

## BAB 2

### TINJAUAN PUSTAKA

#### 2.1 Aksara Sunda

Aksara Sunda merupakan aksara yang digunakan oleh Suku Sunda dahulu untuk berkomunikasi, aksara tradisional dari Indonesia ini merupakan hasil karya ortografi melalui perjalanan sejarahnya sejak 5 abad yang lalu [1]. Secara umum aksara ini dikelompokkan kedalam beberapa bagian yaitu aksara swara (vokal), aksara ngalagena (konsonan), aksara khusus, rarangken (tanga vokalisasi), pasangan, serta angka. Pada penelitian ini dilakukan dengan berfokus pada pengenalan dari aksara sunda ngalagena.

##### 2.1.1 Aksara Sunda Ngalagena

Aksara ngalagena adalah lambang-lambang bunyi yang dapat dipandang sebagai fonem konsonan yang secara silabis mengandung bunyi vokal /a/. Aksara ngalagena terdiri dari 18 aksara dari bunyi bahasa sunda, 5 aksara dari bunyi serapan, dan 2 aksara tambahan. Contoh dari aksara sunda ngalagena dapat dilihat pada Gambar 2.1.



**Gambar 2.1 Aksara Sunda Ngalagena**

## 2.2 Citra Digital

Citra digital adalah citra yang dapat diolah oleh komputer, karena memiliki sistem sampling dan kuantisasi. Sistem sampling adalah sistem yang mengubah citra kontinu menjadi citra digital dengan cara membagi citra analog menjadi  $n$  baris dan  $m$  kolom, sehingga menjadi sebuah citra diskrit. Pertemuan antara baris dan kolom pada citra digital disebut dengan piksel. Sistem kuantisasi adalah sistem yang melakukan perubahan intensitas analog ke intensitas diskrit, sehingga dengan sistem ini memungkinkan perubahan gradasi warna sesuai dengan kebutuhan [11].

Citra digital dipetakan dalam bentuk matriks 2 dimensi ukuran  $n \times m$  yang terdiri dari piksel-piksel yang merupakan elemen terkecil dari sebuah citra. Piksel memiliki dua parameter, yaitu koordinat dan intensitas atau warna. Nilai yang terdapat pada koordinat  $(x, y)$  adalah  $f(x, y)$ , yaitu besar intensitas atau warna dari suatu piksel pada koordinat tersebut. Oleh karena itu, citra dapat dituliskan kedalam sebuah Persamaan ( 2 .1).

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, m-1) \\ f(1,0) & f(1,1) & \dots & f(1, m-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(n-1,0) & f(n-1,1) & \dots & f(n-1, m-1) \end{bmatrix} \quad (2.1)$$

Berdasarkan Persamaan ( 2 .1), suatu citra  $f(x, y)$  dapat dituliskan ke dalam fungsi matematis berikut:

$$\begin{aligned} 0 &\leq x \leq n-1 \\ 0 &\leq y \leq m-1 \\ 0 &\leq f(x, y) \leq g-1 \end{aligned}$$

dimana,

- $m$  = Jumlah kolom pada matriks citra
- $n$  = Jumlah baris pada matriks citra
- $g$  = Nilai skala keabuan (*grayscale*)

## 2.3 Pengolahan Citra Digital

Pengolahan citra digital merupakan istilah untuk melakukan manipulasi atau memodifikasi sedemikian rupa terhadap citra, dengan masukan berupa citra dan keluaran berupa citra juga [12]. Pengolahan citra sering diterapkan dalam berbagai bidang seperti bidang biomedis untuk mengolah citra *x-ray*, bidang penginderaan jauh untuk mengolah citra dari satelit, *optical character recognition* dan lain sebagainya.

Penelitian ini digunakan berbagai metode pengolahan citra untuk memperbaiki kualitas citra sebelum citra digunakan dalam proses pelatihan dan pengujian. Metode-metode yang digunakan dalam penelitian ini adalah pengubahan *channel* citra ke *grayscale*, konversi citra ke bentuk biner (binerisasi) menggunakan teknik *thresholding*, pengubahan ukuran citra (*resize*), dan menerapkan augmentasi data untuk membuat variasi dari sampel citra menjadi lebih banyak.

### 2.3.1 Grayscale

Proses pengubahan *channel* citra dari RGB ke grayscale bertujuan untuk menghemat ukuran citra karena hanya menggunakan satu buah *channel* yang hanya menampilkan nilai intensitas atau derajat keabuannya saja. Ilustrasi perbedaan antara citra RGB dan citra *grayscale* dapat dilihat pada Gambar 2.2.



