

**PENGENALAN POLA TULISAN TANGAN AKSARA BIMA DENGAN
EKSTRAKSI CIRI GLCM DAN ZONING & KLASIFIKASI
PROBABILISTIC NEURAL NETWORK**

Tugas Akhir

Untuk memenuhi sebagian persyaratan
mencapai derajat Sarjana S-1 Program Studi Teknik Informatika



Oleh:

Muhammad Naufal

F1D016060

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

LAPORAN TUGAS AKHIR

PENGENALAN POLA TULISAN TANGAN AKSARA BIMA DENGAN EKSTRAKSI CIRI GLCM DAN ZONING & KLASIFIKASI PROBABILISTIC NEURAL NETWORK

Oleh:

**Muhammad Naufal
F1D 016 060**

Telah diperiksa dan disetujui oleh Tim Pembimbing:

1. Pembimbing Utama



**Fitri Bimantoro, S.T., M.Kom.
NIP. 19860622 201504 1 002**

Tanggal: 16 September 2020

2. Pembimbing Pendamping



**Arik Aranta, S.Kom., M.Kom.
NIP. 19940220 201903 1 004**

Tanggal: 21 September 2020

Mengetahui,
Ketua Program Studi Teknik Informatika
Fakultas Teknik
Universitas Mataram

**Prof. Dr.Eng. I Gede Pasek Suta Wijaya, ST., MT.
NIP: 19731130 200003 1 001**

DAFTAR ISI

| | |
|---|----|
| DAFTAR ISI..... | i |
| DAFTAR GAMBAR | iv |
| DAFTAR TABEL..... | v |
| ABSTRAK | vi |
| BAB I PENDAHULUAN..... | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Rumusan Masalah | 3 |
| 1.3 Batasan Masalah..... | 3 |
| 1.4 Tujuan..... | 4 |
| 1.5 Manfaat..... | 4 |
| 1.6 Sistematika Penulisan..... | 4 |
| BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI | 6 |
| 2.1 Tinjauan Pustaka | 6 |
| 2.2 Dasar Teori..... | 8 |
| 2.2.1 Aksara Bima..... | 8 |
| 2.2.2 Ekstraksi Ciri | 9 |
| 2.2.3 <i>Gray Level Co-occurrence Matrix (GLCM)</i> | 9 |
| 2.2.4 <i>Zoning</i> | 10 |
| 2.2.5 <i>Artificial Neural Network</i> | 11 |
| 2.2.6 <i>Probabilistic Neural Network (PNN)</i> | 11 |
| 2.2.7 <i>Confusion Matrix</i> | 12 |
| 2.2.8 Akurasi, Presisi, dan <i>Recall</i> | 13 |
| BAB III METODOLOGI PENELITIAN | 14 |
| 3.1 Alat dan Bahan Penelitian | 14 |
| 3.2 Diagram Alir Penelitian | 15 |
| 3.3 Pengumpulan Dataset..... | 16 |
| 3.4 Pembangunan Model..... | 16 |
| 3.4.1 <i>Data Pre-processing</i> | 17 |
| 3.4.2 <i>Zoning</i> | 18 |
| 3.4.3 Ekstraksi Ciri GLCM..... | 18 |
| 3.4.4 <i>K-Fold Cross Validation</i> | 20 |

| | | |
|-----------------------------------|---|----|
| 3.4.5 | Model PNN | 21 |
| 3.5 | Pengujian Skenario..... | 27 |
| BAB IV HASIL DAN PEMBAHASAN | | 31 |
| 4.1 | Akuisisi Data | 31 |
| 4.2 | Data Pre-processing..... | 35 |
| 4.3 | Ekstraksi Ciri..... | 35 |
| 4.4 | Mekanisme Pengujian | 36 |
| 4.5 | Hasil Pengujian | 37 |
| 4.5.1 | Pengujian <i>Smoothing Parameter</i> | 37 |
| 4.5.2 | Pengujian Kombinasi Sudut dan Jarak GLCM..... | 38 |
| 4.5.3 | Pengujian <i>Zoning</i> | 39 |
| 4.5.4 | Pengujian Pembagian Data | 39 |
| BAB V KESIMPULAN DAN SARAN | | 41 |
| 5.1 | Kesimpulan..... | 41 |
| 5.2 | Saran..... | 41 |
| DAFTAR PUSTAKA | | 42 |

DAFTAR GAMBAR

| | |
|--|----|
| Gambar 2.1 Aksara Bima baru atau bentuk garis-garis | 9 |
| Gambar 2.2 <i>Co-occurrence matrix</i> | 10 |
| Gambar 2.3 <i>Zoning</i> | 11 |
| Gambar 2.4 Arsitektur ANN..... | 11 |
| Gambar 2.5 Arsitektur PNN [33]..... | 12 |
| Gambar 2.6 <i>Confusion matrix</i> | 13 |
| Gambar 3.1 Diagram alir penelitian..... | 15 |
| Gambar 3.2 Form dataset Aksara Bima..... | 16 |
| Gambar 3.3 Diagram alir perancangan model | 17 |
| Gambar 3.4 <i>Zoning</i> pada citra..... | 18 |
| Gambar 3.5 Pencarian hubungan ketertanggaan antar nilai piksel dalam citra | 19 |
| Gambar 3.6 Matriks kookurensi GLCM..... | 19 |
| Gambar 3.7 Matriks kookurensi GLCM setelah normalisasi | 19 |
| Gambar 3.8 <i>K-fold cross validation</i> | 21 |
| Gambar 3.9 Contoh model PNN sederhana | 23 |
| Gambar 3.10 Pengujian skenario | 27 |
| Gambar 4.1 Hasil <i>scan</i> form pengambilan dataset | 31 |
| Gambar 4.2 <i>Cropping</i> awal dataset..... | 31 |
| Gambar 4.3 Pembagian panjang dan lebar citra per karakter | 32 |
| Gambar 4.4 Hasil <i>cropping</i> gambar..... | 32 |
| Gambar 4.5 Hasil <i>cropping</i> sebelum dicari area ROI..... | 33 |
| Gambar 4.6 Hasil <i>invert</i> citra asli | 33 |
| Gambar 4.7 Ilustrasi pencarian batas ROI | 34 |
| Gambar 4.8 Hasil <i>crop</i> berdasarkan ROI..... | 34 |
| Gambar 4.9 Citra hasil re-size | 35 |

DAFTAR TABEL

| | |
|---|----|
| Tabel 3.1 Data <i>dummy</i> PNN | 22 |
| Tabel 3.2 <i>Confusion matrix 3x3</i> | 28 |
| Tabel 4.1 Hasil pengujian <i>smoothing parameter</i> | 37 |
| Tabel 4.2 Hasil pengujian kombinasi sudut dan jarak GLCM..... | 38 |
| Tabel 4.3 Hasil pengujian <i>zoning</i> | 39 |
| Tabel 4.4 Hasil pengujian pembagian data | 40 |

ABSTRAK

Aksara Bima merupakan salah satu warisan budaya Nusantara yang perlu dilestarikan. Berdasarkan hasil kuesioner yang dilakukan penulis secara *online* dengan jumlah responden sebanyak 81 orang yang berasal dari Bima, terdapat 66.7% orang yang belum familiar dengan bentuk karakter Aksara Bima dan 45.7% orang bahkan tidak mengetahui keberadaan Aksara Bima. Terdapat berbagai upaya dalam pelestarian Aksara di Nusantara, antara lain pembuatan media pembelajaran di antaranya permainan, transliterasi, dan pengenalan pola Aksara. Penelitian ini bertujuan untuk membangun suatu model *machine learning* yang mampu mengenali pola tulisan tangan Aksara Bima dengan menggunakan ekstraksi ciri *Gray Level Co-occurrence Matrix* (GLCM) yang dipadukan dengan *Zoning* dan klasifikasi *Probabilistic Neural Network* (PNN). Model terbaik yang dihasilkan pada penelitian ini mampu melakukan klasifikasi karakter Aksara Bima dengan akurasi mencapai 81.35%, dimana diterapkan penggunaan *zoning* dengan ukuran 4x4 dengan perbandingan pembagian data *train* dan *test* sebesar 3:1.

Keyword: Aksara Bima, Pengenalan Pola, *Gray Level Co-occurrence Matrix*, *Zoning*, *Probabilistic Neural Network*

ABSTRACT

The Bima's Script, natively known as Aksara Bima is one of Indonesia's local heritage that needs to be preserved. Based on an online questionnaire of 81 respondent from Bima, there were 66.7% of people who were not familiar with the Bima's Script and 45.7% of people did not even know the existence of the Bima's Script. There are various ways of preserving the local script in Indonesia, such as the creation of learning media like games, transliteration, and Script's pattern recognition. This research aims to build a machine learning model that is able to recognize the Bima Script's handwriting pattern by using the Gray Level Co-occurrence Matrix (GLCM) feature extraction combined with Zoning and Probabilistic Neural Network (PNN) classification. The best model obtained accuracy up to 81.35% by applying 4x4 zoning size and the 75% of the dataset as training data.

Keywords: Bima's Script, Pattern Recognition, Gray Level Co-occurrence Matrix, Zoning, Probabilistic Neural Network

BAB I

PENDAHULUAN

1.1 Latar Belakang

Aksara Daerah merupakan salah satu warisan Budaya yang banyak terdapat di Indonesia. Setiap daerah memiliki berbagai jenis bahasa yang berbeda-beda dan aksara yang berbeda pula. Provinsi Nusa Tenggara Barat pada umumnya terbagi menjadi tiga jenis suku, yaitu suku Sasak, Samawa, dan Mbojo. Suku Mbojo memiliki bahasa daerah sendiri, yaitu Bahasa Mbojo, atau Bahasa 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 [1]. Penggunaan bahasa daerah sering digunakan, namun untuk penggunaan aksara daerah masih minim di Indonesia, termasuk Aksara Bima. Bali merupakan salah satu daerah yang mengeluarkan kebijakan untuk menggunakan Aksara Bali pada berbagai fasilitas umum, seperti nama jalan, prasasti peresmian gedung, dan sarana pariwisata melalui Peraturan Gubernur Bali [2]. Pembelajaran Aksara Bima di sekolah-sekolah Negeri di Bima pernah diajarkan sejak tahun 1994 namun pada akhirnya sempat terhenti dan semenjak 3 tahun terakhir pembelajaran mulai digalakkan [1]. Terdapat berbagai upaya dalam pelestarian Aksara di Nusantara, antara lain transliterasi, pengenalan pola citra aksara, dan pembuatan media pembelajaran melalui permainan [2]–[6].

Berdasarkan hasil kuesioner yang dilakukan oleh penulis secara *online* dengan jumlah responden sebanyak 87 orang dengan rentang usia 17 sampai 38 tahun, dimana terdapat 81 orang yang berasal dari Bima, sebanyak 45.7% masih belum mengetahui tentang keberadaan Aksara Bima, dan sebanyak 48.1% belum pernah mempelajari Aksara Bima. Adapun dari 81 responden tersebut, yang pernah mempelajari Aksara Bima di bangku sekolah tingkat SD-SMA sebanyak 49.4% dan 2.5% pernah belajar di Perguruan Tinggi. Terdapat 66.7% atau 54 orang Bima masih tidak familiar dengan penggunaan Aksara Bima. Dari data tersebut dapat disimpulkan bahwa masih banyak penduduk Bima yang belum familiar dengan penggunaan Aksara Bima, bahkan di pendidikan formal pun tidak seluruhnya mendapatkan pelajaran Muatan Lokal yang mempelajari Aksara Bima. Hal ini menjadi masalah yang dapat mengakibatkan hilangnya warisan Budaya Bima yaitu Aksara Bima. Dengan berkembangnya teknologi digital, media pembelajaran pun mulai beralih ke media digital. Pengenalan pola tulisan tangan Aksara Bima merupakan salah satu solusi yang dapat dihadirkan untuk menciptakan suatu model pembelajaran mesin yang dapat mengenali pola tulisan tangan

Aksara Bima, dimana nantinya model yang dibuat dapat digunakan dalam pengembangan aplikasi edukasi.

Pengenalan pola bertujuan untuk mengenali atau mengklasifikasikan suatu citra tertentu dalam suatu kelas, dimana pengenalan pola terbagi menjadi tiga tahapan, yaitu pemrosesan awal, ekstraksi fitur, dan klasifikasi atau pengenalan [7]. Pada saat penelitian ini dilakukan, pengenalan pola tulisan tangan Aksara Bima belum pernah ditemukan. Pengenalan pola tulisan tangan Aksara di Nusantara pernah dilakukan pada beberapa Aksara, antara lain Aksara Bali, Aksara Sunda, Aksara Jawa, dan Aksara Lampung [6]–[9].

Neural Network adalah suatu model klasifikasi dalam *machine learning* yang dapat digunakan untuk melakukan prediksi. Paradigma *Neural Network* terinspirasi dari cara otak manusia memproses suatu informasi. Arsitektur suatu *Neural Network* pada umumnya terdiri dari *input layer*, *hidden layer*, dan *output layer* yang masing-masing terdapat *neuron* di dalamnya. *Probability Neural Network* (PNN) adalah salah satu model *classifier* yang menggunakan sebuah *Probability Density Function* dalam suatu *pattern layer*, dimana pada model ini diusung sebuah konsep “*winner take all*” yang berdasarkan pada estimasi probabilitas multivariat [10]. PNN dapat dimanfaatkan dalam beberapa kasus klasifikasi pada beberapa penelitian sebelumnya, antara lain pengenalan pola tulisan tangan Bahasa Arab[11], klasifikasi kapal dengan citra ISAR[12], dan klasifikasi jenis batu kapur[13].

Gray Level Co-occurrence Matrix (GLCM) adalah suatu metode ekstraksi ciri tekstur yang dimiliki oleh suatu citra. Perhitungan nilai ciri pada GLCM berdasarkan suatu *co-occurrence matrix* yang menunjukkan probabilitas ketetanggaan antar dua level keabuan pada piksel yang bertetanggaan [14]. Ekstraksi ciri GLCM dapat dimanfaatkan dalam beberapa kasus pada beberapa penelitian sebelumnya, antara lain pendeteksian kolestrol melalui iris mata [15] dan pengenalan citra tanda tangan [14].

Zoning merupakan suatu pembagian citra menjadi beberapa zona atau daerah, dimana *zoning* digunakan untuk mendapatkan nilai ciri per bagian yang merupakan bagian dari ciri citra secara utuh [16]. *Zoning* dapat dimanfaatkan dalam beberapa kasus pada penelitian sebelumnya, antara lain pengenalan Aksara Lampung[7] dan pengenalan Aksara Sasak[16].

Untuk itu, dilakukan sebuah penelitian untuk merancang sebuah model *machine learning* untuk pengenalan pola Aksara Bima menggunakan metode klasifikasi *Probabilistic Neural Network* dan ekstraksi ciri GLCM dan *Zoning*. Adapun pertimbangan penulis dalam memilih PNN sebagai *classifier* adalah PNN tidak memerlukan proses *train* yang lama dan dapat bekerja dengan baik dalam jumlah data yang sedikit yang ditunjukkan pada penelitian untuk pendeteksian teks pada citra dengan 300 buah citra [17] dan pengenalan tulisan Bahasa

Arab dengan jumlah citra sebanyak 100 [11]. Pemilihan GLCM yang dipadukan dengan *Zoning* sebagai ekstraksi ciri yang digunakan merujuk pada penelitian tentang pengenalan tanda tangan dengan PNN dan ekstraksi fitur GLCM yang memperoleh hasil akurasi sebesar 71% [14]. Terdapat beberapa kesalahan dalam klasifikasi pada penelitian tersebut karena kemiripan nilai ciri yang diperoleh, dimana nilai ciri didapatkan secara global. Penelitian tentang pengenalan pola Aksara Sasak dengan menggunakan *Moment Invariant* dan *Zoning* menunjukkan bahwa *Zoning* dapat meningkatkan nilai akurasi dalam pengenalan pola Aksara Sasak hingga 15,61% [16]. Oleh karena itu, sudah dilakukan penelitian oleh penulis dengan ekstraksi ciri GLCM dengan harapan akurasi yang diperoleh dapat lebih baik jika dipadukan dengan *Zoning* dan klasifikasi PNN dapat melakukan klasifikasi dengan baik pada jumlah data yang sedikit dalam pengenalan pola tulisan tangan Aksara Bima.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, terdapat beberapa rumusan masalah yang akan diselesaikan dalam penelitian ini. Adapun rumusan masalah pada penelitian ini adalah sebagai berikut:

1. Bagaimana merancang suatu model pengenalan pola tulisan tangan Aksara Bima dengan menggunakan metode ekstraksi ciri GLCM dan *Zoning* dan klasifikasi *Probabilistic Neural Network*
2. Bagaimana tingkat akurasi dari perpaduan metode ekstraksi ciri GLCM dan *Zoning* dan klasifikasi *Probabilistic Neural Network* dalam pengenalan pola Aksara Bima.

1.3 Batasan Masalah

Penelitian ini memiliki beberapa batasan masalah atau ruang lingkup tersendiri demi tercapainya penelitian yang memfokuskan pada beberapa masalah. Adapun batasan masalah pada penelitian ini adalah sebagai berikut:

1. Karakter atau huruf dalam Aksara Bima yang digunakan dalam klasifikasi berjumlah 22 karakter dasar dan beberapa karakter sengau.
2. Ekstraksi fitur yang digunakan pada penelitian adalah GLCM dan *Zoning*.
3. Metode klasifikasi yang digunakan adalah *Probabilistic Neural Network*.
4. Data citra aksara diperoleh dari tulisan tangan pada kertas dan di-*scan* yang didapat dari sepuluh orang sukarelawan yang masing-masing menuliskan tiap huruf sebanyak dua belas kali.

