

# Pengenalan Pola Huruf pada Kata dengan Menggunakan Algoritma Backpropagation dan Hybrid Feature

## IDENTIFICATION LETTER PATTERNS WORD USING BACKPROPAGATION AND HYBRID FEATURE

Novan Wijaya<sup>\*1</sup>, Kevin Susanto<sup>2</sup>, Jefry<sup>3</sup>

AMIK MDP: Jl. Rajawali No. 14, Palembang 30113, Indonesia

STMIK GI MDP: Jl. Rajawali No. 14, Palembang 30113, Indonesia

<sup>\*1</sup>Manajemen Informatika: AMIK MDP

<sup>2,3</sup> Teknik Informatika: STMIK GI MDP

e-mail: novan.wijaya@mdp.ac.id<sup>\*1</sup>, kevinsusanto11@mhs.mdp.ac.id<sup>2</sup>, jefry@mhs.mdp.ac.id<sup>3</sup>

### Abstrak

Dengan kemajuan teknologi informasi saat ini, khususnya dalam proses penulisan kata maupun kalimat dimana seseorang akan lebih mudah dalam proses penulisan tangan dan rapi tetapi proses penulisan yang dilakukan pada saat ini juga tidak dapat ditinggalkan. Beberapa aktivitas menuntut seseorang untuk menggunakan tulisan tangan secara manual seperti pembuatan surat pernyataan ataupun dokumen penting lainnya. Dalam implementasi yang akan dilakukan pada sistem pengenalan pola huruf pada kata dengan menggunakan tulisan tangan akan menghasilkan tingkat keakuratan yang lebih rendah jika dibandingkan dengan pengenalan yang dilakukan pada sistem yang ada dikomputer seperti Times New Roman, Calibri, dan lain sebagainya. Dengan menggunakan ekstraksi *Hybrid Features* dan algoritma backpropagation diharapkan mampu mengurangi masalah dalam proses pengenalan pola tulisan tangan seseorang. Untuk mendapatkan pola fitur, digunakan ekstraksi *Hybrid Features* dari fitur diagonal, gradien sumbu x dan y serta rata-rata. Fitur akan digunakan menggunakan jaringan saraf tiruan dengan algoritma backpropagation. Parameter pengujian yang digunakan adalah ketepatan jumlah citra tulisan yang dikenali oleh sistem. Berdasarkan pengujian kinerja models, didapatkan akurasi tertinggi 93,86% untuk huruf uppercase dengan 45 dataset, 80,46% dengan 30 dataset untuk huruf *lowercase* dan 78,205 dengan 45 dataset untuk seluruh huruf. Pengujian program yang dibuat mendapatkan nilai akurasi sebesar 72,92% untuk citra kata baru, 83,33% untuk citra huruf baru dan 82,56% untuk citra huruf yang telah dipelajari.

**Kata kunci**— Jaringan Saraf Tiruan, *Backpropagation*, *Hybrid Features*

### Abstract

With the advancement of information technology today, especially in the process of writing words and sentences where a person will be easier in the process of writing using hands and neat but the writing process carried out at this time also cannot be abandoned. Some activities require us to use handwriting manually such as making statement letters or other important documents. In the implementation that will be carried out on the system of letter pattern recognition in words using handwriting will produce a lower level of accuracy compared to the introduction made on systems that is in computers such as Times New Roman, Calibri, and so on. By using Hybrid Features extraction and backpropagation algorithm, it is expected to reduce problems in the process of recognizing one's handwriting pattern. To get a feature pattern, Hybrid Features extraction used from diagonal features, x and y axis gradients and averages are used. The feature will be used is artificial neural networks with the backpropagation algorithm. The test parameters used are the accuracy of the number of written images recognized by the system. Based on models' performance tests, the highest accuracy was 93.86% for uppercase letters with 45 datasets, 80.46% with 30 datasets for lowercase letters and 78.205 with 45 datasets for all letters. The testing of the program made obtained an accuracy value of 72.92% for the image of the new word, 83.33% for the image of the new letter and 82.56% for the image of the letter that had been studied.

**Keywords**— Artificial Neural Networks, Backpropagation, Hybrid Features

## 1. PENDAHULUAN

**C**haracter Recognition merupakan pengenalan yang membandingkan antara teknologi komputerisasi data digital dan *non-digital*. Pengenalan pola huruf pada kata dianggap masih dibutuhkan seiring perkembangan teknologi informasi yang terus berkembang, akan tetapi proses dokumentasi secara tertulis yang dilakukan belum bisa ditinggalkan dalam kehidupan sehari-hari. Dengan adanya *Character Recognition* proses pendataan dari data *non-digital* ke bentuk digital dapat dipermudah. *Character Recognition* adalah sebuah proses mengenali karakter tulisan dari gambar masukan dan mengubahnya kedalam *American Standard Code for Information Interchange (ASCII)* atau bahasa mesin lainnya. Dalam metode penulisan, *character recognition* dikelompokkan menjadi 2 yaitu Handwritten Character Recognition (HCR) dan Optical Character Recognition (OCR). HCR memiliki tingkat ketepatan berkisar 90%, dikarenakan ada beberapa faktor penyebabnya diantara lain perbedaan bentuk tulisan, gaya penulisan, dan tipe dari tulisan tersebut. Dalam bahasa tertentu seperti bahasa Bangla, Tamil, Hindi, Gujarati, dan Punjabi ketidaksamaan karakter penulisan berpengaruh besar, dikarenakan bahasa tersebut masih sangat jarang digunakan secara luas [1]. Ini menyebab HCR mengalami kesulitan dalam melakukan proses pengenalan tulisan tangan dikarenakan hampir setiap orang memiliki gaya atau cara penulisan yang berbeda-beda walaupun karakter penulisan tersebut memiliki arti yang sama. Sementara tipe OCR mempunyai tata cara penulisan yang baku seperti Times New Roman, Calibri, dan sebagainya. Sehingga untuk proses pengenalan karakter, OCR lebih baik dibandingkan dengan HCR.

