

APLIKASI PENGENALAN POLA PADA HURUF TULISAN TANGAN MENGUNAKAN JARINGAN SARAF TIRUAN DENGAN METODE EKSTRAKSI FITUR GEOMETRI

^[1]Herviana Masrani,^[2]Ilhamsyah,^[3]Ikhwan Ruslianto

^[1]^[3]Jurusan Sistem Komputer, Fakultas MIPA Universitas Tanjungpura

^[2]Jurusan Sistem Informasi, Fakultas MIPA Universitas Tanjungpura

Jl. Prof. Dr. H. Hadari Nawawi, Pontianak

Telp./Fax.: (0561) 577963

e-mail:

^[1]hervinanamasrani@gmail.com, ^[2]ilhamsyah@sisfo.untan.ac.id,

^[3]ikhwanruslianto@siskom.untan.ac.id

Abstrak

Pengenalan huruf tulisan tangan merupakan salah satu bentuk dari pengenalan pola. Pemecahan masalah terhadap tulisan tangan yang bervariasi akan sangat terkait dengan pengenalan pola yang bertujuan untuk menghasilkan dan memilih pola-pola yang bisa dimanfaatkan pada saat identifikasi. Pengenalan dalam penelitian ini akan memelalui beberapa tahapan, seperti praproses, segmentasi, dan proses pengenalan. Praproses yang digunakan dalam penelitian ini meliputi mengkonversi citra ke grayscale, threshold dan binerisasi. Hasil dari praproses harus di segmentasi melalui tahap deteksi tepi (edge detection), penebalan citra (dilation), dan pengisian citra (filling) agar citra dapat diekstraksi bentuk geometrinya. Jaringan syaraf tiruan perceptron akan menyesuaikan nilai bobot data yang telah dilatih dari 25 jenis setiap 26 huruf dengan nilai bobot citra masukan. Nilai keluaran dari pengklasifikasian jaringan syaraf tiruan disesuaikan dengan nilai ekstraksi fitur sehingga menghasilkan keluaran yang diharapkan. Aplikasi dengan menerapkan jaringan syaraf tiruan dengan ekstraksi fitur geometri yang dilakukan dari data pengujian dengan tingkat keberhasilan 86,535% terhadap 30 data pengujian dalam mengenali kata.

Kata Kunci : *Pengenalan huruf tulisan tangan, pengolahan citra, ekstraksi fitur geometri, perceptron, matlab.*

1. PENDAHULUAN

Untuk dapat mengenali huruf tulisan tangan, harus ada beberapa data yang akan mewakili setiap bentuk huruf tersebut yang diambil dari beberapa objek, yang merupakan suatu data digital. Data digital tersebut dapat diperoleh dengan melakukan scan gambar dengan format bitmap. Dari hasil scan tersebut, diperoleh citra yang kemudian dilakukan operasi praproses yang terdiri dari pola dan bentuk huruf tulisan tangan. Dari proses input citra ini, maka dapat dibangun suatu sistem cerdas yang memanfaatkan komputer untuk membaca tulisan tangan.

Pada penelitian yang dilakukan oleh Andi Prasojjo yang berjudul "Pengenalan Karakter Alfabet Menggunakan Jaringan Syaraf Tiruan". Metode klasifikasi yang digunakan adalah jaringan saraf tiruan perambatan-balik (*backpropagation*). Dalam penelitian yang dilakukannya, jaringan saraf

tiruan menggunakan 234 karakter huruf besar untuk tiap jenis font dan diuji dengan 702 karakter yang terdiri atas 234 karakter data latih dan 468 karakter data baru untuk setiap jenis font.[1]

Penelitian lainnya mengenai pengenalan huruf adalah penelitian oleh Anchemona Yudha yang melakukan penelitian tentang pengenalan tulisan tangan menggunakan *fuzzy feature extraction* dengan pendekatan *radial basis function neural*. Dimana *fuzzy feature extraction* menganggap huruf tulisan tangan sebagai sebuah graf berarah, yang *node*-nya terdiri atas titik ujung dan titik cabang. *Edge*-nya berupa garis lurus, kurva, dan *loop*. [2]

Dalam pemecahan permasalahan terhadap tulisan tangan tersebut, akan sangat terkait dengan pengenalan pola yang bertujuan untuk menghasilkan dan memilih pola-pola yang bisa dimanfaatkan pada saat identifikasi. Metode pengenalan pola citra

pada penelitian yang ditulis oleh Aris Fanani, Anny Yuniarti, dan Nanik Suciati yang berjudul “Ekstraksi Fitur Geometri Pada Citra Batik Menggunakan Representasi *Cardinal Spline*”. Ekstraksi fitur geometri citra batik digunakan untuk membantu komputer dalam mengenali pola atau motif batik. Penelitian tersebut menghasilkan sistem ekstraksi fitur geometri citra batik dan merepresentasikan fitur geometri tersebut menggunakan kurva *cardinal spline*. [3]

