

## **USULAN TUGAS AKHIR**

### **PENERAPAN METODE *BACKPROPAGATION* DAN *IMAGECENTROID ZONE- ZONE CENTROID ZONE* PADA PENGENALAN POLA TULISAN TANGAN AKSARA BIMA**



Oleh :  
**RIZKA AMALIA**  
**F1D 015 077**

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS MATARAM**  
**2020**

## ABSTRAK

Suku Bima memiliki sistem komunikasi atau bahasa tersendiri yang dilengkapi dengan sistem penulisan atau Aksara yang biasa disebut dengan aksara Bima atau Mbojo. Sebagai bentuk pelestarian aksara, perlu dilakukan digitalisasi alih aksara untuk memudahkan masyarakat mengenal dan mempelajari aksara Bima. Tujuan dari penelitian ini adalah menciptakan alih aksara Bima menjadi tulisan latin serta untuk menentukan tingkat akurasi dengan metode *Backpropagation* dan *Image Centroid Zone – Zone Centroid Zone (ICZ-ZCZ)*. Karakter yang akan dikenali adalah sebanyak 22 huruf aksara Bima dalam bentuk tulisan tangan dari 50 orang yang berbeda. Masing-masing citra karakter yang ditulis sebanyak 12 kali oleh tiap orang kemudian diubah dalam ukuran 128x128, 64 x 64, dan 32x32 $pixel$ . Ekstraksi fitur yang digunakan adalah metode *Zoning* gabungan ICZ-ZCZ. Nilai akurasi dari setiap percobaan nantinya akan diolah untuk menentukan nilai rata rata kemudian hasilnya akan menjadi nilai akhir akurasi sistem.

**Kata kunci:** Aksara Bima, Pengenalan Pola, Jaringan Syaraf Tiruan, *Backpropagation*, *Zoning*, *Image Centroid Zone*, *Zone Centroid Zone*, ICZ-ZCZ.

## DAFTAR ISI

ABSTRAK .....	i
DAFTAR ISI .....	ii
DAFTAR GAMBAR .....	iii
DAFTAR TABEL .....	iv
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan Penelitian .....	3
1.5 Manfaat .....	3
1.6 Sistematika Penulisan .....	3
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI .....	5
2.1 Tinjauan Pustaka .....	5
2.2 Dasar Teori .....	6
2.2.1 Citra Digital .....	6
2.2.2 Pengolahan Citra .....	6
2.2.3 Pengenalan Pola .....	6
2.2.4 Aksara Bima .....	7
2.2.5 Jaringan Syaraf Tiruan <i>Backpropagation</i> .....	7
2.2.6 Ekstraksi Ciri .....	11
2.2.7 ICZ-ZCZ .....	11
2.2.8 <i>Confusion Matrix</i> .....	13
BAB III METODOLOGI PENELITIAN .....	15
3.1 Alat dan Bahan Penelitian .....	15
3.2 Rancangan Penelitian .....	15
3.3 Perancangan Algoritma .....	17
3.4 <i>Data Acquisition</i> .....	18
3.5 <i>Pre-processing</i> .....	19
3.6 Ekstraksi Fitur .....	22
3.7 Klasifikasi .....	26
3.7.1 Inisialisasi Bobot .....	26
3.7.2 <i>Feed Forward</i> .....	28

3.7.3	<i>Backpropagation</i> .....	30
3.8	Teknik Pengujian Sistem .....	36
3.9	Skenario Pengujian Sistem .....	38
3.10	Jadwal Kegiatan.....	39
DAFTAR PUSTAKA.....		40

## DAFTAR GAMBAR

Gambar 2.1 Bentuk aksara Bima modern .....	7
Gambar 2.2 Arsitektur jaringan <i>backpropagation</i> . ....	8
Gambar 2.3 Bentuk <i>confussion matrix</i> .....	12
Gambar 3.1 Diagram alir perancangan sistem. ....	14
Gambar 3.2 <i>Diagram block system</i> .....	16
Gambar 3.3 Form tulisan tangan aksara Bima.....	18
Gambar 3.4 Proses <i>resizing</i> aksara Bima.....	19
Gambar 3.5 Citra aksara <i>Mpa</i> .....	22
Gambar 3.6 Pembagian zona ICZ.....	23
Gambar 3.7 Pembagian zona ZCZ.....	24

## DAFTAR TABEL

Tabel 3.1 Matriks A.....	20
Tabel 3.2 Normalisasi matriks A.....	21
Tabel 3.3 Data <i>dummy</i> ekstraksi fitur .....	25
Tabel 3.4 Bias dan Bobot awal dari <i>input layer</i> ke <i>hidden layer</i> pertama .....	26
Tabel 3.5 Bias dan Bobot awal dari <i>hidden layer</i> pertama ke <i>hidden layer</i> kedua.....	27
Tabel 3.6 Bias dan Bobot awal dari <i>hidden layer</i> kedua ke <i>output layer</i> .....	27
Tabel 3.7 Bias dan Bobot akhir dari <i>input layer</i> ke <i>hidden layer</i> pertama .....	34
Tabel 3.8 Bias dan Bobot akhir dari <i>hidden layer</i> pertama ke <i>hidden layer</i> kedua .....	34
Tabel 3.9 Bias dan Bobot akhir dari <i>hidden layer</i> kedua ke <i>output layer</i> .....	34
Tabel 3.10 <i>Output</i> data latih.....	35
Tabel 3.11 Contoh nilai <i>confussion matrix</i> .....	36
Tabel 3.12 Jadwal kegiatan pengembangan sistem.....	39

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Provinsi Nusa Tenggara Barat memiliki tiga suku yakni suku Sasak, Samawa, dan Mbojo atau biasa disingkat menjadi SaSaMbo. Masing – masing suku mempunyai peninggalan aksara yaitu aksara sasak (Lombok), Satera Jontal (Sumbawa), dan aksara Mbojo (Bima). Suku Bima memiliki sistem komunikasi atau bahasa tersendiri yang dilengkapi dengan sistem penulisan atau aksara yang biasa disebut dengan Aksara Mbojo atau Aksara Bima. Bima memiliki peradaban yang luar biasa, namun sempat “hilang” terpendam selama ratusan tahun. Peradaban tersebut bukan berupa tugu monumental atau candi sebagai salah satu keajaiban dunia, tapi berupa ilmu pengetahuan dalam bentuk aksara, tepatnya aksara Bima. Aksara yang sampai saat ini belum banyak kalangan yang mengenal apalagi menggunakannya dalam bentuk tulis menulis [1].

Apabila menurunnya penggunaan aksara Bima dalam kehidupan sehari-hari tidak diimbangi dengan upaya untuk melestarikan penggunaan aksara Bima dikhawatirkan dapat menyebabkan aksara Bima masuk dalam kategori *endangered language*. *Endangered languages* adalah bahasa-bahasa yang terancam punah yang tidak memiliki generasi muda sebagai penuturnya dan penutur yang fasih hanyalah kelompok generasi menengah atau dewasa [2].

Menteri Pendidikan dan Kebudayaan telah berupaya untuk mengenalkan aksara Bima kepada generasi muda dengan mengeluarkan peraturan Nomor: 79 Tahun 2014 tentang Muatan Lokal Kurikulum 2013. Pada tahun 2017 Dinas Pendidikan dan Kebudayaan Provinsi NTB merumuskan kurikulum muatan lokal jenjang pendidikan SMA dan SMK Etnis Mbojo dan pada tahun 2019, aksara Bima masuk dalam kurikulum Mulok [3].

Pelestarian aksara Bima dapat dilakukan dengan memanfaatkan kemajuan teknologi dalam bidang pengenalan pola dan pemrosesan citra digital. Penelitian terkait pengenalan pola pada aksara Bima sendiri pertama kali dilakukan dengan menggunakan metode *Shearlet Transform* dan *Support Vector Machine* yang mampu menghasilkan akurasi terbaik sebesar 90% [4]. Penelitian terkait pengenalan pola tulisan tangan aksara berupa aksara Jawa telah dilakukan dengan

menggunakan algoritma *Backpropagation* dan *Intency of Character* sebagai ekstraksi cirinya dan menghasilkan akurasi tertinggi yaitu sebesar 83% [5]. Penelitian berikutnya terkait alih aksara menggunakan metode *Image Centroid Zone – Zone Centeroid Zone (ICZ-ZCZ)* didapatkan akurasi sebesar 97.87% [6]. Sedangkan Penelitian terkait aksara Bima yang telah dilakukan sebelumnya dengan menerapkan algoritma *Rule Base* dan pendekatan Hexadesimal pada transliterasi Aksara Bima menjadi huruf latin diperoleh tingkat akurasi sebesar 90.64% [12].

Berdasarkan penelitian yang telah dipaparkan serta ditinjau dari tingkat akurasi yang dihasilkan, maka penulis bermaksud menerapkan gabungan metode *Backpropagation* untuk proses klasifikasi serta metode ICZ-ZCZ untuk ekstraksi cirinya dengan menggunakan data dan zona ekstraksi yang berbeda. Alasan dipilihnya kedua metode tersebut karena tingginya akurasi yang dihasilkan pada penelitian terkait alih aksara dan metode tersebut belum pernah diterapkan pada pengenalan aksara Bima. Hasil dari penelitian ini diharapkan dapat meningkatkan minat masyarakat luas untuk mengetahui ataupun melakukan penelitian terkait dengan aksara Bima.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang yang telah dipaparkan, rumusan masalah pada penelitian ini adalah sebagai berikut.

1. Bagaimana proses perancangan sistem pengenalan pola yang baik untuk mengenali tulisan tangan aksara Bima dengan metode *Backpropagation* dan ICZ-ZCZ.
2. Bagaimana performa terbaik yang dapat dihasilkan dalam pengenalan pola aksara Bima menggunakan *Image Centroid Zone - Zone Centroid Zone* dan *Backpropagation*.

## **1.3 Batasan Masalah**

Batasan masalah dari tugas akhir ini sebagai berikut:

1. Data yang digunakan adalah tulisan tangan aksara Bima.
2. Data yang digunakan terdiri dari 22 karakter aksara dan masing masing aksara terdiri dari 12 data.



3. Setiap aksara dilakukan ekstraksi ciri 5x7 zona atau 35 ciri per huruf.
4. Format citra akan diolah dalam bentuk .JPG

#### **1.4 Tujuan Penelitian**

Tujuan yang diharapkan dari penelitian ini adalah sebagai berikut :

1. Membangun model klasifikasi yang baik untuk mengenali aksara Bima menggunakan metode ICZ-ZCZ dan *Backpropagation*.
2. Mendapatkan performa terbaik dalam pengenalan pola aksara Bima menggunakan ICZ-ZCZ dan *Backpropagation*.

#### **1.5 Manfaat**

Manfaat dari penelitian tugas akhir ini sebagai berikut :

1. Penelitian ini diharapkan dapat memberikan pengetahuan tentang pengenalan pola menggunakan metode *Backpropagation* dan ICZ-ZCZ bagi penulis.
2. Sebagai dasar pertimbangan atau referensi bagi para peneliti khususnya di bidang ilmu pengetahuan dan pendidikan yang ingin melakukan penelitian terkait pengenalan pola dan klasifikasi aksara.
3. Sebagai salah satu media penerjemah aksara Bima ke dalam bentuk tulisan latin untuk berbagai kalangan masyarakat.

