

# PENGENALAN CITRA HURUF HIJIAH MENGGUNAKAN METODE GRAY LEVEL CO-OCCURRENCE MATRICES (GLCM) DENGAN 4 SUDUT ORIENTASI DAN JARINGAN SYARAF TIRUAN BACKPROPAGATION

*(Hijaiiyah Pattern Recognition Using Grey Level Co-Occurrence Matrices (GLCM) Method With 4 Oriented Angle and Backpropagation Artificial Neural Network)*

Muhlis Fathurrahman\*, Ramaditia Dwiyanaputra

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Mataram

Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA

Email: muhlisfathurrahman@gmail.com, rama@unram.ac.id

## Abstract

Arabic is one of the international languages according to the United Nations (UN) which was adopted by General Council resolution 3190 (XXVIII) as the official language and working language of the General Counsel and Main Committees on 18 December 1973. Arabic can be found in the holy book Al - Qur'an. For a Muslim, it is obligatory to learn and master Arabic in order to read and understand the contents of the Al-Qur'an. job applicants from Indonesia also have to learn Arabic. The Hijaiiyah letter has the same role as the alphabet, which is to compose every word and sentence. Humans have a natural intelligence to be able to recognize each Hijaiiyah letter based on the special characteristics or patterns contained in each letter. However, natural intelligence has deficiencies such as inconsistencies in assessing the similarity of each handwritten Hijaiiyah letter from different people. Therefore, this research will develop a system for identifying or recognizing Hijaiiyah handwritten patterns using the Gray Level Co-occurrence Matrices (GLCM) method with 4 orientation angles and Backpropagation Artificial Neural Network (ANN). Data was collected using the Autodesk Sketchbook application so that can reduce the noise. The purpose of this research is to know the level of accuracy and precision of the classification of the Hijaiiyah letter pattern. In this research, the numbers of data were 1500 images of Hijaiiyah letters. The highest accuracy is 45.1111% with a precision of 45.1111%.

**Keywords:** Arabic, Hijaiiyah letters, GLCM, Artificial Neural Networks, Backpropagation

\*Penulis Korespondensi

## 1. PENDAHULUAN

Bahasa Arab merupakan salah satu Bahasa internasional menurut Perserikatan Bangsa Bangsa (PBB) yang ditetapkan melalui resolusi Majelis Umum 3190 (XXVIII) sebagai bahasa resmi dan bahasa kerja Majelis Umum dan Komite – Komite Utama pada tanggal 18 Desember 1973 [1]. Bahasa Arab dapat ditemui di dalam kitab suci Al – Qur'an. Bagi seorang muslim wajib untuk mempelajari dan menguasai Bahasa Arab agar dapat membaca dan memahami isi dari Al – Qur'an.

Dalam mempelajari Bahasa Arab harus mengenali setiap huruf tunggal yang disebut dengan huruf hijaiyah. Huruf hijaiyah memiliki peran yang sama dengan *alphabet* yaitu untuk menyusun setiap kata dan kalimat. Jika tidak mengenali dan/atau memahami setiap huruf dengan baik, maka sulit untuk berkomunikasi dengan menggunakan sebuah Bahasa

tertentu khususnya Bahasa Arab yang memiliki bentuk huruf yang berbeda dengan huruf yang lazim digunakan di Indonesia. Oleh karena itu, mengenali huruf merupakan salah satu faktor utama untuk dapat memahami suatu Bahasa yang baru dipelajari.

Manusia memiliki kecerdasan alami untuk dapat membedakan dan mengenali setiap huruf hijaiyah berdasarkan ciri atau pola khusus yang terdapat pada masing – masing hurufnya. Namun kecerdasan alami memiliki kekurangan seperti inkonsistensi dalam menilai tingkat kemiripan dari setiap huruf hijaiyah. Salah satu solusi untuk dapat mengenali tulisan tangan huruf hijaiyah adalah dengan memanfaatkan pengolahan citra digital. Oleh karena itu penelitian ini akan mengembangkan sistem untuk melakukan identifikasi atau pengenalan pola tulisan tangan huruf hijaiyah.

Pada proses ekstraksi fitur GLCM, parameter ciri yang digunakan yaitu tekstur dari citra huruf hijaiyah. Ekstraksi GLCM menghasilkan fitur antara lain energi, kontras, homogenitas, entropi dan korelasi dalam 4 orientasi sudut ( $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  dan  $135^\circ$ ) [2]. Metode Jaringan Syaraf Tiruan (JST) *Backpropagation* merupakan metode yang sangat baik dalam menangani masalah pengenalan pola yang kompleks, seperti pengoperasian data, peramalan (prediksi) dan pengenalan pola [3].

Berdasarkan uraian di atas penulis mengajukan sebuah penelitian untuk merancang sebuah model pembelajaran mesin (*machine learning*) untuk mengklasifikasikan pola huruf hijaiyah menggunakan metode GLCM dan JST *Backpropagation* sebagai media pengenalan huruf Arab (hijaiyah). Tingkat akurasi dan presisi dari hasil klasifikasi adalah parameter hasil uji atas metode JST *Backpropagation*.