2. LANDASAN TEORI

2.1. Teknik Pengenalan Pola

Teknik pengenalan pola huruf tulisan tangan adalah sebuah teknik dimana input data yang berisi tulisan tangan dari lembaran kertas hasil scan menggunakan scanner dan menghasilkan gambar pada komputer yang dikenali sebagai titik-titik (bitmap), bitmap inilah yang kemudian diproses lebih lanjut menggunakan algoritma tertentu menjadi karakter, sehingga dapat dikenali dan diolah menjadi informasi.[4]

Pengenalan pola merupakan langkah perantara bagi proses menghilangkan dan menormalkan gambar dalam satu cara (pemrosesan gambar (*image processing*), teks dll.), pengiraan ciri-ciri, pengkelasan dan akhirnya post-pemrosesan berdasarkan kelas pengenalan dan aras keyakinan. Pengenalan pola berkaitan dengan langkah pengklasifikasian. Dalam kasus tertentu, sebagaimana dalam jaringan syaraf (*neural networks*), pemilihan ciri-ciri dan pengambilan juga boleh dilaksanakan secara semi otomatis atau otomatis sepenuhnya.[5]

2.2. Citra Digital

Ditinjau dari sudut pandang matematis, citra merupakan fungsi terusan (continue) dari intensitas cahaya pada bidang dua dimensi. Citra digital dapat didefinisikan sebagai fungsi dua variabel $f(x,y)$, dimana x dan y adalah koordinat spasial dan nilai $f(x,y)$ adalah intensitas citra pada koordinat tersebut. Sebuah citra diubah ke bentuk digital agar dapat disimpan dalam memori komputer atau media lain.[6]

$$f(x,y) \begin{pmatrix} f(0,0) & f(0,1) & f(0,2) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & f(1,2) & \dots & f(1,N-1) \\ f(2,0) & f(2,1) & f(2,2) & \dots & f(2,N-1) \\ \vdots & \vdots & \vdots & \dots & \vdots \\ f(M-1,0) & f(M-1,1) & f(M-1,2) & \dots & f(M-1,N-1) \end{pmatrix}$$

Dimana :

M = banyaknya baris pada array citra

N = banyaknya kolom pada array citra

2.3. Citra RGB

Citra RGB mempunyai array berukuran $m \times n \times 3$ yang mendefinisikan warna merah, warna hijau, dan warna biru untuk setiap pikselnya. Warna pada setiap piksel ditentukan dari kombinasi merah, hijau, dan biru. RGB merupakan citra 24 bit dengan komponen merah, hijau, dan biru disimpan masing-masing berukuran 8 bit yang berarti memiliki tingkat kecerahan warna sampai 256 level.

2.4. Konversi RGB ke Grayscale

Citra digital keabuan adalah citra yang mana nilai tiap piksel di dalamnya adalah berupa sampel tunggal. Hasil citra yang ditunjukkan jenis ini secara khusus terdiri atas warna kelabu yang terbagi dalam rentang warna dari hitam di intensitas yang paling lemah sampai putih dengan intensitas yang paling kuat. Perhitungan konversi RGB ke *grayscale* dirumuskan sebagai berikut: [7]

$$Grayscale = (0.299 * R) + (0.587 * G) + (0.114 * B) \quad (1)$$

Dimana, R (*Red*) ; G (*Green*) ; B (*Blue*)

2.5. Thresholding dan Binerisasi

Threshold (pengambangan) merupakan proses menjadikan citra skala abu-abu menjadi citra dengan nilai biner. Nilai satu mewakili warna putih, sedangkan nilai nol mewakili warna hitam. Berikut merupakan persamaan untuk menentukan nilai binerisasi berdasarkan nilai *threshold*. [6]

$$g(x,y) = \begin{cases} 1, & \text{if } f(x,y) \geq T \\ 0, & \text{if } f(x,y) < T \end{cases} \quad (2)$$

Dengan $g(x,y)$ adalah citra biner dari citra *grayscale* $f(x,y)$ dan T menyatakan nilai ambang (*threshold*). Nilai T memegang peranan yang sangat penting dalam proses pengambangan. Kualitas hasil citra biner sangat tergantung pada nilai T yang digunakan.

2.6. Segmentasi

Pada proses segmentasi ini dibagi menjadi dua bagian, yaitu segmentasi baris dan segmentasi karakter.

Segmentasi baris menelusuri citra dan melakukan pemotongan baris secara horizontal. Sedangkan segmentasi karakter melakukan pemotongan karakter dari hasil segmentasi baris.[8]

Tabel 1 Citra *input* sebelum proses pemotongan.

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

Setelah proses pemotongan, citra akan disegmentasi secara baris dengan cara menelusuri citra secara horizontal dan menjumlahkan setiap pikselnya. Apabila hasil penjumlahan tidak 0 maka terdapat baris teks disana.