1.4 Tujuan

Tujuan yang diharapkan dari penelitian ini adalah sebagai berikut:

1. Menciptakan suatu model *machine learning* untuk pengenalan pola Aksara Bima berbasis *Probabilistic Neural Network* dan GLCM dan *Zoning*.
2. Mengetahui performa dari model *machine learning* yang dibuat dalam pengenalan pola Aksara Bima.

1.5 Manfaat

Manfaat dari penelitian ini dapat diperoleh oleh dua subjek antara lain sebagai berikut.

1. Bagi penulis
 - a. Dapat menerapkan metode *Probabilistic Neural Network* dan GLCM dan *Zoning* dalam pengenalan pola tulisan tangan Aksara Bima.
 - b. Dapat menambah pengetahuan di bidang *machine learning*.
2. Bagi pembaca
 - a. Dapat mengetahui bagaimana cara mendapatkan model *Probabilistic Neural Network* dalam pengenalan pola Aksara Bima.
 - b. Dapat dijadikan sebagai rujukan dalam pengembangan model *Probabilistic Neural Network* agar memperoleh performa yang lebih baik.
 - c. Model yang dihasilkan dari penelitian ini dapat digunakan untuk mengembangkan sistem pengenalan pola Aksara Bima berbasis *mobile* atau platform lainnya.

1.6 Sistematika Penulisan

Bab I Pendahuluan

Bab ini menjelaskan dasar-dasar dalam penulisan laporan tugas akhir, yang terdiri atas latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penulisan laporan.

Bab II Tinjauan Pustaka dan Dasar Teori

Bab ini membahas tentang penelitian-penelitian terdahulu yang berkaitan dengan implementasi *Probabilistic Neural Network* dan GLCM dan *Zoning* serta dasar teori sebagai referensi penulis dalam melakukan penelitian.

Bab III Metodologi Penelitian

Bab ini membahas tentang metodologi yang digunakan dalam membangun model *machine learning* dalam pengenalan pola Aksara Bima.

Bab IV Hasil dan Pembahasan

Bab ini membahas tentang hasil yang diperoleh dari berbagai skenario pengujian dari model yang dibuat disertai pembahasan tentang hasil tersebut.

Bab V Implementasi dan Pengujian Metode

Bab ini membahas tentang implementasi yang dilakukan terhadap perancangan model atau sistem *machine learning* dengan metode *Probabilistic Neural Network*.

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Upaya dalam pelestarian warisan budaya Aksara dapat dilakukan dengan berbagai cara, beberapa di antaranya antara lain transliterasi, pembuatan media pembelajaran melalui permainan, dan pengenalan pola citra Aksara [2]–[6]. Penelitian mengenai pengenalan pola tulisan tangan Aksara Bima belum ditemukan saat penelitian ini dilakukan. Adapun penelitian terkait tentang pengenalan pola Aksara Bima dengan menggunakan *Optical Character Recognition* pernah dilakukan dengan menggunakan *Shearlet Transform* dan *Support Vector Machine* dengan akurasi mencapai 90%. Adapun data yang digunakan berupa 150 buah citra Aksara Bima yang didapatkan dengan cara diketik menggunakan *font* Aksara Bima dengan variasi tulisan terletak di berbagai posisi [18].

Penelitian pengenalan pola pada citra dengan *Probabilistic Neural Network* (PNN) pernah dilakukan sebelumnya. Penelitian pertama tentang klasifikasi jenis batu kapur dengan menggunakan fitur *Histogram* dengan akurasi sebesar 94% [13]. Penelitian kedua tentang pengenalan kapal berdasarkan citra *Inverse Synthetic Aperture Radar* (ISAR) dengan menggunakan ekstraksi ciri *Zernike Moments* memperoleh hasil akurasi rata-rata sebesar 96,25% [12].

Pada bidang kesehatan, penelitian dengan PNN pernah dilakukan untuk mengklasifikasi jenis *Arrhythmia*, suatu kondisi detak jantung dengan menggunakan *Wavelet Transform* dengan hasil akurasi rata-rata sebesar 92,746% [19]. Penelitian lainnya di bidang kesehatan dengan PNN dilakukan untuk mengidentifikasi dan klasifikasi jenis tumor otak dengan citra *Magnetic Resonance Imaging* (MRI) dan ekstraksi ciri GLCM dan DWT dengan akurasi mencapai hampir 100% [20]. Penelitian untuk mendeteksi penyakit *Coronary Artery Disease* (CAD) dengan menggunakan PNN dan ekstraksi fitur GLCM dan segmentasi dengan *K-Means Clustering* dengan akurasi yang tinggi dan waktu yang singkat tanpa adanya klasifikasi yang bernilai *false positive* atau *false negative* [21].

Penelitian tentang verifikasi bio metrik berbasis bibir menggunakan PNN dan *Region of Interest* (ROI) diperoleh hasil akurasi mencapai 87,26% [22]. Penelitian tentang *Genome-enabled Prediction* dengan menggunakan PNN dan *Multi Layer Perceptron* (MLP) pada dua spesies, yaitu jagung dan gandum. Pada penelitian ini diperoleh hasil PNN lebih akurat dibanding MLP dengan berbagai jenis skenario uji [23].

Selain penelitian tentang pengenalan pola pada citra, PNN digunakan dalam penelitian untuk mengestimasi pola cuaca di Wilayah Pekanbaru, rata-rata akurasi keseluruhan diperoleh sebesar 88%, dimana akurasi tertinggi diperoleh sebesar 96,77% untuk bulan Januari, April, dan Agustus. Sedangkan akurasi terendah terdapat pada bulan November sebesar 70% [24]. Penelitian lainnya yaitu pendeteksian serangan *Denial of Service* (DoS) memperoleh akurasi sebesar 98,06% [25].

Pada penelitian tentang pola tulisan tangan terdapat berbagai penelitian yang telah dilakukan dengan menggunakan PNN. Pada penelitian tentang pengenalan pola tanda tangan dengan menggunakan *Wavelet Entropy* dan PNN diperoleh akurasi mencapai 92% [26]. Pengenalan pola tanda tangan dengan menggunakan GLCM dan PNN dengan jumlah data citra sebanyak 100 untuk 10 jenis tanda tangan diperoleh akurasi rata-rata sebesar 71%, dengan akurasi terendah sebesar 50% dan tertinggi sebesar 90% [14]. Penelitian lainnya tentang pengenalan pola tulisan tangan terdapat pada pengenalan tulisan Bahasa Arab. Dengan menggunakan PNN, diperoleh hasil akurasi rata-rata sebesar 90,14% dengan jumlah data sebanyak 100 data citra untuk 10 huruf Bahasa Arab [11]. Selain itu pada penelitian yang dilakukan oleh Bondan Sebastian pada tahun 2015 untuk mengenali dokumen beraksara Jawa dengan menggunakan PNN diperoleh hasil akurasi mencapai 93,12% [27].

Penelitian lainnya tentang pengenalan Aksara yaitu pengenalan Aksara Bali dengan menggunakan *Local Binary Pattern* dan klasifikasi dengan *K-Nearest Neighbor* dan *Support Vector Machine* diperoleh akurasi tertinggi sebesar 74,06% dengan rata-rata waktu komputasi 2,3203 detik [8]. Pengenalan Aksara Sunda dengan menggunakan *Modified Direction Feature* dan *Learning Vector Quantization* (LVQ) memperoleh akurasi sebesar 78,57% [6]. Penelitian tentang pengenalan pola tulisan tangan Aksara Lontara dengan menggunakan LVQ diperoleh akurasi sebesar 66,66% [28]. Pengenalan Aksara Jawa dengan menggunakan Artificial Neural Network Backpropagation dan ekstraksi fitur *Canny Edge Detection* diperoleh akurasi rata-rata sebesar 74% [29]. Adapun penelitian dengan menggunakan metode *Convolution Neural Network* (CNN) dalam pengenalan Aksara Jawa diperoleh hasil akurasi sebesar 95,04%. Namun dalam penggunaan CNN diperlukan data yang besar, pada penelitian tersebut terdapat 53424 jumlah citra [30].

Berdasarkan pemaparan di atas, dilakukan suatu penelitian dengan menggunakan klasifikasi PNN dan ekstraksi fitur GLCM dan *Zoning*. Adapun pertimbangan penulis dalam memilih PNN sebagai *classifier* adalah PNN tidak memerlukan proses *train* yang lama dan dapat bekerja dengan baik dalam jumlah data yang sedikit yang ditunjukkan pada penelitian untuk pendeteksian teks pada citra dengan 300 buah citra [17] dan pengenalan tulisan Bahasa

Arab dengan jumlah citra sebanyak 100 [11]. Pemilihan GLCM yang dipadukan dengan *Zoning* sebagai ekstraksi ciri yang digunakan merujuk pada penelitian tentang pengenalan tanda tangan dengan PNN dan ekstraksi fitur GLCM yang memperoleh hasil akurasi sebesar 71% [14]. Terdapat beberapa kesalahan dalam klasifikasi pada penelitian tersebut karena kemiripan nilai ciri yang diperoleh, dimana nilai ciri didapatkan secara global. Penelitian tentang pengenalan pola Aksara Sasak dengan menggunakan *Moment Invariant* dan *Zoning* menunjukkan bahwa *Zoning* dapat meningkatkan nilai akurasi dalam pengenalan pola Aksara Sasak hingga 15,61% [16]. Oleh karena itu, dilakukan penelitian dengan ekstraksi ciri GLCM dengan harapan akurasi yang diperoleh dapat lebih baik jika dipadukan dengan *Zoning* dan klasifikasi PNN dapat melakukan klasifikasi dengan baik pada jumlah data yang sedikit dalam pengenalan pola tulisan tangan Aksara Bima.

2.2 Dasar Teori

2.2.1 Aksara Bima

Bima memiliki peradaban yang luar biasa, Aksara Bima merupakan salah satu warisan budaya yang sempat hilang selama ratusan tahun. Menurut catatan Thomas Stanford Raffles (1978), Bima memiliki dua jenis Aksara, yaitu Aksara huruf lengkung dan Aksara huruf garis-garis [1]. Aksara huruf lengkung menurutnya sudah tidak terpakai lagi. Sampai saat ini belum ditemukan naskah yang ditulis dengan huruf lengkung karena diduga naskah tersebut musnah terbakar bersamaan dengan terbakarnya Istana Bima tempo dulu. Namun, jenis Aksara ini pernah diajarkan di Sekolah Dasar Negeri di Bima dan Dompu hingga akhirnya sempat terhenti selama beberapa lama kemudian dilanjutkan kembali. Aksara lengkung tersebut hanya terdapat dalam catatan Raffles di dalam lampiran buku *The History of Java* [1].

Adapun Aksara Bima yang baru berbentuk garis-garis. 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]. Bentuk huruf Aksara Bima dapat dilihat pada Gambar 2.1.

| AKSARA MBOJO | | | | | |
|---|---|---|---|---|---|
| HURUF DASAR | | | | | |
|  |  |  |  |  |  |
| A | BA | CA | DA | FA | GA |
|  |  |  |  |  |  |
| HA | JA | KA | LA | MA | NA |
|  |  |  |  |  |  |
| PA | RA | SA | TA | WA | YA |
| HURUF SENGAU | | | | | |
|  |  |  |  |  |  |
| MBA | NCA | NDA | NGA | NGGA | NTA |
|  |  |  |  |  |  |
| MPA | NCA | NDA | NGA | NGGA | NTA |

Gambar 2.1 Aksara Bima baru atau bentuk garis-garis

Seorang ahli bernama Henry Chambert-Loir pernah beranggapan bahwa aksara Bima tidak pernah ada karena tidak digunakan sebagai bahasa yang resmi secara tertulis, namun hal tersebut terbantahkan setelah penemuan sebuah catatan tentang aksara Bima di Perpustakaan Nasional Indonesia berupa selembarnya dokumen dari H Zollinger seorang peneliti dari Belanda melalui penelitian yang dilakukan oleh Hj. Siti Maryam R. Salahuddin, seorang Puteri dari Sultan Bima ke-14, Sultan Muhammad Salahuddin [31].

Pada Aksara Bima berbentuk garis-garis, terdapat dua jenis huruf yaitu huruf dasar dan huruf sengau atau huruf yang lebih dari dua. Huruf dasar memiliki bentuk satu huruf mulai dari A sampai Y kecuali huruf Q, V, X, dan Z. Namun, dalam beberapa kalimat huruf Q diganti dengan huruf K seperti kata “Mutlaq” menjadi “Mutlak”. Begitu pula dengan huruf Z diganti dengan J, seperti kata “zaman” menjadi “jaman” [1].

2.2.2 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. Beberapa contoh ekstraksi ciri antara lain adalah ciri bentuk, ukuran, geometri, dan tekstur. Nilai ciri yang didapatkan dapat digunakan menjadi masukan dalam model pembelajaran mesin [32].

2.2.3 Gray Level Co-occurrence Matrix (GLCM)

Gray Level Co-Occurrence Matrix (GLCM) adalah salah satu jenis ekstraksi fitur. GLCM pertama kali dikenalkan oleh Haralick, dimana GLCM adalah salah satu metode statistika yang digunakan untuk menganalisis tekstur. Langkah yang dilakukan dalam mendapatkan ciri pada GLCM adalah menentukan *Co-occurrence Matrix*, yaitu suatu matriks

yang menunjukkan hubungan spasial antara *gray level* dalam ciri tekstur [14]. Bentuk dari *co-occurrence matrix* dapat dilihat pada Gambar 2.2.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|-------|-------|-------|-------|-------|-------|---|-------|
| 0 | 0 | 0 | 0.083 | 0 | 0.083 | 0 | 0 | 0 |
| 1 | 0.167 | 0 | 0 | 0 | 0 | 0.083 | 0 | 0 |
| 2 | 0 | 0.083 | 0 | 0.083 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.083 |
| 4 | 0 | 0.083 | 0 | 0.083 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0.083 | 0.083 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Gambar 2.2 *Co-occurrence matrix*

Setiap citra akan menghasilkan suatu *Co-occurrence Matrix* dengan setiap element p dengan posisi (i,j) menunjukkan frekuensi relative dua piksel *gray level* i dan j . Kemudian setiap nilai pada matriks akan dinormalisasi dengan persamaan (2-1) menjadi suatu matriks probabilitas [14].

$$p(i,j) = \frac{p_d(i,j)}{\sum_{i,j=0}^{N-1} p_d(i,j)} \quad (2-1)$$

Fitur tekstur yang diekstraksi dengan menggunakan GLCM pada penelitian ini adalah *Dissimilarity*, *Entropy*, dan *Homogeneity* yang perhitungannya ditunjukkan pada Persamaan (2-2), (2-3), dan (2-4).

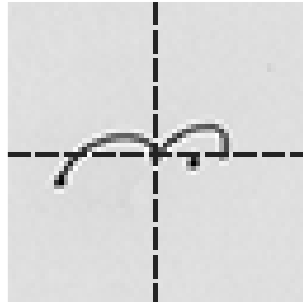
$$f(x) = \sum_i \sum_j |i - j| \cdot p(i,j) \quad (2-2)$$

$$f(x) = \sum_i \sum_j p(i,j) \cdot \log p(i,j) \quad (2-3)$$

$$f(x) = \sum_i \sum_j \frac{1}{1+(i-j)^2} p(i,j) \quad (2-4)$$

2.2.4 Zoning

Zoning merupakan suatu pembagian citra menjadi $m \times n$ daerah, dimana setiap daerah yang telah terbagi memiliki nilai fitur tersendiri. Tujuan dari *zoning* adalah mendapatkan nilai ciri bagian atau zona sejumlah $m \times n$ yang merupakan bagian dari ciri citra secara utuh [16].

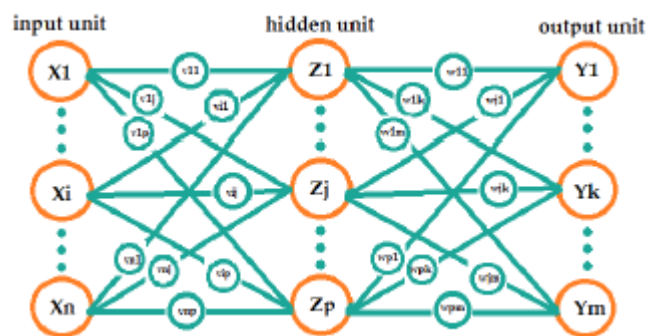


Gambar 2.3 Zoning

2.2.5 Artificial Neural Network

Artificial Neural Network (ANN) adalah suatu model *Machine Learning* dimana arsitektur dari ANN sendiri direpresentasikan sebagai jaringan sel syaraf biologis pada otak, yang selalu menstimulasi proses pembelajaran pada otak manusia [32].