Pengenalan pola (*pattern recognition*) adalah suatu ilmu untuk mengklasifikasikan atau menggambarkan sesuatu berdasarkan pengukuran kuantitatif fitur (ciri) atau sifat utama dari suatu objek [2]. Dengan ciri yang cukup unik, proses pengklasifikasian dari huruf dapat menghasilkan hasil yang akurat.

Terdapat banyak metode ekstraksi fitur yang dapat digunakan pada data citra, namun metode ekstraksi tersebut harus menghasilkan fitur yang bersifat unik sehingga bisa dipakai sebagai bahan pembelajaran yang dapat membedakan tiap karakter diantaranya *Feature Extraction* [3] dan *Diagonal Feature Extraction* [4]. Sedangkan penelitian yang terkait dalam metode pengenalan diantaranya menggunakan *Naïve Bayes Classifier* [5], *Backpropagation* dalam pola aksara Hanacaraka [6] dan algoritma *Backpropagation FeedForward* [7].

Dari beberapa penelitian yang telah dilakukan belum ada yang menggunakan segmentasi Profile Projection Histogram dalam proses pemisahan huruf dari kata pada penulisan tangan dan Hybrid Features sebagai ekstraksi fitur yang diperoleh dari nilai gradien, rata-rata dan diagonal dan algoritma *Backpropagation* sebagai pembelajaran JST pada jenis huruf atau karakter dari tulisan tangan.

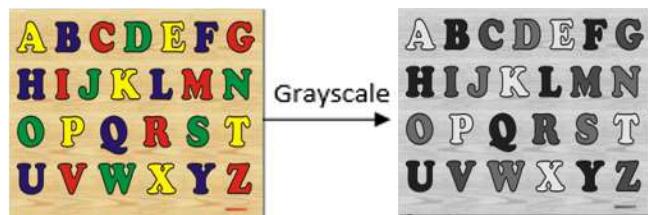
## 2. METODE PENELITIAN

### 2.1 Preprocessing Image

*Preprocessing Image* adalah ilmu yang mengolah sinyal yang berupa citra secara spesifik. Dalam artian yang sebenarnya, citra merupakan gambar yang dipetakan dalam dua dimensi. Dan dalam arti matematis citra itu fungsi kontinyu dari intesitas cahaya pada bidang 2 dimensi [8]. Manfaat dari pengolahan citra adalah untuk memperbaiki bentuk citra, menganalisis dan mendeteksi citra. Terdapat jenis-jenis citra diukur dari nilai suatu piksel dalam rentang tertentu, antara rentang 0-255 [2]. Pengolahan Citra yang digunakan dalam penelitian ini ialah pengubahan ke citra keabuan (*Grayscale*) dan hitam putih.

#### 2.1.1 *Grayscale*

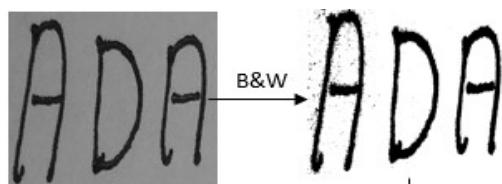
*Grayscale* merupakan proses merubah gambar menjadi monokrom dengan nilai intensitas 0 untuk hitam, 255 untuk putih, dan abu-abu untuk nilai yang berada pada rentang nilai 0 sampai 255.



Gambar 1. Contoh Proses *Grayscale*

#### 2.1.2 *Thresholding*

*Thresholding* merupakan metode untuk mengelompokkan piksel dalam sebuah citra berdasarkan batas nilai intensitas tertentu. Metode *Thresholding* yang digunakan adalah metode otsu [2]. *Thresholding* akan membandingkan nilai masing-masing piksel dari sebuah citra dengan nilai *Thresholding* ( $T$ ). Apabila nilai sebuah piksel kurang dari nilai  $T$ , maka nilai piksel akan menjadi 0, sedangkan untuk nilai piksel yang lebih dari  $T$  akan diganti nilainya menjadi 1.



Gambar 2. Contoh Proses *Thresholding*

#### 2.1.3 *Cropping*

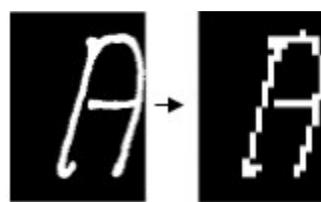
*Cropping* pada pengolahan citra berarti memotong suatu bagian dari citra pada koordinat tertentu sehingga diperoleh citra yang diharapkan. Koordinat yang dimaksud adalah koordinat piksel yang ingin diambil untuk dijadikan piksel baru [2].



Gambar 3. Contoh Proses *Cropping*

#### 2.1.4 *Scaling*

Operasi penskalaan (*scaling*) dimaksudkan untuk memperbesar (*zoom-in*) atau memperkecil (*zoom-out*) citra sesuai dengan faktor skala  $K$  yang diinginkan. Pada prinsipnya, operasi penskalaan menggandakan jumlah piksel sebesar  $K$  kali semula bila  $K > 1$  dan  $1/K$  kali semula bila  $0 < K < 1$  dalam arah vertikal dan horizontal [9].