#### **1.6 Sistematika Penulisan**

Untuk mencapai tujuan yang diharapkan, maka sistematika penulisan yang disusun dalam tugas akhir ini dibagi menjadi 5 bab sebagai berikut :

##### **BAB I : PENDAHULUAN**

Bab ini membahas tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, dan manfaat penelitian.

##### **BAB II : TINJAUAN PUSTAKA DAN DASAR TEORI**

Bab ini membahas tentang tinjauan pustaka yang digunakan sebagai acuan dalam penyusunan tugas akhir dan landasan teori yang berkaitan dengan tugas akhir diantaranya: pengenalan pola, *Backpropagation*, ICZ-ZCZ (*Image Centroid Zone –Zone Centroid Zones*)

##### **BAB III : METODE PENELITIAN**

Bab ini membahas tentang metodologi yang digunakan dalam penelitian dan pengembangan perangkat lunak.

#### **BAB IV : HASIL DAN PEMBAHASAN**

Bab ini membahas tentang hasil pengujian sistem yang telah dibuat dan analisa dari hasil pengujian yang telah dilakukan.

#### **BAB V : KESIMPULAN DAN SARAN**

Bab ini membahas tentang kesimpulan dari penelitian yang dilakukan serta saran dari penulis untuk pengembangan sistem selanjutnya.

## BAB II

### TINJAUAN PUSTAKA DAN DASAR TEORI

#### 2.1 Tinjauan Pustaka

Penelitian terkait pengenalan pola menggunakan metode *Backpropagation* dengan menggabungkan metode *Image Centroid Zone – Zone Centroid Zone* (ICZ-ZCZ) sebagai ekstraksi cirinya pernah dilakukan sebelumnya. Penelitian tersebut menggunakan ciri ICZ-ZCZ 4\*5 zona dan dengan data sebanyak 340 citra untuk 15 aksara jawa cetak dengan jumlah data *training* 230 serta data testing sebanyak 110 dengan metode pengujian *3-fold cross validation* menghasilkan akurasi sebesar 97,87% [6]. Selanjutnya penelitian terkait pengenalan pola dengan metode *Backpropagation*. Penelitian dilakukan dengan menggunakan Jaringan Syaraf Tiruan *Backpropagation* dan *Intency Of Character, Mark direction* sebagai ekstraksi cirinya dan menghasilkan akurasi tertinggi yaitu sebesar 83% [5]. Penelitian terkait pengenalan karakter ciri plat nomor kendaraan menggunakan metode *Momentum Backpropagation* dan *Haar Wavelet*. Hasil uji coba memiliki tingkat akurasi hingga 97,10% [7]. Penelitian tentang pengenalan pola tulisan tangan dengan metode *Backpropagation* dan menghasilkan tingkat akurasi sebesar 96%. Arsitektur jaringan yang digunakan adalah dengan variasi jumlah iterasi 22, *learning rate* 0,05 dan jumlah *neuron hidden layer* 40 [8]. Penelitian terkait pengenalan huruf dan angka Arab atau *hijaiyah* dengan Jaringan Syaraf Tiruan *Backpropagation* menghasilkan tingkat akurasi sebesar 91,66% untuk angka arab dan 92% untuk pengenalan huruf [13].

Penelitian berikutnya terkait metode ekstraksi ciri ICZ-ZCZ dan Klasifikasi *K-Nearest Neighbour* pada pengenalan aksara Jawa. Nilai akurasi tertinggi diperoleh pada 12 zona dengan nilai  $k = 1$  pada *K-Nearest Neighbour* sebesar 71.5% [9]. Penelitian terkait metode ZCZ (*Zone Centroid Zone*) dan metode *Backpropagation* pada pengenalan angka arab timur menghasilkan tingkat akurasi sebesar 100% dengan 100 data sampel dan akurasi sebesar 86% dari 500 data sampel. Nilai *learning rate* yang digunakan 0,5 dan max Epoch 3000 [10].

Penelitian berikutnya terkait identifikasi Karakter Pada Plat Nomor Kendaraan Menggunakan Ekstraksi Fitur ICZ dan ZCZ dengan Metode Klasifikasi K-NN dengan menggunakan zona berjumlah 14 dan ekstraksi ciri ICZ adalah 97.00% sedangkan akurasi dengan menggunakan ekstraksi ciri ICZ+ZCZ adalah 98.17%. Akurasi indentifikasi per plat nomor kendaraan menggunakan ekstraksi ciri ICZ+ZCZ tanpa toleransi kesalahan adalah 72.31% dan 89.23% untuk toleransi kesalahan sebanyak 1 [11].

Berdasarkan tinjauan Pustaka yang tertera di atas, metode zoning ICZ-ZCZ terbukti baik dalam proses ekstraksi ciri serta metode *Backpropagation* terbukti dapat menghasilkan tingkat akurasi yang tinggi dalam proses klasifikasi. Oleh sebab itulah penulis memutuskan untuk menggunakan gabungan metode ICZ-ZCZ sebagai metode ekstraksi ciri tulisan tangan aksara Bimaserta dikombinasikan dengan metode *Backpropagation* sebagai metode klasifikasi pengenalan pola aksara Bima.

## **2.2 Dasar Teori**

### **2.2.1 Citra Digital**

Citra dapat didefinisikan sebagai fungsi dua dimensi dari  $f(a,b)$  dimana  $a$  dan  $b$  adalah koordinat bidang dan amplitudo dari  $f$  pada pasangan koordinat apapun adalah intensitas atau tingkat keabu-abuan dari citra pada titik tersebut. Jika  $a$ ,  $b$ , dan nilai intensitas dari  $f$  tersebut berhingga, bernilai diskrit, citra tersebut disebut citra digital[14].

### **2.2.2 Pengolahan Citra**

Pengolahan gambar atau pengolahan citra yang sering disebut *image processing*, merupakan suatu proses yang mengubah sebuah gambar menjadi gambar lain yang memiliki kualitas lebih baik untuk tujuan tertentu[15].

### **2.2.3 Pengenalan Pola**

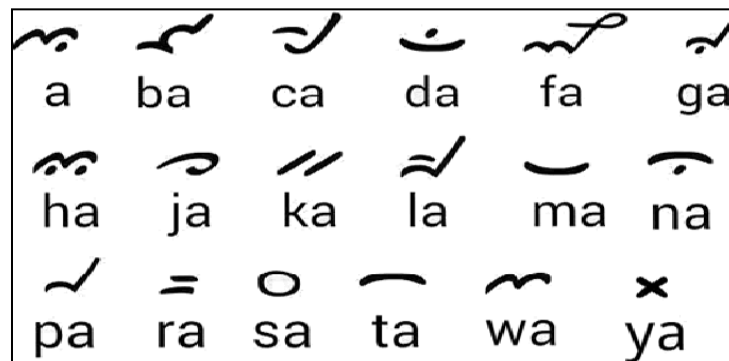
Pengenalan pola merupakan disiplin ilmu yang bertujuan untuk klasifikasi objek kedalam sejumlah kategori atau kelas. Adapun objek-objek tersebut adalah citra, gelombang sinyal, *database*, atau segala jenis ukuran yang lain yang perlu

diklasifikasikan [5]. Pengenalan pola adalah proses memberikan label berbagai golongan objek pada setiap piksel citra berdasarkan pemetaan jaringan keras dan pewilayahan berbagai jaringan lunak pada citra *biometric* [16] atau proses pengolahan data mentah yang akan diklasifikasikan sesuai ciri yang ada pada data.

#### 2.2.4 Aksara Bima

Bima memiliki peradaban yang luar biasa, Aksara Bima merupakan salah satu warisan budaya yang sempat hilang selama ratusan tahun. Aksara tersebut tersimpan dalam waktu yang lama dalam suatu naskah, yang kemudian pada tanggal 28 Juli 2007 dideklarasikan pada Simposium Internasional Pernaskahan Nusantara XI yang dilaksanakan di Bima [1].

Bima memiliki dua aksara, yaitu aksara huruf lengkung dan aksara huruf garis-garis. Aksara huruf lengkung, menurutnya sudah tidak terpakai lagi. Penulis juga telah berupaya untuk mendapatkan naskah yang ditulis dengan huruf lengkung tersebut, namun hingga saat ini belum dapat menemukannya dalam bentuk naskah. Aksara lengkung tersebut, hanya terdapat dalam catatan Thomas Raffles yang terdapat dalam lampiran 3b buku *the History of Java* [17].



Gambar 2.1 Bentuk Aksara Bima Modern

Aksara Bima Kuno, sebagaimana diakui oleh Raffles, *An alphabet formerly adopted in Bima but not now used*, pernah dipakai di Bima tapi saat ini tidak digunakan lagi[17] Adapun sampai saat ini, jenis aksara Bima yang digunakan adalah aksara Bima modern.

#### 2.2.5 Jaringan Syaraf Tiruan *Backpropagation*

*Backpropagation* adalah metode sistematis pada jaringan saraf tiruan yang menggunakan algoritma pembelajaran terawasi yang biasanya digunakan oleh

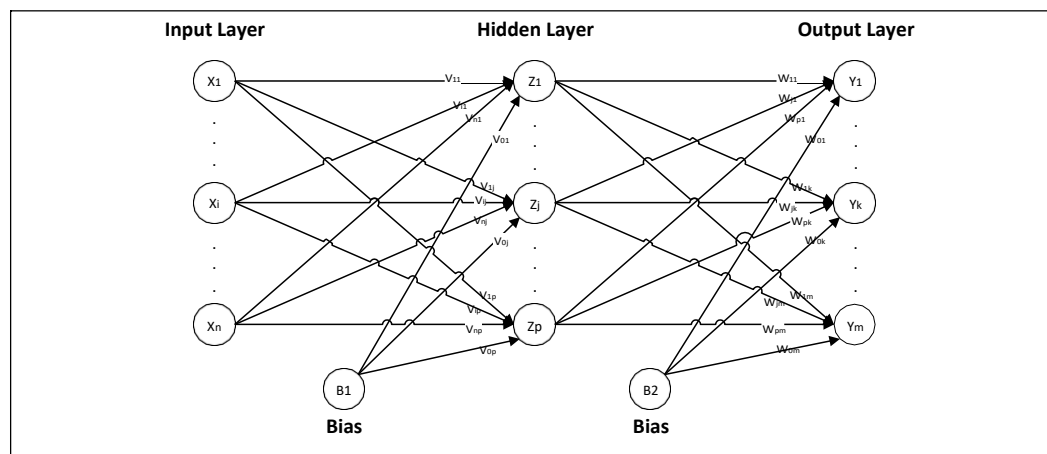
algoritma perceptron dengan banyak *layer* untuk mengubah bobot-bobot yang ada pada lapisan tersembunyi [18].

JST model *Backpropagation* terdiri dari dua tahap :

- 1) Tahap pembelajaran atau pelatihan,  
tahap ini diberikan sejumlah data input pelatihan dan juga target pada pembobotan jaringan syaraf.
- 2) Tahap pengujian atau simulasi,  
tahap setelah melalui proses pembelajaran dan pelatihan maka dilakukanlah proses pengujian sistem jaringan syaraf yang dibangun menggunakan data latih sebelumnya.