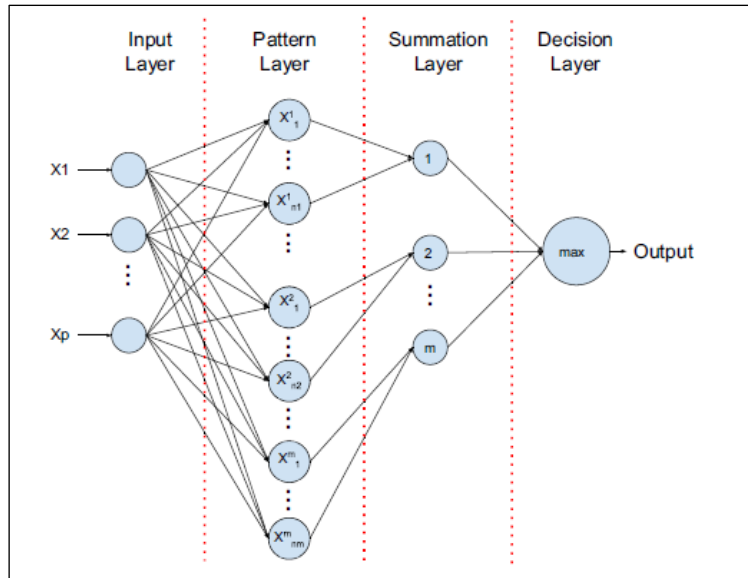
Pada umumnya suatu *Artificial Neural Network* memiliki ratusan *neuron*. *Neural Network* memiliki banyak jenis, salah satu model original dari *Neural Network* adalah *Perceptron* yang ditemukan oleh Frank Rosenblatt pada tahun 1958, sampai saat ini terdapat berbagai model dari *Neural Network*. Arsitektur suatu *Neural Network* pada umumnya terdiri dari *input layer*, *hidden layer*, dan *output layer* yang masing-masing terdapat *neuron* di dalamnya [10]. Arsitektur dari ANN dapat dilihat pada Gambar 2.4.



Gambar 2.4 Arsitektur ANN

2.2.6 Probabilistic Neural Network (PNN)

Probabilistic Neural Network adalah salah satu varian dari ANN yang dibentuk berdasarkan suatu pendekatan dengan *Probabilistic Density Function* (PDF). Pada PNN tidak diperlukan proses komputasi yang berat seperti pada *Backpropagation Neural Network*, melainkan dengan mengukur nilai kesamaan data pola dengan data masukan [33]. Arsitektur PNN dapat dilihat pada Gambar 2.5.



Gambar 2.5 Arsitektur PNN [33]

PNN terdiri dari 4 buah *layer* atau lapisan, antara lain *Input Layer*, *Pattern Layer*, *Summation Layer*, dan *Decision Layer* atau *Output Layer*. *Input layer* terdiri dari beberapa *Neuron* yang jumlahnya sesuai dengan jumlah data *input*. Pada *Pattern Layer* dilakukan perhitungan nilai PDF pada masing-masing *node* dengan Persamaan (2-5).

$$f(x, m) = \frac{1}{(2\pi)^{\frac{n}{2}} \sigma^n} e^{\left(\frac{-\|x-m\|^2}{2\sigma^2}\right)} \quad (2-5)$$

Dimana x merupakan vektor *input*, dan m merupakan vektor pada *pattern node*, dan n merupakan panjang vektor [33].

Pada *Summation Layer* dilakukan proses penjumlahan nilai PDF dari masing-masing *node* yang bersesuaian dengan kelasnya, dimana nilai pada *node* untuk kelas yang sama akan dijumlahkan. Jumlah *node* pada *Summation Layer* setara dengan jumlah *ouput* atau kelas, dimana dalam penelitian ini terdapat 22 kelas, sehingga terdapat 22 *node* pada *Summation Layer*. Kemudian pada *Decision Layer* atau *Output Layer* dihitung nilai rata-rata untuk masing-masing kelas, kemudian nilai tertinggi diambil untuk menentukan klasifikasi kelas.

2.2.7 Confusion Matrix

Confusion Matrix adalah suatu matriks berukuran 2×2 , dimana secara umum tiap blok terbagi menjadi 4 bagian. *True Positives* (TP) menunjukkan prediksi bernilai benar sesuai dengan kondisi yang sesungguhnya bernilai benar. *False Positives* (FP) menunjukkan prediksi bernilai benar dimana pada kondisi yang sesungguhnya bernilai salah. *True Negatives* (TN) menunjukkan prediksi bernilai salah sesuai dengan kondisi yang sesungguhnya bernilai salah.

False Negatives (FN) menunjukkan prediksi bernilai salah dimana pada kondisi yang sesungguhnya bernilai benar [34]. Bentuk dari *confusion matrix* dapat dilihat pada Gambar 2.6.

| | | 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 2.6 *Confusion matrix*

2.2.8 Akurasi, Presisi, dan Recall

Akurasi merupakan rasio prediksi benar (*True Positives* dan *True Negatives*) dengan keseluruhan data. Adapun perhitungan yang digunakan dalam menghitung akurasi dapat dilihat pada Persamaan (2-6) [35].

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \quad (2-6)$$

Presisi merupakan rasio prediksi benar positif (*True Positives*) dengan keseluruhan hasil yang diprediksi positif (*True Positives* dan *False Positives*). Adapun perhitungan yang digunakan dalam menghitung presisi dapat dilihat pada Persamaan (2-7) [35].

$$Presisi = \frac{TP}{TP+FP} \quad (2-7)$$

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 (2-8) [35].

$$Recall = \frac{TP}{TP+FN} \quad (2-8)$$

BAB III

METODOLOGI PENELITIAN

3.1 Alat dan Bahan Penelitian

Alat dan bahan yang diperlukan dalam melakukan penelitian berupa *hardware*, *software*, dan data berupa citra tulisan tangan Aksara Bima.

1. Alat Penelitian

Alat-alat yang digunakan dalam penelitian ini berupa *hardware* dan *software* sebagai berikut.

- a. Laptop ASUS X450JN Intel® Core™ i7-4720HQ @2.60 GHz (8 CPUs), dengan GPU NVIDIA GEFORCE 840M, dan RAM 12 GB
- b. Sistem operasi Windows 8.1
- c. Jupyter Notebook
- d. Google Colab
- e. Microsoft Office 2016

Adapun penggunaan dari masing-masing alat di atas adalah sebagai berikut.

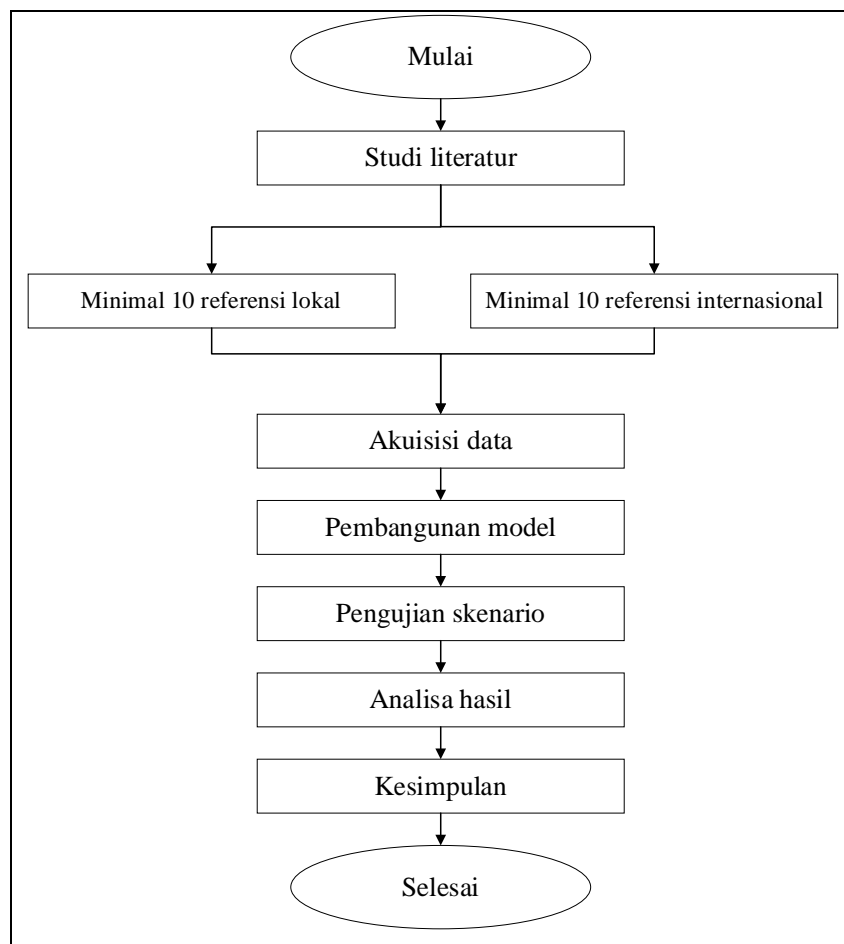
- a. Sebagai *hardware* yang digunakan dalam melakukan penelitian.
- b. Sistem operasi yang digunakan di dalam laptop
- c. Sebagai alat untuk mengeksekusi *code* dalam melaksanakan penelitian, dimana Jupyter Notebook digunakan untuk melakukan komputasi secara lokal di *hardware* dalam melakukan ekstraksi ciri
- d. Sebagai alat untuk mengeksekusi *code* dalam melaksanakan penelitian, dimana penggunaan Google Colab mirip dengan Jupyter Notebook, namun Google Colab digunakan dalam tahap klasifikasi untuk kemudahan instalasi *library* tambahan serta menjadi alternatif jika terdapat masalah dalam menggunakan Jupyter Notebook.
- e. Sebagai alat untuk menulis laporan penelitian

2. Bahan Penelitian

Bahan penelitian yang digunakan dalam penelitian ini adalah citra tulisan tangan Aksara Bima yang didapatkan dari sukarelawan dari kategori masyarakat umum sejumlah 10 orang yang kemudian di-*scan* menjadi citra digital sejumlah 2640 citra untuk 22 karakter dengan dengan format .jpg dan ukuran 64x64 piksel, dimana masing-masing sukarelawan menulis sebanyak 12 kali untuk tiap karakter.

3.2 Diagram Alir Penelitian


Terdapat beberapa tahapan dalam pelaksanaan penelitian ini. Pertama dilakukan studi literatur untuk memperoleh wawasan atau dasar dalam melakukan penelitian. Studi literatur dilakukan dengan membaca jurnal, buku, dan penelitian-penelitian terkait tentang pengenalan pola dan Aksara Bima. Selanjutnya dilakukan pengumpulan *dataset*, dimana dataset tersebut akan digunakan dalam pembuatan model, baik sebagai data latih maupun data uji. Setelah dataset berhasil dikumpulkan, dilakukan pembuatan model *machine learning* PNN untuk pengenalan pola Aksara Bima. Setelah model selesai dibuat, dilanjutkan dengan menguji berbagai jenis skenario pengujian antara lain penentuan *smoothing parameter*, variasi sudut GLCM, *zoning*, dan rasio perbandingan data latih dan data uji. Setelah pengujian skenario selesai, maka dilakukan analisa terhadap hasil pengujian skenario yang selanjutnya dapat diambil suatu kesimpulan skenario model manakah yang memiliki akurasi terbaik. Diagram alir penelitian dapat dilihat pada Gambar 3.1.















Gambar 3.1 Diagram alir penelitian

3.3 Pengumpulan Dataset

Dataset yang digunakan dalam penelitian ini adalah tulisan tangan Aksara Bima, dimana data yang didapatkan berasal dari sukarelawan masyarakat umum sebanyak 10 orang. Pengambilan data digunakan dengan menggunakan kertas berukuran F4 yang dibagi menjadi 12 buah kolom dan ditulis dengan bolpoin gel ukuran 1.0 mm. Masing-masing orang menuliskan tiap karakter sebanyak 12 kali, dengan jumlah karakter sebanyak 22 karakter. Data tulisan tangan karakter Aksara Bima yang telah ditulis kemudian di-*scan* dan di-*crop* dengan format .jpg dan ukuran 64x64 piksel. Sehingga dihasilkan $10 \times 22 \times 12 = 2640$ data citra Aksara Bima.

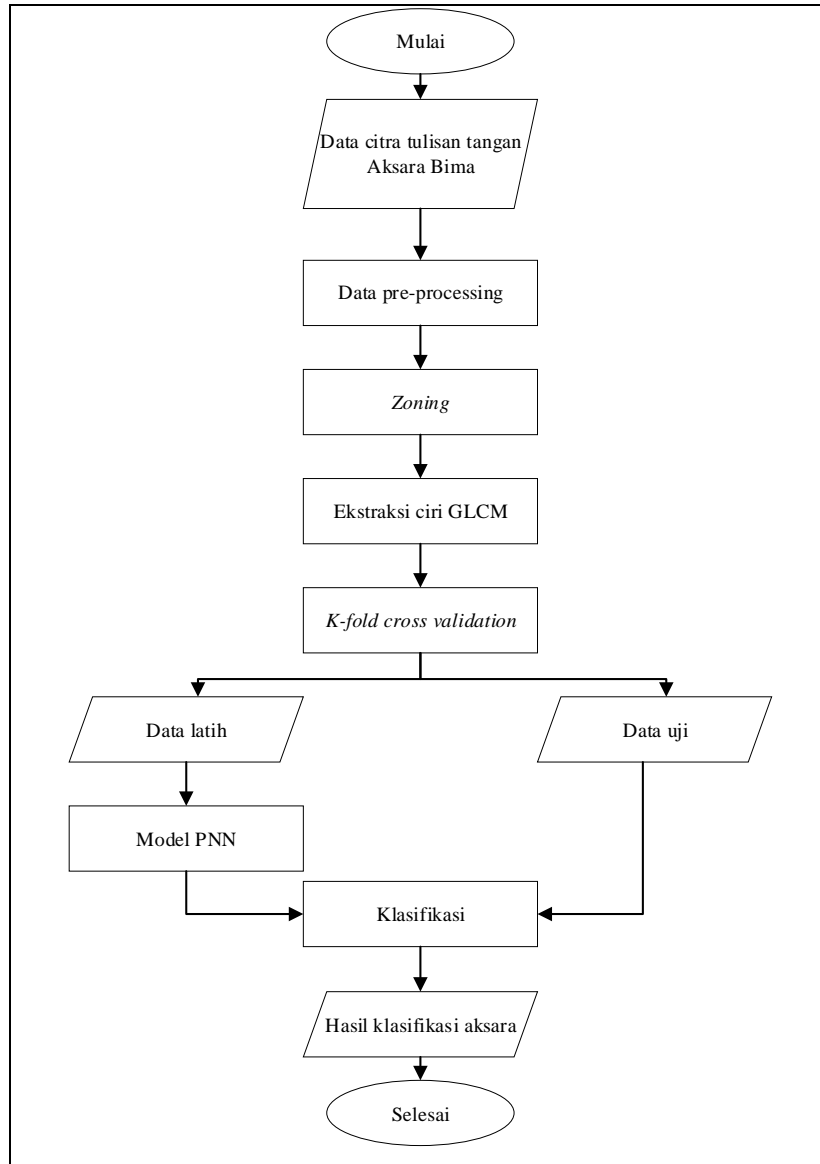
| | |
|--|---|
|  a | Nama : Tanggal : 03/03/2020 Orang Bima / Bukan Orang Bima |
|--|---|

| | | | | | |
|---|---|---|---|---|---|
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Gambar 3.2 Form dataset Aksara Bima

3.4 Pembangunan Model

Dalam pembangunan model *machine learning* dengan ekstraksi GLCM dan Zoning dan klasifikasi PNN, terdapat berbagai tahapan yang dilakukan, yaitu *data pre-processing*, ekstraksi ciri, pembagian data, dan pembuatan model PNN. Diagram alir pembangunan model dapat dilihat pada Gambar 3.3.



Gambar 3.3 Diagram alir perancangan model

3.4.1 Data Pre-processing

Setelah seluruh data citra diimpor dilakukan *pre-processing* pada citra, dimana dalam pre-processing citra asli diubah ke dalam ruang warna *grayscale* terlebih dahulu dan di-*resize* dengan ukuran yang sama yaitu 64x64 piksel untuk seluruh citra. Konversi ruang warna dari citra asli RGB menjadi *grayscale* 8-bit dilakukan dengan perhitungan pada Persamaan (3-1).

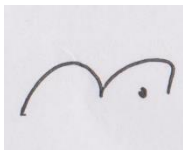
$$I(x, y) = \frac{I(x, y) - \text{lowest pixel value}}{\text{max pixel value} - \text{lowest pixel value}} \times 255 \quad (3-1)$$

Preprocessing ke dalam ruang warna *grayscale* dilakukan agar dapat dilakukan ekstraksi ciri GLCM. Proses *resize* menjadi ukuran 64x64 piksel dimaksudkan agar citra memiliki ukuran yang sama dan ukuran matriks yang digunakan tidak terlalu besar sehingga dapat meringankan proses komputasi.

3.4.2 Zoning

Data citra yang telah dilakukan pre-processing diproses dengan *zoning* sehingga menghasilkan beberapa daerah yang merupakan sub-citra bagian dari citra utuh, pada penelitian ini *zoning* dibagi menjadi 4 daerah. Contoh *zoning* dapat dilihat pada Gambar 3.4 berikut.

Citra asli



Matriks citra asli

| | | | |
|---|---|---|---|
| 1 | 0 | 2 | 3 |
| 4 | 1 | 0 | 4 |
| 2 | 1 | 5 | 5 |
| 5 | 4 | 3 | 7 |

Matriks citra zoning-1

| | |
|---|---|
| 1 | 0 |
| 4 | 1 |

Matriks citra zoning-2

| | |
|---|---|
| 2 | 3 |
| 0 | 4 |

Matriks citra zoning-3

| | |
|---|---|
| 2 | 1 |
| 5 | 4 |

Matriks citra zoning-4

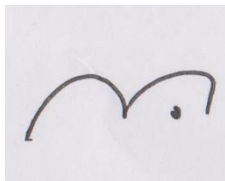
| | |
|---|---|
| 5 | 5 |
| 3 | 7 |

Gambar 3.4 Zoning pada citra

3.4.3 Ekstraksi Ciri GLCM

Citra yang telah melewati tahap *pre-processing* dan *zoning* akan diambil ciri teksturnya menggunakan ekstraksi ciri GLCM, dimana tahap pertama dalam ekstraksi fitur GLCM adalah pembuatan matriks kookurensi, dimana matriks kookurensi ini merupakan matriks yang menunjukkan jumlah pasangan ketetanggaan nilai piksel baris ke-i dengan nilai piksel kolom ke-j dengan derajat tertentu dan jarak ketetanggaan tertentu. Contoh pembuatan matriks kookurensi dengan derajat 0 dan tetangga berjarak 1 piksel dapat dilihat pada Gambar 3.5 dan Gambar 3.6.