Gambar 4. Contoh Proses *Scaling*

#### 2.1.5 Segementasi

Segmentasi citra merupakan suatu proses yang membagi citra ke dalam beberapa bagian, yaitu bagian yang diperlukan dan bagian yang tidak diperlukan oleh sistem. Segmentasi karakter perlu dilakukan sebelum proses pengenalan karakter. Biasanya citra yang akan

disegmentasi sudah berupa citra biner untuk memudahkan pemisahan karakter [2]. Metode yang digunakan adalah metode *Projection Profile Histogram* dengan menjadikan huruf atau objek kedalam bentuk garis-garis histogram vertikal dan horizontal. Profil dari proyeksi merupakan sebuah struktur data yang digunakan untuk menyimpan sejumlah piksel hitam yang merupakan objek ketika suatu citra diproyeksikan melalui sumbu-x maupun sumbu-y. Perhitungan dilakukan dengan cara menghitung jumlah piksel hitam pada suatu citra secara horizontal dan juga secara vertikal. Untuk menggunakan metode *Projection Profile Histogram* menggunakan persamaan (1)(2)(3).

$$X, Y \rightarrow M(x, y) \quad (1)$$

$$X_n = \sum_{i=0}^h Y_i, n \in [0, v] \quad (2)$$

$$Y_n = \sum_{i=0}^v X_i, n \in [0, h] \quad (3)$$

Dengan  $X_n$  dan  $Y_n$  adalah nilai projeksi piksel horizontal dan vertikal,  $v$  dan  $h$  adalah jarak vertikal dan horizontal dan  $M(x, y)$  adalah matriks citra gambar [10].

#### 2.1.6 Thinning

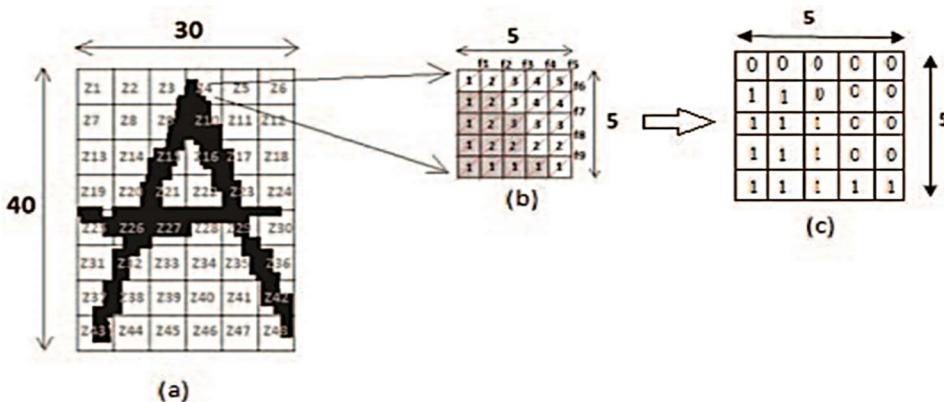
*Thinning* merupakan suatu operasi *morphologi*, terkadang seperti erosi atau *opening*. *Thinning* mengubah bentuk asli citra biner menjadi citra yang menampilkan batas-batas objek/*foreground* hanya setebal satu piksel [11]. Tujuan *thinning* adalah untuk menghilangkan piksel tepi objek pada citra biner yang berlebihan.

#### 2.1.7 Noise Reduction

Dalam proses *noise reduction* ada dua tipe metode kedekatan piksel yang bisa digunakan yaitu *4-Connected* dan *8-Connected*. Dalam kedekatan *4-Connected*, piksel dianggap terhubung apabila piksel berbagi tepi atas, kanan, kiri dan bawah yang sama, sedangkan pada *8-Connected* dengan tambahan 4 tepi pojok yang juga bertetangga. Secara umum, *4-Connectedness* lebih baik digunakan pada objek gambar "*foreground*" dan *8-Connectedness* lebih baik pada "*background*". Dikarenakan penulis akan menggunakan metode ini untuk mengambil objek "*foreground*", maka digunakan *4-Connected neighbourhood* [11].

## 2.2 Ekstraksi Fitur

Ekstraksi Fitur merupakan tahap yang paling penting dalam mencapai performa yang baik dari metode pengenalan data terutama karakter tulisan tangan. Secara singkat, ekstraksi fitur menghasilkan suatu informasi yang penting dan mewakili data secara keseluruhan sehingga dapat mengurangi banyaknya data yang perlu diproses. Dalam proses ekstraksi fitur ini, setiap karakter gambar berukuran 30x40 piksel terbagi menjadi 48 zona yang sama, masing-masing berukuran 5x5 piksel seperti yang ditunjukkan pada gambar 5. Dalam Penelitian ini digunakan metode ekstraksi fitur gabungan (*Hybrid Features*) dari metode ekstraksi diagonal, gradien dan rata-rata untuk mendapat 158 fitur



**Gambar 5.** Diagonal Ekstrasi Fitur

### 2.3 JST Backpropagation

Jaringan saraf tiruan (JST) adalah sebuah sistem pengolahan informasi yang karakteristik kinerjanya menyerupai jaringan saraf biologis. Jaringan saraf tiruan telah banyak dikembangkan sebagai generalisasi *model* matematika dari pengertian manusia atau saraf biologi, berdasarkan pada asumsi-asumsi bahwa [12]:

- Pemrosesan informasi terjadi pada banyak elemen-elemen sederhana yang disebut saraf (*neuron*).
- Sinyal-sinyal disampaikan antar saraf atas/pada jalur-jalur hubungan (*connection link*).
- Setiap jalur hubungan mempunyai sebuah bobot hubungan (*associated weight*), yang mana di dalam jaringan saraf yang khas, ini menggandakan.
- Setiap saraf menggunakan fungsi aktivasi (*activation function*), biasanya *non-linier*, untuk jaringan masukannya (penjumlahan dari bobot sinyal masukan) untuk menentukan sinyal keluarannya.