#### 2.2.5.1 Arsitektur *Backpropagation*

*Backpropagation* memiliki beberapa unit (*neuron*) yang ada dalam satu atau lebih *layer* tersembunyi. Gambar 2.3 adalah arsitektur *backpropagation multilayer* dengan 1 *hidden layer*. Pada gambar, unit *input* dilambangkan dengan X, unit *hidden* dilambangkan dengan Z, dan unit *output* dilambangkan dengan Y. Bobot antara unit *input* (X) dan unit *hidden* (Z) dilambangkan dengan V, sedangkan bobot antara unit *hidden* (Z) dan unit *output* (Y) dilambangkan dengan W.



Gambar 2.2 Arsitektur jaringan *backpropagation*

#### 2.2.5.2 Fungsi Aktivasi

Dalam *backpropagation*, fungsi aktivasi yang dipakai harus memenuhi beberapa syarat yaitu: kontinu, terdiferensial dengan mudah, dan merupakan fungsi yang tidak turun. Salah satu fungsi yang memenuhi ketiga syarat tersebut sehingga

sering dipakai adalah fungsi *sigmoid biner* yang memiliki range (0, 1) [19].  
 Persamaan fungsi aktivasi *sigmoid biner* yaitu sebagai berikut:

$$f(x) = \frac{1}{1+e^{-x}} \dots\dots\dots (2-1)$$

$$f'(x) = f(x)(1 - f(x)) \dots\dots\dots (2-2)$$

### 2.2.5.3 Algoritma *Backpropagation*

Algoritma pelatihan *backpropagation* terdiri dari proses *feed forward* dan *backpropagation*. Algoritma tersebut yaitu sebagai berikut [20]:

Langkah 0: Inisialisasi bobot (ambil bobot awal dengan nilai acak yang cukup kecil)

Langkah 1: Jika kondisi penghentian belum terpenuhi, lakukan langkah 2 sampai 9

Langkah 2: Untuk setiap pasang data pelatihan, lakukan langkah 3 sampai 8

#### **Fase I: *Feed forward***

Langkah 3: Tiap unit masukan ( $x_i, i = 1, 2, \dots, n$ ) menerima sinyal dan meneruskannya ke unit selanjutnya, yaitu lapisan tersembunyi

Langkah 4 : Hitung semua keluaran pada lapisan tersembunyi ( $Z_j, j = 1, 2, \dots, p$ )

$$Z\_net_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \dots\dots\dots (2-3)$$

Gunakan fungsi aktivasi untuk menghitung sinyal *output*-nya:

$$Z_j = f(Z\_net_j) \dots\dots\dots (2-4)$$

dan kirimkan sinyal tersebut ke semua unit lapisan atasnya (unit-unit *output*).

Langkah ini dilakukan sebanyak jumlah lapisan tersembunyi.

Langkah 5 : Hitung semua keluaran jaringan di lapisan *output* ( $Y_k, k = 1, 2, \dots, m$ )

$$Y\_net_k = w_{0k} + \sum_{j=1}^p Z_j w_{jk} \dots\dots\dots (2-5)$$

Gunakan fungsi aktivasi untuk menghitung sinyal *output*-nya:

$$Y_k = f(y\_net_k) \dots\dots\dots (2-6)$$

#### **Fase II: *Backpropagation***

Langkah 6: Hitung faktor  $\delta$  unit keluaran berdasarkan kesalahan di setiap unit keluaran ( $y_k, k = 1, 2, \dots, m$ )

$$\delta_k = (t_k - y_k) f'(y\_net_k) \dots\dots\dots (2-7)$$

$\delta$  merupakan unit kesalahan yang akan dipakai dalam perubahan bobot *layer* di bawahnya (langkah 7).  $f'(y_{net_k})$  merupakan fungsi turunan dari fungsi aktivasi *sigmoid biner*.

Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki  $w_{jk}$ ) dengan laju percepatan  $\alpha$

$$\Delta w_{jk} = \alpha \cdot \delta_k \cdot z_j \dots \dots \dots (2-8)$$

Kemudian hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai  $w_{0k}$ )

$$\Delta w_{0k} = \alpha \cdot \delta_k \dots \dots \dots (2-9)$$

Langkah 7: Hitung faktor  $\delta$  unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi ( $z_j, j = 1, 2, \dots, p$ )

$$\delta_{net_j} = \sum_{k=1}^m \delta_k \cdot w_{jk} \dots \dots \dots (2-10)$$

Faktor  $\delta$  unit tersembunyi:

$$\delta_j = \delta_{net_j} f'(z_{net_j}) \dots \dots \dots (2-11)$$

Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai  $v_{ij}$ )

$$\Delta v_{ij} = \alpha \cdot \delta_j \cdot x_i \dots \dots \dots (2-12)$$

Kemudian hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai  $v_{0j}$ )

$$\Delta v_{0j} = \alpha \cdot \delta_j \dots \dots \dots (2-13)$$

### Fase III: Perubahan Bobot

Langkah 8: Tiap-tiap unit *output* ( $Y_k, k = 1, 2, \dots, m$ ) memperbaiki bobotnya ( $j = 0, 1, 2, \dots, p$ )

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \dots \dots \dots (2-14)$$

Tiap-tiap unit tersembunyi ( $Z_j, j = 1, 2, 3, \dots, p$ ) memperbaiki bobotnya ( $j = 0, 1, 2, 3, \dots, n$ )

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \dots \dots \dots (2-15)$$

Langkah 9: Kondisi pelatihan berhenti

Ketiga fase tersebut diulang terus menerus hingga kondisi penghentian dipenuhi. Umumnya kondisi penghentian yang sering dipakai adalah jumlah iterasi atau kesalahan. Iterasi akan dihentikan jika jumlah iterasi yang dilakukan sudah



melebihi jumlah maksimum iterasi yang ditetapkan, atau jika kesalahan yang terjadi sudah lebih kecil dari batas toleransi yang diijinkan[21].

### 2.2.6 Ekstraksi Ciri

Ekstraksi ciri adalah suatu proses atau cara untuk mendapatkan informasi dari suatu objek, baik berupa citra, tulisan, ataupun suara, dimana informasi ciri yang didapatkan dapat membedakan antara satu objek dengan objek yang lainnya [8].

### 2.2.7 ICZ-ZCZ

ICZ-ZCZ (*Image Centroid Zone – Zone Centroid Zone*) merupakan metode ekstraksi zoning. Metode ekstraksi zoning akan membagi citra menjadi beberapa zona yang berukuran sama, untuk kemudian dari setiap zona akan diambil cirinya. Ada beberapa algoritme untuk metode ekstraksi ciri zoning, di antaranya metode ekstraksi ciri jarak metrik ICZ (*image centroid and zone*), metode ekstraksi ciri jarak metrik ZCZ (*zone centroid and zone*), dan metode ekstraksi ciri gabungan ICZ + ZCZ. Ketiga algoritma tersebut menggunakan citra digital sebagai input dan menghasilkan fitur untuk klasifikasi dan pengenalan sebagai output-nya. Berikut merupakan tahapan dalam proses ekstraksi ciri ICZ, ZCZ dan ICZ+ZCZ [22]

Algoritme 1: *Image Centroid and Zone (ICZ)* berdasarkan jarak metrik.

1. Hitung *centroid* dari citra masukan
2. Bagi citra masukan ke dalam n zona yang sama
3. Hitung jarak antara *centroid* citra dengan masing-masing piksel yang ada dalam zona
4. Ulangi langkah ke 3 untuk setiap piksel yang ada di zona
5. Hitung rata-rata jarak antara titik-titik tersebut
6. Ulangi langkah-langkah tersebut untuk keseluruhan zona
7. Hasilnya adalah n fitur yang akan digunakan dalam klasifikasi dan pengenalan

Algoritma 2: *Zone Centroid Zone (ZCZ)* berdasarkan jarak metrik.

1. Bagi citra masukan ke dalam sejumlah n bagian yang sama
2. Hitung *centroid* dari masing-masing zona
3. Hitung jarak antara *centroid* masing-masing zona dan piksel yang ada di zona

4. Ulangi langkah ke 3 untuk seluruh piksel yang ada di zona
5. Hitung rata-rata jarak antara titik-titik tersebut
6. Ulangi langkah 3-7 untuk setiap zona secara berurutan
7. Hasilnya adalah n fitur yang akan digunakan dalam klasifikasi dan pengenalan

Algoritme 3: ICZ+ZCZ berdasarkan jarak metrik.

1. Hitung *centroid* dari citra masukan
2. Bagi citra masukan ke dalam sejumlah n bagian yang sama
3. Hitung jarak antara *centroid* citra dengan setiap piksel yang ada dalam zona
4. Ulangi langkah 3 untuk semua piksel yang ada dalam zona
5. Hitung jarak rata-rata antara titik-titik tersebut
6. Hitung *centroid* tiap zona
7. Hitung jarak antara *centroid* zona dengan setiap piksel yang ada dalam zona
8. Ulangi langkah 7 untuk semua piksel yang ada dalam zona
9. Hitung jarak rata-rata antara titik-titik tersebut
10. Ulangi langkah 3-9 untuk semua zona secara berurutan
11. Hasilnya akan didapatkan 2n ciri untuk klasifikasi dan pengenalan

### 2.2.8 Confusion Matrix

Matriks Kebingungan adalah suatu matriks yang digunakan untuk menunjukkan ringkasan dari hasil predeksi dalam permasalahan klasifikasi. Matriks Kebingungan dapat menunjukkan cara dimana suatu model kalsifikasi disalahartikan ketika melakukan prediksi [17]. Bentuk dari *confusion matrix* sendiri dapat dilihat pada Gambar 2.3 berikut.

		Kondisi sebenarnya	
		Kondisi benar	Kondisi salah
Prediksi	Prediksi benar	<i>True Positives</i>	<i>False Positives</i>
	Prediksi salah	<i>False Negatives</i>	<i>True Negatives</i>

Gambar 2.3 Bentuk *confussion matrix*

*Confussion matrix* merupakan suatu matriks yang berukuran 2 x 2, yang pada umumnya setiap blok pada matrikstersebut terbagi menjadi 4 bagian. *True*

*Positive* (TP) merupakan bagian yang menunjukkan prediksi bernilai benar untuk kondisi yang sesungguhnya bernilai benar. *False Positive* (FP) merupakan bagian yang menunjukkan prediksi yang bernilai benar yang dimana pada kondisi sebenarnya bernilai salah. *True Negative* (TN) merupakan bagian menunjukkan prediksi bernilai salah pada kondisi yang sesungguhnya bernilai salah. *False Negative* (FN) merupakan bagian yang menunjukkan prediksi yang bernilai salah, yang dimana pada kondisi sebenarnya bernilai benar [23].