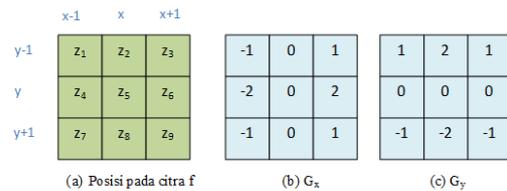
Tabel 2. Citra *input* setelah proses *cropping*

	2	3	4	5
2	1	1	1	1
3	0	0	0	0
4	0	0	0	0
5	0	1	1	0
6	1	1	1	1

Setelah disegmentasi, baris pertama dilakukan segmentasi terhadap karakter-karakter yang terdapat didalamnya. Segmentasi selanjutnya adalah segmentasi karakter, memisahkan tiap karakter berdasarkan baris yang telah dipotong. Segmentasi dilakukan dengan menjumlahkan setiap kolom.

2.7. Deteksi Tepi Sobel

Operator Sobel adalah salah satu operator yang digunakan dalam pengolahan citra dan visi komputer (*computer vision*), terutama dalam algoritma deteksi tepi, dan menciptakan sebuah gambar yang menekankan tepi dan transisi.



Gambar 1. Matrik Operator Sobel (Sumber : Crane, 1997)

Berdasarkan susunan piksel tetangga tersebut, operator Sobel menggunakan pembobotan pada piksel-piksel yang lebih dekat dengan titik pusat kernel. Oleh karena itu, pengaruh piksel-piksel tetangga akan berbeda sesuai dengan letaknya terhadap titik di mana gradient dihitung.[9]

$$G = \sqrt{Gx^2 + Gy^2} \quad (3)$$

Dimana M adalah besar gradient di titik tengah kernel dan turunan parsial dihitung menggunakan persamaan berikut:

2.7.1. Penebalan Citra (Dilasi)

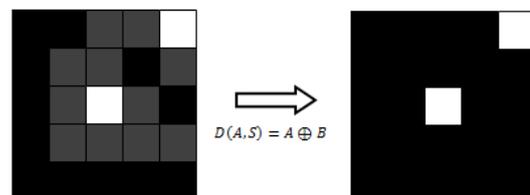
Dalam praproses citra digital, penebalan citra digunakan untuk objek piksel yang diinginkan pada citra biner dimana semua garis direduksi sehingga ketebalannya menjadi satu pixel. Cara dilasi adalah:

$$D(A,S)=A \oplus B \quad (4)$$

Untuk setiap titik pada A, lakukan hal berikut:

- Letakkan titik poros S pada titik A tersebut
- Beri nilai 1 untuk semua titik (x,y) yang terkena/tertimpa oleh struktur S pada posisi tersebut.

Ilustrasi dari proses penebalan dapat dilihat pada gambar 2 dan gambar 3.

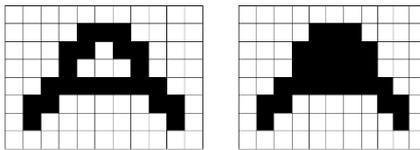


Gambar 2. Proses Dilasi

2.8. Proses Pengisian (Filling)

Proses *filling* dimulai dengan menentukan titik awal pengisian yang

terletak di dalam objek, kemudian bergerak ke arah titik-titik tetangganya. Citra yang memiliki lubang region akan diisi dengan menentukan titik awal pengisian yang terletak di dalam objek, kemudian bergerak ke arah titik-titik tetangganya. Operasi dilakukan secara rekursif dan berhenti jika sampai di batas objek. Perbedaan lokasi titik awal ini dapat mengakibatkan citra hasil pengisian yang diperoleh berbeda satu sama lain.



Gambar 3. Proses *Filling*

2.9. Ekstraksi Fitur Geometri

Ciri geometri merupakan ciri yang didasarkan pada hubungan antara dua buah titik, garis, atau bidang dalam citra digital. Ada beberapa cara umum yang digunakan untuk mengukur jarak dua buah titik pada citra. Salah satunya dengan menggunakan persamaan *euclidean* yang dapat dihitung dengan persamaan berikut :[10]

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (5)$$

Kombinasi beberapa sifat geometri objek dapat digunakan untuk menganalisis bentuk objek tersebut melalui perhitungan faktor bentuk tak berdimensi. Salah satu dari sifat geometri tersebut dapat diukur melalui analisis faktor bentuk tak berdimensi ketidaksamaan *isoperimetric* yang diwakili persamaan berikut :

$$R = \frac{4\pi A}{P^2} \quad (6)$$

Dimana, R = Nilai kekompakan ; A = Area (Jumlah piksel tiap baris) ; P = Parameter (Jumlah piksel dari batas daerah).

Dengan menganalisis faktor bentuk kekompakan, objek-objek dengan tepi yang rata akan memperlihatkan nilai yang berbeda dengan objek yang memiliki tepi bergerak. Objek dengan tepi yang rata akan memberikan nilai R sekitar 1, dan semakin mengecil bila tepinya tidak rata atau

membentuk lintasan yang berliku-liku.