Sejak publikasi yang dilakukan *Parallel distributed processing (PDP)* mengenai algoritma *backpropagation* pada tahun 1986, pembelajaran dengan metode *Backpropagation* telah menjadi metode yang paling populer dalam melatih jaringan saraf tiruan. Alasan untuk popularitas itu ialah kesederhanaan dan kekuatan *relatif* algoritma yang mendasarinya. Kekuatan yang dimaksud berasal dari kenyataan bahwa, tidak seperti aturan belajar *Perceptron* dan peraturan pembelajaran *Windows-Hoff*, *Backpropagation* dapat digunakan untuk melatih jaringan *non-linier* dari konektivitas sembarang [13]. Algoritma pelatihan untuk *Backpropagation* adalah sebagai berikut:

Langkah 0: inisialisasi bobot (harga acak kecil)

Langkah 1: selama kondisi berhenti salah, kerjakan langkah 2-9

Langkah 2: untuk setiap pasangan, lakukan langkah 3-8

*Feedforward:*

Langkah 3: setiap unit masukan ( $X_i$ ,  $i = 1, \dots, n$ ), menerima sinyal masukan  $x_i$ , dan mengirimkan ke semua unit lapisan tersembunyi.

Langkah 4: setiap lapisan tersembunyi ( $Z_j$ ,  $j = 1, \dots, p$ ), jumlahkan sinyal masukkan bobotnya (4):

$$z\_in_j = V_{0j} + \sum_{i=1}^n X_i V_{ij} \quad (4)$$

Gunakan fungsi aktivasi untuk menghitung sinyal keluaran (5).

$$z_j = f(z\_in_j) \quad (5)$$

Dan kirim sinyal ini ke semua unit keluaran.

Langkah 5: setiap unit keluaran ( $Y_k$ ,  $k = 1, \dots, m$ ), jumlahkan sinyal masukkan bobotnya (6),

$$y_{in_k} = W_{0j} + \sum_{j=1}^p Z_j W_{jk} \quad (6)$$

Dan gunakan fungsi aktivasinya untuk menghitung sinyal keluaran (7).

$$y_k = f(y_{in_k}) \quad (7)$$

kesalahan *Backpropagation* :

Langkah 6: setiap unit keluaran ( $Y_k$ ,  $k = 1, \dots, m$ ), menerima pola target yang berhubungan dengan pola pelatihan masukan, hitung informasi kesalahannya (8)

$$\delta_k = (t_k - y_k) f'(y_{in_k}) \quad (8)$$

Menghitung koreksi bobot (9)

$$\Delta w_{jk} = a \delta_k z_j \quad (9)$$

Menghitung koreksi bias (10)

$$\Delta w_{ok} = a \delta_k \quad (10)$$

Mengirim harga  $\delta_k$  ke unit-unit lapisan bawah

Langkah 7: setiap unit tersembunyi ( $Z_j$ ,  $j=1, \dots, p$ ) jumlahkan masukan delta (11)

$$\delta_{in_k} = \sum_{j=1}^m \delta_k w_{jk} \quad (11)$$

kalikan dengan turunan fungsi aktivasi untuk menghitung informasi *error* (12)

$$\delta_j = \delta_{in_j} f'(z_{in_j}) \quad (12)$$

menghitung koreksi bobot (13):

$$\Delta v_{ij} = \alpha \delta_j X_i \quad (13)$$

dan menghitung koreksi bias (14)

$$\Delta v_{oj} = \alpha \delta_j \quad (14)$$

Perbarui bobot dan bias :

Langkah 8: setiap unit keluaran ( $Y_k$ ,  $k=1, \dots, m$ ) memperbarui bias dan bobot ( $j=0, \dots, p$ ) (15):

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (15)$$

Setiap unit tersembunyi ( $Z_j$ ,  $j=1, \dots, p$ ) memperbaui bias dan bobot ( $i=0, \dots, n$ ) (16)

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (16)$$

Langkah 9: Pengujian kondisi berhenti.

#### 2.4 K-Fold Cross Validation

*K-Fold Cross Validation* merupakan salah satu metode yang digunakan untuk mengetahui rata-rata keberhasilan dari suatu sistem dengan cara melakukan perulangan dengan mengacak atribut masukan sehingga sistem tersebut teruji untuk beberapa atribut masukan yang acak [14], dengan  $K=5$  atau 10 dapat digunakan untuk memperkirakan tingkat kesalahan yang terjadi, sebab data *training* pada setiap *fold* cukup berbeda dengan data *training* yang asli. Secara keseluruhan, 5 atau 10-fold cross validation sama-sama direkomendasikan dan disepakati bersama. Menghitung nilai akurasinya dapat dilakukan dengan menggunakan persamaan (17) [15]:

$$\text{Akurasi} = \frac{\text{Recognition Rate}}{\text{Jumlah data uji}} \times 100\% \quad (17)$$

Perhitungan akurasi tingkat klasifikasi akan dilakukan dengan cara menghitung jumlah *recognition rate* dari data yang diklasifikasi dengan benar dibagi total data yang diuji dan dikali 100% sehingga didapat presentasi klasifikasi yang benar dibanding seluruh data pengujian.