**Gambar 2.2 Ilustrasi Perbedaan Citra RGB dan Grayscale**

Pada penelitian ini digunakan metode rata-rata (*averaging*) untuk mengubah *channel* citra dari citra RGB ke citra *grayscale*. Berdasarkan pada artikel berikut [13], persamaan untuk mengubah channel citra dari citra RGB ke

citra *grayscale* menggunakan metode rata-rata dapat dilihat pada Persamaan ( 2.2).

$$G_{x,y} = \frac{r+g+b}{3} \quad (2.2)$$

dimana,

$G_{x,y}$  = Nilai derajat keabuan yang dicari

$r$  = Nilai piksel dari lapisan citra berwarna merah (*red*)

$g$  = Nilai piksel dari lapisan citra berwarna hijau (*green*)

$b$  = Nilai piksel dari lapisan citra berwarna biru (*blue*)

### 2.3.2 *Thresholding*

*Thresholding* merupakan metode untuk mengubah citra *grayscale* ke dalam citra biner. Pengubahan citra ke bentuk biner bertujuan agar setiap nilai pada matriks citra hanya memiliki dua nilai antara 0 dan 255, hal ini juga dapat menghemat ukuran citra sehingga beban komputasi yang diperlukan lebih sedikit dan waktu komputasi semakin cepat. Perbedaan antara citra *grayscale* dan citra biner dapat dilihat pada Gambar 2.3.



**Gambar 2.3 Ilustrasi Perbedaan Citra *Grayscale* dan Biner**

Untuk mengubah citra *grayscale* ke dalam bentuk citra biner digunakan proses *thresholding*, yaitu menggunakan ambang batas sebagai batas penentu nilai piksel *grayscale* ke nilai biner antara nilai 0 dan 1. Berdasarkan pada penelitian berikut [14], persamaan untuk mengubah citra *grayscale* ke biner dapat dilihat pada Persamaan ( 2.3).

$$G_{x,y} \begin{cases} 0, & \text{if } G_{x,y} \leq t \\ 255, & \text{otherwise} \end{cases} \quad (2.3)$$

Dimana,

$G_{x,y}$  = Nilai derajat keabuan pada koordinat  $(x, y)$

$t$  = Nilai *threshold* atau ambang batas

Untuk menentukan nilai ambang batas  $t$  pada penelitian ini digunakan metode *NICK thresholding* yang merupakan jenis *local adaptive thresholding*. Adapun rumus dari metode *NICK thresholding* dapat dilihat pada Persamaan ( 2.4).

$$t = \bar{x} + k \sqrt{\sigma_i} \quad (2.4)$$

dimana,

$k$  = *Niblack Factor* dengan rentang 0.1 sampai 0.9

$\bar{x}$  = Rata-rata nilai piksel

$g_i$  = Nilai piksel pada citra *grayscale*

$n$  = Banyaknya nilai piksel

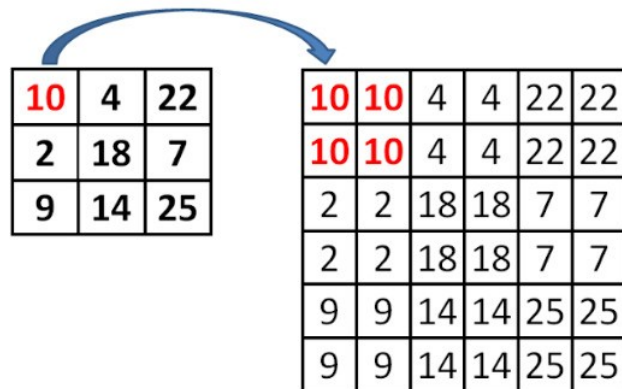
### 2.3.3 *Resize*

*Resize* merupakan metode untuk mengubah ukuran citra ke dalam ukuran yang diinginkan. *Resize* bertujuan untuk memperbesar ukuran citra ataupun sebaliknya yaitu memperkecil ukuran citra. Perbedaan antara citra biner sebelum dilakukan *resize* dan setelah dilakukan *resize* dapat dilihat pada Gambar 2.4.



**Gambar 2.4 Ilustrasi Perbedaan Citra Setelah Dilakukan *Resize***

Pada penelitian ini digunakan metode *Nearest Neighbour Interpolation* yaitu metode sederhana untuk mengubah ukuran citra berdasarkan nilai piksel terdekat. Contoh ilustrasi dari metode *Nearest Neighbour Interpolation* pada matriks citra dapat dilihat pada Gambar 2.5.