2.10. JST Perceptron

Perceptron adalah salah satu model yang dapat diselesaikan secara sederhana dengan Matlab. Fungsi aktivasi sebuah perceptron dapat berupa nilai -1, 0 atau 1 (tapi akan berbeda saat melakukan implementasi pada Matlab, karena masukan dan target yang dipakai dapat bernilai bebas dan threshold yang dipakai adalah 0), dengan penentuan θ (*threshold*) sebagai berikut:[11]

$$f(net) = \begin{cases} 1, & \text{jika } net > \theta \\ 0, & \text{jika } -\theta \leq net \leq \theta \\ -1, & \text{jika } net < -\theta \end{cases} \quad (7)$$

Perceptron melakukan perubahan bobot secara terus menerus hingga menemukan bobot yang dipakai untuk mengenali pola secara maksimal. Perubahan bobot untuk seluruh pola disebut dengan epoch.

3. METODE PENELITIAN

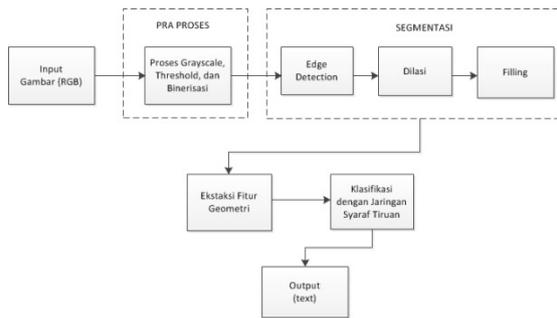
Penelitian ini menggunakan metode penelitian yang mencakup studi literatur, kemudian dilakukan analisa kebutuhan akan komponen-komponen yang dibutuhkan dalam pembuatan aplikasi pengenalan setiap karakter huruf alfabet berdasarkan pola citra dari huruf tulisan tangan hasil segmentasi dan ekstraksi fitur geometri setiap karakter huruf, selanjutnya perancangan sistem, kemudian dilakukan tahap implementasi dan pengujian, serta analisis. Tahap terakhir adalah kesimpulan dan saran.

4. PERANCANGAN SISTEM

Perancangan pada penelitian ini meliputi perancangan sistem pengenalan pola, praproses, segmentasi, proses, dan perancangan perangkat lunak.

4.1. Perancangan Sistem

Untuk mengenali pola pada huruf tulisan tangan dengan ekstraksi fitur geometri, dibutuhkan tahapan algoritma sistem. Algoritma dari sistem pengenalan pola pada huruf tulisan tangan yang akan dibangun seperti pada gambar 6:



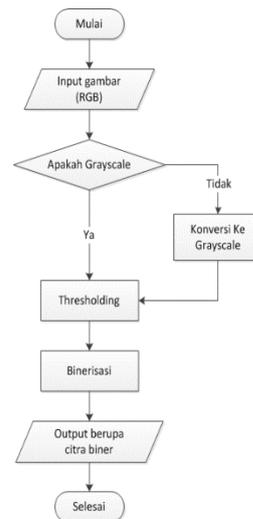
Gambar 4. Flowchart Sistem

Algoritma dari sistem pengenalan pola pada huruf tulisan tangan yang akan dibangun seperti pada gambar 6:

1. Melakukan input berupa citra tulisan tangan, jika citra merupakan citra RGB.
2. Sebelum diproses, Citra harus melalui beberapa tahapan, seperti *Grayscale*, *Thresholding*, dan Binerisasi.
3. Tahapan selanjutnya adalah proses segmentasi masing-masing objek yang akan dikenali yang terdiri dari proses deteksi tepi (*Edge Detection*) terhadap citra hasil segmentasi objek, proses dilasi yakni dengan melakukan penebalan piksel sehingga citra menjadi lebih besar dari citra sebelumnya, serta dilakukan proses *Filling* yakni dengan mengisi ruang kosong pada detail citra.
4. Citra yang dihasilkan dari praproses dan segmentasi disimpan ke dalam file net.mat.
5. Hasil praproses kemudian diekstraksi fitur dengan geometri dan akan menghasilkan data angka yang disimpan di featurout.mat
6. Untuk dapat mengenali karakter citra pada pola huruf tulisan tangan dengan jaringan syaraf tiruan, maka digunakan data dari featureout.mat.
7. Jaringan syaraf tiruan akan mengenali citra sesuai dengan mencocokkan pelatihan dari net.mat dengan data featureout.mat

4.2. Praproses

Praproses merupakan langkah yang dilakukan sebelum dilakukannya tahapan segmentasi dan proses.