### 2.2.8.1 Akurasi

Akurasi merupakan salah satu tolak ukur untuk pengujian dari hasil penelitian. Akurasi digunakan untuk menghitung besarnya rasio prediksi benar dari keseluruhan data uji [24]. Adapun Persamaan yang digunakan untuk menghitung akurasi terdapat pada Persamaan (2-17)

$$Akurasi = \frac{\text{jumlah pengujian yang benar}}{\text{total data uji}} \times 100\% \dots\dots\dots (2-17)$$

### 2.2.8.2 Presisi

Presisi merupakan besarnya rasio prediksi benar (*True Positive*) dari keseluruhan hasil prediksi benar (*True Positive* dan *False Positive*) [24]. Adapun cara untuk menghitung presisi, digunakan Persamaan (2-18)

$$Presisi = \frac{TP}{TP+FP} \dots\dots\dots (2-18)$$

### 2.2.8.3 Recall

*Recall* merupakan rasio prediksi benar positif (*True Positives*) dengan keseluruhan hasil yang benar positif (*True Positives* dan *False Negatives*). Adapun perhitungan yang digunakan dalam menghitung *recall* dapat dilihat pada Persamaan (2-19) [24].

$$Recall = \frac{TP}{TP+FN} \dots\dots\dots (2-19)$$

## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Alat dan Bahan Penelitian

Alat-alat yang digunakan dalam penelitian ini yaitu sebagai berikut:

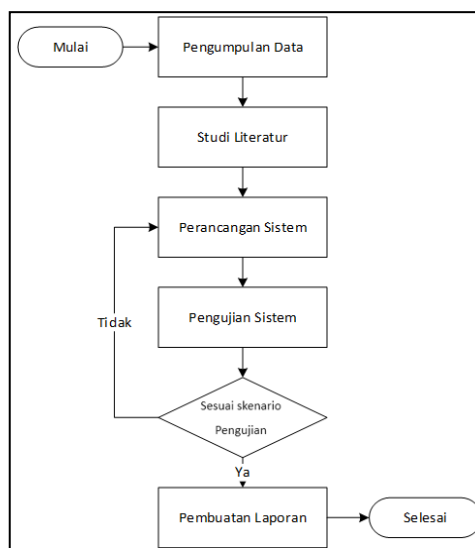
1. Laptop Asus X454YA AMD A8-7410.
2. Sistem operasi Windows 10 Professional Edition.
3. Perangkat lunak: Jupiter Lab, Microsoft Office 2016, Microsoft Excel 2016.
4. Canon MP287 untuk proses *scanning*.

Bahan-bahan yang digunakan dalam penelitian ini yaitu sebagai berikut:

1. Jurnal, buku, serta penelitian-penelitian yang berkaitan dengan pengenalan pola, *neural network*, aksara Bima, metode *backpropagation* dan metode ICZ-ZCZ.
2. Dataset tulisan tangan aksara Bima sebanyak 10560 yang terdiri dari 22 karakter dalam format JPG.

#### 3.2 Rancangan Penelitian

Pada tahap rancangan penelitian dilakukan beberapa tahap secara sistematis, dimulai dari tahap studi literatur hingga tahap penarikan kesimpulan dari sistem yang telah dibuat dan diuji. Tahap proses penelitian ini dapat dilihat pada diagram alir Gambar 3.1



Gambar 3.1 Diagram alir tahap proses penelitian

Tahapan-tahapan dari diagram alir tahap proses penelitian pada Gambar 3.1 dapat dijelaskan sebagai berikut:

#### 1. Pengumpulan Data

Pengumpulan data dilakukan untuk memperoleh data-data yang akan digunakan dalam penelitian. Data yang dikumpulkan adalah tulisan tangan aksara Bima dari sejumlah mahasiswa baik itu mahasiswa asal Bima maupun non-Bima. Setiap mahasiswa menulis 22 huruf aksara Bima dimana setiap huruf ditulis sebanyak 12 kali. Data yang sudah terkumpul selanjutnya akan melalui proses *scanning* lalu *cropping* per-huruf dalam format .JPG berukuran 128 x 128 pixel.

#### 2. Studi Literatur

Studi literatur dilakukan dengan mempelajari jurnal penelitian, buku-buku, serta sumber lain yang berkaitan dengan permasalahan yang diangkat. Adapun materi yang dipelajari dalam studi literatur berkaitan dengan aksara Bima, pengenalan pola, *artificial neural network*, dan metode *backpropagation* serta metode ICZ-ZCZ.

#### 3. Perancangan Sistem

Tahap perancangan sistem yang berfungsi untuk pengolahan data terdiri dari proses *training* dan *testing*. Adapun proses training terdiri dari Akuisisi data, *cropping*, *Preprocessing*, dan ekstraksi ciri kemudian pembentukan dataset aksara. Sedangkan tahapan *testing* yaitu Akuisisi data, *cropping*, *Preprocessing*, Ekstraksi ciri, klasifikasi dan pengujian. Perbedaan proses training dan testing adalah pada proses klasifikasi dan pengujian dimana hasil klasifikasi pada proses *testing* diperlukan dataset aksara yang telah dibuat pada proses *training*

#### 4. Pengujian Sistem

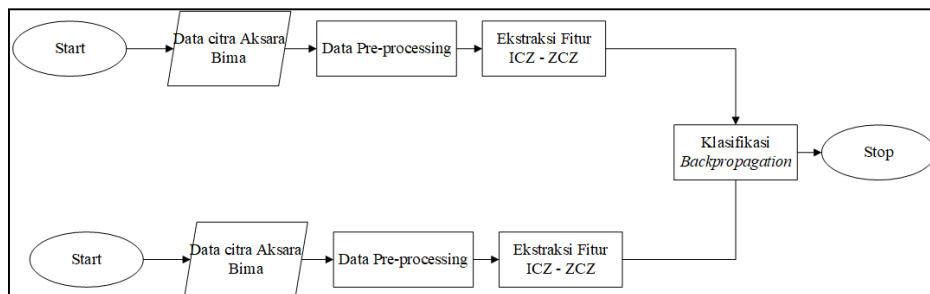
Pengujian sistem dilakukan untuk mengetahui kelayakan sistem berdasarkan tingkat akurasi dari proses alih aksara. Metode yang digunakan untuk menghitung akurasi adalah dengan *confusion matrix* dengan membagi 2/3 data sebagai data *training* dan 1/3 data sebagai data *testing*. Hasil dari perhitungan akurasi merupakan acuan untuk penarikan kesimpulan. Jika sistem belum sesuai standar, maka penelitian perlu diulang ke tahap perancangan sistem untuk memastikan sistem yang lebih baik dan akurat. Tahapan pengujian sistem akan dijelaskan lebih lanjut pada Bab 4.

## 5. Pembuatan Laporan

Tahap akhir yaitu pembuatan laporan dan penarikan kesimpulan yang didapatkan berdasarkan hasil pengujian dari sistem dan apakah sistem sudah mampu menjawab kebutuhan sistem berdasarkan dari tujuan penelitian. Kesimpulan akan dijabarkan pada Bab 5.

### 3.3 Perancangan Algoritma

Sub bab ini akan menjelaskan bagaimana sistem dirancang mulai dari tahap pelatihan sistem sampai dengan sistem dapat mengklasifikasikan aksara. Blok diagram sistem terlihat pada Gambar 3.2.



Gambar 3.2 Diagram Block System

Proses pelatihan dan proses pengujian akan dijelaskan sebagai berikut:

#### a. Proses pelatihan

Proses pelatihan atau *training* mengikuti tahap tahap sebagai berikut:

1. Citra yang diinput ke dalam sistem merupakan citra yang diambil secara langsung dari tulisan tangan yang ditulis dalam selebar kertas F4. Tulisan tersebut di-*scan* dan disimpan dalam bentuk format .JPG kemudian di-*cropping* sesuai dengan banyaknya huruf aksara yaitu sebanyak 22 huruf kemudian akan dipotong menjadi ukuran 128x128 pixel dalam bentuk format “jpg”.
2. Tahap *preprocessing*, yakni tahap manipulasi gambar sesuai keinginan. Karena proses *crop* dan *resize* telah dilakukan di luar sistem maka proses selanjutnya yang akan dilakukan didalam sistem yakni proses konversi ruang warna menjadi *grayscale*.
3. Gambar/citra akan diekstraksi menggunakan metode ICZ-ZCZ untuk mendapatkan ciri dari masing-masing citra aksara Bima dan kemudian hasil dari

ekstraksi ciri tersebut akan menjadi data latih untuk sistem dan sebagai data masukan untuk proses klasifikasi dengan metode *backpropagation*.


4. Proses klasifikasi yang digunakan pada penelitian ini adalah *Backpropagation*.
  5. Hasil klasifikasi disimpan oleh sistem sebagai proses pelatihan kemudian hasil klasifikasi yang telah diproses oleh sistem dibandingkan dengan target apakah sudah sesuai atau tidak. Kemudian tahap selanjutnya yaitu proses pengujian.
- b. Proses Pengujian

Proses pengujian atau *testing* terdiri dari tahap-tahap sebagai berikut:

1. *Input citra* aksara untuk klasifikasi (citra pengujian). Citra yang dimasukkan yaitu citra aksara yang telah di-*crop* dan di-*resize* di luar sistem.
2. Tahap *preprocessing* yang dilakukan di dalam sistem pada proses klasifikasi sama dengan pada proses pelatihan yakni konversi ruang warna.
3. Ekstraksi fitur dilakukan untuk menentukan ciri dari masing-masing aksara sebanyak 22 karakter. ICZ-ZCZ adalah metode ekstraksi fitur dan reduksi dimensi yang digunakan di dalam sistem.
4. Tahap klasifikasi dilakukan dengan metode *Backpropagation* untuk mengetahui karakter aksara Bima. Data hasil pelatihan dimuat untuk dibandingkan dengan data uji.
5. Keluaran akhir dari proses klasifikasi berupa jenis huruf dari karakter aksara Bima.

### **3.4 Data Acquisition**

Data *acquisition* adalah proses pengambilan atau pengumpulan data yang akan digunakan pada proses *training* dan *testing*. Pada penelitian ini, data yang dibutuhkan adalah citra aksara Bima yang nantinya akan di proses. Pengambilan citra aksara Bima menggunakan *template* dari kolom tabel dengan ukuran panjang dan lebar sebesar 4cmx4cm sebanyak 12 kotak sebanyak 22 lembar sesuai dengan karakter huruf. Kertas yang digunakan yaitu HVS F4 80GSM. *Template* terdiri dari dua baris dan enam kolom dan ditulis oleh narasumber menggunakan pulpen yang memiliki ketebalan 1.0mm.

		Nama : <i>Muhammad</i> Tanggal : <i>18-03-2020</i> Orang Bima / <i>Bekas-Orang-Bima</i>															
<table border="1"> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </table>																	

Gambar 3.3 *Form* tulisan tangan aksara Bima

Pengumpulan dataset tulisan tangan aksara Bima dibagi menjadi dua kategori yaitu mahasiswa yang bersal dari Bima dan non Bima. Pembagian kategoribertujuan untuk membandingkan tingkat keakuratan penulisan antara orang Bima dengan non Bima yang mungkin belum pernah mengenal atau mempelajari aksara Bima sebelumnya.

Data diperoleh dari total 40 orang dan masing-masing orang menuliskan 22 karakter masing masing sebanyak 12 kali. Jumlah total dataset yang terkumpul sebanyak 10560.

### 3.5 *Pre-processing*

*Pre-processing* dilakukan untuk memperbaiki citra agar citra yang diolah memiliki hasil yang optimal, oleh karena itu *pre-processing* yang dilakukan di penelitian ini sebagai berikut :

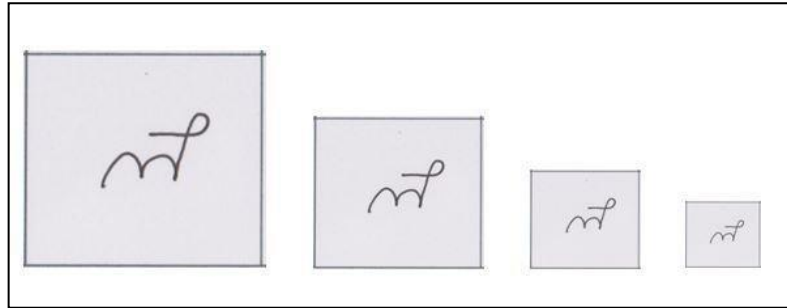
#### 1. *Cropping*

*Cropping* adalah proses pemotongan citra pada elemen tertentu pada area citra. Proses ini bertujuan untuk mengambil elemen yang diinginkan dari citra.