## 2. TINJAUAN PUSTAKA

Penelitian mengenai pengolahan citra digital (*digital image processing*) sudah pernah dilakukan oleh beberapa peneliti. Penelitian – penelitian sebelumnya akan dijadikan sebagai rujukan ketika penelitian ini dilaksanakan.

Penelitian – penelitian tersebut antara lain tentang pengenalan pola huruf hijaiyah tulisan tangan menggunakan logika *fuzzy* dengan JST *backpropagation* memiliki akurasi 69% dengan menggunakan 420 data set [4], parameter *value* setiap huruf dan matriks *interrelationship*, tidak cukup mewakili ciri identik dari setiap huruf dengan baik. Penelitian berikutnya tentang identifikasi tulisan tangan huruf hijaiyah dengan metode *Euclidian Distance* menggunakan 84 data set memiliki akurasi sebesar 84.52% [5]. Penelitian berikutnya tentang pengenalan tulisan tangan bahasa arab menggunakan metode *Probabilistic Neural Network* menggunakan 100 data set dengan tingkat akurasi 90.15%, namun sistem hanya dapat mengenali 10 dari 28 huruf hijaiyah [6]. penelitian lain tentang pengenalan angka arab timur tulisan tangan menggunakan *Zone Centroid Zone* (ZCZ) dan *Backpropagation* menggunakan 500 data set memiliki akurasi sebesar 86% [7]. Penelitian berikutnya tentang identifikasi tulisan Arab dengan menggunakan GLCM dan RNN menggunakan 1219 data set dengan tingkat akurasi 78.75% [8].

Penelitian terkait lain tentang identifikasi huruf hijaiyah berbasis GLCM menggunakan jaringan syaraf tiruan *backpropagation* menggunakan 600 data set dengan tingkat akurasi 96.11% [3], penelitian menggunakan 5 fitur yaitu entropi, energi, korelasi,

kontras dan homogenitas dengan sudut orientasi  $0^\circ$ . Perpaduan metode ekstraksi fitur GLCM dan metode klasifikasi *backpropagation* mampu mengenali citra huruf hijaiyah dengan sangat baik. Penelitian lain tentang pengenalan tulisan tangan huruf hijaiyah sambung menggunakan algoritme *template matching correlation* menggunakan 84 data set dengan tingkat akurasi sebesar 56.41% [9]. Penelitian lain tentang implementasi metode *k-nearest neighbor* pada pengenalan pola tekstur citra saliva untuk deteksi ovulasi menggunakan 45 data set dengan tingkat akurasi 93.3% [10], implementasi pengolahan citra dan klasifikasi *k-nearest neighbor* untuk membangun aplikasi pembeda daging sapi dan babi menggunakan 60 data set dengan tingkat akurasi 93.33% [11] dan identifikasi tumbuhan obat herbal berdasarkan citra daun menggunakan algoritme GLCM dan *k-nearest neighbor* menggunakan 90 data set dengan tingkat akurasi 83.33% [12], namun ke tiga penelitian terakhir belum teruji untuk data set dengan jumlah yang banyak.

Berdasarkan referensi yang didapatkan oleh penulis tersebut, penggunaan metode ekstraksi fitur GLCM dengan 5 fitur dan 4 sudut orientasi mampu menghasilkan akurasi yang tinggi [11], sehingga metode GLCM akan digunakan sebagai metode ekstraksi fitur pada penelitian ini. Metode JST *Backpropagation* mampu melakukan klasifikasi dengan tingkat akurasi yang tinggi saat dipasangkan dengan metode ekstraksi fitur GLCM. Penelitian ini akan menggunakan metode ekstraksi fitur GLCM dan metode klasifikasi JST *Backpropagation* dengan *preprocessing* data menggunakan pola tulisan tangan pada layar *smart phone* (*Autodesk SketchBook*) diharapkan dapat melakukan klasifikasi dengan tingkat akurasi dan presisi yang tinggi. Sehingga penulis bermaksud menggunakan metode tersebut untuk melakukan klasifikasi pola huruf hijaiyah.

## 3. METODE PENELITIAN

### 3.1. Data

Pada penelitian ini terdapat beberapa peralatan pendukung yang digunakan antara lain:

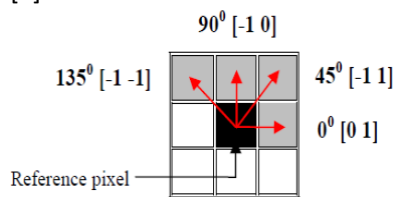
- Matlab R2017a.
- Microsoft Office 2019.
- Microsoft Visio 2019.
- Autodesk Sketchbook*.

Data yang digunakan pada penelitian ini adalah citra huruf hijaiyah. Pengumpulan data dilakukan dengan cara penulisan oleh responden dengan menggunakan aplikasi *Autodesk Sketchbook* pada *smart phone Android*. Pada *Autodesk Sketchbook* jenis

pen yang digunakan adalah *Inking Pen* dengan tinta berwarna hitam, ukuran 12 dan tingkat kegelapan 100%. Data citra yang telah didapatkan dari responden disimpan dalam format ekstensi *file* (.PNG). Dalam penelitian ini, responden yang dimaksud merupakan pelajar atau mahasiswa yang berusia lebih dari 12 tahun atau minimal telah berada pada jenjang pendidikan SMP sederajat serta cakap dalam menggunakan *smart phone Android*. Pengambilan data dengan cara tersebut dimaksudkan agar dapat mengurangi *noise* pada citra huruf hijaiyah.