**Gambar 2.5 Ilustrasi Metode *Nearest Neighbour Interpolation***

Untuk mendapatkan nilai piksel  $x$  dan  $y$  pada citra untuk kemudian digunakan pada citra *resize* dapat menggunakan rumus pada Persamaan (2.5).

$$\begin{aligned} x_1 &= \text{round}\left(x \frac{2 * w_i}{w_t}\right) \\ y_1 &= \text{round}\left(y \frac{2 * h_i}{h_t}\right) \end{aligned} \quad (2.5)$$

dimana,

$x_1, y_1$  = Titik koordinat pada citra

$x_2, y_2$  = Titik koordinat pada citra *resize*

$w_i$  = Lebar dari citra asal

$w_t$  = Lebar dari citra *resize*

$h_i$  = Tinggi dari citra asal

$h_t$  = Tinggi dari citra *resize*

### 2.3.4 Augmentasi Data

Augmentasi data merupakan metode termudah dan paling umum digunakan untuk mengurangi masalah *overfitting* pada data citra dengan memperbesar sampel data secara artifisial [15], penerapan augmentasi data

memberikan pengaruh signifikan terhadap akurasi dalam hal *image recognition* [8]–[10]. Dengan membuat sampel data secara artifisial berarti menambah jumlah sampel data sehingga sangat bermanfaat dalam untuk memberikan variasi yang lebih baik terhadap sampel data yang akan digunakan pada proses pengenalan. Pada penelitian ini digunakan metode augmentasi data tradisional seperti *flipping*, *rotation*, dan *translation*.

#### 2.3.4.1 *Flipping*

Pada penelitian ini digunakan metode *flipping* yaitu membalikkan citra baik secara horizontal maupun vertikal, contoh ilustrasi penggunaan augmentasi data *flipping* pada citra dapat dilihat pada Gambar 2.6.



**Gambar 2.6 Ilustrasi Augmentasi Data *Flipping***

Penelitian ini hanya menggunakan teknik *horizontal flip* yaitu membalikkan citra secara *horizontal* atau ke arah sumbu  $x$ , untuk melakukan *horizontal flip* digunakan rumus pada Persamaan (2.6).

$$F_{x,y} = I_{x,(w-1-y)} \quad (2.6)$$

dimana,

$F$  = Matriks citra *flip*

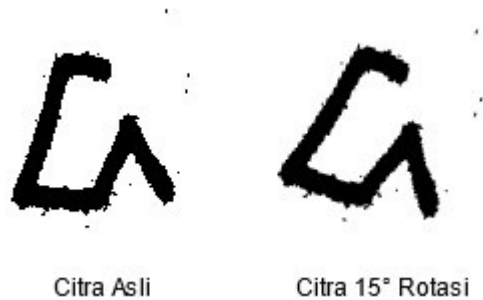
$I$  = Matriks citra asli

$w$  = Lebar citra

#### 2.3.4.2 *Rotation*

*Rotation* merupakan sebuah metode untuk memutar gambar dengan arah searah dengan jarum jam dengan nilai derajat rotasi yang dapat diterapkan antara

$1^\circ - 365^\circ$ . Contoh penggunaan augmentasi data *rotation* pada citra dapat dilihat pada Gambar 2.7.



**Gambar 2.7 Ilustrasi Augmentasi Data *Rotation***

Untuk menerapkan rotasi pada citra, digunakan persamaan matematis sebagai berikut pada Persamaan (2.7).

$$x_1 = \text{integer} \left( \cos \theta * \left( x_2 - \left( \frac{w}{2} \right) \right) - \sin \theta * \left( y_2 - \left( \frac{h}{2} \right) \right) + \left( \frac{w}{2} \right) \right) \quad y_1 = \text{integer} \left( \sin \theta * \left( x_2 - \left( \frac{w}{2} \right) \right) + \cos \theta * \left( y_2 - \left( \frac{h}{2} \right) \right) + \left( \frac{h}{2} \right) \right) \quad (2.7)$$

dimana,

$x_1, y_1$  = Titik koordinat pada citra

$x_2, y_2$  = Titik koordinat pada citra *rotation*

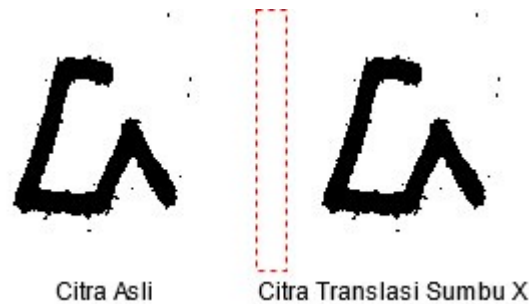
$\theta$  = Derajat rotasi dalam radian

$w, h$  = Panjang dan lebar citra

### 2.3.4.3 *Translation*

*Translation* merupakan metode untuk menggeser posisi citra ke arah sumbu  $x$  ataupun sumbu  $y$ . Ketika citra asli dilakukan *translation* ke arah sumbu tertentu, ruang yang tersisa dapat diisi dengan nilai konstan seperti 0 atau 255. Contoh penggunaan augmentasi data *translation* pada citra dapat dilihat pada Gambar 2.8. Ruang yang ditandai dengan garis putus merah merupakan ruang yang tersisa hasil pergeseran citra dengan nilai 255.





**Gambar 2.8 Ilustrasi Augmentasi Data *Translation***

Untuk menerapkan translasi pada citra digunakan persamaan matematis sebagai berikut pada Persamaan ( 2.8).