Citra asli



Matriks citra asli

| | | | |
|---|---|---|---|
| 1 | 0 | 2 | 3 |
| 4 | 1 | 0 | 4 |
| 2 | 1 | 5 | 5 |
| 5 | 4 | 3 | 7 |

Pencarian tetangga piksel 0 dan 1 pada derajat 0 dan tetangga 1 piksel.

| | | | |
|---|---|---|---|
| 1 | 0 | 2 | 3 |
| 4 | 1 | 0 | 4 |
| 2 | 1 | 5 | 5 |
| 5 | 4 | 3 | 7 |

Pada citra tidak ada pasangan piksel 0 dan 1 yang bertetangga.

Pencarian tetangga piksel 0 dan 2 pada derajat 0 dan tetangga 1 piksel.

| | | | |
|---|---|---|---|
| 1 | 0 | 2 | 3 |
| 4 | 1 | 0 | 4 |
| 2 | 1 | 5 | 5 |
| 5 | 4 | 3 | 7 |

Pada citra terdapat 1 pasangan piksel 0 dan 2.

Gambar 3.5 Pencarian hubungan ketertanggaan antar nilai piksel dalam citra

Hasil dari perhitungan jumlah ketertanggaan tersebut disimpan dalam suatu tabel kookurensi, seperti pada Gambar 3.6.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 2 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 2 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 4 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Keterangan:

Baris

Kolom

Gambar 3.6 Matriks kookurensi GLCM

Setelah diperoleh matriks kookurensi, perlu dilakukan normalisasi sehingga tabel kookurensi terdiri atas probabilitas munculnya kookurensi antar nilai piksel dengan membagi nilai jumlah kookurensi tiap hubungan ketertanggaan terhadap total jumlah kookurensi seperti pada Gambar 3.7.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|-------|-------|-------|-------|-------|-------|---|-------|
| 0 | 0 | 0 | 0.083 | 0 | 0.083 | 0 | 0 | 0 |
| 1 | 0.167 | 0 | 0 | 0 | 0 | 0.083 | 0 | 0 |
| 2 | 0 | 0.083 | 0 | 0.083 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.083 |
| 4 | 0 | 0.083 | 0 | 0.083 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0.083 | 0.083 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Keterangan:

Baris

Kolom

Gambar 3.7 Matriks kookurensi GLCM setelah normalisasi

Langkah selanjutnya dalam ekstraksi ciri GLCM adalah perhitungan nilai ciri dengan persamaan seperti pada Persamaan (2-2) untuk *Dissimilarity*, Persamaan (2-3) untuk *Entropy*, dan Persamaan (2-4) untuk *Homogeneity*.

$$Dissimilarity = \sum_i \sum_j |i - j| \cdot p(i, j) = 1.91$$

Perhitungan nilai *dissimilarity* dilakukan dengan menjumlahkan seluruh hasil kali dari nilai mutlak pengurangan tingkat keabuan i dan tingkat keabuan j yang bertetangaan, dimana selanjutnya dikalikan dengan nilai probabilitas kemunculan ketetangannya.

$$Entropy = \sum_i \sum_j p(i, j) \cdot \log p(i, j) = -1.019$$

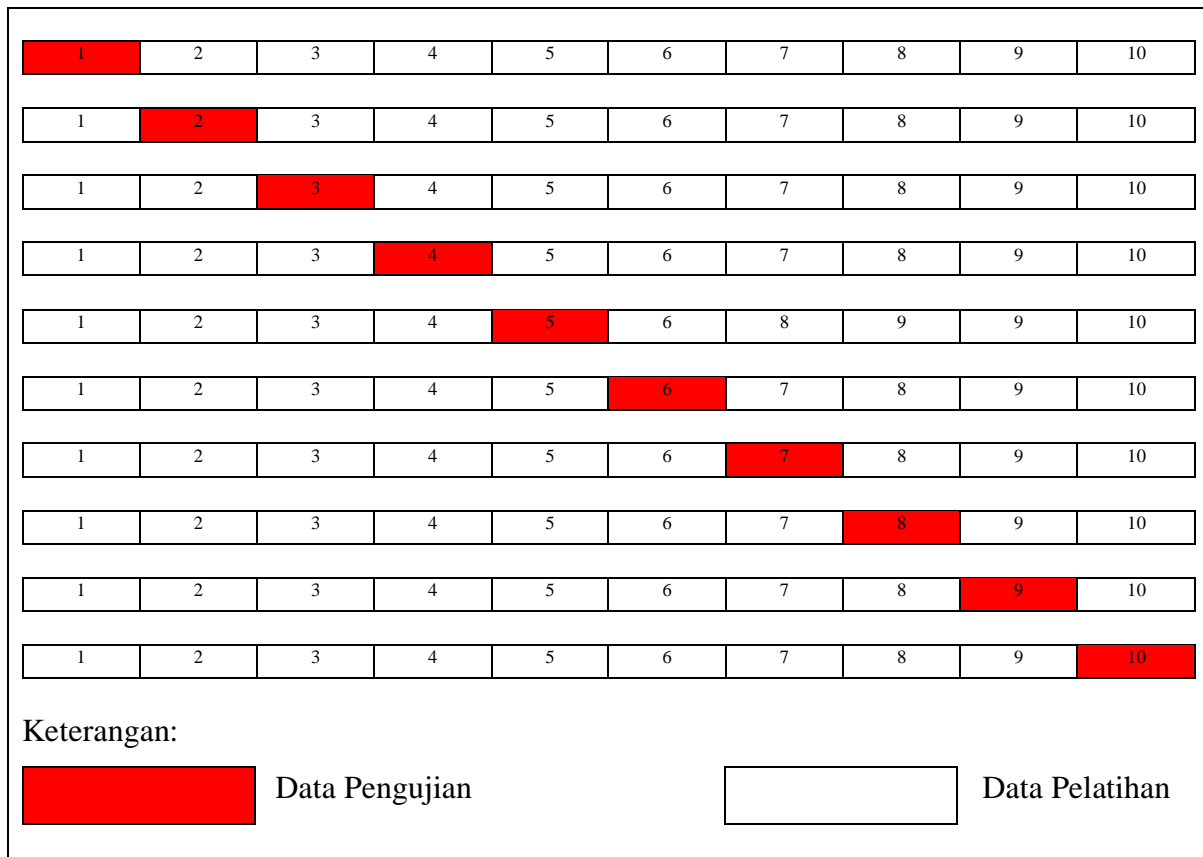
Perhitungan nilai *entropy* dilakukan dengan menjumlahkan seluruh hasil kali dari nilai probabilitas ketetangaan tingkat keabuan dengan nilai logaritma dari nilai probabilitasnya.

$$Homogeneity = \sum_i \sum_j \frac{1}{1 + (i - j)^2} p(i, j) = 0.4417$$

Perhitungan nilai *Homogeneity* dilakukan dengan menjumlahkan seluruh nilai dari probabilitas ketetangaan tingkat keabuan i dan tingkat keabuan j, kemudian dibagi dengan nilai kuadrat dari selisih tingkat keabuan yang ditambahkan dengan nilai 1.

3.4.4 *K-Fold Cross Validation*

K-Fold Cross Validation adalah suatu cara pembagian data dimana data dibagi menjadi beberapa kluster kemudian didapatkan berbagai jenis kombinasi data dengan rasio yang sama sesuai dengan kluster yang sudah ada, contoh *K-fold Cross Validation* dapat dilihat pada Gambar 3.8 dengan nilai K bernilai 10.



Gambar 3.8 K-fold cross validation

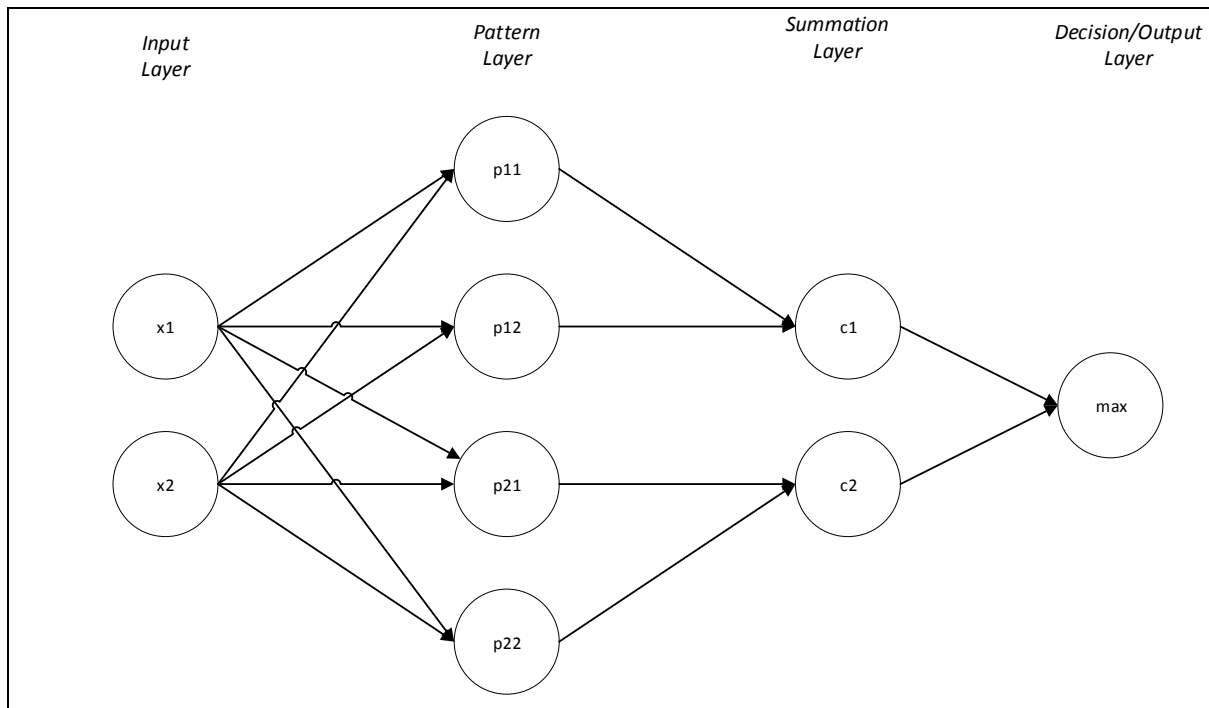
3.4.5 Model PNN

Pada penelitian ini digunakan model PNN dengan arsitektur terdiri atas *Input Layer*, *Pattern Layer*, *Summation Layer*, dan *Output Layer*. Jumlah node pada *Input Layer* terdiri atas jumlah fitur yang akan digunakan di dalam model, pada penelitian ini minimal digunakan 3 nilai fitur dimana tidak diterapkan *zoning* dan maksimal sejumlah 15 fitur untuk *zoning* dengan 4 area. *Pattern Layer* terdiri atas beberapa node, dimana jumlah *node* tergantung pada jumlah data *train* yang akan digunakan. *Summation layer* terdiri atas 22 *node*, dimana 22 *node* ini merupakan representasi dari jumlah kelas pada klasifikasi. *Output layer* hanya terdiri dari 1 buah *node* yang menunjukkan hasil klasifikasi. Berikut adalah contoh data dan arsitektur sederhana dari model PNN dan bagaimana cara melakukan klasifikasi dengan menggunakan PNN.

Tabel 3.1 Data *dummy* PNN

| No | Nilai Fitur 1 (x1) | Nilai Fitur 2 (x2) | Kelas |
|----|--------------------|--------------------|-------|
| 1 | 1 | 0 | 1 |
| 2 | 0 | 1 | 1 |
| 3 | -1 | 0 | 2 |
| 4 | 0 | -1 | 2 |

Selanjutnya dibuat suatu model berdasarkan data di atas, dimana data di atas akan digunakan sebagai *data train* yang akan membentuk *node* dari *pattern layer*. Pada *pattern layer* terdiri atas beberapa nilai fitur, dimana pada *pattern layer* akan dilakukan perhitungan nilai PDF antara nilai *input* yang di-*forward* dari *node* pada *input layer* menuju *node* pada *pattern layer*. Setelah nilai PDF di *pattern layer* dihitung, dilakukan *forward* nilai PDF pada masing-masing *node pattern* menuju *node* di *summation layer*, dimana nilai pada *node pattern* akan diteruskan ke *node* dengan kelas yang bersesuaian. Pada Gambar 3.8 dapat dilihat pada *node* p11 dan *node* 12 merupakan *pattern* dari kelas 1, nilai dari *node* tersebut diteruskan pada *node* c1 yang merupakan *node summation* untuk kelas 1. Hal yang sama dilakukan pada nilai *node* p21 dan p22 yang akan diteruskan ke *node* c2 yang merupakan *node* untuk kelas 2. Selanjutnya pada *summation layer* untuk masing-masing *node* akan dijumlahkan nilai dari *node pattern layer* yang masuk ke *summation layer*. Setelah seluruh jumlah nilai PDF di masing-masing *node* untuk tiap kelas didapatkan, nilai tersebut diteruskan ke *decision/output layer*, dimana pada *layer* ini akan dilakukan proses perhitungan rata-rata PDF untuk masing-masing kelas dengan membagi nilai total PDF terhadap jumlah *node pattern* untuk masing-masing kelas. Kelas dengan nilai rata-rata tertinggi akan dipilih sebagai *output*.



Gambar 3.9 Contoh model PNN sederhana

Berikut contoh perhitungan pada model PNN di atas.

Data pada Tabel 3.1 akan digunakan sebagai *data train*, dimana seluruh data akan dimasukkan ke dalam *pattern layer*. Selanjutnya akan dilakukan klasifikasi terhadap data *input* dengan nilai $x_1 = 0.5$ dan $x_2 = 0.5$ dengan nilai standar deviasi/*smoothing parameter* bernilai 1. Sehingga pada masing-masing *node* di *pattern layer* memiliki kumpulan nilai yang dibentuk menjadi suatu vektor sebagai berikut.

$$\text{vektor } p_{11} = [1, 0]$$

$$\text{vektor } p_{12} = [0, 1]$$

$$\text{vektor } p_{21} = [-1, 0]$$

$$\text{vektor } p_{22} = [0, -1]$$

Nilai pada x_1 dimasukkan ke dalam *input node* x_1 dan nilai pada x_2 dimasukkan ke dalam *input node* x_2 . Kemudian nilai pada *node* x_1 dan x_2 diteruskan menuju seluruh *node* pada *pattern layer*. Kumpulan nilai pada *input layer* yang diteruskan menuju *pattern layer* akan dibaca sebagai suatu vektor. Pada masing-masing *node* di *pattern layer* dilakukan perhitungan nilai PDF dengan perhitungan sesuai pada Persamaan (2-5). Berikut perhitungan nilai PDF pada masing-masing *node* di *pattern layer*.

Sebelum masuk ke perhitungan nilai PDF, untuk memudahkan perhitungan lakukan perhitungan jarak *Euclidean* terlebih dahulu untuk vektor masukan dengan vektor pada *pattern node* menggunakan Persamaan (3-2).

$$\|x - y\| = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2} \quad (3-2)$$

Nilai jarak *Euclidean* pada masing-masing *pattern node* dapat dilihat pada contoh perhitungan sebagai berikut.

Perhitungan nilai jarak *Euclidean* pada *pattern node* p11 dapat dilihat pada perhitungan berikut.

$$\begin{aligned} \|x - p11\| &= \sqrt{(x_1 - p11_1)^2 + (x_2 - p11_2)^2} \\ &= \sqrt{(0.5 - 1)^2 + (0.5 - 0)^2} \\ &= \sqrt{0.25 + 0.25} \\ &= \sqrt{0.5} \end{aligned}$$

Didapatkan nilai jarak *Euclidean* pada *pattern node* p11 dengan nilai $\sqrt{0.5}$.

Perhitungan nilai jarak *Euclidean* pada *pattern node* p12 dapat dilihat pada perhitungan berikut.

$$\begin{aligned} \|x - p12\| &= \sqrt{(x_1 - p12_1)^2 + (x_2 - p12_2)^2} \\ &= \sqrt{(0.5 - 0)^2 + (0.5 - 1)^2} \\ &= \sqrt{0.25 + 0.25} \\ &= \sqrt{0.5} \end{aligned}$$

Didapatkan nilai jarak *Euclidean* pada *pattern node* p12 dengan nilai $\sqrt{0.5}$.

Perhitungan nilai jarak *Euclidean* pada *pattern node* p21 dapat dilihat pada perhitungan berikut.

$$\begin{aligned} \|x - p21\| &= \sqrt{(x_1 - p21_1)^2 + (x_2 - p21_2)^2} \\ &= \sqrt{(0.5 - (-1))^2 + (0.5 - 0)^2} \\ &= \sqrt{2.25 + 0.25} \\ &= \sqrt{2.5} \end{aligned}$$

Didapatkan nilai jarak *Euclidean* pada *pattern node* p21 dengan nilai $\sqrt{2.5}$.