**3.2. Gray Level Co-occurrence Matrices (GLCM)**

Matriks *co-occurrence* adalah salah satu metode statistik yang dapat digunakan untuk analisis tekstur. Matriks *co-occurrence* dibentuk dari suatu citra dengan melihat pada piksel – piksel yang berpasangan yang memiliki intensitas tertentu. Penggunaan metode ini berdasar pada hipotesis bahwa dalam suatu tekstur akan terjadi perulangan konfigurasi atau pasangan aras keabuan. Misal, *d* didefinisikan sebagai jarak antara dua posisi piksel, yaitu  $(x_1, y_1)$ ,  $(x_2, y_2)$  dan  $\theta$  didefinisikan sebagai sudut diantara keduanya. Maka matriks *co-occurrence* didefinisikan sebagai matriks yang menyatakan distribusi spasial antara dua piksel yang bertetangga yang memiliki intensitas *i* dan *j*, yang memiliki jarak *d* di antara keduanya, dan sudut  $\theta$  diantara keduanya. Matriks *co-occurrence* dinyatakan dengan  $P_{ij}$ . Suatu piksel yang bertetangga yang memiliki jarak *d* di antara keduanya, dapat terletak di delapan arah yang berlainan, hal ini ditunjukkan pada Gambar 1 [8].



Gambar 1. Piksel bertetangga dalam empat arah [8]

Dalam matriks *co-occurrence*, terdapat ciri tekstur yang dapat diperoleh dari suatu citra yang digunakan sebagai pembeda antara citra dengan kelas tertentu, dengan kelas lainnya. Ciri – ciri tersebut adalah [13]:

- a. Energi (*Energy*)
 
$$f_1 = \sum_i \sum_j P_{ij}^2 \tag{1}$$
- b. Entropi (*Entropy*)
 
$$f_2 = - \sum_i \sum_j P_{ij} \log P_{ij} \tag{2}$$
- c. Kontras (*Contrast*)
 
$$f_3 = \sum_i \sum_j (i - j)^2 P_{ij} \tag{3}$$
- d. Homogenitas (*Homogeneity*)

$$f_4 = \sum_i \sum_j \frac{P_{ij}}{1+|i-j|} \tag{4}$$

e. Korelasi (*Correlation*)

$$f_5 = \sum_i \sum_j \frac{(i-\mu_i)(j-\mu_j)(P_{ij})}{\sigma_i \sigma_j} \tag{5}$$

$\mu_i$  adalah nilai rata – rata elemen kolom ke – *i* pada matriks  $P_{i,j}$

$\mu_j$  adalah nilai rata – rata elemen kolom ke – *j* pada matriks  $P_{i,j}$

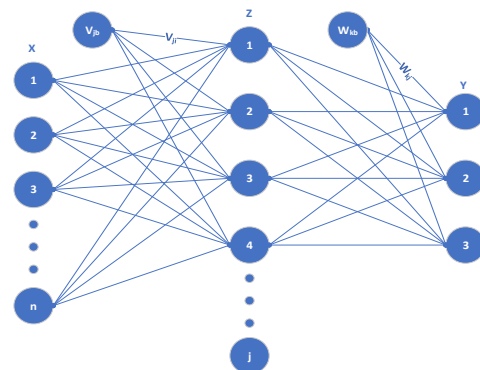
$\sigma_i$  adalah nilai standar deviasi elemen kolom ke – *i* pada matriks  $P_{i,j}$

$\sigma_j$  adalah nilai standar deviasi elemen kolom ke – *j* pada matriks  $P_{i,j}$

**3.3. Jaringan Syaraf Tiruan (JST) Backpropagation**

JST dengan *layer* tunggal memiliki keterbatasan dalam pengenalan pola. Kelemahan ini bisa ditanggulangi dengan menambahkan satu atau beberapa *layer* tersembunyi di antara *layer* masukan dan *layer* keluaran. JST *backpropagation* melatih jaringan mendapatkan keseimbangan antara kemampuan jaringan untuk mengenali pola yang digunakan selama pelatihan serta kemampuan untuk memberikan respons yang benar terhadap pola masukan yang serupa dengan pola yang dipakai selama pelatihan [13].

*Backpropagation* memiliki beberapa unit (*neuron*) yang ada dalam satu atau lebih *layer* tersembunyi. Gambar 3 adalah arsitektur *backpropagation multilayer* dengan 1 *hidden layer*. Pada gambar, unit *input* dilambangkan dengan *X*, unit *hidden* dilambangkan dengan *Z*, dan unit *output* dilambangkan dengan *Y*. Bobot antara unit *input* (*X*) dan unit *hidden* (*Z*) dilambangkan dengan *V*, sedangkan bobot antara unit *hidden* (*Z*) dan unit *output* (*Y*) dilambangkan dengan *W*.