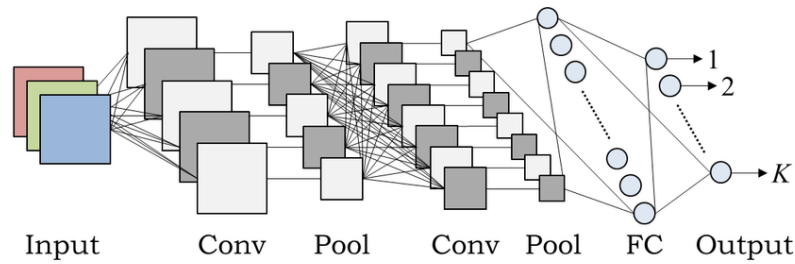
$$A'_{(x+a,y+b)} = A_{(x,y)} \quad (2.8)$$

dimana,

- $A$  = Titik pada ujung citra
- $A'$  = Titik pada ujung citra *translation*
- $a$  = Panjang translasi sumbu  $y$
- $b$  = Panjang translasi sumbu  $x$

#### **2.4 Convolutional Neural Network**

*Convolutional Neural Network* (CNN) merupakan pengembangan dari metode JST yang didesain untuk mengolah data multi dimensi seperti citra. CNN termasuk kedalam metode *deep neural network* karena memiliki kedalaman jaringan yang lebih dalam dibandingkan dengan JST pada umumnya. Secara umum CNN memiliki 3 lapisan jaringan atau *layer* utama yaitu *convolution layer*, *pooling layer*, dan *fully connected layer*. Lapisan *convolution* dan *pooling* melakukan ekstraksi ciri dari sebuah citra, sedangkan lapisan *fully connected* memberikan hasil keluaran berupa klasifikasi terhadap *label* atau kelas yang telah ditentukan [16]. Berikut merupakan contoh arsitektur dasar dari jaringan metode CNN pada Gambar 2.9.



**Gambar 2.9** Contoh Arsitektur *Convolutional Neural Network*

### 2.4.1 Convolution Layer

*Convolutional Layer* merupakan lapisan dari metode CNN untuk melakukan ekstraksi ciri dari citra masukan. Untuk melakukan ekstraksi ciri dari citra, digunakan teknik konvolusi yaitu *dot product* dari citra masukan  $I$  dan kernel konvolusi  $F$  yang lebih kecil dimensinya dari citra masukan. Keluaran dari teknik konvolusi yaitu adalah *feature map*  $FM$ . Rumus dari teknik konvolusi dapat dilihat pada Persamaan ( 2.9).

$$N_{x,y} = \sum_m \sum_n I_{m+x,n+y} \times F_{m,n} + b \quad (2.9)$$

dimana,

$N$  = Matriks hasil konvolusi tanpa fungsi aktivasi

$I$  = Matriks citra masukan

$F$  = Matriks *filter*

$b$  = Nilai bias *filter*

#### 2.4.1.1 ReLU

Fungsi aktivasi atau *activation function* berguna untuk melakukan rekayasa fitur pada citra, pada penelitian ini digunakan fungsi aktivasi ReLU atau *Rectified Linear Unit*. Tujuan menggunakan fungsi aktivasi ReLU adalah untuk menambah *non-linearity* pada citra, alasan menggunakan ReLU adalah karena citra pada dasarnya adalah *non-linear*. Persamaan matematis dari fungsi aktivasi ReLU dapat dilihat pada Persamaan 2.7.

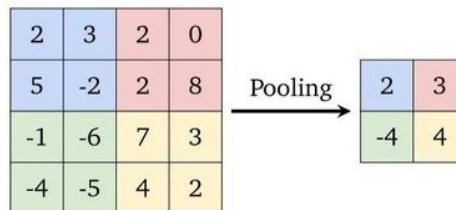
$$\sigma(x) = \begin{cases} x & \text{jika } x \geq 0 \\ 0 & \text{lainnya} \end{cases} \quad (2.7)$$

## 2.4.2 Pooling Layer

Pooling layer merupakan lapisan dari metode CNN untuk mengurangi dimensi dari *feature map* yang dihasilkan dari *convolution layer*. Keluaran dari *pooling layer* adalah *pooled feature map* yaitu *feature map* dengan dimensi lebih kecil dari *feature map* yang dihasilkan *convolution layer* sebelumnya.

### 2.4.2.1 Average Pooling

Pada penelitian ini digunakan *average pooling*, yaitu menggunakan nilai rata-rata dari suatu jendela tertentu. Proses *pooling layer* mirip seperti *convolutional layer*, yaitu menggeser jendela ke seluruh permukaan citra, namun disini jendela digunakan sebagai acuan untuk memilih nilai rata-rata pada area tertentu pada permukaan citra. Sebagai ilustrasi, matriks berdimensi  $4 \times 4$  pada Gambar 2.10 akan dilakukan proses *pooling* menggunakan *average pooling* menjadi berdimensi  $2 \times 2$ .