#### 2. *Resize*

*Resize* merupakan proses perubahan ukuran *pixel* citra. Proses ini dilakukan untuk mengubah citra *pixel* menjadi 128x128, 64x64 dan 32x32 untuk mempermudah proses pada sistem dan melihat pengaruh *size* terhadap penelitian. Semakin kecil *pixel* pada citra maka proses yang ada pada sistem akan lebih cepat.





Gambar 3.4 Proses *resizing* aksara Bima

### 3. *Grayscale*

*Grayscale* adalah merubah citra warna menjadi citra berwarna keabuan. *Grayscale* memungkinkan nilai minimal dan warna putih untuk nilai maksimal. Banyaknya warna tergantung pada jumlah bit yang disediakan di memori untuk menampung kebutuhan warna tersebut. Semakin besar jumlah bit warna yang disediakan di memori, maka semakin halus gradasi warna yang terbentuk [12]. Sehingga pada penelitian ini dibutuhkan proses *greyscale* untuk meringkan kinerja sistem saat proses pelatihan dan pengujian.

### 4. Reduksi *dimensi*

Reduksi dimensi dilakukan pada penelitian ini untuk mengubah citra 2D menjadi citra 1D. Reduksi dimensi dilakukan untuk mempermudah proses perhitungan rata-rata baris pada citra. Berikut persamaan untuk reduksi dimensi pada persamaan (3-1)

$$S = \text{citra}_{(ix,j,l)} \dots \dots \dots (3-1)$$

Gambar 3.3 merupakan contoh reduksi dimensi pada matriks 3x3 dengan persamaan 3-1:

$$C1 = \begin{pmatrix} 101 & 125 & 143 \\ 155 & 99 & 114 \\ 135 & 166 & 127 \end{pmatrix} \quad C2 = \begin{pmatrix} 102 & 103 & 150 \\ 95 & 97 & 111 \\ 123 & 130 & 114 \end{pmatrix}$$

Matriks A direduksi menjadi 1 dimensi menjadi

$$C1 = \begin{bmatrix} 101 \\ 125 \\ 143 \\ 155 \\ 99 \\ 114 \\ 135 \\ 166 \\ 127 \end{bmatrix} \quad C2 = \begin{bmatrix} 102 \\ 103 \\ 150 \\ 95 \\ 97 \\ 111 \\ 123 \\ 130 \\ 114 \end{bmatrix}$$

Reduksi dimensi dilakukan pada semua citra yang ada pada dataset untuk mengubahnya menjadi 1 dimensi. Kemudian semua citra yang ada digabungkan menjadi satu agar terbentuk matriks A menggunakan persamaan (3-2)

$$A = [C1, C1 + 1, \dots \dots \dots Cn]^T \dots \dots \dots (3-2)$$

Pada tabel 3.1 yang berisi 8 citra berbeda yang sudah di reduksi menjadi 1 dimensi disusun menggunakan persamaan (3-2).

Tabel 3.1 Matriks A

Citra	Fitur								
c1	101	125	143	155	99	114	135	166	127
c2	102	103	150	95	97	111	123	130	114
c3	170	180	183	185	165	153	144	153	171
c4	119	127	145	177	120	144	177	147	166
c5	164	142	111	149	152	146	143	111	146
c6	123	137	171	141	137	163	134	168	134
c7	152	138	109	149	146	155	155	140	109
c8	121	130	179	152	120	152	130	138	146

## 5. Normalisasi

Normalisasi adalah proses mengubah nilai agar nilai bernilai antara 0 dan 1. Hal ini dilakukan pada setiap citra untuk mempermudah perhitungan. Berikut persamaan (3-3) untuk proses normalisasi data.

$$N_{(ij)} = \frac{fitur_{(ij)} - min_{(j)}}{max_{(j)} - min_{(j)}} \dots \dots \dots (3-3)$$

Berikut contoh perhitungan menggunakan persamaan (3-3)

$$N_{(1,1)} = \frac{101 - 99}{166 - 99} = 0,185714286$$

$$N_{(2,1)} = \frac{125 - 99}{166 - 99} = 0,357142857$$

Selanjutnya setiap baris dan kolom dilakukan perhitungan seperti di atas pada matriks A. Tabel 3.2 merupakan hasil normalisasi dari matriks A.

Tabel 3.2 Normalisasi matriks A

Fitur								
0.0298507	0.3880597	0.6567164	0.8358208	0	0.2238805	0.5373134	1	0.4179104
0.1272272	0.1454545	1	0.	0.0363636	0.1090909	0.5090909	0.6363636	0.3454545
0.6341463	0.8780487	0.9512195	1	0.5121951	0.2195121	0	0.2195121	0.6585365
0	0.1379310	0.4482758	1	0.0172413	0.4310344	1	0.4827586	0.7758620
1	0.5849056	0	0.7169811	0.7735849	0.6603773	0.6037735	0	0.6603773
0	0.2083333	1	0.2916666	0.2083333	0.8333333	0.2291666	0.9375	0.2291666
0.9347826	0.6304347	0	0.8695652	0.8043478	1	1	0.6739130	0
0	0.1694915	1	0.5423728	0	0.5423728	0.1694915	0.3050847	0.4406779

### 3.6 Ekstraksi Fitur

Dalam metode ZCZ, pusat dari gambar dihitung sebagai ganti *centroid* dari masing-masing zona. Untuk masing-masing zona, jarak rata-rata antara piksel gambar hitam dan pusat gambar adalah dihitung. Jarak rata-rata kemudian digunakan sebagai fitur untuk klasifikasi [25]. Untuk menemukan titik *centroid* dapat dinyatakan dengan persamaan rumus sebagai berikut:

$$XC = \frac{\sum_{i=1}^n x_i \cdot p_i}{\sum_{i=1}^n p_i} \dots\dots\dots (3-11)$$

$$YC = \frac{\sum_{i=1}^n y_i \cdot p_i}{\sum_{i=1}^n p_i} \dots\dots\dots (3-12)$$

dimana:

xc : *centroid* pada koordinat x

yc : *centroid* pada koordinat y

$x_i$  : koordinat titik x ke-i  
 $y_i$  : koordinat titik y ke-i  
 $p_i$  : nilai piksel pada titik ke-i

Dalam perhitungan jarak antara *centroid* dengan nilai piksel yang ada di dalam zona, digunakan perhitungan jarak *euclidean distance* dengan rumus pada persamaan (3 – 1.3)

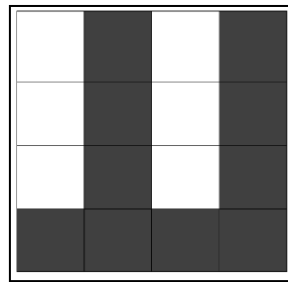
$$d(x, y) = \sqrt{\sum_{k=1}^n (x_k - y_k)^2} \dots\dots\dots (3-13)$$

dimana:

$d(x,y)$  : jarak antara vektor objek x dan y

$n$  : jumlah dimensi objek

$x_k, y_k$  : nilai vektor objek x dan y ke-k



Gambar 3.5 Citra Aksara *Mpa*

Terdapat sebuah citra aksara Bima *mpa* seperti Gambar 3.5. Proses ekstraksi ciridengan menggunakan metode ICZ yaitu pertama-tama menentukan titik *centroid* zona berdasarkan citra warna putih yang direpresentasikan dengan nilai biner 1 dan untuk citra berwarna hitam direpresentasikan dengan nilai 0. Untuk menentukan titik *centroid* pada sumbu  $x$  dan sumbu  $y$  digunakan rumus persamaan (3-11) dan (3-12) sebagai berikut

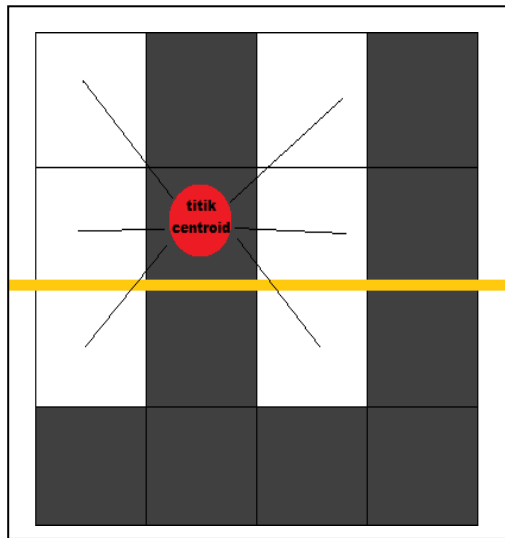
$$x_c = \frac{1+3++1+3+1+3}{6} \dots\dots\dots (3 - 11)$$

$$x_c = \frac{12}{6} = 2$$

$$y_c = \frac{1+1+2+2+3+3}{6} \dots\dots\dots (3 - 12)$$

$$y_c = \frac{12}{6} = 2$$

Berdasarkan hasil yang diperoleh maka koordinat *centroid* citra aksara tersebut berada di titik (2 , 2). Setelah titik *centroid* diperoleh maka objek tersebut dibagi menjadi beberapa zona. Sebagai contoh citra aksara *mpa* dibagi menjadi 2 zona secara horizontal selanjutnya menentukan jarak masing-masing piksel yang ada disetiap zona terhadap titik *centroid*-nya. Total nilai jarak yang ada disetiap zona merupakan nilai rata-rata dari jarak semua piksel pada zona tersebut. Hasil dari ekstraksi ciri ICZ akan menghasilkan 2 fitur.



Gambar 3.6 Pembagian Zona ICZ

Gambar 3.4 menunjukkan representasi zonasi pada aksara *mpa* menggunakan metode *Image Centroid Zone* (ICZ) yang dibagi menjadi 2 zona secara horizontal. Berikut ini merupakan hasil perhitungan fitur menggunakan *Euclidean distance* pada Zona I menggunakan rumus persamaan (3 – 13):

$$d_1 = \sqrt{(1 - 2)^2 + (1 - 2)^2} = 1.41$$

$$d_2 = \sqrt{(3 - 2)^2 + (1 - 2)^2} = 1.41$$

$$d_3 = \sqrt{(1 - 2)^2 + (2 - 2)^2} = 1.00$$

$$d_4 = \sqrt{(3 - 2)^2 + (2 - 2)^2} = 1.00$$

kemudian untuk mendapatkan hasil akhir nilai fitur digunakan persamaan (3-14) :

$$ICZ \text{ fitur}_1 = \frac{d_1 + d_2 + d_3 + \dots + d_n}{n}$$

$$ICZ \text{ fitur}_1 = \frac{1.41 + 1.41 + 1 + 1}{4}$$

$$ICZ \text{ fitur}_1 = \frac{4.82}{4} = 1.205$$

Untuk menentukan fitur ICZ zona II dilakukan perhitungan yang sama dengan menggunakan rumus persamaan *Euclidean distance* sebagai berikut:

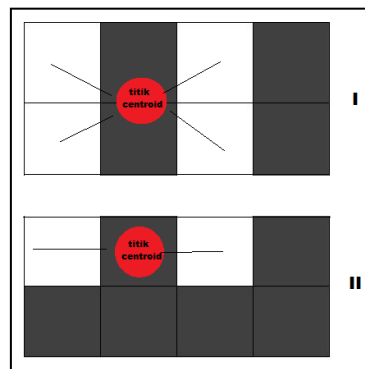
$$d_1 = \sqrt{(1 - 2)^2 + (3 - 2)^2} = 1.41$$

$$d_2 = \sqrt{(3 - 2)^2 + (3 - 2)^2} = 1.41$$

$$ICZ\ fitur_2 = \frac{1.41 + 1.41}{2}$$

$$ICZ\ fitur_2 = \frac{2.82}{2} = 1.41$$

Nilai hasil fitur zona I dan zona II menggunakan metode ICZ yang diperoleh berdasarkan nilai rata-rata berturut-turut adalah sebesar 1.205 dan 1.41 sehingga jika digabungkan menghasilkan 2 fitur [1.205 , 1.410].