Gambar 2. Model JST *backpropagation*

Dalam *backpropagation*, fungsi aktivasi yang dipakai harus memenuhi beberapa syarat yaitu

kontinu, terdiferensial dengan mudah dan merupakan fungsi yang tidak turun. Salah satu fungsi yang memenuhi ketiga syarat tersebut sehingga sering dipakai adalah fungsi *sigmoid biner* yang memiliki *range* (0, 1) [13]. Persamaan fungsi aktivasi *sigmoid biner* yaitu sebagai berikut:

$$f(x) = \frac{1}{1+e^{-x}} \tag{6}$$

$$f'(x) = f(x)(1 - f(x)) \tag{7}$$

Algoritme pelatihan *backpropagation* terdiri dari proses *feedforward* dan *backpropagation*. Algoritme tersebut yaitu sebagai berikut [14]:

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

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

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

Fase I: *Step Forward*

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

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

$$Z_{net_j} = v_{0j} + \sum_i^n x_i v_{ij} \tag{8}$$

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

$$Z_j = f(Z_{net_j}) \tag{9}$$

Dan kirimkan sinyal tersebut ke semua unit lapisan atasnya (unit – unit *output*). Langkah ini dilakukan sebanyak jumlah lapisan tersembunyi.

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

$$Y_{net_k} = W_{0k} + \sum_j^p Z_j W_{jk} \tag{10}$$

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

$$Y_k = f(Y_{net_k}) \tag{11}$$

Fase II: *Backpropagation*

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

$$\delta_k = (T_k - Y_k) f'(Y_{net_k}) \tag{12}$$

$\delta$  merupakan unit kesalahan yang akan dipakai dalam perubahan bobot *layer* di bawahnya (langkah 7).

$f'(Y_{net_k})$  merupakan fungsi turunan dari fungsi aktivasi *sigmoid biner*.

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

$$\Delta W_{jk} = \alpha \delta_k Z_j \tag{13}$$

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

$$\Delta W_{0k} = \alpha \delta_k \tag{14}$$

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

$$\delta_{net_j} = \sum_k^m \delta_k W_{jk} \tag{15}$$

Faktor  $\delta$  unit tersembunyi:

$$\delta_j = \delta_{net_j} f'(Z_{net_j}) \tag{16}$$

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

$$\Delta V_{ij} = \alpha \delta_j x_i \tag{17}$$

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

$$\Delta V_{0j} = \alpha \delta_j \tag{18}$$

Fase III: Perubahan Bobot

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

$$W_{jk}(\text{baru}) = W_{jk}(\text{lama}) + \Delta W_{jk} \tag{19}$$

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

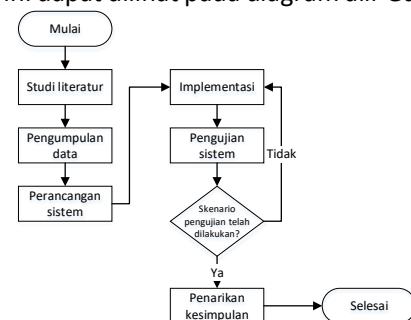
$$V_{ij}(\text{baru}) = V_{ij}(\text{lama}) + \Delta V_{ij} \tag{20}$$

Langkah 9: Kondisi pelatihan berhenti

Ketiga fase tersebut diulang terus menerus hingga kondisi penghentian dipenuhi. Umumnya kondisi penghentian yang sering dipakai adalah jumlah iterasi atau kesalahan. Iterasi akan dihentikan jika jumlah iterasi yang dilakukan sudah melebihi jumlah maksimum iterasi yang ditetapkan, atau jika kesalahan yang terjadi sudah lebih kecil dari batas toleransi yang diizinkan [15].

### 3.4. Proses Penelitian

Pada tahap proses penelitian dilakukan beberapa tahap secara sistematis, dimulai dari tahap studi literatur hingga tahap penarikan kesimpulan dari sistem yang telah dibuat dan diuji. Tahapan proses penelitian ini dapat dilihat pada diagram alir Gambar 3.



Gambar 3. Diagram alir tahapan proses penelitian

- a. Studi literatur  
Studi literatur dilakukan dengan mempelajari buku – buku, jurnal penelitian serta sumber lain yang berkaitan dengan permasalahan yang diangkat. Adapun materi yang dipelajari dalam studi literatur berkaitan dengan Bahasa Arab, huruf hijaiyah, kecerdasan buatan, pengolahan citra digital, *neural network*, metode *gray level co-occurrence matrices* dan metode *backpropagation*.
- b. Pengumpulan data  
Pengumpulan data dilakukan untuk memperoleh data yang akan digunakan dalam penelitian. Data yang berhasil dikumpulkan sebanyak 1500 citra huruf hijaiyah dari 50 orang responden.
- c. Perancangan sistem  
Pada tahapan perancangan sistem ini terdiri dari perancangan arsitektur sistem dan perancangan desain sistem. Sistem dirancang agar dapat melakukan klasifikasi citra huruf hijaiyah menggunakan metode GLCM dan *backpropagation*.
- d. Implementasi  
Pada tahapan implementasi akan dilakukan pembuatan program sistem berdasarkan perancangan yang telah dibuat pada tahap sebelumnya.
- e. Pengujian Sistem  
Pengujian sistem dilakukan untuk mengetahui kelayakan sistem berdasarkan tingkat akurasi. Jika seluruh skenario pengujian telah dilakukan, maka dapat dilanjutkan ke tahap selanjutnya.
- f. Penarikan Kesimpulan  
Penarikan kesimpulan didapatkan berdasarkan hasil pengujian dari sistem dan apakah sistem sudah mampu menjawab kebutuhan sistem berdasarkan dari tujuan penelitian.