Perhitungan nilai jarak *Euclidean* pada *pattern node* p22 dapat dilihat pada perhitungan berikut.

$$\begin{aligned} \|x - p22\| &= \sqrt{(x_1 - p22_1)^2 + (x_2 - p22_2)^2} \\ &= \sqrt{(0.5 - 0)^2 + (0.5 - (-1))^2} \\ &= \sqrt{0.25 + 2.25} \\ &= \sqrt{2.5} \end{aligned}$$

Didapatkan nilai jarak *Euclidean* pada *pattern node* p22 dengan nilai $\sqrt{2.5}$.

Selanjutnya dilakukan perhitungan nilai PDF untuk masing-masing *node* di *pattern layer* dengan perhitungan sebagai berikut.

Perhitungan nilai PDF pada *pattern node* p11 dapat dilihat pada perhitungan berikut.

$$\begin{aligned} f(p11) &= \frac{1}{(2\pi)^{\frac{n}{2}} \sigma^n} e^{\left(\frac{-\|x-p11\|^2}{2\sigma^2}\right)} \\ &= \frac{1}{(2\pi)^{\frac{2}{2}} x 1^2} e^{\left(\frac{-\|x-p11\|^2}{2 x 1^2}\right)} \\ &= \frac{1}{2\pi} e^{\left(\frac{-0.5}{2}\right)} \\ &= 0.12 \end{aligned}$$

Didapatkan nilai PDF pada *pattern node* p11 atau $f(p11)$ dengan nilai 0.12.

Perhitungan nilai PDF pada *pattern node* p12 dapat dilihat pada perhitungan berikut.

$$\begin{aligned} f(p12) &= \frac{1}{(2\pi)^{\frac{n}{2}} \sigma^n} e^{\left(\frac{-\|x-p12\|^2}{2\sigma^2}\right)} \\ &= \frac{1}{(2\pi)^{\frac{2}{2}} x 1^2} e^{\left(\frac{-\|x-p12\|^2}{2 x 1^2}\right)} \\ &= \frac{1}{2\pi} e^{\left(\frac{-0.5}{2}\right)} \\ &= 0.12 \end{aligned}$$

Didapatkan nilai PDF pada *pattern node* p12 atau $f(p12)$ dengan nilai 0.12.

Perhitungan nilai PDF pada *pattern node* p21 dapat dilihat pada perhitungan berikut.

$$\begin{aligned} f(p21) &= \frac{1}{(2\pi)^{\frac{n}{2}} \sigma^n} e^{\left(\frac{-\|x-p21\|^2}{2\sigma^2}\right)} \\ &= \frac{1}{(2\pi)^{\frac{2}{2}} x 1^2} e^{\left(\frac{-\|x-p21\|^2}{2 x 1^2}\right)} \\ &= \frac{1}{2\pi} e^{\left(\frac{-0.5}{2}\right)} \\ &= 0.05 \end{aligned}$$

Didapatkan nilai PDF pada *pattern node* p21 atau $f(p21)$ dengan nilai 0.05.

Perhitungan nilai PDF pada *pattern node* p22 dapat dilihat pada perhitungan berikut.

$$f(p22) = \frac{1}{(2\pi)^{\frac{n}{2}} \sigma^n} e^{\left(\frac{-\|x-p22\|^2}{2\sigma^2}\right)}$$

$$\begin{aligned}
&= \frac{1}{(2\pi)^{\frac{2}{2}} \times 1^2} e^{\left(\frac{-\|x-p_{22}\|^2}{2 \times 1^2}\right)} \\
&= \frac{1}{2\pi} e^{\left(\frac{-0.5}{2}\right)} \\
&= 0.05
\end{aligned}$$

Didapatkan nilai PDF pada *pattern node* p22 atau $f(p_{22})$ dengan nilai 0.05.

Setelah nilai PDF pada masing-masing *node* di *pattern layer* dihitung, nilai tersebut akan diteruskan ke *summation layer*, dimana nilai masing-masing *node* akan diteruskan menuju *node* yang berhubungan dengan kelas masing-masing *node* di *pattern layer*. Pada contoh di atas nilai dari *node* p11 dan p12 akan diteruskan menuju *node* c1 dan nilai dari *node* p21 dan p22 akan diteruskan menuju *node* c2. Pada *node* di *summation layer*, seluruh nilai yang masuk akan dijumlahkan terlebih dahulu. Sehingga nilai yang masuk pada masing-masing *node* di *pattern layer* sebagai berikut.

$$\begin{aligned}
c1 &= p11 + p12 \\
&= 0.12 + 0.12 \\
&= 0.24 \\
c2 &= p21 + p22 \\
&= 0.05 + 0.05 \\
&= 0.1
\end{aligned}$$

Setelah dilakukan penjumlahan terhadap nilai yang masuk menuju *node* di *summation layer*, pada masing-masing *node* tersebut akan dihitung rata-ratanya dengan membagi hasil penjumlahan dengan jumlah *node* di *pattern layer* yang terhubung ke *node* di *summation layer*, sehingga nilai keluaran dari masing-masing *node* di *summation layer* sebagai berikut.

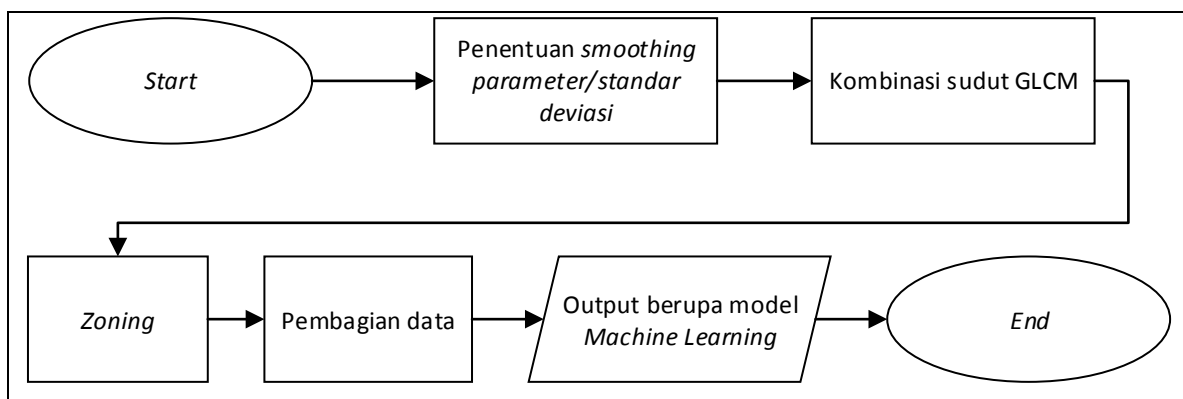
$$\begin{aligned}
f(c1) &= \frac{(p11 + p12)}{2} \\
&= \frac{0.24}{2} \\
&= 0.12 \\
f(c2) &= \frac{(p21 + p22)}{2} \\
&= \frac{0.1}{2} \\
&= 0.05
\end{aligned}$$

Selanjutnya nilai pada masing-masing *node* di *summation layer* diteruskan menuju *decision/output layer*, dimana pada *decision/output layer* akan dilakukan penentuan

untuk *input* yang diterima akan ditentukan ke kelas mana. Pada *node* di *output layer* akan dihitung nilai maksimum yang masuk, dimana nilai maksimum tersebut akan menunjukkan *input* yang diterima akan digolongkan ke kelas yang sesuai dengan indeks *node* yang mengirimkan nilai tersebut. Pada nilai $f(c1)$ diperoleh nilai 0.12, sedangkan pada $f(c2)$ diperoleh nilai 0.05. Maka untuk menentukan kelas dicari nilai maksimum pada nilai masing-masing *node* yang diteruskan menuju *decision/output layer*, dimana nilai maksimum adalah $f(c1)$ yaitu 0.12, maka *input* $x1=0.5$ dan $x2=0.5$ diklasifikasi menjadi kelas yang berkorespondensi dengan $c1$ yaitu kelas 1.

3.5 Pengujian Skenario

Pada penelitian yang dilakukan, penulis merangkai skenario pengujian yang dilakukan sebagai berikut ditunjukkan oleh Gambar 3.10.



Gambar 3.10 Pengujian skenario

Pengujian skenario yang dilakukan penulis dibagi menjadi 4 tahapan. Pada tahapan skenario awal, akan digunakan pembagian data yang umum yaitu 9:1. Adapun tahapan dalam pengujian skenario antara lain sebagai berikut.

1. *Smoothing parameter* atau standar deviasi yang digunakan dalam model PNN. Nilai *smoothing parameter* bergantung pada *range input* fitur. Misalkan nilai fitur berada di antara 0-20, maka nilai *smoothing parameter* yang digunakan berada di antara 0-20, seperti 10 atau 15.
2. Sudut GLCM, dimana pada penelitian yang dilakukan digunakan beberapa sudut dalam perhitungan ekstraksi ciri dalam GLCM di tahap pembuatan matriks kookurensi. Nilai sudut yang digunakan antara lain 0, 45, 90, dan 135.
3. Penggunaan *Zoning*, dimana pada tahapan ini dilakukan pengujian dengan menggunakan *zoning* dan tidak menggunakan *zoning*, dimana *zoning* dibagi menjadi 4 daerah dan 16 daerah.

- Pembagian data, dimana pada tahapan ini *dataset* dibagi menjadi beberapa kombinasi pembagian data *train* dan data *test*. Kombinasi yang digunakan bervariasi mulai dari perbandingan data *train:test* sebesar 9:1 hingga 1:9.

Output atau keluaran yang didapatkan pada pengujian skenario adalah model *machine learning* PNN dengan performa terbaik. Adapun teknik pengujian yang dilakukan adalah dengan menghitung nilai akurasi, presisi, dan *recall*, dimana nilai tersebut didapatkan dari suatu *confusion matrix*. Persamaan yang digunakan dalam menghitung nilai akurasi, presisi, dan *recall* sesuai pada Persamaan (2-6) – (2-7). Berikut contoh perhitungan nilai akurasi, presisi, dan *recall* dengan menggunakan 3 kelas berbeda.

Tabel 3.2 *Confusion matrix 3x3*

| | | <i>Predict Class</i> | | |
|---------------------|---|----------------------|-------------------|-------------------|
| | | A | B | C |
| <i>Actual Class</i> | A | 30 (AA) | 20 (AB) | 10 (AC) |
| | B | 50 (BA) | 60 (BB) | 10 (BC) |
| | C | 20 (CA) | 20 (CB) | 80 (CC) |

Berdasarkan *confusion matrix* pada Tabel 3.2, dapat dihitung nilai akurasi, presisi, dan *recall*, dimana nilai akurasi dihitung secara keseluruhan, sedangkan untuk presisi dan *recall* perlu dilakukan perhitungan per kelas.

Nilai akurasi dapat dihitung dengan membandingkan nilai prediksi yang bernilai benar pada seluruh kelas dengan keseluruhan jumlah data, sebagai berikut.

$$\begin{aligned}
 \text{Akurasi} &= \frac{AA + BB + CC}{\text{total data}} \\
 &= \frac{30 + 60 + 80}{300} \\
 &= 0.567
 \end{aligned}$$

Nilai presisi pada kelas A dapat dihitung dengan membandingkan nilai prediksi yang bernilai benar terhadap kelas A dengan keseluruhan yang diprediksi sebagai kelas A.

$$\begin{aligned}
 \text{Presisi A} &= \frac{AA}{\text{Prediksi sebagai A (AA + BA + CA)}} \\
 &= \frac{30}{30 + 50 + 20}
 \end{aligned}$$

$$\begin{aligned}
&= \frac{30}{100} \\
&= 0.3
\end{aligned}$$

Nilai presisi pada kelas B dapat dihitung dengan membandingkan nilai prediksi yang bernilai benar terhadap kelas B dengan keseluruhan yang diprediksi sebagai kelas B.

$$\begin{aligned}
\text{Presisi } B &= \frac{BB}{\text{Prediksi sebagai B } (AB + BB + CB)} \\
&= \frac{60}{20 + 60 + 20} \\
&= \frac{60}{100} \\
&= 0.6
\end{aligned}$$

Nilai presisi pada kelas C dapat dihitung dengan membandingkan nilai prediksi yang bernilai benar terhadap kelas C dengan keseluruhan yang diprediksi sebagai kelas C.

$$\begin{aligned}
\text{Presisi } C &= \frac{CC}{\text{Prediksi sebagai C } (AC + BC + CC)} \\
&= \frac{80}{10 + 10 + 80} \\
&= \frac{80}{100} \\
&= 0.8
\end{aligned}$$

Nilai *recall* pada kelas A dapat dihitung dengan membandingkan nilai prediksi yang bernilai benar terhadap kelas A dengan keseluruhan yang secara aktual merupakan kelas A.

$$\begin{aligned}
\text{Recall } A &= \frac{AA}{\text{Aktual kelas A } (AA + AB + AC)} \\
&= \frac{30}{30 + 20 + 10} \\
&= \frac{30}{60} \\
&= 0.5
\end{aligned}$$

Nilai *recall* pada kelas B dapat dihitung dengan membandingkan nilai prediksi yang bernilai benar terhadap kelas B dengan keseluruhan yang secara aktual merupakan kelas B.

$$\begin{aligned}
\text{Recall } B &= \frac{BB}{\text{Aktual kelas B } (BA + BB + BC)} \\
&= \frac{60}{50 + 60 + 10} \\
&= \frac{60}{120}
\end{aligned}$$

$$= 0.5$$

Nilai *recall* pada kelas C dapat dihitung dengan membandingkan nilai prediksi yang bernilai benar terhadap kelas C dengan keseluruhan yang secara aktual merupakan kelas C.

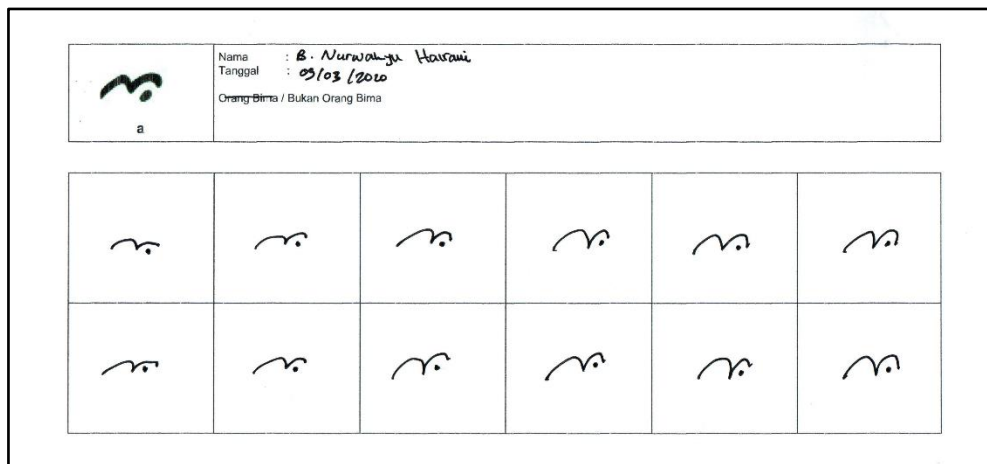
$$\begin{aligned} \text{Recall } C &= \frac{CC}{\text{Aktual kelas } C (CA + CB + CC)} \\ &= \frac{80}{20 + 20 + 80} \\ &= \frac{80}{120} \\ &= 0.67 \end{aligned}$$

BAB IV

HASIL DAN PEMBAHASAN

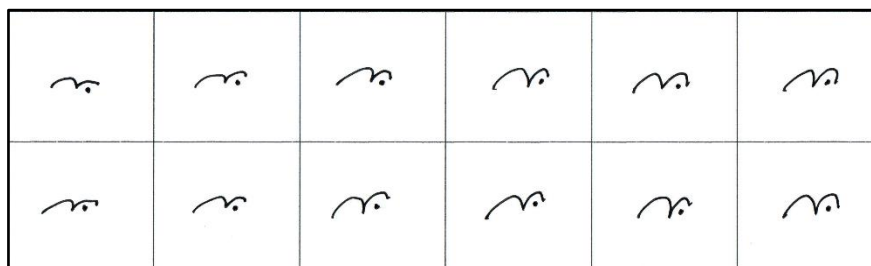
4.1 Akuisisi Data

Pada penelitian ini dilakukan pengumpulan dataset dari 20 orang, namun pada pelaksanaannya hanya didapatkan dataset dari 10 orang saja. Dataset tersebut terdiri dari 2640 citra tulisan tangan karakter Aksara Bima, dimana setiap orang menuliskan 22 jenis karakter Aksara Bima sebanyak 12 kali untuk tiap karakter. Pengambilan dataset tulisan tangan dilakukan menggunakan media kertas berukuran F4 dengan bolpoin gel berukuran 1.0 mm. Adapun cara untuk mengakuisisi citra ke dalam bentuk digital dengan menggunakan *scanner*, dimana citra yang diperoleh memiliki resolusi 600 dpi. Contoh hasil *scan* dari tulisan tangan yang didapatkan dapat dilihat pada Gambar 4.1.



