

Terbit online pada laman web jurnal : <http://teknosi.fti.unand.ac.id/>

Jurnal Nasional Teknologi dan Sistem Informasi

| ISSN (Print) 2460-3465 | ISSN (Online) 2476-8812 |



Artikel Ilmiah

Ekstraksi Citra Digital Tulisan Mandarin Ke Dalam Terjemahan Tulisan Latin Bahasa Indonesia Dengan Stroke Extraction Dan Template Matching

Robert Robert¹, Ananda Ananda¹, Ibnu Surya¹

¹Politeknik Caltex Riau, Jl.Umban Sari No. 1, Pekanbaru, Indonesia

INFORMASI ARTIKEL

Sejarah Artikel:

Diterima Redaksi: 09 Maret 2018

Revisi Akhir: 29 Desember 2018

Diterbitkan Online: 31 Desember 2018

KATA KUNCI

*Digital Image Processing,**Stroke extraction,**Template Matching,**Receiver Operating Characteristic*

KORESPONDENSI

Telepon: 0819 9327 0562

E-mail: roberttjif@gmail.com

A B S T R A C T

Bahasa adalah sistem simbolik dari suara acak, yang digunakan oleh orang untuk berkomunikasi. Mandarin berada di peringkat pertama bahasa yang paling banyak digunakan menurut survei Nationalencyklopedin, dan Mandarin menggunakan huruf yang disebut Hanzi. Manusia memiliki berbagai cara berkomunikasi, salah satunya dengan menulis. Perangkat lunak ini dapat mengenali tulisan melalui gambar dengan menerapkan Pemrosesan Gambar Digital. Penelitian ini bertujuan untuk membangun aplikasi yang dapat mendeteksi Hanzi Mandarin dengan menerapkan *Digital Image Processing*. *Digital Image Processing* mengacu pada pemrosesan gambar dua dimensi menggunakan komputer. Metode pemrosesan gambar dengan *Stroke Extraction* dan *Template Matching* diterapkan solusi untuk masalah ini. *Stroke Extraction* digunakan untuk melakukan ekstraksi stroke dasar (pi hua) pada gambar Mandarin Hanzi dan kemudian mengenalinya menggunakan *Template Matching*. Aplikasi diuji dengan *Receiver Operating Characteristic* sehingga kurva yang dapat membantu proses analisis dapat ditarik. Berdasarkan hasil kurva ROC yang dijelaskan, metode *Stroke Extraction* masih membutuhkan pengembangan karena tingkat keberhasilan yang rendah dalam menghitung sejumlah stroke. Aplikasi yang dibangun berhasil mengekstraksi Mandarin Hanzi baik tulisan tangan dan dicetak dan diterjemahkan ke dalam bahasa Indonesia dengan akurasi 76,47%. Dapat disimpulkan bahwa metode *Stroke Extraction* dan *Template Matching* cukup efisien dalam mengolah gambar digital Mandarin Hanzi untuk diterjemahkan ke dalam bahasa Indonesia.

1. PENDAHULUAN

Bahasa merupakan sistem lambang bunyi arbitrer, yang digunakan oleh anggota suatu masyarakat untuk bekerja sama (KBBI). Banyaknya Bahasa di dunia kini berjumlah sekitar 7000 Bahasa yang mempunyai cara penulisan berbeda dan bahkan beberapa Bahasa seperti Bahasa Mandarin tidak menggunakan karakter latin melainkan karakter yang disebut dengan huruf hanzi. Perbedaan karakter yang digunakan mungkin tidak menjadi masalah besar pada kehidupan sehari-hari tetapi akan dapat menyulitkan kita ketika ingin digunakan pada komputer.

Mandarin merupakan salah satu Bahasa yang paling sering digunakan di dunia dan mempunyai tingkat kompleksitas yang sangat tinggi. Huruf yang terdapat pada Bahasa Mandarin saat

akan diterjemahkan ke Bahasa lain akan selalu terjadi kesulitan. Kesulitan yang terjadi timbul karena misalnya ketika penerjemah ingin menterjemahkan Bahasa Mandarin ke Bahasa Indonesia, penerjemah mengalami kendala karena ada 2 jenis aksara Mandarin (hanzi) yaitu aksara tradisional dan aksara sederhana yang memiliki arti sedikit berbeda satu sama lain.