**3.5. Pengujian Sistem**

Pengujian dilakukan menggunakan citra yang ada pada data uji. Hasil pengujian ini nantinya akan didapatkan tingkat akurasi dan presisi dari model yang dihasilkan ketika pelatihan *backpropagation*. Pada penelitian ini akan dilakukan pengujian kepada nilai ekstraksi fitur GLCM, pengaruh normalisasi terhadap nilai GLCM dan pengaruh *cropping* terhadap citra. Normalisasi mengubah nilai fitur ke dalam rentang 0 hingga 1, sehingga kinerja *machine learning* akan lebih optimal karena melakukan perhitungan dengan skala data yang cukup kecil. Proses *cropping* pada citra masukkan dilakukan agar mendapatkan ukuran tulisan

yang sama untuk setiap citra masukkan. Setelah mendapatkan model pengujian terbaik, maka dilakukan pengujian untuk mengetahui pengaruh *hidden layer*. Terdapat beberapa skenario yang dirancang agar mendapatkan hasil akurasi dan presisi terbaik antara lain:

- a. Pelatihan *backpropagation* menggunakan 20 *node input layer*, 30 *output layer* dan 3 *hidden layer* yaitu (50, 50, 50, 50), (100, 100, 100, 100), (150, 150, 150, 150) dan (150, 50, 100).
- b. Pelatihan *backpropagation* menggunakan 20 *node input layer*, 30 *output layer* dan 2 *hidden layer* yaitu (50, 50), (100, 100), (150, 150) dan (150, 50).
- c. Pelatihan *backpropagation* menggunakan 20 *node input layer*, 30 *output layer* dan 1 *hidden layer* (50, 100 dan 150).

**3.5.1. Confusion Matrix**

*Confusion matrix* berfungsi untuk melakukan analisis apakah *classifier* dapat mengenali *tuple* dari kelas berbeda. Nilai dari *True-Positive* dan *True-Negative* memberikan informasi ketika *classifier* dalam melakukan klasifikasi bernilai benar, sedangkan *False-Positive* dan *False-Negative* memberikan informasi ketika *classifier* salah dalam melakukan klasifikasi data [16].

		Prediksi	
		Benar	Salah
Aktual	Benar	TP	FN
	Salah	FP	TN

Gambar 4. *Confusion matrix*

TP (*True Positive*) merupakan jumlah data dengan nilai sebenarnya positif dan nilai prediksi positif. FP (*False Positive*) merupakan jumlah data dengan nilai sebenarnya negatif dan nilai prediksi positif. FN (*False Negative*) merupakan jumlah data dengan nilai sebenarnya negatif dan nilai prediksi negatif. TN (*True Negative*) merupakan jumlah data dengan nilai sebenarnya positif dan nilai prediksi negatif. Perhitungan akurasi dan presisi model dapat dilakukan dengan menggunakan persamaan (21) dan (22) berikut ini:

$$Akurasi = \frac{\sum TP_i + TN_i}{\sum TP_i + FN_i + FP_i + TN_i} \times 100\% \quad (21)$$

$$Presisi = \frac{\sum TP_i}{\sum TP_i + FP_i} \times 100\% \quad (22)$$

**4. HASIL DAN PEMBAHASAN**

**4.1. Pengujian Terhadap Nilai Ekstraksi Fitur GLCM**

Pada penelitian ini dilakukan pengujian *backpropagation* terhadap nilai hasil ekstraksi fitur GLCM. Pengujian dilakukan dengan berbagai parameter antara lain *learning rate*, jumlah *epoch* dan pembagian data set. Berikut hasil pengujian terbaik disajikan pada Tabel I.

TABEL I. PENGUJIAN TERHADAP NILAI GLCM

<i>Learning rate</i>	<i>Epoch</i>	Variasi data set	Akurasi (%)	Presisi (%)
<b>0.1</b>	<b>1000</b>	<b>7:3</b>	<b>8.4444</b>	<b>8.4444</b>
0.3	2500	7:3	6	6
0.5	1000	7:3	6	6
0.1	2750	8:2	5.6667	5.6667
0.3	2750	8:2	5.6667	5.6667
0.5	1000	8:2	5.6667	5.6667
0.1	2500	9:1	6.6667	6.6667
0.3	2500	9:1	3.3333	3.3333
0.5	1000	9:1	6	6

Berdasarkan hasil pengujian *backpropagation* terhadap nilai hasil ekstraksi fitur GLCM pada Tabel I semakin banyak jumlah data uji yang digunakan maka rata – rata hasil akurasi *testing* meningkat, sedangkan pengaruh *learning rate* dan variasi data set cukup fluktuatif. Hasil terbaik terdapat pada pembagian jumlah data 7:3, *learnig rate* 0.1 dan jumlah *epoch* 1000 dengan akurasi pelatihan sebesar 5.8095%, akurasi pengujian 8.4444% dan presisi sebesar 8.4444%.