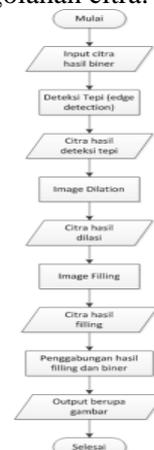
Gambar 5. Flowchart Praproses

Algoritma praproses dari gambar 4.2 adalah sebagai berikut :

1. Melakukan *grayscale* terhadap citra *input*, jika citra merupakan citra RGB, maka citra terlebih dahulu dikonversi ke skala abu-abu.
2. Melakukan proses pengambungan atau *thresholding* terhadap setiap piksel citra kemudian melakukan proses binerisasi yaitu mengubah citra ke skala hitam putih.
3. Binerisasi dihasilkan berdasarkan nilai *threshold*. Jika nilai nilai piksel *grayscale* > T, maka akan menghasilkan nilai 1. Jika *grayscale* < T, maka akan menghasilkan nilai 0.

4.3. Segmentasi

Sebelum melalui tahap proses, citra terlebih dulu disegmentasikan dengan operasi morfologi pengolahan citra.



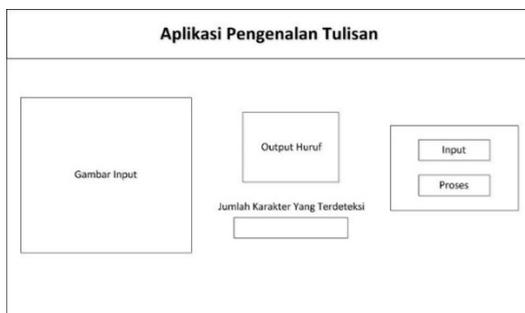
Gambar 6. Flowchart Segmentasi

Algoritma segmentasi dari gambar 8 adalah sebagai berikut:

1. Proses deteksi tepi (edge detection) dilakukan terhadap citra hasil biner.
2. Setelah proses deteksi tepi dilakukan maka citra akan dilakukan proses morfologi dilasi (image dilation) dengan melakukan penambahan piksel gambar sehingga citra gambar terlihat lebih besar dari ukuran sebelumnya.
3. Filling juga dilakukan pada tahapan ini, yakni dengan melakukan rekonstruksi ulang terhadap citra hasil dilasi dengan menutupi ruang kosong pada citra gambar.
4. Hasil citra dari hasil segmentasi akan disimpan di `featureout.mat`.

4.4. Perancangan Antarmuka

Perancangan antarmuka merupakan gambaran rancangan awal bagaimana struktur dari aplikasi pengenalan pola pada huruf tulisan tangan yang akan dibuat. Antarmuka dibuat dengan menggunakan GUI pada MATLAB R2015a. Perancangan aplikasi dapat dilihat pada gambar 9.



Gambar 7. Perancangan Antarmuka

Keterangan :

1. Tombol 'INPUT' : Tombol ini digunakan untuk mengunggah gambar dari direktori *folder* yang akan dijadikan *input* pengolahan.
2. Tombol 'PROSES' : Untuk menjalankan proses pengenalan pola pada huruf tulisan tangan.
3. Gambar Input : Axes untuk menampilkan gambar yang dipilih dari folder (gambar input).
4. Output Huruf : Axes untuk menampilkan gambar setelah proses binerisasi.
5. Jumlah karakter yang terdeteksi : Axes

untuk menampilkan jumlah karakter huruf yang terdeteksi pada citra input.

5. IMPLEMENTASI DAN PENGUJIAN

Implementasi dan pengujian sistem meliputi pelatihan jaringan syaraf tiruan, pelatihan sistem, dan proses pengujian sistem.

5.1. Implementasi Antarmuka

Setelah melalui tahap pelatihan data, diimplementasikan antarmuka atau interface untuk melakukan pengujian. Antar muka dibuat menggunakan MATLAB R2014a. Tampilan antar muka dapat dilihat pada gambar 11. Di bagian header aplikasi terdapat judul aplikasi yaitu "Aplikasi Pengenalan Tulisan Tangan NDOK".



Gambar 8. Tampilan Aplikasi

Dibagian bawah *header* sebelah kanan terdapat tombol navigasi "Gambar Input" untuk mengunggah gambar dari direktori folder. Gambar yang berhasil diunggah, akan tampil di kotak sebelah kiri yaitu axes "Foto Input". Aplikasi akan berjalan ketika tombol "Proses" ditekan. Tahap pertama citra *input* akan mengalami proses *pre-processing*, *threshold* dan penghapusan piksel, hasil dari proses tersebut akan tampil pada kotak yang berada ditengah atau axes "Hasil Ekstraksi Karakter". Selanjutnya aplikasi mulai mengenali satu-persatu karakter dan akan menampilkan jumlah objek yang terdeteksi tiap baris pada kotak axes "Jumlah Objek yang Terdeteksi Tiap Baris". Setelah proses pengenalan selesai, seluruh hasil teks akan tampil didalam *notepad* dengan nama *file* `output.txt`.

5.2. Implementasi Citra

Citra terlebih dahulu dinormalisasikan sehingga lebar piksel pada citra menjadi <700 piksel.