Manusia dapat dengan mudah mengenali karakter Bahasa yang tak dikenalnya berdasarkan contoh yang ada dan Bahasa dengan menggunakan karakter latin mempunyai ciri karakter utama yang disebut Aplhabet yang merupakan ciri dari karakter latin, berbeda halnya dengan huruf Mandarin. Dalam proses pengenalan Citra huruf Mandarin tidak bisa hanya dengan template per huruf yang ada karena setiap satu huruf Mandarin merupakan kombinasi dari beberapa goresan dasar yang ada. Proses ekstraksi huruf

mandarin memiliki keakurasian tertinggi saat digunakan untuk hanzi mandarin dengan rata-rata tujuh goresan [3].

Bahasa mandarin menduduki peringkat pertama bahasa yang paling banyak digunakan dalam survey internasional. Karena banyaknya penggunaan Bahasa Mandarin diluar negara China aplikasi penerjemah seringkali digunakan untuk membantu pihak yang tidak bisa berbahasa Mandarin. Salah satu solusi melakukan terjemahan adalah dengan bantuan website penerjemah Bahasa seperti google-translate. Dalam melakukan terjemahan Bahasa Mandarin biasanya mandarin dibagi menjadi dua jenis yaitu tradisional dan modern. Terdapat beberapa huruf yang memiliki arti yang sama tetapi bentuk hanzi yang berbeda. Mandarin modern memiliki jumlah goresan yang lebih sedikit sehingga ada kalanya website tertentu yang tidak bisa melakukan terjemahan Bahasa Mandarin modern karena memiliki hanzi yang berbeda dari hanzi tradisionalnya.

2. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Penelitian [1] meneliti model baru yang dapat melakukan Pengolahan Citra Digital terhadap tulisan tangan menggunakan pendekatan SVM (Support Vector Machine). Dalam penelitian ini 4067 tulisan tanganhuruf alphabet dan 1857 tulisan tangan angka digunakan sebagai data training. Melalui penelitian ini peneliti berhasil menciptakan model ekstraksi tulisan tangan huruf alphabet yang mempunyai akurasi sebesar 96%, tetapi dalam penelitian yang dilakukan huruf alphabet yang dapat dikenali hanyalah huruf kapital.

Penelitian [2] yang bertujuan membangun aplikasi mobile yang dapat melakukan pengenalan karakter dan terjemahan dengan teknologi web service dan api google. Aplikasi yang dibangun berhasil melakukan ekstraksi terhadap Citra huruf alphabet dan menterjemahkannya menggunakan google translate dengan terhubung ke jaringan internet. Sistem yang dibangun memiliki rata-rata keberhasilan pengenalan sebesar 92,90%. Sistem yang dibangun memiliki kekurangan yaitu untuk melakukan proses terjemahan membutuhkan koneksi ke internet.

Adapun penelitian [3] yang membangun sebuah model ekstraksi khusus untuk huruf huruf Mandarin. Model yang diteliti berhasil melakukan ekstraksi goresan dasar pada huruf Mandarin. Ekstraksi akan lebih akurat apabila digunakan untuk melakukan ekstraksi goresan untuk huruf Mandarin yang memiliki rata-rata tujuh goresan dasar. Akan tetapi penelitian yang dilakukan hanyalah sebatas melakukan ekstraksi goresan dasar pada Citra huruf Mandarin sehingga untuk pengenalan hurufnya masih perlu diteliti lagi.