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Preprocessing

Pada tahapan preprocessing akan dilakukan 6 langkah, diantaranya akan mengubah ukuran resolusi dari citra asli menjadi 100x200 piksel. Kemudian langkah kedua akan dilakukan proses *grayscale*, yaitu mengubah citra digital yang telah dimasukkan menjadi citra keabuan (*grayscale*). Dari proses citra keabuan, akan dilakukan proses *thresholding*, dimana citra tersebut akan diubah menjadi citra biner. Proses selanjutnya akan menghilangkan coretan (*noise*) yang tidak dibutuhkan sehingga aplikasi yang akan dibangun lebih mudah untuk mengenali tulisan tangan tersebut. metode yang digunakan untuk menghilangkan coretan atau kotoran (*noises*) menggunakan metode *Connectivity 4-Connected Neighborhood*. Langkah berikutnya dilakukan proses pemisahan tiap baris terhadap tulisan tangan menjadi karakter dengan menggunakan Projection Profile Histogram, metode ini akan mengubah huruf atau karakter menjadi potongan berdasarkan garis potong dari histogram baik itu vertikal maupun horizontal. Potongan horizontal menghasilkan potongan baris dan vertikal menghasilkan potongan huruf. Langkah terakhir pada proses *preprocessing* yaitu standarisasi yang bertujuan untuk memperkecil tingkat *bias* dari citra huruf yang telah dilakukan pada proses sebelumnya sehingga mudah untuk lebih dikenali.

Proses ekstraksi fitur dari *dataset* hasil *preprocessing* menggunakan metode ekstraksi fitur gabungan (*Hybrid Features*) dari metode ekstraksi diagonal (*D*), gradien (*Gx* dan *Gy*) dan rata-rata ( $\mu$ ) yang dimulai dengan membagi citra huruf menjadi 4 segmen berukuran sama ( $25 \times 25$  piksel). Kemudian pada tiap segmen akan dilakukan ekstraksi fitur diagonal, gradien x dan y serta rata-rata. Pembagian segmen ini dilakukan untuk memperbanyak fitur *neuron* masukan dan memastikan tidak ada ciri yang terlewat serta mempercepat pemrosesan citra huruf. Total fitur berjumlah 72 fitur yang didapat dari tiap huruf berdasarkan jumlah 16 fitur untuk gradien sumbu-x (*Gx*), 16 fitur untuk gradien sumbu-y (*Gy*), 16 fitur rata-rata ( $\mu$ ) dan 24 fitur diagonal (*D*). Berikut adalah contoh hasil ekstraksi fitur menggunakan fitur *Hybrid* yang dapat dilihat pada Tabel 1.

**Tabel 1.** Hybrid Features

Index Fitur	<i>Gx</i>	<i>Gy</i>	$\mu$	<i>D</i>
1	0	0	0	0
2	0,3908	0,3048	3,4400	3,4211
3	0,2496	0,2349	2,6800	0,7895
4	0	0	0	0
5	0	0	0	1,0526
6	0,4914	0,3674	4,3600	0
7	0,4914	0,5339	5,9600	3,5263
8	0	0,0558	0,0800	6,3684
9	0,2121	0,2709	1,6000	0
10	0,4715	0,3369	5,7200	2,4737
11	0,4815	0,2847	3,6000	0,5263
12	0	0,0558	0,1600	3,6842
13	0,0988	0,0558	0,0800	2,8421
14	0,3554	0,02423	2,6800	0
15	0,1883	0,0781	0,6400	1,7632

16	0,0558	0,2847	1,4400	0,1053
17				2,8947
18				0
19				1,8947
20				1,2237
21				0,1579
22				3,3816
23				2,5000
24				0,4737

### 3.2 Pengecekan Jenis

Setelah tahap standarisasi akan dilakukan langkah pengecekan jenis dari huruf untuk menentukan *model* mana yang akan digunakan dalam proses pengklasifikasian huruf tersebut. Proses ini dilakukan dengan menghitung selisih tinggi satu huruf dengan huruf lainnya, dikarenakan huruf *uppercase* selalu ditulis lebih tinggi dibandingkan huruf *lowercase* sehingga didapat jenis dari huruf tersebut. Hasil dari proses ini dapat dilihat pada Tabel 2.

**Tabel 2.** Hasil Pengecekan Jenis

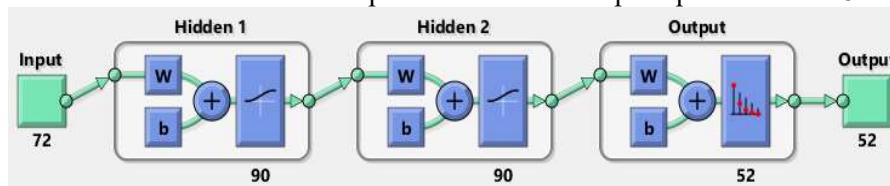
Status Case	1	0	0	0	1	0	0	0
Test	A	n	A	k	M	u	d	a

Keterangan :