Penentuan titik *centroid*  $x_c$  dan  $y_c$  pada zona I menggunakan metode ZCZ

$$: x_c = \frac{1+3+1+3}{4}$$

$$x_c = \frac{8}{4} = 2$$

$$y_c = \frac{1+1+2+2}{4}$$

$$y_c = \frac{6}{4} = 1.5$$

sehingga didapatkan nilai titik *centroid* pada zona I sebesar (2, 1.5) kemudian dengan didapatkan nilai pada zona II sebesar (2, 1).

Selanjutnya dilakukan perhitungan untuk menentukan fitur pada zona I menggunakan rumus *euclidean distance*(3 - 1.3) :

$$d_1 = \sqrt{(1 - 2)^2 + (1 - 1.5)^2} = 1.11$$

$$d_2 = \sqrt{(3 - 2)^2 + (1 - 1.5)^2} = 1.11$$

$$d_3 = \sqrt{(1 - 2)^2 + (2 - 1.5)^2} = 1.11$$

$$d_4 = \sqrt{(3 - 2)^2 + (2 - 1.5)^2} = 1.11$$

$$ZCZ_{fitur_1} = \frac{1.11+1.11+1.11+1.11}{4}$$

$$ZCZ_{fitur_1} = \frac{4.44}{4} = 1.11$$

Untuk menentukan fitur ZCZ zona II dilakukan perhitungan yang sama dengan menggunakan rumus persamaan *Euclidean distance* sebagai berikut:

$$d_1 = \sqrt{(1 - 2)^2 + (1 - 1)^2} = 1$$

$$d_2 = \sqrt{(3 - 2)^2 + (1 - 1)^2} = 1$$

$$ZCZ_{fitur_2} = \frac{1+1}{2}$$

$$ZCZ_{fitur_2} = \frac{2}{2} = 1$$

Nilai hasil fitur zona I dan zona II menggunakan metode ZCZ yang diperoleh berdasarkan nilai rata-rata berturut-turut adalah sebesar 1.11 dan 1 sehingga jika digabungkan menghasilkan 2 fitur [1.11 , 1.00]

### 3.7 Klasifikasi

#### 3.7.1 Inisialisasi Bobot

Berikut merupakan contoh perhitungan metode *backpropagation* dengan menggunakan data *dummy* dari perhitungan fitur – fitur pada metode ICZ-ZCZ yang tertera pada Tabel 3.3.

Tabel 3.3 Data *dummy* ekstraksi fitur

No	X1	X2	X3	Target
1	0	1	0	1
2	0	1	1	0
3	1	0	0	1
4	0	0	1	0

Dalam pelatihan data ini, arsitektur JST *backpropagation* yang digunakan dapat dilihat pada Gambar 3.8





Bias dan bobot awal dari *input layer* ke *hidden layer* pertama dinyatakan dengan  $V_{01}$  sampai  $V_{36}$ , dapat dilihat pada Tabel 3.4. Selanjutnya bias dan bobot awal dari *hidden layer* pertama ke *hidden layer* kedua dinyatakan  $V_{09}$  sampai  $V_{69}$  pada tabel dibawah:

Tabel 3.5 Bias dan bobot awal dari *hidden layer* pertama ke *output layer* kedua

Dari- Ke-	B2	Z1	Z2	Z3	Z4	Z5	Z6
Z7	0.0189	0.0122	0.1423	0.0335	0.0150	0.0181	0.0316
Z8	0.0155	0.0133	0.1130	0.0404	0.1010	0.0296	0.0317
Z9	0.1072	0.0227	0.0811	0.0190	0.0643	0.0099	0.1270

Bias dan bobot awal dari *hidden layer* kedua ke *output layer* dinyatakan dengan  $W_{01}$ ,  $W_{71}$ ,  $W_{81}$  dan  $W_{91}$ , yaitu:

Tabel 3.6 Bias dan bobot awal dari *hidden layer* kedua ke *output layer*

Dari- Ke-	B3	Z7	Z8	Z9
Y1	0.044	0.107	0.111	0.105

Langkah 1: Jika kondisi penghentian belum terpenuhi, lakukan langkah 2 sampai 9.

Kondisi penghentian terpenuhi jika  $error < 0.0001$  atau  $epoch > 100$ .

Langkah 2: Untuk setiap data pelatihan, lakukan langkah 3 sampai Langkah 8

### 3.7.2 Feed Forward

Langkah 3: Tiap unit masukan ( $x_i, i = 1, 2, \dots, n$ ) menerima sinyal dan meneruskannya ke unit selanjutnya, yaitu lapisan tersembunyi.

Langkah 4 : Hitung semua keluaran pada lapisan tersembunyi ( $Z_j, j = 1, 2, \dots, p$ ) menggunakan persamaan dibawah ini

$$Z_{netj} = v_{0j} + \sum_{i=1}^n x_i v_{ij} \dots \dots \dots (2-9)$$

berikut ini merupakan hasil perhitungan untuk *hidden layer* pertama :

$$\begin{aligned} Z_{\text{net}1} &= 0.011 + (0 * 0.123) + (1 * 0.379) + (0 * 0.069) \\ &= 0.390 \end{aligned}$$

$$\begin{aligned} Z_{\text{net}2} &= 0.263 + (0 * 0.487) + (1 * 0.015) + (0 * 0.322) \\ &= 0.278 \end{aligned}$$

$$\begin{aligned} Z_{\text{net}3} &= 0.057 + (0 * 0.191) + (1 * 0.024) + (0 * 0.183) \\ &= 0.081 \end{aligned}$$

$$\begin{aligned} Z_{\text{net}4} &= 0.109 + (0 * 0.165) + (1 * 0.106) + (0 * 0.021) \\ &= 0.215 \end{aligned}$$

$$\begin{aligned} Z_{\text{net}5} &= 0.035 + (0 * 0.081) + (1 * 0.077) + (0 * 0.045) \\ &= 0.112 \end{aligned}$$

$$\begin{aligned} Z_{\text{net}6} &= 0.111 + (0 * 0.123) + (1 * 0.045) + (0 * 0.078) \\ &= 0.156 \end{aligned}$$

Kemudian gunakan fungsi aktivasi untuk menghitung sinyal *output* untuk *hidden layer* pertamanya dengan persamaan (2-10) di bawah ini :

$$Z_j = f'(Z_-) \dots\dots\dots(2-10)$$

$$Z_1 = \frac{1}{1+e^{-0.390}} = 0.5962$$

$$Z_2 = \frac{1}{1+e^{-0.278}} = 0.5690$$

$$Z_3 = \frac{1}{1+e^{-0.081}} = 0.5202$$

$$Z_4 = \frac{1}{1+e^{-0.215}} = 0.5535$$

$$Z_5 = \frac{1}{1+e^{-0.112}} = 0.5279$$

$$Z_6 = \frac{1}{1+e^{-0.156}} = 0.5389$$

Selanjutnya sinyal *output* tersebut dikirim dari *hidden layer* pertama ke semua unit *hidden layer* kedua.

$$\begin{aligned} Z_{\text{net}7} &= 0.0189 + (0.5962 * 0.0122) + (0.5690 * 0.1423) + (0.5202 * 0.0335) + \\ &\quad (0.5535 * 0.0150) + (0.5279 * 0.0181) + (0.5389 * 0.0316) \\ &= 0.1592 \end{aligned}$$

$$Z_{net8} = 0.0155 + (0.5962 * 0.0133) + (0.5690 * 0.1130) + (0.5202 * 0.0404) + (0.5535 * 0.1010) + (0.5279 * 0.0296) + (0.5389 * 0.0317) \\ = 0.1971$$

$$Z_{net9} = 0.1072 + (0.5962 * 0.0227) + (0.5690 * 0.0811) + (0.5202 * 0.0190) + (0.5535 * 0.0643) + (0.5279 * 0.0199) + (0.5389 * 0.1270) \\ = 0.2910$$

Gunakan fungsi aktivasi *sigmoid biner* untuk menghitung sinyal keluaran *hidden layer* kedua.

$$Z_7 = \frac{I}{1 + e^{-0.1592}} = 0.5397$$

$$Z_8 = \frac{I}{1 + e^{-0.1971}} = 0.5491$$

$$Z_9 = \frac{I}{1 + e^{-0.2910}} = 0.5722$$

Langkah ini dilakukan sebanyak jumlah lapisan tersembunyi.

Langkah 5: Hitung semua keluaran jaringan di lapisan *output* ( $Y_k, k = 1, 2, \dots, m$ ) menggunakan Persamaan (2-11).

$$Y_{net1} = 0.044 + (0.5397 * 0.107) + (0.5491 * 0.111) + (0.5722 * 0.105)$$

$$Y_{net1} = 0.2227$$

Kemudian gunakan fungsi aktivasi *sigmoid biner* pada Persamaan (2-7) untuk menghitung sinyal *output*:

$$y_k = f(y_{net_k}) = \frac{1}{1 + e^{-y_{net_k}}}$$

$$Y_1 = \frac{1}{1 + e^{-0.2227}} = 0.5554$$

### 3.7.3 Backpropagation

Langkah 6: Hitung faktor  $\delta$  unit keluaran berdasarkan kesalahan di setiap unit keluaran ( $y_k, k = 1, 2, \dots, m$ ) menggunakan Persamaan (2-13).