**Gambar 2.10 Ilustrasi Metode Average Pooling**

Untuk melakukan proses *average pooling* digunakan persamaan sebagai berikut pada Persamaan (2.10).

$$S_{x,y} = \frac{1}{f_w \times f_h} \sum_{u=x \times_s}^{x \times_s + 1} \sum_{v=y \times_s}^{y \times_s + 1} C_{u,v} \quad (2.10)$$

dimana,

$S$  = Matriks *pooling*

$f_w$  = Lebar dimensi matriks *filter*

$fh$  = Tinggi dimensi matriks *filter*

$C$  = Matriks *convolution*

### 2.4.3 Fully Connected Layer

*Fully Connected Layer* yaitu merupakan lapisan pada metode CNN yang mirip dengan metode JST atau *Multi-Layer Perceptron* (MLP) yang memiliki *hidden layer*, *activation function*, *output layer*, dan *loss function*. Pada lapisan ini dilakukan klasifikasi untuk menghasilkan *output* atau keluaran berupa kelas dari citra masukan. Untuk menghitung keluaran *neuron* pada *fully connected* dapat digunakan rumus pada Persamaan ( 2 .11).

$$o_m = \sum_{i=0} W_{i,m} \times fi + b \quad (2.11)$$

dimana,

$m$  = Banyaknya neuron pada keluaran *fully connected*

$i$  = Banyaknya neuron pada vektor masukan

$o_m$  = Keluaran dari hasil *fully connected*

$b$  = Bias

#### 2.4.3.1 Softmax

Penelitian ini menggunakan fungsi aktivasi *softmax* untuk menormalisasikan keluaran dari proses *fully connected* menjadi nilai probabilitas, yang kemudian akan digunakan sebagai hasil prediksi dari kelas citra. Untuk menghitung fungsi *softmax* dapat digunakan rumus pada Persamaan ( 2 .12).

$$\hat{o}_m = \frac{e^{y_m}}{\sum_{j=1} e^{y_m}} \quad (2.12)$$

#### 2.4.3.2 Cross Entropy Error Function

Untuk menghitung loss atau kerugian, pada penelitian ini digunakan untuk fungsi *cross entropy error function*. Untuk menghitung kerugian menggunakan *cross entropy error function* digunakan rumus pada Persamaan ( 2 .13).

$$E = \frac{-1}{n} \sum_{m=0}^{n-1} ce(m) \quad (2.13)$$

$$ce = \begin{cases} \ln(\hat{o}_m), & \text{jika } (t_m = 1) \\ \ln(1 - \hat{o}_m), & \text{lainnya} \end{cases}$$

dimana,

$\sigma(y)_m$  = Hasil fungsi *softmax* pada keluaran ke- $m$

$t_m$  = Target sebenarnya yang ingin dicapai

$n$  = Jumlah keluaran *neuron*

## 2.5 Confusion Matrix

*Confusion Matrix* merupakan salah satu metode yang digunakan untuk melakukan evaluasi kinerja dari algoritma atau metode. Kinerja dari algoritma atau metode dapat diukur dengan nilai akurasi, *recall*, presisi, ataupun *f-measure*. Contoh confusion matriks dapat dilihat pada Tabel 2.1.

**Tabel 2.1 Contoh Confusion Matrix**

		<i>Predicted Class</i>	
		<b>1</b>	<b>0</b>
<i>Actual Class</i>	<b>1</b>	TP	FN
	<b>0</b>	FP	TN

Dimana, *True Positive* (TP) merupakan kondisi dimana model memprediksi kelas benar pada kelas yang benar, *True Negative* (TN) merupakan kondisi dimana model memprediksi kelas salah pada kelas yang salah, *False Positive* (FP) merupakan kondisi dimana model memprediksi kelas benar pada kelas yang salah, serta *False Negative* (FN) yaitu kondisi dimana model memprediksi salah pada kelas yang benar.

### 2.5.1 Akurasi

Akurasi merupakan nilai yang digunakan untuk mengukur kinerja dari seberapa sering model mengklasifikasi benar. Rumus dari akurasi dapat dilihat pada rumus pada Persamaan ( 2.14).

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.14)$$

### 2.6 Penelitian Sebelumnya

Berikut merupakan hasil ringkasan dari penelitian-penelitian sebelumnya yang menjadi landasan terhadap penelitian yang akan dilakukan pada Tabel 2.2.