Penelitian lainnya dilakukan oleh [4] yang membangun sistem pengenalan karakter pada plat kendaraan bermotor menggunakan profile projection dan algoritma korelasi. Sistem yang dibangun dapat melakukan ekstraksi karakter pada Citra plat kendaraan bermotor. Dalam penelitian yang dilakukan peneliti mengalami kesulitan saat melakukan ekstraksi plat apabila lokasi jendela masking tidak sesuai dengan yang diharapkan. Penelitian tentang klasifikasi huruf korea dengan metode template matching [5]. Aplikasi ini dibuat oleh peneliti dengan tujuan mempermudah proses pembelajaran bahasa korea. Dari hasil ujicoba yang dilakukan oleh peneliti berhasil melakukan klasifikasi citra yang diinputkan ke huruf Hangul dengan tingkat keakurasian sebesar 83,33% dari 24 data uji.

Penelitian ini membangun sistem pengenalan karakter yang berfokus kepada huruf Mandarin aksara sederhana yang kemudian dapat diterjemahkan menjadi Bahasa Indonesia. Aplikasi akan dikembangkan dengan menggunakan deteksi tepi dan model Stroke Extraction. Aplikasi ini berfungsi untuk melakukan terjemahan dari Citra Bahasa Mandarin menjadi teks Bahasa Indonesia. Pengguna dapat memilih Citra huruf Mandarin yang ingin diproses, kemudian Citra inputan akan melalui beberapa proses yaitu proses binerisasi, segmentasi Citra, dan proses morfologi. Lalu dari Citra hasil tersebut dilakukan proses ekstraksi goresan sesuai dengan model yang ada pada penelitian yang dilakukan oleh [3]. Karakter yang didapat kemudian diterjemahkan dengan bantuan basis data yang ada. Pengguna kemudian mendapatkan hasil terjemahan dari citra inputannya.

Tabel 1. Penelitian Terdahulu

Peneliti	Metode Ekstraksi Citra
Cao & Tan (2000)	Stroke Extraction
Ayyaz, Javed, & Mahmood (2012)	Hybrid Support Vector Machine
Habibi (2013)	Google Neural Network, Template Matching
(Putri, Rafmadhanty, Megantara, Jayanti, & Hapsari (2014)	Template Matching Correlation
Angraini Silaen (2015)	Profile Projection
Sekarang	Template Matching Correlation, Stroke Extraction

2.2. Mandarin

Kata Mandarin merujuk kepada cara pengucapan Bahasa keluarga penduduk China dalam Bahasa barat. Selama berabad-abad Mandarin sudah dipergunakan sebagai Bahasa utama negara China dan baru diresmikan sebagai Bahasa nasional pada tahun 1955. Mandarin sebagai Bahasa nasional China dinegaranya dikenal dengan sebutan 'Putonghua' yang artinya Bahasa sehari-hari. Cara penulisan dan pengucapan Putonghua dibuat agar sesuai dengan kata-kata yang sering digunakan oleh pekerja atau petani. Pada provinsi Taiwan Mandarin dikenal dengan sebutan 'Guoyu' yang artinya 'Bahasa Nasional'. Putonghua dan Guoyu bersama menghasilkan Mandarin yang kita kenal sekarang ini [6]

2.3. Pengolahan Citra Digital

Pengolahan Citra Digital (*Digital Image Processing*) menunjuk pada pemrosesan gambar 2 dimensi menggunakan komputer. Pengolahan Citra dapat dibagi ke dalam tiga kategori yakni kategori rendah, menengah, dan tinggi [7].

- a. Kategori rendah
Meliputi operasi sederhana seperti prapengolahan Citra untuk mengurangi derau, kontras, dan ketajaman Citra. Output dan input pada kategori ini adalah Citra.
- b. Kategori menengah
Meliputi operasi seperti segmentasi dan klasifikasi Citra. Proses pada kategori menengah ini meliputi input berupa Citra dan output berupa atribut (fitur) Citra yang didapat dari Citra input.
- c. Kategori tinggi
Meliputi proses pengenalan dan deskripsi Citra.

2.4. Deteksi Tepi

Tepi (Edge) merupakan bagian dari Citra yang memiliki intensitas kecerahan berubah secara drastis. Deteksi tepi adalah proses untuk mendeteksi garis tepi yang membatasi dua wilayah pada suatu Citra[8]. Adapun tujuan dilakukan deteksi tepi, yaitu:

1. Untuk mendapatkan bagian yang menjadi fokus dalam Citra.
2. Untuk memperbaiki Citra yang kabur karena error atau efek proses akuisisi Citra.
3. Serta untuk mengubah Citra 2D menjadi bentuk kurva.