$$\delta_1 = (t_1 - y_1) f'(y_{net1}) \\ = (1 - 0.5554) f'(0.2227) \\ = 0.2469$$

$\delta$  merupakan unit kesalahan yang akan dipakai dalam perubahan bobot *layer* di bawahnya (langkah 7). Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki  $w_{jk}$ ) dengan laju percepatan (*learning rate*) dengan persamaan (2-14)

$$\Delta w_{jk} = \alpha \cdot W_j \dots \dots \dots (2-14)$$

berikut perhitungannya :

$$\Delta W_{71} = 0.01 * 0.2469 * 0.5397 = 0.0013$$

$$\Delta W_{81} = 0.01 * 0.2469 * 0.5491 = 0.0013$$

$$\Delta W_{91} = 0.01 * 0.2469 * 0.5722 = 0.0014$$

Kemudian hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai  $w_{0k}$ ) menggunakan persamaan di bawah ini :

$$\Delta v_{0k} = \alpha \cdot \delta_k \dots \dots \dots (2-15)$$

berikut perhitungannya :

$$\Delta W_{01} = 0.01 * 0.2469 = 0.0024$$

Langkah 7: Hitung faktor  $\delta$  unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi ( $z_j, j = 1, 2, \dots, p$ ) menggunakan persamaan di bawah ini

$$\delta_{net_j} = \sum_{k=1}^m \delta_k \cdot w_{jk} \dots \dots \dots (2-16)$$

berikut perhitungannya :

$$\delta_{net_7} = 0.2469 * 0.1592 = 0.0397$$

$$\delta_{net_8} = 0.2469 * 0.1971 = 0.0491$$

$$\delta_{net_9} = 0.2469 * 0.2910 = 0.0726$$

Faktor  $\delta$  unit tersembunyi pada *hidden layer* pertama dihitung menggunakan persamaan di bawah ini

$$\delta_j = \delta_{net_j} f'(Z_{net_j}) \dots \dots \dots (2-17)$$

berikut perhitungannya :

$$\begin{aligned} \delta_7 &= 0.0397 * f'(0.1592) \\ &= 0.0214 \end{aligned}$$

$$\begin{aligned} \delta_8 &= 0.0491 * f'(0.1971) \\ &= 0.0269 \end{aligned}$$

$$\begin{aligned}\delta_9 &= 0.0726 * f'(0.2910) \\ &= 0.0415\end{aligned}$$

Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai  $v_{ij}$ ) dari *input layer* ke *hidden layer* pertama menggunakan persamaan di bawah ini

$$\Delta v_{ij} = \alpha \cdot \delta_j \cdot x_i \dots\dots\dots (2-18)$$

berikut perhitungannya :

$$\begin{aligned}\Delta V_{17} &= 0.01 * 0.0214 * 0.5962 = 0.000127 \\ \Delta V_{27} &= 0.01 * 0.0214 * 0.5690 = 0.000121 \\ \Delta V_{37} &= 0.01 * 0.0214 * 0.5202 = 0.000111 \\ \Delta V_{47} &= 0.01 * 0.0214 * 0.5535 = 0.000118 \\ \Delta V_{57} &= 0.01 * 0.0214 * 0.5279 = 0.000112 \\ \Delta V_{67} &= 0.01 * 0.0214 * 0.5389 = 0.000115 \\ \Delta V_{18} &= 0.01 * 0.0269 * 0.5962 = 0.000116 \\ \Delta V_{28} &= 0.01 * 0.0269 * 0.5690 = 0.000153 \\ \Delta V_{38} &= 0.01 * 0.0269 * 0.5202 = 0.000139 \\ \Delta V_{48} &= 0.01 * 0.0269 * 0.5535 = 0.000148 \\ \Delta V_{58} &= 0.01 * 0.0269 * 0.5279 = 0.000142 \\ \Delta V_{68} &= 0.01 * 0.0269 * 0.5389 = 0.000144 \\ \Delta V_{19} &= 0.01 * 0.0415 * 0.5962 = 0.000247 \\ \Delta V_{29} &= 0.01 * 0.0415 * 0.5690 = 0.000236 \\ \Delta V_{39} &= 0.01 * 0.0415 * 0.5202 = 0.000215 \\ \Delta V_{49} &= 0.01 * 0.0415 * 0.5535 = 0.000229 \\ \Delta V_{59} &= 0.01 * 0.0415 * 0.5279 = 0.000219 \\ \Delta V_{69} &= 0.01 * 0.0415 * 0.5389 = 0.000223\end{aligned}$$

Kemudian hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai  $v_{0j}$ ) menggunakan persamaan di bawah ini,

$$\Delta v_{0j} = \alpha \cdot \delta_j \dots\dots\dots (2-19)$$

berikut perhitungannya :

$$\Delta V_{07} = 0.01 * 0.0214 = 0.000214$$

$$\Delta V_{08} = 0.01 * 0.0269 = 0.000269$$

$$\Delta V_{09} = 0.01 * 0.0415 = 0.000415$$

Lalu hitung faktor  $\delta$  unit tersembunyi *hidden layer* pertama berdasarkan kesalahan di setiap unit tersembunyi ( $Z_j, j = 1, 2, \dots, p$ ) menggunakan persamaan (2-19)

$$\begin{aligned}\delta_{net_1} &= (0.000214 * 0.0122) + (0.000269 * 0.0133) + (0.000415 * 0.0227) \\ &= 0.000123\end{aligned}$$

$$\begin{aligned}\delta_{net_2} &= (0.000214 * 0.1423) + (0.000269 * 0.1130) + (0.000415 * 0.0811) \\ &= 0.000397\end{aligned}$$

$$\begin{aligned}\delta_{net_3} &= (0.000214 * 0.0335) + (0.000269 * 0.0404) + (0.000415 * 0.0190) \\ &= 0.000161\end{aligned}$$

$$\begin{aligned}\delta_{net_4} &= (0.000214 * 0.0150) + (0.000269 * 0.1010) + (0.000415 * 0.0643) \\ &= 0.000326\end{aligned}$$

$$\begin{aligned}\delta_{net_5} &= (0.000214 * 0.0181) + (0.000269 * 0.0296) + (0.000415 * 0.0099) \\ &= 0.000159\end{aligned}$$

$$\begin{aligned}\delta_{net_6} &= (0.000214 * 0.0316) + (0.000269 * 0.0317) + (0.000415 * 0.1270) \\ &= 0.000679\end{aligned}$$

$$\delta_j = \delta_{net_j} f'(Z_{net_j}) \dots\dots\dots(2-17)$$

$$\begin{aligned}\delta_1 &= 0.000123 * f'(0.390) \\ &= 0.000073\end{aligned}$$

$$\begin{aligned}\delta_2 &= 0.000397 * f'(0.278) \\ &= 0.000225\end{aligned}$$

$$\begin{aligned}\delta_3 &= 0.000161 * f'(0.081) \\ &= 0.000083\end{aligned}$$

$$\begin{aligned}\delta_4 &= 0.000326 * f'(0.215) \\ &= 0.000180\end{aligned}$$

$$\begin{aligned}\delta_5 &= 0.000159 * f'(0.112) \\ &= 0.000083\end{aligned}$$

$$\begin{aligned}\delta_6 &= 0.000679 * f'(0.156) \\ &= 0.000365\end{aligned}$$

### Fase III: Perubahan Bobot

Langkah 8: Tiap-tiap unit *output* ( $k = 1, 2, \dots, m$ ) memperbaiki bobotnya ( $j = 0, 1, 2, \dots, p$ ) menggunakan persamaan di bawah ini

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \dots \dots \dots (2-20)$$

berikut perhitungannya,

$$W_{01}(\text{baru}) = 0.044 + 0.1531 = 0.1971$$

$$W_{71}(\text{baru}) = 0.107 + 0.0839 = 0.1909$$

$$W_{81}(\text{baru}) = 0.111 + 0.0856 = 0.1966$$

$$W_{91}(\text{baru}) = 0.105 + 0.0891 = 0.1941$$

Tiap-tiap unit tersembunyi ( $Z_j, j = 1, 2, 3, \dots, p$ ) memperbaiki bobotnya ( $j = 0, 1, 2, 3, \dots, n$ ) menggunakan persamaan di bawah ini

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \dots \dots \dots (2-21)$$

$$V_{01}(\text{baru}) = 0.011 + 0.000001 = 0.011001$$

$$V_{02}(\text{baru}) = 0.263 + 0.000002 = 0.263002$$

$$V_{03}(\text{baru}) = 0.057 + 0.000000 = 0.057000$$

$$V_{04}(\text{baru}) = 0.109 + 0.000001 = 0.109001$$

$$V_{05}(\text{baru}) = 0.035 + 0.000000 = 0.035000$$

$$V_{06}(\text{baru}) = 0.111 + 0.000001 = 0.111001$$

Perhitungan dilakukan sampai mendapatkan nilai  $V_{36}$  baru.

Langkah 9: Kondisi pelatihan berhenti jika  $error \leq 0.0001$  atau jumlah *epoch* mencapai 10.000

*Software Jupyterlab* digunakan untuk pembuatan jaringan *backpropagation* yang sesuai dengan inisialisasi awal dan parameter dari pelatihan manual yang telah dilakukan. Sehingga diperoleh bias dan bobot akhir yang dapat dilihat pada Tabel 3.7 dan Tabel 3.8

Tabel 3.7 Bias dan bobot akhir dari *input layer* ke *hidden layer* pertama

Dari- Ke-	Bias (B1)	X1	X2	X3
Z1	0.571	0.989	1.127	1.760
Z2	0.904	1.240	0.686	-1.545
Z3	-1.270	-1.169	-1.332	-2.551
Z4	-1.114	-1.174	-1.269	-2.722
Z5	-1.336	-1.382	-1.443	-2.924
Z6	2.227	2.363	2.294	4.612

Tabel 3.8 Bias dan bobot akhir dari *hidden layer* pertama ke *hidden layer* kedua

Dari- Ke-	Bias (B2)	Z1	Z2	Z3	Z4	Z5	Z6
Z7	-1.012	-1.563	1.115	1.988	1.830	2.225	-4.272
Z8	-0.084	-1.510	1.275	1.964	1.965	2.133	-4.642
Z9	1.179	-1.570	-1.110	-2.064	-2.111	-2.370	4.771

Bias dan bobot awal dari *hidden layer* kedua ke *output layer* dinyatakan dengan  $W_{01}$ ,  $W_{71}$ ,  $W_{81}$  dan  $W_{91}$ , yaitu:

Tabel 3.9 Bias dan bobot awal dari *hidden layer* kedua ke *output layer*

Dari- Ke-	Bias (B3)	Z7	Z8	Z9
Y1	-3.355	9.383	10.136	-12.4691

Dari hasil yang sudah didapatkan sebelumnya, nilai akan dibulatkan ke integer terdekat (*threshold*) dan menghasilkan tabel 3.10.



Tabel 3.10 *Output* data latih

Data ke-	<i>Output</i>	Hasil <i>threshold</i>	Target
1	0.9999990	1	1
2	0.0000190	0	0
3	0.9999000	1	1
4	0.0000191	0	0

Untuk mendapatkan tingkat akurasi dari hasil pelatihan, jumlah hasil *threshold* yang sesuai target dibagi dengan jumlah data. Tingkat akurasi =  $(4/4) * 100 \% = 100 \%$ .

### 3.8 Teknik Pengujian Sistem

Tahapan pengujian sistem digunakan untuk mengetahui dan mencoba sistem apakah berjalan baik serta untuk mengetahui kekurangan sistem pada saat terjadi kesalahan. Berikut perhitungan akurasi, presisi dan *recall* dalam pengujian sistem:

$$Akurasi = \frac{\text{jumlah data sesuai target}}{\text{Total keseluruhan data}} \quad (3-15)$$

$$Presisi = \frac{\text{jumlah data yang sesuai target di satu kelas}}{\text{jumlah seluruh data yang sesuai target}} \quad (3-16)$$

$$Recall = \frac{\text{jumlah data yang sesuai target di satu kelas}}{\text{jumlah data di satu kelas}} \quad (3-17)$$

Pembagian data pada dataset yang merupakan hasil ekstraksi ciri dari citra tulisan tangan aksara Bima dibagi menjadi beberapa kombinasi data *train* dan *testing* dengan metode *K-fold Cross Validation*. Metode ini digunakan untuk mevariasikan dataset yang didapatkan, sehingga nantinya dapat dilihat hasil model pengenalan pola dalam menghadapi variasi jumlah data yang diterima walaupun dengan dataset yang tidak terlalu banyak. Keluaran dari skenario pengujian ini ialah berupa model pengenalan pola *Backpropagation Neural Network* dengan performa terbaik. Dimana untuk mendapatkan model BPNN dengan performa terbaik tersebut, dihitung nilai akurasi, presisi dan *recall*, dimana nilai masing-masing dari variabel tersebut didapatkan dari suatu *confusion matrix*.