**Tabel 2.2 Ringkasan Penelitian Sebelumnya**

<b>Literatur ke-1</b>	
Judul Penelitian	Pengenalan Pola Aksara Sunda dengan Metode <i>Convolutional Neural Network</i>
Penulis	Alif Kirana, Hanny Hikmayanti H, dan Jamaludin Indra
Tahun	2020
Publikasi	Scientific Student Journal for Information, Technology, and Science, Vol. 1, No. 2
Ikhtisar	Penulis melakukan penelitian untuk melakukan pengenalan citra Aksara Sunda menggunakan metode CNN dengan data yang diambil dari beberapa sumber diantaranya diambil dari citra buku, citra komputer, tulisan tangan yang diambil menggunakan kamera <i>smartphone</i> , dan tulisan tangan yang diambil menggunakan pemindai. Metode CNN berhasil mendapatkan akurasi 72.41% pada cita buku, 100% pada citra komputer, 87.1% pada citra tulisan tangan yang

	diambil menggunakan <i>smartphone</i> , dan 85.5% pada citra tulisan tangan yang diambil menggunakan pemindai
Hasil, Kesimpulan, dan Saran	<ol style="list-style-type: none"> <li>a. Hasil: Metode CNN mendapatkan akurasi sebesar 85.5% pada citra tulisan tangan Aksara Sunda yang diambil menggunakan pemindai</li> <li>b. Kesimpulan: Metode CNN berhasil melakukan klasifikasi Aksara Sunda</li> <li>c. Saran: Menambahkan proses segmentasi serta penghilangan noise pada citra, memperbanyak sampel data, ditulis menggunakan berbagai alat tulis selain <i>board marker</i>, penggunaan kamera dengan resolusi yang lebih baik</li> </ol>
Persamaan dan Perbedaan dengan Penelitian	<ol style="list-style-type: none"> <li>a. Persamaan: Objek penelitian serta metode ekstraksi ciri dan klasifikasi yang sama</li> <li>b. Perbedaan: Metode pra proses yang berbeda serta tidak menggunakan augmentasi data</li> </ol>
Komentar	Literatur memberikan gambaran mengenai bagaimana penggunaan metode CNN untuk pengenalan pola citra tulisan tangan Aksara Sunda
<b>Literatur ke-2</b>	
Judul Penelitian	Pengenalan Aksara Sunda Menggunakan Metode Jaringan Saraf Tiruan <i>Backpropagation</i> dan Deteksi Tepi Canny
Penulis	Nisa Amalia, Eka Wahyu Hidayat, dan Aldy Putra Aldya
Tahun	2020
Publikasi	CESS (Journal of Computer Engineering System

	and Science), Vol. 5, No. 1
Ikhtisar	Penulis melakukan penelitian untuk melakukan pengenalan dari tulisan tangan Aksara Sunda menggunakan metode <i>JST Backpropagation</i> dan Deteksi Tepi Canny, dengan menggunakan total 91 sampel data yang diambil menggunakan kamera atau pemindai. Dari hasil pengujian menggunakan 21 citra uji, metode <i>JST Backpropagation</i> berhasil mendapatkan akurasi sebesar 76.21%
Hasil, Kesimpulan, dan Saran	<ul style="list-style-type: none"> <li>a. Hasil: Metode <i>JST Backpropagation</i> mendapatkan akurasi sebesar 76.2% pada 21 citra uji</li> <li>b. Kesimpulan: Metode JST dan deteksi tepi Canny dapat diterapkan untuk pengenalan Aksara Sunda</li> <li>c. Saran: -</li> </ul>
Persamaan dan Perbedaan dengan Penelitian	<ul style="list-style-type: none"> <li>a. Persamaan: Menggunakan objek penelitian yang sama</li> <li>b. Perbedaan: Menggunakan metode ekstraksi ciri dan klasifikasi yang berbeda</li> </ul>
Komentar	Literatur ini memberikan gambaran mengenai bagaimana melakukan pengenalan dari citra tulisan tangan Aksara Sunda menggunakan metode JST
<b>Literatur ke-3</b>	
Judul Penelitian	Pengenalan Pola Citra Tulisan Tangan Aksara Sunda dengan Metode <i>Convolutional Neural Network</i>
Penulis	Imam Ramadhan dan Galih Hermawan
Tahun	2018



Publikasi	SKRIPSI: Program Studi Teknik Informatika UNIKOM
Ikhtisar	Penulis melakukan penelitian untuk pengenalan pola citra tulisan tangan Aksara Sunda menggunakan metode CNN dengan menggunakan total sampel data sebanyak 250 citra. Metode CNN berhasil mendapatkan akurasi terbaik sebesar 62%
Hasil, Kesimpulan, dan Saran	<ol style="list-style-type: none"> <li>a. Hasil: Metode CNN mendapatkan akurasi sebesar 62%</li> <li>b. Kesimpulan: Sistem pengenalan tulisan tangan dapat mengenali Aksara Sunda</li> <li>c. Saran: Menggunakan metode ekstraksi ciri dan klasifikasi yang berbeda, menggunakan metode pra proses yang berbeda, memperbanyak sampel data latih, membuat sampel data latih yang bervariasi</li> </ol>
Persamaan dan Perbedaan dengan Penelitian	<ol style="list-style-type: none"> <li>a. Persamaan: Objek penelitian dan metode ekstraksi ciri serta klasifikasi yang sama</li> <li>b. Perbedaan: Metode pra proses yang berbeda serta tidak menggunakan augmentasi data</li> </ol>
Komentar	Literatur memberikan gambaran mengenai bagaimana penggunaan metode CNN untuk pengenalan pola citra tulisan tangan Aksara Sunda pada sampel data yang sedikit
<b>Literatur ke-4</b>	
Judul Penelitian	Sistem Pengenalan Aksara Sunda Menggunakan Metode <i>Modified Direction Feature</i> dan <i>Learning</i>