- 1 : Uppercase
- 2 : Lowercase

### 3.3 Klasifikasi

Setelah dilakukan ekstraksi fitur, proses selanjutnya memasuki tahap *testing* seluruh fitur terhadap *model* dari hasil pembelajaran Jaringan Saraf Tiruan *Backpropagation*. *Model* berisi jaringan yang sudah dilatih terlebih dahulu yang bentuknya dapat dilihat pada Gambar 6. Setelah fitur dimasukkan kedalam *network model*, didapat 52 keluaran *neuron* yang masih berbentuk *vector* yang lalu dibulatkan menjadi *Round* (Keluaran) dan dikonversi kebentuk *index* untuk mendapatkan hasil klasifikasi seperti pada Gambar 7. Kelas 1-26 adalah *uppercase* (A-Z) dan kelas 27-52 adalah *lowercase* (a-z). Setelah didapat hasil klasifikasi akan dicari tingkat kebenaran klasifikasi untuk menentukan persentase akurasi seperti pada Gambar 8.

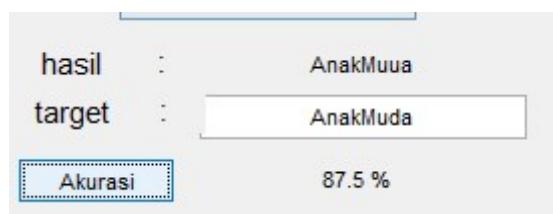


**Gambar 6.** Tampilan Network

$$\begin{array}{c}
 \text{Output neuron} \\
 n - \text{huruf} \\
 \hline
 i \\
 1 \quad [0.9480 \quad 6.927e-06 \quad 0.0001 \quad \dots \quad 6.027e-05] \\
 2 \quad 0.0005 \quad 0.0025 \quad 8.056e-07 \quad \dots \quad 4.590e-05 \\
 3 \quad 6.484e-06 \quad 0.0033 \quad 0.0254 \quad \dots \quad 0.0096 \\
 \vdots \quad \vdots \quad \vdots \quad \dots \quad \vdots \\
 52 \quad 0.00011 \quad 9.074e-05 \quad 3.097e-05 \quad \dots \quad 1.974e-04
 \end{array} \rightarrow
 \begin{array}{c}
 \text{Round}(Output) \\
 n - \text{huruf} \\
 \hline
 i \\
 1 \quad [1 \quad 0 \quad 0 \quad \dots \quad 0] \\
 2 \quad 0 \quad 0 \quad 0 \quad \dots \quad 0 \\
 3 \quad 0 \quad 0 \quad 0 \quad \dots \quad 0 \\
 \vdots \quad \vdots \quad \vdots \quad \dots \quad \vdots \\
 52 \quad 0 \quad 0 \quad 0 \quad \dots \quad 0
 \end{array}$$

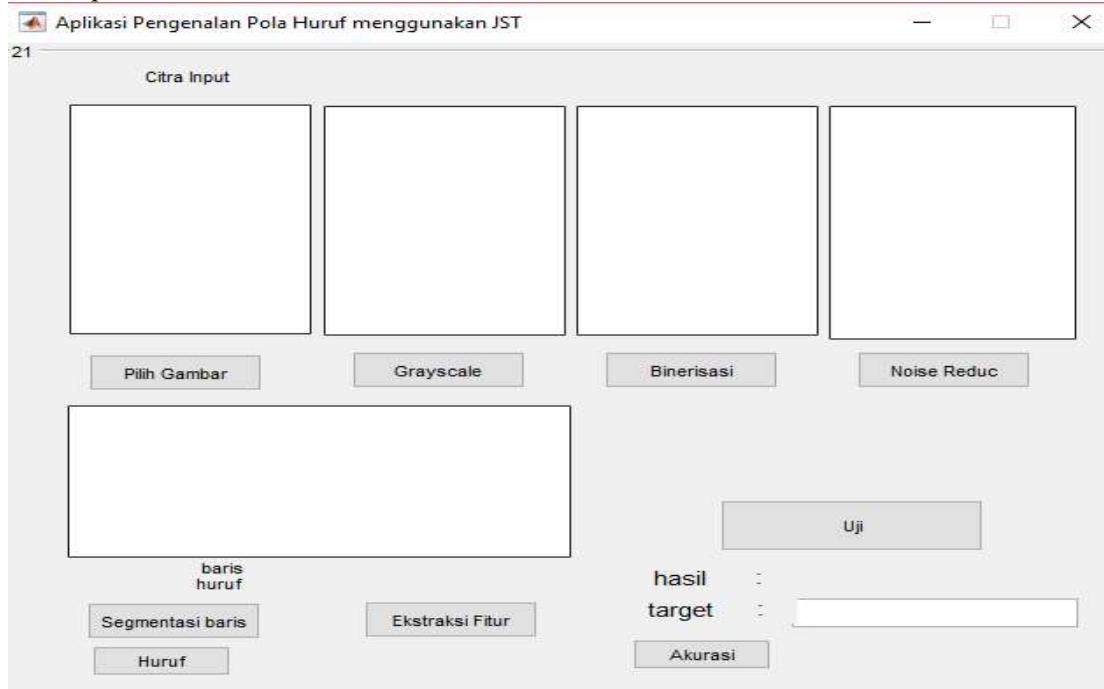
$huruf - n \mid 1 \quad 2 \quad 3 \quad \dots \quad n \mid$   
 $index (kelas) \mid 1 \quad 40 \quad 27 \quad \dots \quad 27 \mid$

Gambar 7. Neuron Keluaran, nilai vector dan hasil kelas



Gambar 8. Perhitungan Akurasi

### 3.4 Tampilan GUI



Gambar 9. GUI

Penggunaan program untuk mengenali huruf dapat dilakukan dengan 8 langkah, yaitu :

1. Pengguna harus memasukkan gambar yang diuji dengan cara menekan tombol “Pilih Gambar”.
2. Mengubah gambar yang dimasukkan menjadi *grayscale* dengan menekan tombol “*Grayscale*”.
3. Mengubah citra yang didapat menjadi citra biner dengan menekan tombol “*Binerisasi*”.
4. Menghilangkan coretan atau kotoran yang tidak dibutuhkan pada citra dengan menekan tombol “*Noise Reduction*”.