Tabel 3.11 Contoh nilai *confusion matrix* untuk menghitung akurasi, presisi dan *recall*.

		Predict Class		
		A	B	C
Actual Class	A	45	10	30
	B	20	60	20
	C	35	30	50

Dengan menggunakan *confusion matrix* di atas maka dapat dihitung besarnya akurasi, presisi dan *recall*. Besarnya akurasi dapat dihitung menggunakan persamaan .

$$Akurasi = \frac{AA+BB+CC}{total\ data}$$

$$\begin{aligned}
 Akurasi &= \frac{AA+BB+CC}{total\ data} \\
 &= \frac{45+60+50}{300} \\
 &= \frac{170}{300} \\
 &= 0,5166 \text{ atau } 51,66\%
 \end{aligned}$$

Besarnya presisi pada kelas A dapat dihitung menggunakan persamaan.

$$Presisi = \frac{AA}{(AA+BA+CA)}$$

$$\begin{aligned}
 Presisi\ A &= \frac{AA}{(AA+BA+CA)} \\
 &= \frac{45}{(45+20+35)} \\
 &= \frac{45}{100} \\
 &= 0,45 \text{ atau } 45\%
 \end{aligned}$$

Besarnya presisi pada kelas B dapat dihitung menggunakan persamaan.

$$Presisi = \frac{BB}{(AB+BB+CB)}$$

$$\begin{aligned}
 Presisi\ B &= \frac{BB}{(AB+BB+CB)} \\
 &= \frac{60}{(10+60+30)} \\
 &= \frac{60}{100} \\
 &= 0,60 \text{ atau } 60\%
 \end{aligned}$$

Besarnya presisi pada kelas C dapat dihitung menggunakan persamaan.

$$Presisi = \frac{CC}{(AC+BC+CC)}$$

Maka besar presisi pada kelas adalah.

$$\begin{aligned} Presisi C &= \frac{CC}{(AC+BC+CC)} \\ &= \frac{50}{(30+20+70)} \\ &= \frac{50}{100} \\ &= 0,5 \text{ atau } 50\% \end{aligned}$$

Besar nilai *recall* pada masing-masing kelas dapat dihitung menggunakan persamaan. Maka besar nilai *recall* pada kelas A adalah.

$$\begin{aligned} Recall A &= \frac{AA}{(AA+AB+AC)} \\ &= \frac{45}{(45+10+30)} \\ &= \frac{45}{85} \\ &= 0,529 \text{ atau } 52,9\% \end{aligned}$$

$$\begin{aligned} Recall B &= \frac{BB}{(BA+BB+BC)} \\ &= \frac{60}{(20+60+20)} \\ &= \frac{60}{100} \\ &= 0,6 \text{ atau } 60\% \end{aligned}$$

$$\begin{aligned} Recall C &= \frac{CC}{(CA+CB+CC)} \\ &= \frac{50}{(35+30+50)} \\ &= \frac{50}{115} \\ &= 0,434 \text{ atau } 43,4\% \end{aligned}$$

### 3.9 Skenario Pengujian Sistem

Pada sistem pengenalan pola aksara digunakan metode *backpropagation* untuk tahap pengujiannya, untuk mengetahui pengaruh size citra jika ukuran *pixel* dibuat berbeda yaitu dengan ukuran 32x32, 64x64, dan 128x128. Selain itu, Penentuan parameter-parameter yang diperlukan pada *Backpropagation Neural Network* antara lain:

1. *Neuron output*: 22 neuron
2. Jumlah *hidden layer* : 1, 2, dan 3 layers.
3. Rentang *Learning rate* sebagai parameter uji : 0.1 ~ 0.9
4. Batas *epoch* sebagai parameter uji : 1000
5. Batas *error* sebagai parameter uji : 0.001
6. Fungsi aktivasi : sigmoid biner
7. Pembagian data *training* dan *testing* yaitu 70-30%, 75-25% dan 80-20%
8. Ekstraksi ICZ-ZCZ dibagi menjadi 5x7 zona atau 35 ciri.

### 3.10 Jadwal Kegiatan

Jadwal kegiatan yang menampilkan rentang waktu pengerjaan sistem dapat dilihat pada Tabel 3.12.

Tabel 3.12 Jadwal kegiatan pengerjaan sistem

No	Kegiatan	Waktu (Bulan)						Keterangan
		I	II	III	IV	V	VI	
1	Analisis							Analisa kebutuhan
2	Perancangan							Perancangan sistem
3	<i>Coding</i>							Pengkodean sistem
4	<i>Testing</i>							Pengujian sistem
5	Dokumentasi							Dokumentasi sistem

## DAFTAR PUSTAKA

- [1] S. Abubakar, “Aksara Bima Usaha Menemukanali dan Mengembangkannya,” UIN Mataram DPK Sekol. Tinggi Ilmu Tarb. Sunan Giri Bima, pp. 1–30, 2018.
- [2] F. H. Tondo, “Kepunahan Bahasa-Bahasa Daerah: Faktor Penyebab Dan Implikasi Etnolinguistis,” *J. Masy. Budaya*, vol. 11, no. 2, pp. 277–296, 2009.
- [3] Peraturan Menteri Pendidikan dan Kebudayaan Republik Indonesia Nomor 79/Permendikbud/SR.79/2014 tentang Muatan Lokal Kurikulum 2013.
- [4] M. Rizky, I. Nurtanio, and I. S. Areni, “Mbojo Character Recognition Using Shearlet Transform and Support Vector Machine,” *Proceeding - 2018 Int. Semin. Intell. Technol. Its Appl. ISITIA 2018*, pp. 339–344, 2018.
- [5] A. Tanto Wibowo, “Pengenalan Pola Tulisan Tangan Aksara Jawa Dengan Algoritma Backpropagation,” Universitas Sanata Dharma. Yogyakarta, 2018.
- [6] I. Osmond Giovanni “Metode Backpropagation Untuk Alih Aksara Jawa Cetak Menggunakan Ciri ICZ ZCZ” Universitas Sanata Dharma. Yogyakarta, 2019.
- [7] D. Arianto “Pengenalan Pola Karakter Plat Nomor Kendaraan Menggunakan Algoritma Momentum Backpropagation Neural Network”, *Jurnal Informatika.*, vol.10, pp 1–11,2016.
- [8] A. Andana, R.Widyati, and M.Irzal, “Pengenalan Citra Tulisan Tangan Menggunakan Metode Backpropagation,” *J. Mat. Terap.*, vol. 2, no. 1, pp. 36–44, 2018.
- [9] R. Muhammad Syam, “Pengenalan Aksara Jawa Tulisan Tangan Dengan Menggunakan Ekstraksi Fitur Zoning Dan Klasifikasi K-Nearest Neighbor,” Departemen Ilmu Komputer, Institut Pertanian Bogor. 2013.
- [10] Sutarno and P. Ardhyia Garini, “Pengenalan Angka Arab Timur Tulisan

- Tangan Menggunakan Zone Centroid Zone dan Backpropagation ,”*Computer Science and ICT*,. vol. 3, no.1. pp. 162–164, 2017.
- [11] R. Putri Ayu Pramesti, “Identifikasi Karakter Plat Nomor Kendaraan Menggunakan Ekstraksi Fitur ICZ Dan ZCZ Dengan Metode Klasifikasi K-NN,” Departemen Ilmu Komputer, Institut Pertanian Bogor. 2013. Departemen Ilmu Komputer, Institut Pertanian Bogor. 2013.
  - [12] A. Aranta, F. Bimantoro, and I. P. T. Putrawan, “Penerapan Algoritma Rule Base dengan Pendekatan Hexadesimal pada Transliterasi Aksara Bima Menjadi Huruf Latin,” *J. Teknol. Informasi, Komputer, dan Apl. (JTika)* , vol. 2, no. 1, pp. 130–141, 2020.
  - [13] O. Aro Taye, Y. Musa Abdullah, and S. Abdul Kasir Ikeola “Recognition of Alphabet Characters and Arabic Numerals Using Backpropagation Neural”. Universitas of Ilorin, vol. 11, no. 2. 2018.
  - [14] R. Gonzales “*Digital Image Processing Using Matlab*”. Prentis Hall, Inc. Upper Saddle River, NJ, USA, 2002.
  - [15] E. Ardianto, S. Munawaroh, and Prihandono “Pengolahan Citra Digital Untuk Identifikasi Ciri Sidik Jari Berbasis Minutiae ”.*Jurnal Teknologi Informatika Dinamik*, vol. 16, no. 1, 2011.
  - [16] A. Agung Wisnu, N. Muhammad Zidny, and U. Elisa, “Identifikasi Citra Tanda Tangan Berdasarkan Grid Entropy dan PCA Menggunakan Multi Layer Perceptron ”., Institut Teknologi Telkom Purwokerto, 2019.
  - [17] T. Stamford Raffles, “ *The History of Java*” John Murray Albemarle Street, London. vol. 1 no.1, 1817.
  - [18] Supardi, Julian and A. Utami, “*Development Of Artificial Neural Network Architecture For Face Recognition In Real Time*”.International Journal Machine Learning and Computing. Singapore. vol. 04. 2014
  - [19] J. J. Siang, *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab*, 1st ed. Yogyakarta: ANDI, 2005.

- [20] L. Fausett, *Fundamentals of neural networks: architectures, algorithms, and applications*. Melbourne: Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1994.
- [21] A. Jumarwanto, R. Hartanto, and D. Prastiyanto, “Aplikasi Jaringan Saraf Tiruan Backpropagation Untuk Memprediksi Penyakit THT Di Rumah Sakit Mardi Rahayu Kudus,” *J. Tek. Elektro*, vol. 1, no. 1, pp. 11–21, 2009.
- [22] Rajashekararadhya SV, and Ranjan PV. Efficient zone based feature extraction algorithm for handwritten numeral recognition of four popular south indian scripts. *Journal of Theoretical and Applied Information Technology*. 4(12):1171-1181. 2008.
- [23] J. Davis and M. Goadrich, “The relationship between precision-recall and ROC curves,” *ACM Int. Conf. Proceeding Ser.*, pp. 233–240, 2006.
- [24] R. Arthana, “Mengenal Accuracy, Precision, Recall dan Specificity serta yang diprioritaskan dalam Machine Learning,” Medium.Com, 2018. [Online]. Available: <https://medium.com/@rey1024/mengenal-accuracy-precision-recall-dan-specificity-septa-yangdiprioritaskan-b79ff4d77de8>. [Accessed: 15-Apr-2020].
- [25] Budhi, G. satia, Adipranata, and Rudy. *Handwritten Javanese Character Recognition Using several Artificial Neural Network Methods*, J.ITC Re. Appl, Vol.8 No.3: 195-212. 2015.