**4.2. Pengaruh Normalisasi Hasil Ekstraksi Fitur GLCM**

Pada penelitian ini dilakukan pengujian *backpropagation* terhadap nilai hasil ekstraksi fitur GLCM yang dilakukan normalisasi terhadap nilai tersebut ke dalam rentang nilai 0 hingga 1. Pengujian dilakukan dengan berbagai parameter antara lain *learning rate*, jumlah *epoch* dan pembagian data set. Berikut hasil pengujian terbaik disajikan pada Tabel II.

TABEL II. PENGUJIAN TERHADAP DATA SETELAH DINORMALISASI

<i>Learning rate</i>	<i>Epoch</i>	Variasi data set	Akurasi (%)	Presisi (%)
0.1	2000	7:3	22.4444	22.4444
0.3	2500	7:3	22.2222	22.2222
<b>0.5</b>	<b>2000</b>	<b>7:3</b>	<b>23.7778</b>	<b>23.7778</b>
0.1	2000	8:2	21.3333	21.3333

0.3	2750	8:2	20	20
0.5	2500	8:2	23.3333	23.3333
0.1	2750	9:1	20	20
0.3	1000	9:1	20	20
0.5	2500	9:1	20.6667	20.6667

Berdasarkan hasil pengujian *backpropagation* terhadap nilai hasil ekstraksi fitur GLCM yang dilakukan normalisasi ke dalam rentang nilai 0 hingga 1 pada Tabel II semakin banyak jumlah data uji yang digunakan maka rata – rata hasil akurasi *testing* meningkat, sedangkan pengaruh *learning rate* dan variasi data set cukup fluktuatif. Hasil terbaik terdapat pada pembagian jumlah data 7:3, *learnig rate* 0.5 dan jumlah *epoch* 2000 dengan akurasi pelatihan sebesar 49.5190%, akurasi pengujian 23.7778% dan presisi sebesar 23.7778%. Normalisasi terhadap hasil ekstraksi fitur GLCM berdampak cukup baik, akurasi pengujian secara rata – rata meningkat 15.0216% dibandingkan dengan akurasi pada proses sebelumnya.

**4.3. Pengaruh Penambahan Cropping Pada Preprocessing**

Pada penelitian ini dilakukan pengujian *backpropagation* terhadap nilai hasil ekstraksi fitur GLCM yang dilakukan normalisasi terhadap nilai tersebut ke dalam rentang nilai 0 hingga 1 pada citra yang di – *crop* untuk mengurangi pengaruh latar putih pada citra masukkan. Pengujian dilakukan dengan berbagai parameter antara lain *learning rate*, jumlah *epoch* dan pembagian data set. Berikut hasil pengujian terbaik disajikan pada Tabel III.

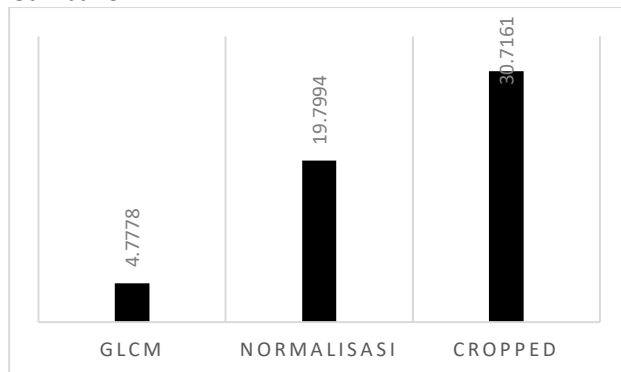
TABEL III. PENGUJIAN TERHADAP PENGARUH CROPPING SAAT PREPROCESSING

<i>Learning rate</i>	<i>Epoch</i>	Variasi data set	Akurasi (%)	Presisi (%)
<b>0.1</b>	<b>2500</b>	<b>7:3</b>	<b>41.1111</b>	<b>41.1111</b>
0.3	2500	7:3	40	40
0.5	2500	7:3	38.6667	38.6667
0.1	2750	8:2	32	32
0.3	2750	8:2	35.6667	35.6667
0.5	2750	8:2	33.3333	33.3333
0.1	2750	9:1	24.6667	24.6667
0.3	2000	9:1	20.6667	20.6667
0.5	2500	9:1	20.6667	20.6667

Berdasarkan hasil pengujian *backpropagation* terhadap nilai hasil ekstraksi fitur GLCM yang dilakukan normalisasi ke dalam rentang nilai 0 hingga 1 pada citra yang di – *crop* untuk mengurangi pengaruh latar putih pada citra masukkan, pada Tabel III semakin tinggi *learning rate* yang digunakan maka rata – rata akurasi

testing menurun, sedangkan semakin banyak jumlah data uji yang digunakan maka rata – rata hasil akurasi testing meningkat, kemudian pengaruh variasi data set cukup fluktuatif. Hasil terbaik terdapat pada pembagian jumlah data 7:3, *learnig rate* 0.1 dan jumlah *epoch* 2500 dengan akurasi pelatihan sebesar 89.1429%, akurasi pengujian 41.1111% dan presisi sebesar 41.1111%, dengan demikian maka konfigurasi yang akan digunakan pada pengujian berikutnya adalah pembagian data set 7:3, *learning rate* 0.1 dan jumlah *epoch* 2500.