Kode Program 1. Menampilkan citra input

```
%% Membaca Foto
global foto
imagen=imread(foto);
[baris,kolom]=size(imagen);
while(kolom > 700)
    imagen = imresize(imagen,0.5);
    [baris,kolom]=size(imagen);
end
%% Menampilkan Foto
figure(1)
imshow(imagen);
title('INPUT IMAGE WITH NOISE')
```

5.2.1. Konversi Citra *Grayscale*

Citra RGB dikonversi ke skala abu-abu, dari citra dengan 3 komponen warna menjadi 1 komponen warna dengan rentang nilai 0-255 piksel.

Kode Program 1. Melakukan konversi ke grayscale

```
%% Konversi ke grayscale
if size(imagen,3)==3
    imagen=rgb2gray(imagen);
end
```

Nilai RGB pada citra yang dimasukkan bernilai sama dengan nilai *grayscale* karena citra masukkan hasil *scan* berwarna hitam dan putih.

5.2.2. *Threshold* dan Binerisasi

Fungsi *threshold* akan melakukan pengambangan terhadap citra hasil *grayscale*. Setelah nilai ambang didapat, *greyscale* dibandingkan dengan nilai *threshold* untuk menghasilkan nilai biner. Diketahui nilai piksel citra *grayscale* 15 dan nilai ambang (*Threshold*) dari 0-255 adalah 128, maka 15 < 128 menghasilkan biner bernilai 0.

Kode Program 2. Melakukan pengambangan dan binerisasi

```
%% Konversi ke biner
threshold = graythresh(imagen);
imagen = ~im2bw(imagen,threshold);

%% Menghapus objek yang dibawah 128 pixel
imagen = bwareaopen(imagen,15);
pause(1)
```

5.2.3. Deteksi Tepi Sobel

Metode yang digunakan untuk deteksi adalah metode sobel. Operator Sobel menggunakan pembobotan pada piksel-piksel yang lebih dekat dengan titik pusat kernel yaitu dengan matriks 3x3.

Kode Program 3. Proses Deteksi Tepi (*edge detection*)

```
%% Deteksi Tepi
Iedge = edge(uint8(imagen));
```

Apabila didapatkan nilai hasil perhitungan lebih dari 255, maka nilai yang akan digunakan adalah 255.

5.2.4. Penebalan Citra (*Dilasi*)

Pada bagian ini dilakukan proses penambahan atau penebalan piksel pada batasan dari objek terhadap citra *input* setelah proses deteksi tepi.

Kode Program 4. Proses Penebalan Citra (*image dilation*)

```
%% Morphology
% Dilasi
se = strel('square',3);
Iedge2 = imdilate(Iedge, se);
```

Setelah dilakukannya proses *image dilation*, maka citra *input* lebih besar ukurannya dibandingkan dengan citra hasil deteksi tepi dengan persamaan $D(A,S) = A \oplus B$ dan akan terlihat seperti gambar 13.

5.2.5. Pengisian (*Filling*)

Proses *filling* akan merekonstruksi citra untuk menutupi lubang-lubang pada citra.

Kode Program 5. Proses pengisian lubang citra (*image filling*)

```
% Filling
Ifill= imfill(Iedge2,'holes');
```

Image filling, akan menghasilkan sebuah bentuk karakter yang dapat dibedakan dengan bentuk karakter pada huruf lainnya.

5.3. Implementasi Antarmuka

Pengujian terhadap antar muka dilakukan dengan metode *blackbox*. Metode ini menguji apakah hasil *input* atau masukan sesuai dengan hasil *output* atau keluaran yang diharapkan.

Tabel 3. Tabel hasil pengujian antar muka aplikasi.

No	Masukan	Keluaran	Keluaran yang diharapkan	Keterangan
1	Data axes kosong	Tidak Terjadi Proses	Tidak Terjadi Proses	Berhasil
2	Tombol Proses	Hasil pengenalan pada <i>textbox</i>	Menampilkan karakter hasil pengenalan pada <i>textbox</i>	Berhasil
3	Data sesuai	Gambar tampil pada <i>axes</i>	Gambar unggahan tampil pada <i>axes</i>	Berhasil

5.4. Pelatihan Jaringan Syaraf Tiruan

Proses pelatihan data dilakukan membuat citra karakter yang terdiri dari huruf besar dalam alphabet (A-Z). Pelatihan data selanjutnya dengan melakukan pemotongan karakter terhadap karakter yang berjumlah 650 data. Data tersebut merupakan data latih huruf besar dengan masing-masing karakter terdiri dari 25 jenis tulisan yang berbeda. Tabel 4 menunjukkan pelatihan data terhadap citra input.

Tabel 4. Jaringan Pelatihan Data

PELATIHAN DATA	
Input	85
Hidden	35
Output	26

5.5. Ekstraksi Fitur Geometri

Fitur diekstrak secara terpisah untuk setiap zona 50x50px. Pada citra *input* terdapat 8 elemen pada vektor fitur untuk setiap zona yang dikategorikan dengan membagi matriks gambar dan pelabelan. Fitur itu kemudian diekstraksi untuk setiap zona. Langkah selanjutnya adalah mengekstraksi keseluruhan citra berdasarkan sifat regional yang didefinisikan dengan panjang dan lebar dari suatu objek dan luas area objek berdasarkan akumulasi nilai piksel yang merupakan bagian dari pada objek.