5. Menekan tombol “Segmentasi baris” dan tombol “Huruf” maka akan terjadi proses segmentasi yang digunakan untuk memisahkan tiap baris pada tulisan setelah itu tiap baris tersebut kata dipisah menjadi perkarakter.
6. Ekstraksi fitur dari ketiga *dataset* menggunakan metode ekstraksi fitur gabungan (*Hybrid Features*) dari metode ekstraksi diagonal, gradien ( $G_x$  dan  $G_y$ ) dan rata-rata dengan menekan tombol “Ekstraksi Fitur”.
7. Menguji gambar dengan menekan tombol “Uji” lalu hasil ujinya ditampilkan pada label hasil.
8. Proses perhitungan akurasi yaitu membandingkan kata yang di dapat dari proses pengujian dengan kata yang target yang diinginkan lalu tekan tombol “Akurasi”.

### 3.5 Pengujian

*K-Fold Cross Validation* merupakan salah satu metode yang digunakan untuk mengetahui rata-rata keberhasilan dari suatu sistem dengan cara melakukan perulangan dengan mengacak atribut masukan sehingga sistem tersebut teruji untuk beberapa atribut masukan yang acak. Pada Tabel 3 dijabarkan penggunaan 5 sampai 50 *dataset* dalam *5-fold Cross Validation* untuk mendapatkan variasi akurasi dari tiap *model*. Pada *model* gabungan akurasi tertinggi didapat pada saat 45 *dataset*, pada *model uppercase (MU)* pada 45 *dataset* dan terakhir pada *model lowercase (ML)* pada saat 30 *dataset*.

**Tabel 3.** K-Fold Cross

Jumlah	Model(%)	MU(%)	ML(%)
5	45,7692	67,6923	45,3846
10	72,1154	76,5385	76,9231
15	73,9744	82,5641	76,6667
20	77,2115	88,0769	76,9231
25	76,8462	88,7692	76,9231
30	76,9231	87,8205	80,4615
35	74,5744	87,8156	75,7143
40	75,7692	91,25	78,2692
45	78,2051	93,8462	79,0598
50	74,2791	92,8517	78,5385

### Keterangan :

Model = *model gabungan*

MU = *model uppercase*

ML = *model lowercase*

### 3.6 Pengujian Kata

Pada pengujian kata, digunakan 15 citra kata dengan gaya penulisan berbeda-beda yang dikumpulkan dari berbagai sukarelawan dimana pemilihan kata yang ingin ditulis dibebaskan namun dengan format kapital yang telah ditentukan yaitu kata pertama dengan *uppercase* di awal, *all caps* atau *all low* dan 2 kata dalam bentuk vertikal. Setelah tahap *preprocessing*, akan dihasilkan potongan citra huruf yang lalu akan dicari jenis hurufnya menggunakan langkah pengecekan jenis yang menentukan tipe kapital dari huruf berdasarkan ekstraksi ketinggian huruf pada tahap standarisasi yang menentukan *model* mana yang akan digunakan sebagai penguji pola fitur huruf. Setelah didapat *model* yang tepat, dilakukan pengujian kata terhadap program yang menghasilkan akurasi pengenalan kata dalam Tabel 4. Program menghasilkan akurasi rataan berupa 73,539%.

Tabel 4. Pengujian Akurasi Kata

Pengujian	Prediction	Actual	Akurasi
1	AnakMuua	AnakMuda	87,50%
2	Doplmaf	Dopemin	71,43%
3	BocqK	Bocah	60%
4	FDN	ADA	33,33%
5	Saatu	Suatu	80%
6	BapaHIka	BapakIbu	62,5%
7	ON	ON	100%
8	Svnny	Sunny	80%
9	UAWG	UANG	75%
10	PENP	PENA	75%
11	tAS	tAS	100%
12	iOTAL	tOTAL	80%
13	nOSS	bOSS	75%
14	Sumier	Sumber	83,33%
15	Mlziu	Mulia	40%

Dari Tabel 4 diketahui bahwa gaya penulisan tiap kata mempengaruhi hasil nilai akurasi dari pengujian kata dikarenakan adanya beberapa huruf yang berhasil dikenali pada suatu kata namun gagal dikenali pada kata lainnya, seperti huruf ‘A’ pada “UANG” berhasil dikenali namun gagal dikenali pada “ADA” dan “PENA”. Kemiripan pola beberapa huruf juga terlihat menyebabkan kegagalan klasifikasi huruf pada pengujian beberapa kata yang dapat dilihat pada pengujian “Sunny” yang menghasilkan “Svnny” dimana adanya kemiripan pola pada ‘u’ dan ‘v’. Kegagalan pada tahapan pengujian jenis juga dapat menyebabkan kegagalan klasifikasi seperti yang terlihat pada pengujian kata “BapakIbu” menjadi “BapaHIka” dimana huruf ‘k’ menghasilkan jenis huruf *uppercase* sehingga *model* yang digunakan adalah MU yang mengakibatkan hasil klasifikasi kelas huruf *uppercase* ‘H’. Hal ini diakibatkan tinggi huruf ‘k’ yang ditulis hamper sama tinggi dengan huruf *uppercase* ‘B’ dan ‘I’ sehingga terjadi kegagalan pengujian jenis.