	<i>Vector Quantization</i>
Penulis	Rizki Rahmat Riansyah, Youllia Indrawaty Nurhasanah, dan Irma Amelia Dewi
Tahun	2017
Publikasi	Jurnal Teknik Informatika dan Sistem Informasi, Vol. 3, No. 1
Ikhtisar	Penulis melakukan penelitian untuk pengenalan Aksara Sunda menggunakan metode MDF dan LVQ, dari hasil pengujian metode MDF dan LVQ didapatkan hasil akurasi sebesar 78.67%
Hasil, Kesimpulan, dan Saran	<ul style="list-style-type: none"> <li>a. Hasil: Metode MDF serta LVQ mendapatkan akurasi sebesar 78.67%</li> <li>b. Kesimpulan: Sistem dapat digunakan digunakan untuk mengenali citra Aksara Sunda</li> <li>c. Saran: Sistem pengenalan dapat mengenali kata bahkan kalimat, serta mampu mengenali berbagai macam citra hingga tulisan tangan, menambahkan atau mengubah metode pengolahan citra</li> </ul>
Persamaan dan Perbedaan dengan Penelitian	<ul style="list-style-type: none"> <li>a. Persamaan: Objek penelitian yang sama</li> <li>b. Perbedaan: Menggunakan metode ekstraksi ciri dan klasifikasi yang berbeda</li> </ul>
Komentar	-
<b>Literatur ke-5</b>	
Judul Penelitian	<i>A Survey of Handwritten Character Recognition with MNIST and EMNIST</i>
Penulis	Alejandro Baldominos, Yago Saez, dan Pedro Isasi
Tahun	2019
Publikasi	Applied Sciences (Switzerland), Vol. 9, No. 15

Ikhtisar	Penulis melakukan penelitian untuk melakukan survey untuk memberikan pemaparan <i>state-of-the-art</i> terbaru mengenai <i>dataset</i> MNIST dan EMNIST. Dari hasil penelitian tersebut, metode CNN sejauh ini masih unggul dalam pengenalan pola gambar dari <i>dataset</i> MNIST dan EMNIST
Hasil, Kesimpulan, dan Saran	<ol style="list-style-type: none"> <li>a. Hasil: Metode CNN mendapatkan <i>error rate</i> yaitu sebesar 0.21% pada data normal dan 0.24% pada data yang sudah digunakan augmentasi data untuk basis data MNIST. Sedangkan untuk basis data EMNIST mendapatkan akurasi sebesar 95.44% untuk basis data huruf ketika menggunakan model <i>Markov Random Field</i> dan CNN</li> <li>b. Kesimpulan: Metode CNN mampu mendapatkan hasil tertinggi pada basis data MNIST dan EMNIST dibandingkan dengan metode yang lain</li> <li>c. Saran: -</li> </ol>
Persamaan dan Perbedaan dengan Penelitian	<ol style="list-style-type: none"> <li>a. Persamaan: Menggunakan salah satu metode ekstraksi ciri dan klasifikasi yang sama</li> <li>b. Perbedaan: Objek penelitian yang berbeda yaitu tulisan tangan latin</li> </ol>
Komentar	Literatur ini memberikan gambaran mengenai bagaimana kinerja metode CNN pada basis data tulisan tangan MNIST dan EMNIST
<b>Literatur ke-6</b>	
Judul Penelitian	Analisis Pengaruh Tingkat Akurasi Klasifikasi