Kode Program 5.6. Mengekstraksi Fitur Citra.

```
axes(handles.axes5);
fid = fopen('output.txt','wt');
for n=1:Ne
    [r,c] = find(L==n);
    nl=imgn(min(r):max(r),min(c):max(c));
    BW2 = bwmorph(nl,'thin',Inf);
    imrotate(BW2,0);
    imshow(~BW2);
    z=imresize(BW2,[50 50]);
```

```
z=feature_extractor(z);

load ('featureout.mat');
featureout=z;
save ('featureout.mat','featureout');
test
pause(0.5);
end
if isempty(re)
    break
end
end
fclose(fid);
winopen('output.txt');
clear all
close (gcbf)
```

$$P = \text{jumlah piksel dari batas daerah} = 21$$

$$A = \text{jumlah piksel pada baris } 1 + \dots + \text{baris } 7$$

$$A = 1 + 3 + 5 + 5 + 7 + 4 + 2 = 27$$

$$R = \frac{4\pi A}{P^2} = \frac{4 * 3.14 * 27}{21^2} = 0,7690$$

Nilai hasil keseluruhan ekstraksi fitur akan tersimpan di *featureout.mat*.

5.6. Hasil Pengenalan Karakter

Hasil ekstraksi setiap karakter huruf pada *featureout.mat* diproses dengan net (hasil pelatihan) untuk menemukan karakter yang tepat.

Pengujian juga dilakukan terhadap sampel lain dengan bentuk jenis tulisan tangan yang berbeda. Hal ini dilakukan untuk menguji sistem apakah berjalan dengan yang diharapkan. Contoh sampel bentuk jenis tulisan tangan yang berbeda dapat dilihat pada tabel 5.

Tabel 5. Contoh Sampel Pengujian

No	Citra Input (.jpg)	Output Aplikasi (.txt)	Keberhasilan (%)
1	HERVIANA	HERVIAN A	100%
2	MASRANI	MASRANI	100%
3	UNIVERSITAS	UNIVERSI TAS	100%
4	TANJUNGPURA	TANJUNG PURA	100%
5	SISTEM KOMPUTER	SISTEMK OMPUTE R	100%
6	HERVIANA MASRANI	HERVIAN AMASRA NI	100%

7	ABC DEFGHIJKL MNOPQRST UVWXYZ	100%
8	ADJUSTMENT	ADJUSTMENT	90%
9	ARTIFICIAL	ARTIFICIAL	90%
10	BLESSINGS	BLESSINGS	77,77 %
11	CHAMPION LEAGUE	CHAMPION LEAGUE	92,85 %
12	FINAL	FIWAL	80%
13	GOOD	GOOD	100%
14	HI HELLO	HXHELLO	85.71 %
15	IN THE MORNING	INTHEMORNING	75%
16	LAST THINK	LAGPTHKNK	66,66 %
17	LIFE	LIFE	100%
18	LISTENING	LISTENDNG	88,88 %
19	LONG PROCESS	LOMGPROCESS	75%
20	MAGNIFICENT	MAGNIFICENT	100%
21	MISTAKE	NIDTRKE	57,14 %
22	NETWORK	KWETWOKE	57,14 %
23	NEURAL	NEUKAL	83,33 %
24	OPEN YOUR BOOK	OFEWYOURBOOK	83,33 %

25	OPPORTUNITY	OPFORTUNXTTY	75%
26	PROJECT	FKOJKCT	57,14 %
27	REAL MADRID	REALMADRID	100%
28	TALENT	TALENT	100%
29	WHO LAUGHS	HHKLAUGHS	77,77 %
30	SISKOM	SKXKOM	66,66 %

5.7. Pembahasan

Pengujian yang telah dilakukan pada penelitian “Aplikasi Pengenalan Pola Pada Huruf Tulisan Tangan Menggunakan Jaringan Saraf Tiruan Dengan Metode Ekstraksi Fitur Geometri” terhadap 30 tulisan tanganyang berbeda-beda mampu menghasilkan persentase keberhasilan 86,535%. Kesalahan pada keluaran dapat disebabkan karena penulisan yang tidak sesuai data pelatihan dan nilai *featureout* yang dihasilkan dari ekstraksi fitur geometri. Selain itu, salah dalam mengekstraksi jumlah fitur juga akan mengakibatkan kesalahan.

Faktor lain yang menunjang keberhasilan aplikasi dalam melakukan pengenalan karakter adalah bentuk penulisan memiliki pola karakter yang mirip dengan data latih. Oleh karena itu, aplikasi ini diperlukan banyak data pelatihan. Dengan semakin mirip penulisan karakter huruf tulisan tangan dan karakter yang tidak berdekatan, nilai akurasi dari identifikasi akan semakin tinggi tingkat keberhasilan yang diperoleh.