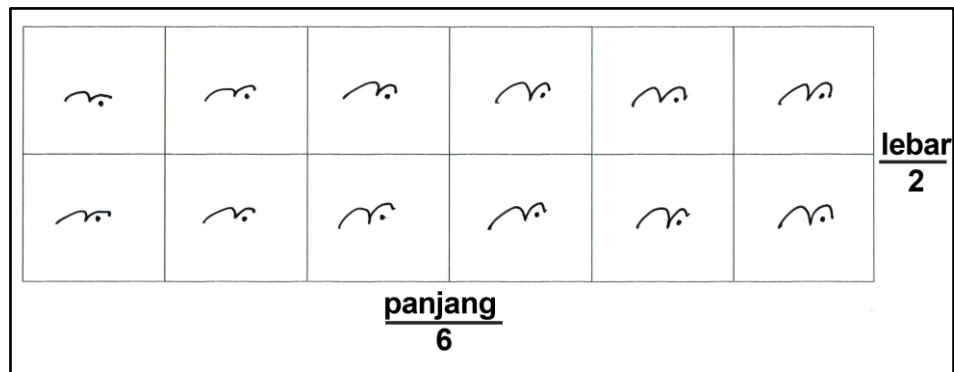
Gambar 4.1 Hasil *scan* form pengambilan dataset

Setelah didapatkan citra utuh melalui proses *scan*, selanjutnya untuk masing-masing citra form dataset, dilakukan proses *cropping* untuk mendapatkan citra masing-masing karakter. Langkah pertama yang dilakukan dalam mempermudah proses *cropping* adalah melakukan *cropping* pada daerah 12 kotak aksara saja sehingga dihasilkan gambar seperti pada Gambar 4.2.



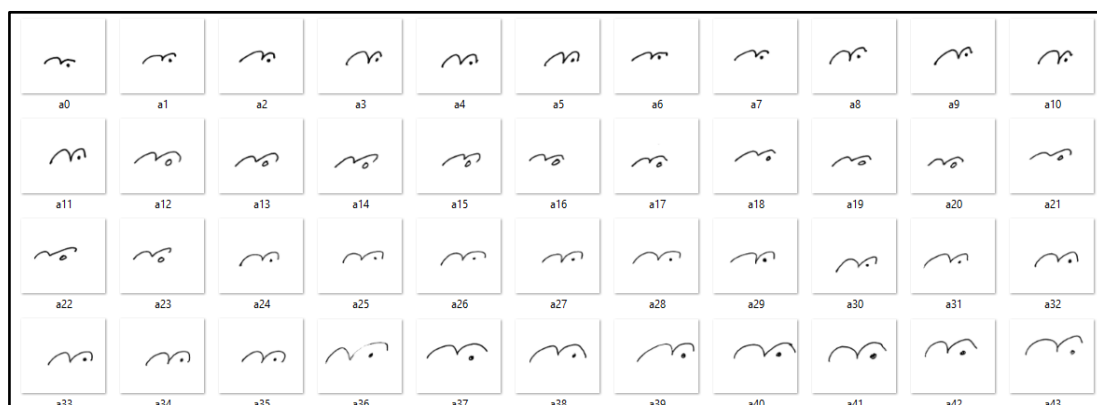
Gambar 4.2 *Cropping* awal dataset

Selanjutnya dilakukan proses *cropping* yang menghasilkan 12 buah citra karakter, dimana proses *cropping* ini dilakukan menggunakan *code* dengan Bahasa pemrograman Python. Langkah yang dilakukan dalam melakukan *cropping* untuk per-kotak karakter adalah mendapatkan ukuran panjang dan lebar, dimana ukuran panjang didapatkan dengan membagi panjang citra untuk 12 kotak karakter dengan pembagi bernilai 6 dan nilai lebar didapatkan dengan membagi lebar citra untuk 12 kotak karakter dengan pembagi bernilai 2 seperti ilustrasi pada Gambar 4.3.



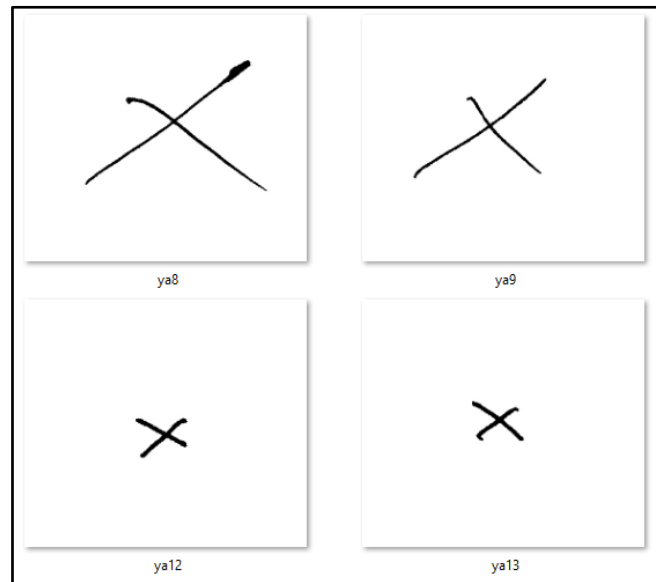
Gambar 4.3 Pembagian panjang dan lebar citra per karakter

Langkah selanjutnya dilakukan proses *slicing* terhadap gambar untuk mendapatkan citra per karakter, proses *slicing* dimulai dari koordinat (0, 0) atau pojok kiri atas citra untuk kotak karakter pertama hingga pojok kanan bawah atau kotak karakter terakhir pada 1 kertas, dengan *slicing* berukuran nilai panjang per kotak yang didapatkan dan lebar per kotak yang didapatkan pada perhitungan sebelumnya pada Gambar 4.3. Pada Gambar 4.3 dapat dilihat terdapat border atau garis yang membatasi tempat penulisan tiap karakter. Untuk menghilangkan border atau garis tersebut dari *cropping*, dilakukan *slicing* kembali dengan *padding* nilai 10 untuk masing-masing citra sehingga dihasilkan sekumpulan citra per karakter seperti pada Gambar 4.4.



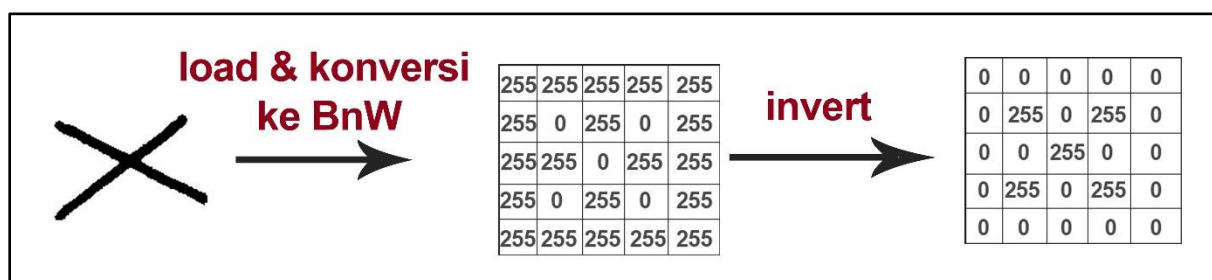
Gambar 4.4 Hasil *cropping* gambar

Proses *cropping* dilanjutkan dengan melakukan *cropping area Region of Interest* (ROI) untuk tiap tulisan agar didapatkan standarisasi ukuran tulisan yang sama. Hal tersebut perlu dilakukan karena terdapat beberapa tulisan yang memiliki ukuran tulisan yang berbeda pada karakter “ya” saat ditulis seperti pada ilustrasi di Gambar 4.5. Terdapat 4 karakter yang sama dengan ukuran penulisan yang berbeda. Perlu dilakukan proses *cropping area* tulisannya saja dengan mencari ROI dari masing-masing karakter.



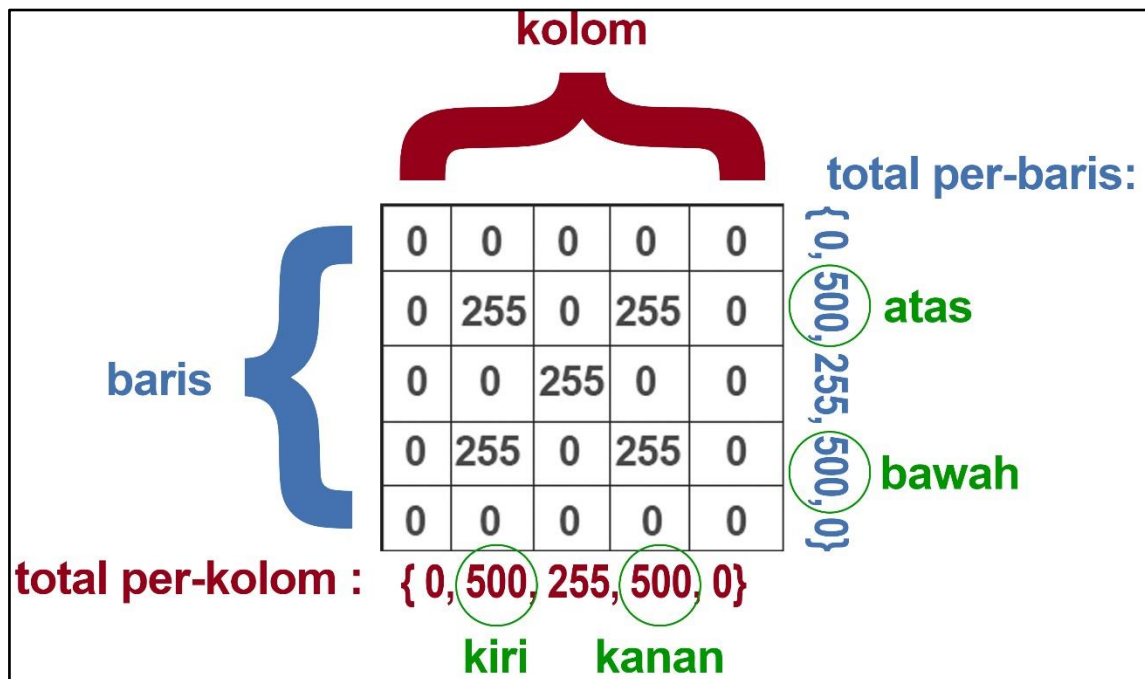
Gambar 4.5 Hasil *cropping* sebelum dicari area ROI

Proses pencarian ROI dan *cropping* dilakukan dengan menggunakan *code*. Tahapan yang dilakukan dalam pencarian ROI dilakukan dengan memuat gambar dan dikonversi ke dalam ruang warna *Black and White*. Adapun konversi ke ruang warna *Black and White* untuk memudahkan mencari area di mana letak awal dan akhir tulisan secara vertikal dan horizontal. Kemudian dilakukan proses *invert*, dimana nilai piksel 0 yang menandakan warna hitam ditukar dengan nilai piksel 255 yang menandakan warna putih, dan sebaliknya nilai 255 ditukar menjadi nilai 0 untuk memudahkan proses pencarian ROI seperti pada ilustrasi di Gambar 4.6.



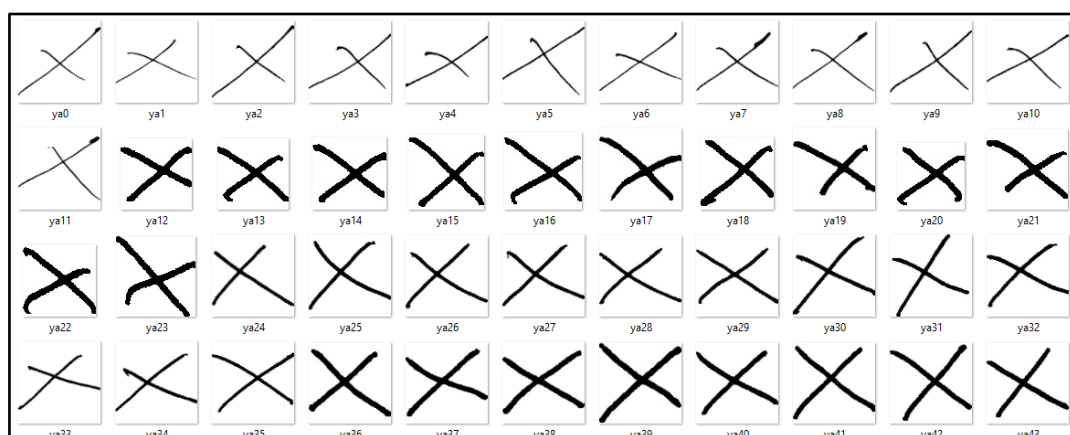
Gambar 4.6 Hasil *invert* citra asli

Langkah selanjutnya dalam mencari ROI adalah mencari nilai total piksel pada baris awal yang bernilai lebih dari nol sebagai batas atas dari ROI dan nilai baris akhir yang bernilai lebih dari nol sebagai batas bawah dari ROI. Kemudian dilakukan pencarian nilai total piksel kolom awal yang bernilai lebih dari nol sebagai batas kiri dari ROI dan nilai kolom akhir yang bernilai lebih dari nol sebagai batas kanan dari ROI seperti ilustrasi pada Gambar 4.7.



Gambar 4.7 Ilustrasi pencarian batas ROI

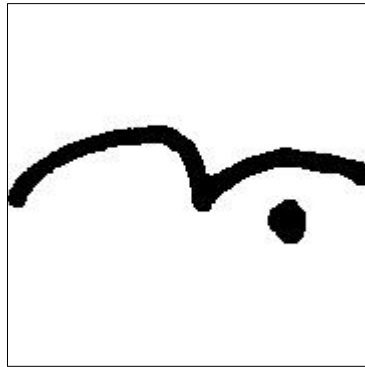
Setelah didapatkan posisi baris atau piksel yang bernilai lebih dari nol untuk batas atas, bawah, kiri, dan kanan, dilakukan proses *slicing* sesuai dengan batas ROI yang didapat, sehingga dihasilkan sekumpulan citra yang memiliki ukuran yang hampir sama seperti pada Gambar 4.7.



Gambar 4.8 Hasil *crop* berdasarkan ROI

4.2 Data Pre-processing

Pada penelitian ini, tahapan pre-processing yang dilakukan terhadap seluruh data citra yang telah di-*scan* dan di-*crop* adalah *resize* ke dalam ukuran 64x64 piksel. Adapun tahap selanjutnya yang dilakukan setelah proses *cropping* ialah konversi ruang warna menjadi ruang warna *greyscale* untuk mempermudah proses dalam tahapan ekstraksi ciri dengan GLCM yang memerlukan citra *greyscale*. Proses re-size dan konversi ruang warna dilakukan sebelum proses ekstraksi ciri dilakukan. Pada penelitian ini digunakan *library* OpenCV pada Python untuk memuat citra, re-size, dan konversi ruang warna.



Gambar 4.9 Citra hasil re-size

4.3 Ekstraksi Ciri

Pada penelitian ini dilakukan ekstraksi ciri GLCM, dimana terdapat tiga buah ciri yang digunakan, antara lain *dissimilarity*, *homogeneity*, dan *entropy*. Ekstraksi ciri dilakukan dengan menggunakan *library* Skimage pada Python, dimana beberapa fungsi yang digunakan adalah *greycomatrix* untuk menghasilkan matriks kookurensi, *greycoprops* untuk mendapatkan nilai ciri *dissimilarity* dan *homogeneity*, dan *shannon_entropy* untuk mendapatkan nilai ciri *entropy*. Proses ekstraksi ciri dilakukan untuk semua citra, dimana terdapat 3 buah tahapan yang dilakukan dalam melakukan ekstraksi ciri. Tahap pertama yaitu ekstraksi nilai ciri citra secara utuh, tahap kedua ekstraksi ciri dengan *zoning* 2x2, dan tahap ketiga yaitu ekstraksi ciri dengan *zoning* 4x4. Pada masing-masing tahapan ekstraksi ciri dilakukan ekstraksi ciri dengan beberapa kombinasi jarak dan sudut. Kombinasi jarak yang digunakan yaitu jarak 1, 2, 3, 4, dan 5. Kombinasi sudut yang digunakan yaitu sudut 0, 45, 90, dan 135. Setiap hasil kombinasi ekstraksi ciri disimpan dalam *file* dengan ekstensi *.csv* untuk mempermudah dalam tahapan selanjutnya.

4.4 Mekanisme Pengujian

Pada penelitian ini terdapat beberapa skenario pengujian yang dilakukan, antara lain sebagai berikut.

1. *Smoothing parameter* atau standar deviasi yang digunakan dalam model PNN. Nilai *smoothing parameter* bergantung pada *range input* fitur. Misalkan nilai fitur berada di antara 0-20, maka nilai *smoothing parameter* yang digunakan berada di antara 0-20, seperti 10 atau 15.
2. Sudut GLCM, dimana pada penelitian yang dilakukan digunakan beberapa sudut dalam perhitungan ekstraksi ciri dalam GLCM di tahap pembuatan matriks kookurensi. Nilai sudut yang digunakan antara lain 0, 45, 90, dan 135.
3. Penggunaan *Zoning*, dimana pada tahapan ini akan dilakukan pengujian dengan menggunakan *zoning* dan tidak menggunakan *zoning*, dimana *zoning* dibagi menjadi 4 daerah dan 16 daerah.
4. Pembagian data, dimana pada tahapan ini *dataset* akan dibagi menjadi beberapa kombinasi pembagian data *train* dan data *test*. Kombinasi yang digunakan bervariasi mulai dari perbandingan data *train:test* sebesar 9:1 hingga 1:9.

Skenario pengujian dijalankan dari skenario pertama hingga skenario terakhir dengan menggunakan Jupyter Notebook, dimana parameter terbaik pada skenario pertama akan digunakan dalam skenario kedua, kemudian parameter terbaik pada skenario kedua akan digunakan dalam skenario ketiga, dan parameter terbaik pada skenario ketiga akan digunakan dalam skenario keempat. Adapun hasil terbaik pada masing-masing skenario mengacu pada nilai akurasi, presisi, dan *recall* terbaik.

Untuk menghindari bias dalam pembagian data, maka pada setiap pengujian skenario digunakan *K-Fold Cross Validation*, dimana pada skenario pertama hingga skenario ketiga digunakan nilai K atau nilai *fold* sebanyak 10, pembagian dilakukan dengan membagi data *train:test* dengan perbandingan 9:1. Pada pengujian skenario pertama kombinasi sudut dan jarak GLCM yang digunakan yaitu sudut 0 dengan jarak 1. Pada penelitian ini *library* untuk klasifikasi yang digunakan adalah *library* NeuPy pada Python. Adapun terdapat fungsi yang dikembangkan oleh penulis yang dapat digunakan dalam melakukan klasifikasi dengan PNN.