#### 4. KESIMPULAN

Berdasarkan hasil implementasi Jaringan Saraf Tiruan menggunakan *hybrid features*, dapat disimpulkan sebagai berikut:

1. Fitur diagonal, rata-rata dan gradien dapat digunakan dalam pengenalan huruf tulisan tangan dengan implementasi Jaringan Saraf Tiruan menggunakan metode *Backpropagation* dan nilai gradien, diagonal dan rata-rata dari bagian-bagian citra tulisan tangan.
2. Berdasarkan analisis pengujian, masing-masing *model* yang akan digunakan pada program mendapatkan akurasi tertinggi 93,86% untuk huruf *uppercase* menggunakan 45 *dataset*, 80,46% menggunakan 30 *dataset* untuk huruf *lowercase* dan 78,205% menggunakan 45 *dataset* untuk seluruh huruf.
3. Program yang dibuat mendapatkan nilai akurasi sebesar 71,54% untuk 15 *dataset* citra kata baru, 83,33% untuk 5 *dataset* citra huruf baru dan 82,56% untuk citra 35 *dataset* huruf yang telah dipelajari.

#### UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada STMIK Global Informatika MDP yang telah memfasilitasi penelitian ini. Penulis juga mengucapkan terima kasih kepada LPPM STMIK

PalComTech yang memberikan yang telah banyak memberikan masukan pada penelitian ini sehingga penelitian ini siap untuk diterbitkan.

#### DAFTAR PUSTAKA

- [1] N. V. Rao, S. Sastry, Chakravarthy, and Kalyanchakravarthi, “Optical Character Recognition Technique Algorithms,” *J. Theor. Appl. Inf. Technol.*, vol. 83, no. 2, pp. 275–282, 2016.
- [2] D. Putra, *Pengolahan Citra Digital*. Yogyakarta: Andi Offset, 2010.
- [3] G. Katiyar and S. Mehfuz, “MLPNN Based Handwritten Character Recognition Using Combined Feature Extraction,” in *International Conference on Computing, Communication & Automation*, 2015, pp. 1155–1159.
- [4] U. Dwivedi, P. Rajput, and M. K. Sharma, “Cursive Handwriting Recognition System Using Feature Extraction and Artificial Neural Network,” *Int. Res. J. Eng. Technol.*, vol. 04, no. 03, pp. 2202–2206, 2017.
- [5] A. P. Wijaya and H. A. Santoso, “Naive Bayes Classification pada Klasifikasi Dokumen Untuk Identifikasi Konten E-Government,” *J. Appl. Intell. Syst.*, vol. 1, no. 1, pp. 48–55, 2016.
- [6] S. Winardi and H. Hamzah, “PENERAPAN METODE JARINGAN SYARAF TIRUAN BACKPROPAGATION DALAM PENGENALAN POLA AKSARA HANACARAKA,” *J. Teknol. Inf. Respati*, vol. 9, no. 27, pp. 33–42, 2017.
- [7] S. Solikhun, M. Safii, and A. Trisno, “aringan Saraf Tiruan Untuk Memprediksi Tingkat Pemahaman Siswa Terhadap Matapelajaran Dengan Menggunakan Algoritma Backpropagation,” *J-SAKTI (Jurnal Sains Komput. dan Inform.)*, vol. 1, no. 1, pp. 24–36, 2017.
- [8] Y. Permadi and M. Murinto, “Aplikasi Pengolahan Citra Untuk Identifikasi Kematangan Mentimun Berdasarkan Tekstur Kulit Buah Menggunakan Metode Ekstraksi Ciri Statistik,” *J. Inform. Ahmad Dahlan*, vol. 9, no. 1, pp. 1028–1038, 2015.
- [9] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing Using Matlab (Second Edition)*. 2004.
- [10] I. M. G. Sunarya, M. W. A. Kesiman, and I. A. P. Purnami, “Segmentasi Citra Tulisan Tangan Aksara Bali Berbasis Proyeksi Vertikal Dan Horizontal,” *J. Inform.*, vol. 9, no. 1, pp. 982–992, 2015.
- [11] D. Putra, *Sistem Biometrika*. Yogyakarta: Andi Offset, 2009.
- [12] S. Mori, “Deep architecture neural network-based real-time image processing for image-guided radiotherapy,” *Phys. Medica*, 2017.
- [13] Y. Chauvin and D. E. Rumelhart, *Backpropagation: Theory, Architectures, and Applications*. New York: Psychology Press, 2013.
- [14] E. Pandie, “Implementasi Algoritma Data mining KNearrest Neighbour (KNN) Dalam Pengambilan Keputusan Pengajuan Kredit,” Universitas Nusa Cendana, Kupang, 2012.
- [15] S. F. Rodiyansyah and E. Winarko, “Klasifikasi Posting Twitter Kemacetan Lalu Lintas Kota Bandung Menggunakan Naive Bayesian Classification,” *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 6, no. 1, pp. 91–100, 2012.