Proses *cropping* terhadap citra masukkan pada tahap *preprocessing* berdampak cukup baik, akurasi pengujian secara rata – rata meningkat 10.9167% dibandingkan dengan akurasi pada proses sebelumnya. Rata – rata akurasi pengujian dapat dilihat pada Gambar 5.



Gambar 5. Rata – rata akurasi pengujian backpropagation

#### 4.4. Pengaruh Variasi Jumlah Hidden Layer

Pada penelitian ini dilakukan pengujian *backpropagation* terhadap pengaruh variasi *hidden layer* dengan konfigurasi sesuai hasil pengujian pada sub bagian 4.3. Berikut hasil pengujian terbaik disajikan pada Tabel IV.

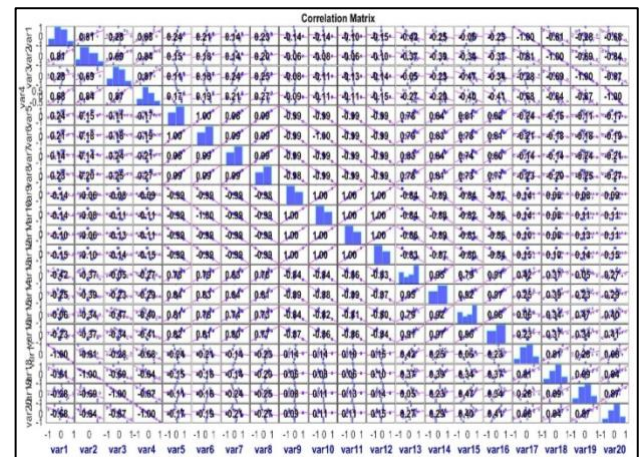
TABEL IV. PENGUJIAN TERHADAP PENGARUH VARIASI HIDDEN LAYER

Jumlah hidden layer	Neuron	Akurasi (%)	Presisi (%)
3	150, 50, 100	41.1111	41.1111
2	50, 50	41.5556	41.5556
<b>1</b>	<b>50</b>	<b>45.1111</b>	<b>45.1111</b>

Pengujian *backpropagation* terhadap pengaruh variasi *hidden layer* dengan konfigurasi parameter terbaik berdasarkan hasil pengujian sebelumnya, mendapatkan hasil terbaik terdapat pada konfigurasi jumlah *hidden layer* sebanyak 1 *layer* dan *neuron*

sejumlah 50 dengan akurasi pelatihan sebesar 56.7619%, akurasi pengujian 45.1111% dan presisi sebesar 45.1111%. Pengurangan jumlah *hidden layer* menyebabkan akurasi pengujian meningkat.

Jumlah data set yang digunakan cukup sedikit untuk jumlah kelas yang banyak seperti pada penelitian ini menyebabkan tingkat akurasi tidak sesuai harapan. Hasil klasifikasi kurang maksimal juga dapat disebabkan fitur – fitur yang kurang relevan dan tidak dapat mewakili keadaan data yang sesungguhnya. Terdapat cukup banyak fitur yang memiliki nilai korelasi yang mendekati 0, seperti terlihat pada Gambar 6.



Gambar 6. Correlation matrix

## 5. KESIMPULAN DAN SARAN

### 5.1. Kesimpulan

Berdasarkan hasil penelitian dan analisis yang telah dilakukan, dapat diambil kesimpulan sebagai berikut:

- Model arsitektur JST *backpropagation* terbaik untuk penelitian pengenalan huruf hijaiyah ini yaitu menggunakan 1 *hidden layer* dengan jumlah *node* sebanyak 50, *learning rate* sebesar 0.1, dengan *epoch* sebanyak 2500 kali perulangan dan pembagian data set 7:3.
- Tingkat akurasi tertinggi dari model arsitektur pengujian terhadap pengaruh variasi *hidden layer* adalah 56.7619% untuk pengenalan data latih, 45.1111% untuk pengenalan data uji dan presisi sebesar 45.1111%.
- Luas latar belakang berwarna putih pada citra masukkan sangat berpengaruh terhadap tingkat akurasi pada klasifikasi citra huruf

- hijaiah menggunakan metode GLCM dan *backpropagation*. Semakin besar latar belakang dari sebuah citra masukkan, maka objek akan semakin sulit dikenali. Semakin kecil latar belakang dari sebuah citra masukkan, maka objek akan semakin mudah untuk dikenali.
- d. Normalisasi data hasil ekstraksi fitur dan proses *cropping* dapat meningkatkan akurasi pengujian dari 8.4444% hingga 45.1111%.
  - e. Perlakuan terhadap data yang tepat seperti *cropping* dan *resize* memberikan dampak yang signifikan terhadap peningkatan akurasi pengenalan citra huruf hijaiah ini.
  - f. Metode GLCM dan *backpropagation* kurang maksimal dalam pengenalan citra huruf hijaiah dengan jumlah data set yang sangat terbatas yaitu hanya 1500 data, dibuktikan dengan terjadinya penurunan tingkat akurasi saat jumlah data uji dikurangi.