Adapun sebelum dilakukan klasifikasi, pada setiap skenario pengujian dilakukan normalisasi terhadap nilai *input* dengan rentang nilai 0 - 1. Normalisasi tersebut dilakukan

karena berdasarkan dokumentasi NeuPy, data harus dinormalisasi untuk mendapatkan hasil terbaik.

4.5 Hasil Pengujian

4.5.1 Pengujian *Smoothing Parameter*

Pada pengujian skenario *smoothing parameter*, variasi nilai *smoothing parameter* yang diujikan yaitu rentang nilai 0.1 – 0.9. Adapun pertimbangan penggunaan nilai 0.1 – 0.9 dikarenakan nilai fitur berada pada rentang 0 – 1 setelah dilakukan proses normalisasi. Berdasarkan dokumentasi dari *library* NeuPy, untuk mendapatkan hasil terbaik diperlukan proses normalisasi nilai *input*, dimana pada penelitian ini nilai *input* berupa nilai ciri GLCM. Kemudian untuk nilai *smoothing parameter* yang disarankan pada dokumentasi NeuPy yaitu sesuai dengan rentang nilai dari *input*, oleh karena itu nilai 0.1 – 0.9 dipilih dalam pengujian nilai *smoothing parameter*.

Adapun pengujian dilakukan dengan menggunakan K-fold *Cross Validation* dengan nilai K atau nilai *fold* senilai 10 dengan perbandingan data *train:test* sebesar 9:1. Hasil dari pengujian variasi *smoothing parameter* antara lain sebagai berikut.

Tabel 4.1 Hasil pengujian *smoothing parameter*

| <i>Smoothing Parameter</i> | Akurasi | Presisi | <i>Recall</i> |
|----------------------------|---------------------|---------------------|---------------------|
| 0.1 | 0.21940603756193106 | 0.21940603756193106 | 0.21940603756193106 |
| 0.2 | 0.209556112455352 | 0.209556112455352 | 0.209556112455352 |
| 0.3 | 0.2046318700311096 | 0.2046318700311096 | 0.2046318700311096 |
| 0.4 | 0.2050092176518032 | 0.2050092176518032 | 0.2050092176518032 |
| 0.5 | 0.20576679340937892 | 0.20576679340937892 | 0.20576679340937892 |
| 0.6 | 0.20501065790989745 | 0.20501065790989745 | 0.20501065790989745 |
| 0.7 | 0.2050120981679917 | 0.2050120981679917 | 0.2050120981679917 |
| 0.8 | 0.2053908860467796 | 0.2053908860467796 | 0.2053908860467796 |
| 0.9 | 0.20614846180435534 | 0.20614846180435534 | 0.20614846180435534 |

Berdasarkan pengujian nilai *smoothing parameter*, diperoleh nilai akurasi tertinggi pada nilai *smoothing parameter* bernilai 0.1 dengan akurasi sebesar 0.21940603756193106. Kenaikan nilai *smoothing parameter* tidak memiliki hubungan dengan kenaikan nilai akurasi, dari hasil pengujian dapat dilihat bahwa nilai akurasi bervariasi. Nilai *smoothing parameter* 0.1 digunakan dalam skenario pengujian berikutnya.

4.5.2 Pengujian Kombinasi Sudut dan Jarak GLCM

Pada pengujian skenario kombinasi sudut dan jarak GLCM, terdapat beberapa variasi sudut dan jarak yang digunakan, antara lain sudut 0, 45, 90, dan 135. Kemudian untuk jarak digunakan variasi jarak 1, 2, 3, 4, dan 5. Pada pengujian skenario sudut dan jarak GLCM, digunakan data pada hasil ekstraksi ciri yang sebelumnya telah dilakukan dan disimpan dalam *file* dengan ekstensi *.csv*. Seluruh *file* dengan kombinasi sudut dan jarak akan dimuat dengan menggunakan *library* Pandas pada Python, kemudian pada masing-masing nilai ciri akan dilakukan normalisasi dengan rentang nilai 0 – 1.

Adapun pengujian dilakukan dengan menggunakan K-fold *Cross Validation* dengan nilai K atau nilai *fold* senilai 10 dengan perbandingan data *train:test* sebesar 9:1. Hasil dari pengujian variasi kombinasi sudut dan jarak GLCM antara lain sebagai berikut.

Tabel 4.2 Hasil pengujian kombinasi sudut dan jarak GLCM

| Jarak | Sudut | Akurasi | Presisi | <i>Recall</i> |
|-------|-------|---------------------|---------------------|---------------------|
| 1 | 0 | 0.21940603756193106 | 0.21940603756193106 | 0.21940603756193106 |
| | 45 | 0.20499913584514343 | 0.20499913584514343 | 0.20499913584514343 |
| | 90 | 0.1477892038253255 | 0.1477892038253255 | 0.1477892038253255 |
| | 135 | 0.17696595229865192 | 0.17696595229865192 | 0.17696595229865192 |
| 2 | 0 | 0.20347534278142637 | 0.20347534278142637 | 0.20347534278142637 |
| | 45 | 0.1811340592234128 | 0.1811340592234128 | 0.1811340592234128 |
| | 90 | 0.13604389906671277 | 0.13604389906671277 | 0.13604389906671277 |
| | 135 | 0.17886709298306255 | 0.17886709298306255 | 0.17886709298306255 |
| 3 | 0 | 0.19817663325267887 | 0.19817663325267887 | 0.19817663325267887 |
| | 45 | 0.16447027307293466 | 0.16447027307293466 | 0.16447027307293466 |
| | 90 | 0.13527624150247725 | 0.13527624150247725 | 0.13527624150247725 |
| | 135 | 0.1735424588086185 | 0.1735424588086185 | 0.1735424588086185 |
| 4 | 0 | 0.13641548565502937 | 0.13641548565502937 | 0.13641548565502937 |
| | 45 | 0.11938011291623458 | 0.11938011291623458 | 0.11938011291623458 |
| | 90 | 0.122774801244383 | 0.122774801244383 | 0.122774801244383 |
| | 135 | 0.14892844797787763 | 0.14892844797787763 | 0.14892844797787763 |
| 5 | 0 | 0.1299746514575412 | 0.1299746514575412 | 0.1299746514575412 |
| | 45 | 0.11329646272612053 | 0.11329646272612053 | 0.11329646272612053 |
| | 90 | 0.13603957829243 | 0.13603957829243 | 0.13603957829243 |

| | | | | |
|--|-----|---------------------|---------------------|---------------------|
| | 135 | 0.14248905403848372 | 0.14248905403848372 | 0.14248905403848372 |
|--|-----|---------------------|---------------------|---------------------|

Berdasarkan pengujian kombinasi sudut dan jarak GLCM, diperoleh nilai akurasi tertinggi pada kombinasi jarak 1 dan sudut 0 dengan akurasi sebesar 0.21940603756193106. Kombinasi jarak 1 dan sudut 0 digunakan dalam pengujian skenario berikutnya.

4.5.3 Pengujian Zoning

Pada pengujian skenario *zoning*, terdapat dua variasi *zoning* yang digunakan, yaitu ukuran 2x2 dan ukuran 4x4. Pada pengujian *zoning* digunakan data pada hasil ekstraksi ciri yang sebelumnya telah dilakukan dan disimpan dalam *file* dengan ekstensi *.csv*. *File* dengan kombinasi sudut 0 dan jarak 1 dengan *zoning* 2x2 dan 4x4 akan dimuat dengan menggunakan *library* Pandas pada Python, kemudian pada masing-masing nilai ciri akan dilakukan normalisasi dengan rentang nilai 0 – 1.

Adapun pengujian dilakukan dengan menggunakan K-fold *Cross Validation* dengan nilai K atau nilai *fold* senilai 10 dengan perbandingan data *train:test* sebesar 9:1. Hasil dari pengujian *zoning* antara lain sebagai berikut.

Tabel 4.3 Hasil pengujian *zoning*

| Ukuran <i>Zoning</i> | Akurasi | Presisi | <i>Recall</i> |
|----------------------|--------------------|--------------------|--------------------|
| 2x2 | 0.6100659638207168 | 0.6100659638207168 | 0.6100659638207168 |
| 4x4 | 0.808274282751469 | 0.808274282751469 | 0.808274282751469 |

Berdasarkan pengujian *zoning*, diperoleh nilai akurasi tertinggi pada ukuran *zoning* 4x4 dengan akurasi sebesar 0.808274282751469. Berdasarkan hasil pengujian, *zoning* memberikan kenaikan akurasi seperti pada penelitian yang sebelumnya dilakukan oleh Riska Yulianti pada pengenalan pola Aksara Sasak [16]. Pada hasil yang didapatkan dapat dilihat bahwa semakin banyak daerah *zoning* nilai akurasi menjadi semakin tinggi. Ukuran *zoning* 4x4 akan digunakan dalam pengujian skenario berikutnya.

4.5.4 Pengujian Pembagian Data

Pada pengujian pembagian data, dilakukan pengujian terhadap berbagai variasi pembagian data. Berdasarkan pengujian yang telah dilakukan sebelumnya telah diperoleh hasil terbaik dari masing-masing skenario dengan nilai *smoothing parameter* 0.1 dengan ukuran *zoning* 4x4, serta jarak 1 dan sudut 0 pada ciri GLCM. Data dimuat dengan menggunakan *library* Pandas pada Python, kemudian pada masing-masing nilai ciri akan dilakukan normalisasi dengan rentang nilai 0 – 1.

Adapun pada pengujian pembagian data dilakukan dengan membagi data dengan menggunakan K-Fold *Cross Validation*. Nilai K atau nilai *fold* menjadi parameter pengujian dengan variasi nilai K yaitu 10, 8, 6, 5, 4, 3, dan 2. Pada masing-masing variasi nilai K akan diambil 1 buah *fold* yang menjadi data *train* atau data *test*, sehingga pada masing-masing nilai K didapatkan 2 buah rasio pembagian data. Hasil dari pengujian pembagian data antara lain sebagai berikut.

Tabel 4.4 Hasil pengujian pembagian data

| K | Rasio | Akurasi | Presisi | <i>Recall</i> |
|----|-------|------------|------------|---------------|
| 10 | 9:1 | 0.80827428 | 0.80827428 | 0.80827428 |
| | 1:9 | 0.6626241 | 0.6626241 | 0.6626241 |
| 8 | 7:1 | 0.81281431 | 0.81281431 | 0.81281431 |
| | 1:7 | 0.68754291 | 0.68754291 | 0.68754291 |
| 6 | 5:1 | 0.80750414 | 0.80750414 | 0.80750414 |
| | 1:5 | 0.70844981 | 0.70844981 | 0.70844981 |
| 5 | 4:1 | 0.80902119 | 0.80902119 | 0.80902119 |
| | 1:4 | 0.71996881 | 0.71996881 | 0.71996881 |
| 4 | 3:1 | 0.81356451 | 0.81356451 | 0.81356451 |
| | 1:3 | 0.74409462 | 0.74409462 | 0.74409462 |
| 3 | 2:1 | 0.79916658 | 0.79916658 | 0.79916658 |
| | 1:2 | 0.75369528 | 0.75369528 | 0.75369528 |
| 2 | 1:1 | 0.78173978 | 0.78173978 | 0.78173978 |
| | 1:1 | 0.78173978 | 0.78173978 | 0.78173978 |

Berdasarkan pengujian pembagian data, diperoleh nilai akurasi tertinggi pada ukuran nilai K = 4, dimana rasio pembagian data *train:test* sebesar 3:1. Nilai akurasi terendah terletak pada nilai K = 10, dimana rasio data *train:test* sebesar 1:9 dengan nilai akurasi 0.6626241. Pada nilai K = 2 dengan rasio *train:test* sebesar 1:1 diperoleh akurasi sebesar 0.78173978. Pada hasil yang didapatkan dapat dilihat bahwa jumlah data *train* yang lebih besar belum tentu memberikan hasil akurasi yang lebih baik. Bahkan dengan data *train* yang lebih sedikit pun model PNN dapat memberikan performa yang lebih baik.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, terdapat beberapa hal yang dapat disimpulkan oleh penulis, antara lain sebagai berikut.

1. Model *machine learning* yang telah dibuat memiliki *input* berupa ciri GLCM dari citra *grayscale* dengan ukuran 64x64 piksel, dimana diperoleh skenario terbaik dengan kombinasi sudut 0 dan jarak 1 dengan pembagian data *train:test* sebesar 3:1 dengan memanfaatkan *zoning* 4x4.
2. GLCM dapat menjadi *input* ciri dalam melakukan klasifikasi tulisan tangan Aksara Bima dengan klasifikasi PNN, dimana diperlukan jumlah *ciri* yang lebih banyak untuk performa yang lebih baik.
3. *Zoning* mampu meningkatkan performa akurasi model dari 21.9% menjadi 61% dengan ukuran *zoning* 2x2 dan 80.82% dengan ukuran *zoning* 4x4.
4. PNN dapat bekerja dengan pembagian data yang sedikit, dengan menggunakan rasio data *train:test* 3:1 diperoleh akurasi sebesar 81.35%.

5.2 Saran

Penelitian yang telah dilakukan masih jauh dari kata sempurna, adapun terdapat beberapa saran yang dapat penulis berikan untuk pengembangan terhadap penelitian ini, antara lain sebagai berikut.

1. Penambahan metode *pre-processing* untuk memperoleh data yang lebih baik dan lebih terstandarisasi.
2. Penambahan ciri GLCM lainnya sebagai *input* pada model PNN.
3. Perbanyak data tulisan tangan Aksara Bima dengan varian yang lebih beragam.
4. Peningkatan ukuran *zoning* agar diperoleh ciri yang lebih rinci pada masing-masing sub-bagian citra.
5. Penerapan *Dimensionality Reduction* untuk ukuran *zoning* yang lebih besar agar beban komputasi menjadi lebih ringan

DAFTAR PUSTAKA

- [1] S. Abubakar, "Aksara Bima Usaha Menemukanali Dan Mengembangkannya," *STIT Sunan Giri Bima*, pp. 1–30, 2018.
- [2] 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, doi: 10.29303/jtika.v2i1.96.
- [3] E. Hara, "Sistem Pengenalan Tulisan Tangan Aksara Lampung Dengan Metode Deteksi Tepi (Canny) Berbasis Jaringan Syaraf Tiruan Backpropagation," *Electr. J. Rekayasa dan Teknol. Elektro*, vol. 10, no. 3, pp. 1–86, 2016, doi: <https://doi.org/10.23960/elc.v10n3.2004>.
- [4] P. L. K. Nisa, "Game Aplikasi Pengenalan Aksara Jawa ' HANACARAKA ' Berbasis Android," *SENASIF*, vol. 1, no. 1, pp. 756–765, 2017.
- [5] K. T. Y. Aditya, M. W. A. Kesiman, and G. A. Pradnyana, "Pengembangan Game Edukasi Tematik Aksara dan Bahasa Bali," *Kumpul. Artik. Mhs. Pendidik. Tek. Inform.*, vol. 8, no. 3, pp. 522–533, 2019.
- [6] R. R. Riansyah, Y. I. Nurhasanah, and I. A. Dewi, "Sistem Pengenalan Aksara Sunda Menggunakan Metode Modified Direction Feature dan Learning Vector Quantization," *J. Tek. Inform. dan Sist. Inf.*, vol. 3, no. 1, pp. 17–30, 2017, doi: 10.28932/jutisi.v3i1.563.
- [7] A. Aryantio and R. Munir, "Pengenalan Aksara Lampung Menggunakan Jaringan Syaraf Tiruan," in *Konferensi Nasional Informatika (KNIF)*, 2015, pp. 34–38.
- [8] I. Ayu Dian Purnama Sari, "Pengenalan Aksara Bali Dengan Metode Local Binary Pattern," *e-Proceeding Eng.*, vol. 2, no. 2, pp. 2697–2704, 2015.
- [9] A. H. Nurul, M. D. Sulistiyo, and R. N. Dayawati, "Pengenalan Aksara Jawa Tulisan Tangan Menggunakan Directional Element Feature Dan Multi Class Support Vector Machine," *Konf. Nas. Teknol. Inf. dan Apl.*, vol. 3, no. September, pp. 13–20, 2014, [Online]. Available: <http://seminar.ilkom.unsri.ac.id/index.php/kntia/article/view/733%0Ahttp://seminar.ilkom.unsri.ac.id/index.php/kntia/article/view/733/409>.
- [10] S. S. Sawant and P. S. Topannavar, "Introduction to Probabilistic Neural Network - Used For Image Classifications," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 5, no. 4, pp. 279–283, 2015.
- [11] Vidia, "Pengenalan Tulisan Tangan Bahasa Arab Menggunakan Metode Probabilistic Neural Network," *J. Ilmu Komput. dan Desain Komun. Vis.*, vol. 4, no. 1, pp. 28–35, 2019.
- [12] V. V. Kumari, "Ship ISAR Image Classification with Probabilistic Neural Network," *Adv. Comput. Sci. Technol.*, vol. 10, no. 4, pp. 537–548, 2017.
- [13] A. K. Patel and S. Chatterjee, "Computer vision-based limestone rock-type