	Citra Wayang dengan Algoritma <i>Convolutional Neural Network</i>
Penulis	Muhammad Resa Arif Yudianto, Kusrini, dan Hanif Al Fatta
Tahun	2020
Publikasi	Jurnal Teknologi Informasi, Vo. 4, No. 2
Ikhtisar	Penulis melakukan penelitian untuk melakukan analisis terhadap pengaruh tingkat akurasi yang didapatkan dari klasifikasi citra wayang dengan menggunakan algoritma CNN, dari hasil penelitian penggunaan augmentasi data dan pengubahan <i>channel</i> citra ke <i>grayscale</i> pada tahap <i>preprocessing</i> berpengaruh signifikan terhadap nilai akurasi
Hasil, Kesimpulan, dan Saran	<ol style="list-style-type: none"> <li>a. Hasil: Skenario terbaik didapatkan dengan nilai akurasi sebesar 97.08%</li> <li>b. Kesimpulan: Penggunaan augmentasi data dan pengubahan <i>channel</i> citra ke <i>grayscale</i> pada <i>preprocessing</i> data latih berpengaruh secara signifikan terhadap nilai akurasi model, penggunaan nilai <i>epoch</i> yang lebih tinggi juga berpengaruh terhadap akurasi yang lebih baik jika sampel data yang digunakan lebih banyak</li> <li>c. Saran: -</li> </ol>
Persamaan dan Perbedaan dengan Penelitian	<ol style="list-style-type: none"> <li>a. Persamaan: Menggunakan augmentasi data dan metode ekstraksi ciri dan klasifikasi yang sama</li> <li>b. Perbedaan: Objek penelitian yang berbeda</li> </ol>
Komentar	Literatur ini memberikan gambaran mengenai bagaimana penggunaan augmentasi data terhadap

	metode ekstraksi ciri dan klasifikasi yang akan digunakan
<b>Literatur ke-7</b>	
Judul Penelitian	Augmentasi Data Pengenalan Citra Mobil Menggunakan Pendekatan <i>Random Crop</i> , <i>Rotate</i> , dan <i>Mixup</i>
Penulis	Sanjaya dan Mewati Ayub
Tahun	2020
Publikasi	Jurnal Teknik Informatika dan Sistem Informasi, Vol. 6, No. 2
Ikhtisar	Penulis melakukan penelitian untuk menerapkan augmentasi data untuk pengenalan citra mobil menggunakan pendekatan <i>Random Crop</i> , <i>Rotate</i> , dan <i>Mixup</i> , dari hasil pengujian terhadap model CNN didapatkan hasil bahwa metode CNN dengan augmentasi data <i>Random Crop</i> , <i>Rotate</i> , dan <i>Mixup</i> memberikan akurasi yang lebih baik daripada model <i>baseline</i>
Hasil, Kesimpulan, dan Saran	<ol style="list-style-type: none"> <li>a. Hasil: Metode CNN dengan augmentasi data mendapatkan akurasi sebesar 82.6%</li> <li>b. Kesimpulan: Penggunaan augmentasi data <i>random crop</i>, <i>rotation</i>, dan <i>mixup</i> memberikan akurasi yang lebih baik pada metode CNN</li> <li>c. Saran: -</li> </ol>
Persamaan dan Perbedaan dengan Penelitian	<ol style="list-style-type: none"> <li>a. Persamaan: Menggunakan augmentasi data untuk meningkatkan kinerja dari metode CNN</li> <li>b. Perbedaan: Objek penelitian yang berbeda</li> </ol>
Komentar	Literatur ini memberikan gambaran mengenai bagaimana penggunaan augmentasi data terhadap

	metode ekstraksi ciri dan klasifikasi yang akan digunakan
<b>Literatur ke-8</b>	
Judul Penelitian	<i>Bangla Handwritten Character Recognition using Convolutional Neural Network with Data Augmentation</i>
Penulis	Rumman Rashid Chowdhury, Mohammad Shahadat Hossain, Rihan Ul Islam, Karl Anderson, dan Sazzad Hossain
Tahun	2019
Publikasi	2019 Joint 8th International Conference on Informatics, Electronics and Vision, ICIEV 2019 and 3rd International Conference on Imaging, Vision and Pattern Recognition, IVPR 2019 with International Conference on Activity and Behavior Computing, ABC 2019
Ikhtisar	Penulis melakukan penelitian untuk pengenalan citra tulisan tangan Aksara Bangla menggunakan metode CNN, dari hasil penelitian menggunakan <i>dataset</i> BanglaLekha-Isolated memperoleh akurasi sebesar 91.81%, setelah ditambahkan sebesar 200.000 citra menggunakan augmentasi data, akurasi yang didapat meningkat sebesar 92.25%
Hasil, Kesimpulan, dan Saran	<ol style="list-style-type: none"> <li>a. Hasil: Metode CNN dengan data augmentasi mendapatkan akurasi sebesar 92.5%</li> <li>b. Kesimpulan: Metode CNN dengan data augmentasi mampu melakukan pengenalan lebih baik dibandingkan dengan metode <i>machine learning</i> yang lain untuk</li> </ol>

	klasifikasi karakter tulisan tangan Bangla c. Saran: -
Persamaan dan Perbedaan dengan Penelitian	a. Persamaan: Menggunakan augmentasi data untuk meningkatkan kinerja dari metode CNN b. Perbedaan: Objek penelitian yang berbeda
Komentar	Literatur ini memberikan gambaran mengenai bagaimana penggunaan augmentasi data terhadap metode ekstraksi ciri dan klasifikasi yang akan digunakan