## 5.2. Saran

Berdasarkan hasil penelitian, analisis dan kesimpulan di atas, penulis dapat memberikan beberapa masukan atau saran untuk penelitian lebih lanjut antara lain:

- a. Memilih fitur yang paling relevan dan dapat mewakili keadaan data yang sesungguhnya sehingga dapat dilakukan klasifikasi dengan baik.
- b. Perlakuan terhadap data pada saat *preprocessing* harus diperhatikan agar bisa melakukan klasifikasi dengan baik dan mendapatkan akurasi yang tinggi.
- c. Menambah jumlah data set dalam jumlah yang besar, agar mendapatkan hasil tingkat akurasi yang lebih maksimal untuk pengenalan citra huruf hijaiah menggunakan metode klasifikasi *backpropagation*.
- d. Pengumpulan data harus dipersiapkan secara maksimal dan dilakukan mengikuti konfigurasi yang telah ditentukan agar mendapatkan data sesuai kondisi yang diinginkan.

## DAFTAR PUSTAKA

- [1] "What are the official languages of the United Nations? - Ask DAG!" <http://ask.un.org/faq/14463> (accessed Oct. 24, 2019).
- [2] A. Haralick, Robert M., Shanmugam. K and I. Dinstein, "Textural Features for Image Classification," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-3, no. 6, pp. 610–621, 1973.
- [3] K. B. Putri and N. L. Marpaung, "Identifikasi Huruf Hijaiah Berbasis GLCM Menggunakan Jaringan Syaraf Tiruan BackPropagation," *Jom FTEKNIK*, vol. 6, no. 1, pp. 2–6, 2019.
- [4] I. A. Kanta, "Pengenalan Pola Huruf Hijaiah Tulisan Tangan Menggunakan Logika Fuzzy Dengan Jaringan Syaraf Tiruan Backpropagation," *Jurusan Teknik Informatika Fakultas Komunikasi dan Informatika Universitas Muhammadiyah Surakarta*. 2013.
- [5] A. Sanjaya and D. W. Widodo, "Identifikasi Tulisan Tangan Huruf Hijaiah," *J. Ilm. NERO*, vol. 4, no. 1, pp. 23–29, 2018.
- [6] 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.
- [7] Sutarno and P. A. Garini, "Pengenalan Angka Arab Timur Tulisan Tangan Menggunakan Zone Centroid Zone ( ZCZ ) dan Backpropagation," 2017.
- [8] A. S. Kholimi and F. Nazihullah, "Identifikasi Tulisan Arab Dengan Menggunakan GLCM Dan RNN," 2018.
- [9] N. R. Angraheni, R. Efendi, and E. P. Purwandari, "Pengenalan Tullisan Tangan Huruf Hijaiah Sambung Menggunakan Algoritma Tamplate Matching Correlation," *J. Rekursif*, vol. 5, no. 1, 2017.
- [10] E. Y. Subairi, Rahmadwati, "Implementasi Metode k-Nearest Neighbor pada Pengenalan Pola Tekstur Citra Saliva untuk Deteksi Ovulasi," *J. EECCIS*, vol. 12, no. 1, pp. 9–14, 2018.
- [11] E. Budianita, Jasril, and L. Handayani, "Implementasi Pengolahan Citra dan Klasifikasi K-Nearest Neighbour Untuk Membangun Aplikasi Pembeda Daging Sapi dan Babi," *J. Sains, Teknol. dan Ind.*, vol. 12, no. 2, pp. 242–247, 2015.
- [12] F. S. Ni'mah, T. Sutojo, and D. R. I. M. Setiadi, "Identifikasi Tumbuhan Obat Herbal Berdasarkan Citra Daun Menggunakan Algoritma Gray Level Co-occurrence Matrix dan K-Nearest Neighbor," *J. Teknol. dan Sist. Komput.*, vol. 6, no. 2, pp. 51–56, 2018, doi: 10.14710/jtsiskom.6.2.2018.51-56.
- [13] Y. K. Ganis, I. Santoso, and R. R. Isnanto, "Klasifikasi Citra Dengan Matriks Ko-okurensi Aras Keabuan ( Gray Level Co-occurrence Matrix -GLCM ) Pada Lima Kelas Biji-Bijian," *Undergrad. thesis, Jur. Tek. Elektro Fak. Tek. Univesitas Diponegoro*, pp. 1–7, 2011.



- [14] L. Fausett, *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*. Melbourne, 1994.
- [15] A. Jumarwanto, R. Hartanto, and D. Prastiyanto, "Aplikasi Jaringan Syaraf Tiruan Backpropagation Untuk Memprediksi Penyakit THT di Rumah Sakit Mardi Rahayu Kudus," *J. Tek. Elektro*, vol. 1, no. 1, pp. 11–21, 2009.
- [16] M. F. Fibrianda and A. Bhawiyuga, "Analisis Perbandingan Akurasi Deteksi Serangan Pada Jaringan Komputer Dengan Metode Naïve Bayes Dan Support Vector Machine ( SVM )," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 9, pp. 3112–3123, 2018.