5.8. Analisa

Dari contoh sampel pengujian dengan bentuk tulisan tangan berbeda, aplikasi dapat mendeteksi citra huruf tulisan tangan yang lebih bervariasi dengan tingkat akurasi pengenalan karakter huruf sebesar 86,535% yang mengenali semua abjad yang terdiri dari 26 huruf. Kelemahan dari aplikasi terdapat pada pengenalan spasi, Tidak terbacanya

spasi maupun terbacanya spasi pada contoh kasus tersebut karena kurangnya data pelatihan. Jika ada kesalahan dalam pengenalan karakter artinya penulisan huruf tidak sesuai dengan pelatihan data dan nilai *featureout* yang dihasilkan.

6. KESIMPULAN DAN SARAN

6.1. Kesimpulan

Dari hasil penelitian yang dilakukan dengan menerapkan jaringan saraf tiruan dengan ekstraksi fitur geometri untuk pengenalan pola pada huruf tulisan tangan, dapat diambil kesimpulan bahwa :

1. Pada pengujian yang telah dilakukan, penulisan harus sesuai dengan data pelatihan pada setiap huruf besar yang jelas dan karakter yang tidak berdekatan mampu dikenali dengan baik oleh aplikasi tanpa terjadi kesalahan. Fitur huruf diekstrak secara terpisah untuk setiap zona 50x50 piksel dan pencocokan fitur geometri tersebut disimpan di dalam *featureout.mat* agar dapat diproses dengan jaringan syaraf tiruan untuk menemukan karakter huruf yang tepat.
2. Pengujian pada aplikasi dengan menerapkan jaringan syaraf tiruan dengan ekstraksi fitur geometri berhasil melakukan pengenalan karakter terhadap citra pada pengujian dengan tingkat keberhasilan 86,535%.
3. Pada pengujian "OPPORTUNITY" berjumlah 11 karakter yang ditulis menghasilkan 12 jumlah ekstraksi. Hal ini terjadi ketika huruf yang di tulis terpisah atau putus. Kesalahan pada identifikasi spasi disebabkan karena tidak terbacanya *database* spasi oleh aplikasi. Contoh pada *input* "SISTEM KOMPUTER" yang terbaca "SISTEMKOMPUTER" oleh aplikasi.

6.2. Saran

Adapun saran untuk penelitian kedepan-nya agar lebih baik adalah sebagai berikut :

1. Adanya pengembangan metode segmentasi khususnya segmentasi karakter untuk melakukan segmentasi

terhadap citra karakter yang berdekatan.

2. Mengenali pola huruf kecil, angka, dan tanda baca dengan tambahan data pada data pelatihan menggunakan metode yang sama dengan penelitian ini.

DAFTAR PUSTAKA

- [1] Prasojo, A. 2003. *Pengenalan Karakter Alfabet Menggunakan Jaringan Saraf Tiruan*. Jurnal Program Studi Teknik Elektro. UNDIP.
- [2] Yudha, A. 2014. *Pengenalan Huruf Tulisan Tangan Menggunakan Fuzzy Feature Extraction Dengan Pendekatan Radial Basis Function Neural*. Jurnal Program Studi Teknik Informatika. Universitas Bengkulu.
- [3] Fanani, A., Yuniarti, A., & Suciati. N. 2013. *Ekstraksi Fitur Geometri Pada Citra Batik menggunakan Representasi Cardinal Spline*. Jurnal Program Studi Teknik Informatika. ITS: Surabaya.
- [4] Kumar, S. 2009. *A Comperhensive On On-Line Handwriting Recognition Technology And Its Real Application To The Nepalese Natural Handwriting*. Kathmandu University Journal of Science, Engineering and Technology.
- [5] Richard O. Duda, Peter E. Hart, David G. Stork. (2001). *Pattern classification (2nd ed.)*. Wiley: New York
- [6] Gonzalez, R., & Woods, R. 2002. *Digital Image Processing (2nd ed.)*. Prentice Hall: New Jersey.
- [7] Kadir, Abdul., dan Susanto, Andi. 2012. *Teori dan Aplikasi Pengolahan Citra*. Yogyakarta: Andipublisher.
- [8] Gonzalez, R, & Wintz, P. 1987. *Digital Image Processing (2nd ed.)*. Addison-Wesley: Boston
- [9] Crane, R. 1997. *A Simplifield Approach to Image Processing Clasical and Modern Techniques*. Inc. Prentice Hall PTR: Upper saddle River.
- [10] Astuti, S. 2008. *Pengenalan Bentuk Geometri Benda Menggunakan Faktor Kebundaran*. Techno Com, Vol 7 No. 3.
- [11] Siang, JJ. 2004. *Aplikasi Jaringan Syaraf Tiruan dan Pemrograman Menggunakan MATLAB*. Yogyakarta: Andi.