang artinya model dan manusia tidak diperbolehkan melihat sampel ini pada saat dilakukan proses Latihan.

Pada umumnya, rasio untuk pembagian antara *dataset* (*training, validation, testing*) adalah ( 80% : 10% : 10%). Pada umumnya *validation set* tidak dapat

digunakan apabila *dataset* berukuran kecil hanya dibagi untuk data *training* dan *testing set* saja [28].