- classification using probabilistic neural network,” *Geosci. Front.*, vol. 7, no. 1, pp. 53–60, 2016, doi: 10.1016/j.gsf.2014.10.005.
- [14] I. Amalia, “Pengenalan Citra Tanda Tangan Menggunakan Gray Level Co-Occurrence Matrix (GLCM) Dan Probabilistic Neural Network (PNN),” *E-Jurnal Politek. negeri Lhokseumawe*, vol. 14, pp. 29–34, 2014.
 - [15] M. M. B, Andriana, and A. S. Hidayat, “Implementasi Algoritma GLCM Dan MED pada,” *MIND J.*, vol. 2, no. 2, pp. 23–41, 2017.
 - [16] R. Yulianti, I. G. P. S. Wijaya, and F. Bimantoro, “Pengenalan Pola Tulisan Tangan Suku Kata Aksara Sasak Menggunakan Metode Moment Invariant dan Support Vector Machine,” *J. Comput. Sci. Informatics Eng.*, vol. 3, no. 2, pp. 91–98, 2019, doi: 10.29303/jcosine.v3i2.181.
 - [17] R. Satria and V. Suhartono, “Pemilihan Parameter Smoothing pada Probabilistic Neural Network dengan Menggunakan Particle Swarm Optimization untuk Pendeteksian Teks Pada Citra,” *J. Intell. Syst.*, vol. 1, no. 1, pp. 22–26, 2015.
 - [18] 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, doi: 10.1109/ISITIA.2018.8710976.
 - [19] J. A. Gutiérrez-Gnecchia *et al.*, “DSP-based arrhythmia classification using wavelet transform and probabilistic neural network,” *Biomed. Signal Process. Control*, vol. 32, pp. 44–56, 2017, doi: 10.1016/j.bspc.2016.10.005.
 - [20] N. Varuna Shree and T. N. R. Kumar, “Identification and classification of brain tumor MRI images with feature extraction using DWT and probabilistic neural network,” *Brain Informatics*, vol. 5, no. 1, pp. 23–30, 2018, doi: 10.1007/s40708-017-0075-5.
 - [21] P. Nivetha, “Construction of predictive modelling for cardiac patient using probabilistic neural network,” *Int. J. Adv. Res. Ideas Innov. Technol.*, vol. 5, no. 2, pp. 874–877, 2019.
 - [22] K. Wrobel, R. Doroz, P. Porwik, J. Naruniec, and M. Kowalski, “Using a Probabilistic Neural Network for lip-based biometric verification,” *Eng. Appl. Artif. Intell.*, vol. 64, no. 1, pp. 112–127, 2017, doi: 10.1016/j.engappai.2017.06.003.
 - [23] J. M. González-Camacho, J. Crossa, P. Pérez-Rodríguez, L. Ornella, and D. Gianola, “Genome-enabled prediction using probabilistic neural network classifiers,” *BMC Genomics*, vol. 17, no. 1, pp. 1–16, 2016, doi: 10.1186/s12864-016-2553-1.
 - [24] Aszaani and Mustakim, “Estimasi Pola Cuaca Wilayah Pekanbaru Menggunakan Probabilistic Neural Network,” in *Seminar Nasional Teknologi Informasi, Komunikasi dan Industri (SNTIKI) 8*, 2016, no. 1, pp. 78–86.
 - [25] A. Cahyaningtyas, P. Sukarno, and M. A. Nugroho, “Deteksi Serangan Denial of Service (DoS) menggunakan Algoritma Probabilistic Neural Network (PNN),”

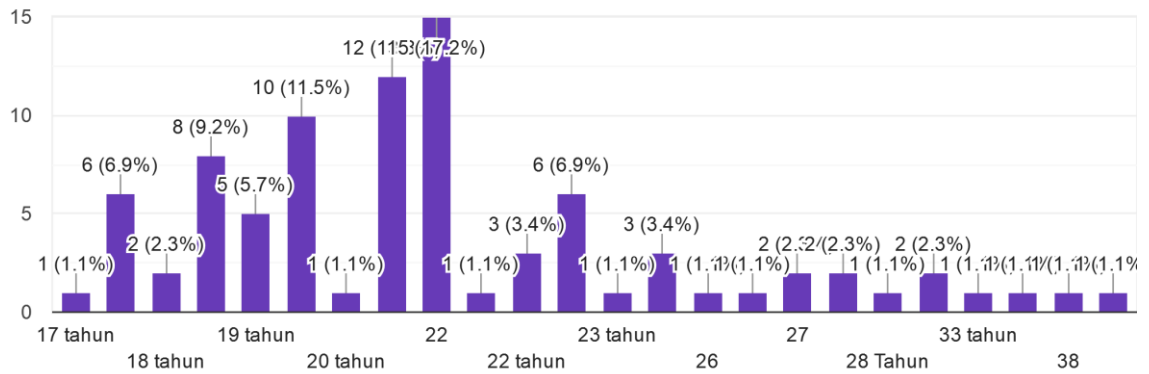
Universitas Telkom, 2019.

- [26] K. Daqrouq, H. Sweidan, A. Balamesh, and M. N. Ajour, "Off-line handwritten signature recognition by wavelet entropy and neural network," *Entropy*, vol. 19, no. 6, 2017, doi: 10.3390/e19060252.
- [27] G. S. Satiabudhi, Gregorius and Adipranata, Rudy and Liliana, and Budhi, "The Use of Probabilistic Neural Network and ID3 Algorithm for Java Character Recognition," *J. Chem. Inf. Model.*, vol. 8, no. 9, pp. 1–58, 2017, doi: 10.1017/CBO9781107415324.004.
- [28] H. Haviluddin, R. Alfred, N. Moham, H. S. Pakpahan, I. Islamiyah, and H. J. Setyadi, "Handwriting Character Recognition using Vector Quantization Technique," *Knowl. Eng. Data Sci.*, vol. 2, no. 2, p. 82, 2019, doi: 10.17977/um018v2i22019p82-89.
- [29] A. Setiawan, A. S. Prabowo, and E. Y. Puspaningrum, "Handwriting Character Recognition Javanese Letters Based on Artificial Neural Network," *Int. J. Comput. Netw. Secur. Inf. Syst. Vol.*, vol. 1, no. 1, pp. 39–42, 2019.
- [30] C. A. Lorentius, R. Adipranata, and A. Tjondrowiguno, "Pengenal Aksara Jawa dengan Menggunakan Metode Convolutional Neural Network," *J. Infra Petra*, vol. 7, no. 1, 2019.
- [31] H. S. M. R. Salahuddin, M. Sulaiman, and S. Abubakar, *Aksara Bima Peradaban Lokal yang Sempat Hilang*, 1st ed. Mataram: Alam Tara Institute, 2013.
- [32] A. Andana, R. Widyati, and M. Irzal, "Pengenal Citra Tulisan Tangan Dengan Metode Backpropagation," *J. Mat. Terap.*, vol. 2, no. 1, pp. 36–44, 2018, [Online]. Available: <http://journal.unj.ac.id/unj/index.php/jmt/article/view/7166>.
- [33] P. Samui, D. T. Bui, S. Chakraborty, and R. C. Deo, *Handbook of Probabilistic Models*. Oxford: Butterworth-Heinemann, 2020.
- [34] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," *ACM Int. Conf. Proceeding Ser.*, vol. 148, pp. 233–240, 2006, doi: 10.1145/1143844.1143874.
- [35] R. Arthana, "Mengenal Accuracy, Precision, Recall dan Specificity serta yang diprioritaskan dalam Machine Learning," *Medium.Com*, 2018. <https://medium.com/@rey1024/mengenal-accuracy-precision-recall-dan-specificity-serta-yang-diprioritaskan-b79ff4d77de8> (accessed Apr. 15, 2020).

LAMPIRAN

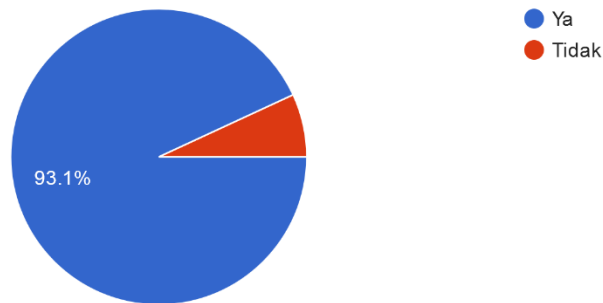
Umur anda saat ini

87 responses



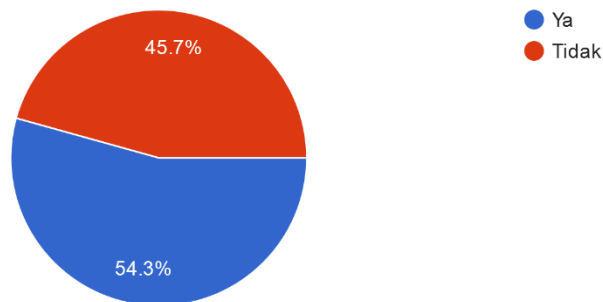
Apakah anda bisa berkomunikasi dengan Bahasa Bima?

87 responses



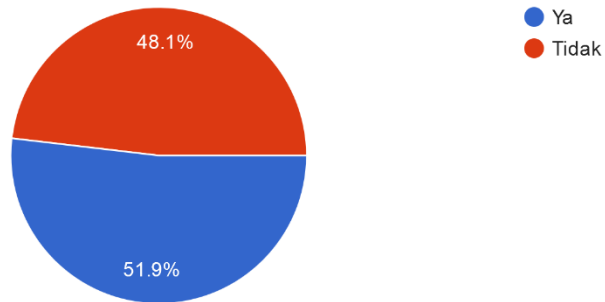
Apakah anda tahu tentang keberadaan Aksara Bima?

81 responses



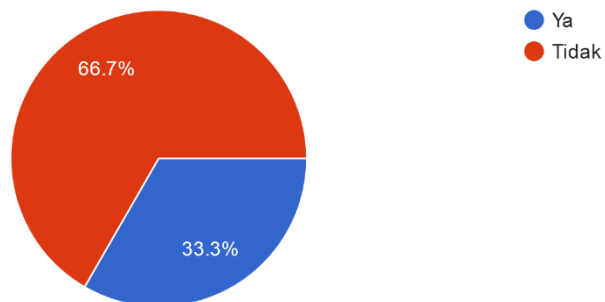
Apakah anda pernah belajar Aksara Bima?

81 responses



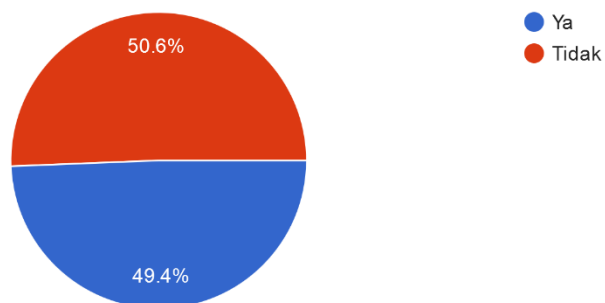
Apakah anda familiar dengan penggunaan Aksara Bima?

81 responses



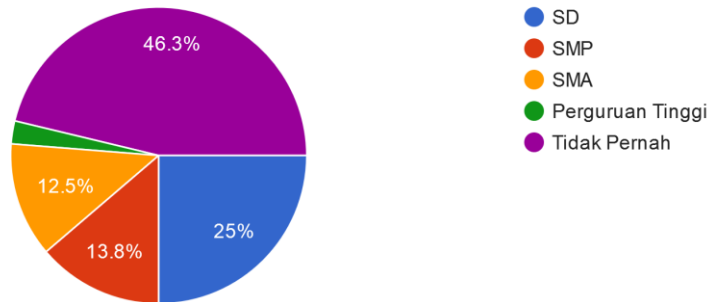
Apakah di sekolah anda pernah diajarkan tentang Aksara Bima?

81 responses



Jika anda pernah belajar Aksara Bima, terakhir kali belajar pada jenjang pendidikan apa?

80 responses



Survey Aksara Bima

Halo, perkenalkan nama saya Muhammad Naufal, mahasiswa yang sedang menempuh Tugas Akhir di Program Studi Teknik Informatika Universitas Mataram. Adapun tujuan saya melakukan survey ini untuk mengetahui bagaimana penggunaan Aksara Bima.

Nama *

Ardiansyah

Umur anda saat ini

28

Apakah anda bisa berkomunikasi dengan Bahasa Bima? *

- ☒ Ya
☐ Tidak

Aksara Bima

Mohon jawab dengan sejujur-jujurnya

Apakah anda tahu tentang keberadaan Aksara Bima?

- ☒ Ya
☐ Tidak

Apakah anda pernah belajar Aksara Bima?

- ☐ Ya
☒ Tidak

Apakah anda familiar dengan penggunaan Aksara Bima?

- ☐ Ya
☒ Tidak

Apakah di sekolah anda pernah diajarkan tentang Aksara Bima?

- ☒ Ya
☐ Tidak

Jika anda pernah belajar Aksara Bima, terakhir kali belajar pada jenjang pendidikan apa?

- ☐ SD
☒ SMP
☐ SMA
☐ Perguruan Tinggi
☐ Tidak Pernah

This content is neither created nor endorsed by Google.

Google Forms

Survey Aksara Bima

Halo, perkenalkan nama saya Muhammad Naufal, mahasiswa yang sedang menempuh Tugas Akhir di Program Studi Teknik Informatika Universitas Mataram. Adapun tujuan saya melakukan survey ini untuk mengetahui bagaimana penggunaan Aksara Bima.

Nama *

Titin Nur Adianti

Umur anda saat ini

23 tahun

Apakah anda bisa berkomunikasi dengan Bahasa Bima? *

- ☒ Ya
☐ Tidak

Aksara Bima

Mohon jawab dengan sejujur-jujurnya

Apakah anda tahu tentang keberadaan Aksara Bima?

- ☒ Ya
☐ Tidak

Apakah anda pernah belajar Aksara Bima?

- ☐ Ya
☒ Tidak

Apakah anda familiar dengan penggunaan Aksara Bima?

- ☒ Ya
☐ Tidak

Apakah di sekolah anda pernah diajarkan tentang Aksara Bima?

- ☐ Ya
☒ Tidak

Jika anda pernah belajar Aksara Bima, terakhir kali belajar pada jenjang pendidikan apa?

- ☐ SD
☐ SMP
☐ SMA
☐ Perguruan Tinggi
☒ Tidak Pernah

This content is neither created nor endorsed by Google.

Google Forms

Survey Aksara Bima

Halo, perkenalkan nama saya Muhammad Naufal, mahasiswa yang sedang menempuh Tugas Akhir di Program Studi Teknik Informatika Universitas Mataram. Adapun tujuan saya melakukan survey ini untuk mengetahui bagaimana penggunaan Aksara Bima.

Nama *

Pandu Hidayat

Umur anda saat ini

20

Apakah anda bisa berkomunikasi dengan Bahasa Bima? *

- ☒ Ya
☐ Tidak

Aksara Bima

Mohon jawab dengan sejujur-jujurnya

Apakah anda tahu tentang keberadaan Aksara Bima?

- ☐ Ya
☒ Tidak

Apakah anda pernah belajar Aksara Bima?

- ☐ Ya
☒ Tidak

Apakah anda familiar dengan penggunaan Aksara Bima?

- ☐ Ya
☒ Tidak

Apakah di sekolah anda pernah diajarkan tentang Aksara Bima?

- ☐ Ya
☒ Tidak

Jika anda pernah belajar Aksara Bima, terakhir kali belajar pada jenjang pendidikan apa?

- ☐ SD
☐ SMP
☐ SMA
☐ Perguruan Tinggi
☒ Tidak Pernah

This content is neither created nor endorsed by Google.

Google Forms



KEMENTERIAN RISTEK DAN PENDIDIKAN TINGGI
UNIVERSITAS MATARAM
FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK INFORMATIKA

Jalan Majapahit No. 62 Mataram 83125 Telepon(0370) 636126 – Fax (0370) 636523

LEMBAR BIMBINGAN TA

NIM : F1D016060

Nama Mahasiswa : Muhammad Naufal

Judul : Pengenalan Pola Tulisan Tangan Aksara Bima dengan
Ekstraksi Ciri GLCM dan *Zoning* & Klasifikasi
Probabilistic Neural Network

Dosen Pembimbing I : Fitri Bimantoro, S.T., M.Kom.

| No. | Tanggal | Uraian | Paraf |
|-----|------------|---|-------|
| 1. | 08/09/2020 | Perbaikan Abstrak Tambahkan Abstrak Bahasa Inggris Perbaikan penulisan | |
| 2. | 14/09/2020 | Siapkan PPT Siapkan video Revisi minor pada paper Revisi minor pada poster | |
| 3. | 15/09/2020 | ACC | |

Mataram, 15 September 2020
Dosen Pembimbing 1,



Fitri Bimantoro, S.T., M.Kom.
NIP. 19860622 201504 1 002



KEMENTERIAN RISTEK DAN PENDIDIKAN TINGGI
UNIVERSITAS MATARAM
FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK INFORMATIKA

Jalan Majapahit No. 62 Mataram 83125 Telepon(0370) 636126 – Fax (0370) 636523

LEMBAR BIMBINGAN TA

NIM : F1D016060

Nama Mahasiswa : Muhammad Naufal

Judul : Pengenalan Pola Tulisan Tangan Aksara Bima dengan Ekstraksi Ciri GLCM dan *Zoning & Klasifikasi Probabilistic Neural Network*

Dosen Pembimbing II : Arik Aranta, S.Kom., M.Kom.

| No. | Tanggal | Uraian | Paraf |
|-----|------------|---|-------|
| 1. | 10/09/2020 | Perbaikan penomoran formula matematis Buat PPT dan video | |
| 2. | 21/09/2020 | ACC | |

Mataram, 21 September 2020
Dosen Pembimbing 2,





Arik Aranta, S.Kom., M.Kom.
NIP. 19940220 201903 1 004