Citra merupakan representasi dari suatu matrix [9], sehingga dalam setiap Citra terdapat nilai tertentu pada tiap pixelnya yang merupakan nilai intensitas warna. Suatu piksel (x,y) dapat dikatakan tepi dari suatu Citra apabila piksel tersebut memiliki perbedaan intensitas nilai yang tinggi jika dibandingkan dengan piksel tetangganya. Berikut beberapa operator yang sering digunakan dalam melakukan proses deteksi tepi:

1. Operator Robert
Operator ini merupakan teknik differensial pada arah horizontal dan vertikal, dimana ditambahkan proses konversi biner setelah proses differensial. Konversi biner dilakukan guna mengubah warna dari suatu piksel menjadi hitam dan putih. Operator ini mengambil arah diagonal dalam melakukan perhitungan nilai gradient dalam matrik 2x2. Gx merupakan mask untuk sudut horizontal sedangkan Gy untuk sudut vertical.

$$Gx = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \dots\dots\dots(1)$$

$$Gy = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \dots\dots\dots(2)$$

2. Operator Prewitt
Operator Prewitt adalah pengembangan dari operator Robert dengan menggunakan High Pass Filter (HPF) yang memiliki satu angka 0 sebagai penyangga dan juga menggunakan persamaan yang sama dengan Operator sobel yang menggunakan matrik 3x3.

$$Gx = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \dots\dots\dots(3)$$

$$Gy = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \dots\dots\dots(4)$$

3. Operator Sobel
Operator sobel merupakan pengembangan dari operator Robert sama halnya dengan operator prewitt yang menggunakan filter HPF dengan satu angka nol sebagai penyangga. Kelebihan dari operator ini adalah adanya proses untuk mengurangi noise dalam Citra sebelum melakukan perhitungan deteksi tepi. Operator sobel menggunakan matrik konvolusi 3x3 dan susunan pixelnya di sekitar piksel (x,y) seperti pada formula matrik berikut:

$$Gx = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \dots\dots\dots(5)$$

$$Gy = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \dots\dots\dots(6)$$

Dengan bantuan Gx dan Gy maka nilai magnitude dapat ditentukan dengan rumus berikut:

$$|G| = \sqrt{Gx^2 + Gy^2} \dots\dots\dots(7)$$

Keterangan:

Gx = Nilai piksel hasil konvolusi secara horizontal

Gy = Nilai piksel hasil konvolusi secara vertikal

|G| = Nilai magnitude dari gradient

4. Operator Laplace
Operator Laplace merupakan operator turunan. Operator ini dapat mendeteksi tepi lebih akurat khususnya pada tepi yang curam. Pada operator ini bagian tepi curam, turunan keduanya memiliki *zerocrossing* (persilangan nol), dimana adanya pergantian tanda nilai turunan kedua dan pada tepi landau tidak ada persilangan nol. Matrik dibuat dalam 3x3 seperti berikut:

$$L = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \dots\dots\dots(8)$$

5. Operator Canny
Deteksi tepi Canny merupakan perkembangan dari deteksi tepi Sobel. Operator canny dikenal optimal dalam mendeteksi tepi. Dalam penggunaannya Canny menambahkan beberapa tahapan setelah mendapat output dari menerapkan operator sobel.

2.5. Deteksi tepi Canny

Deteksi tepi Canny dikembangkan oleh John F.Canny pada tahun 1986. Dikenal sebagai deteksi tepi yang bekerja optimal, berikut proses yang harus dilalui dalam menerapkan deteksi tepi canny:

1. Proses menghilangkan daerah dengan menggunakan Gaussian filter. Kernel Gaussian dengan besar matrik 5x5 yang akan digunakan. Kernel memiliki standar deviasi 1,4.

$$B = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} \dots\dots\dots(1)$$

2. Kemudian diterapkan operator Sobel dan mencari *gradient magnitude* Citra untuk mendapatkan wilayah dengan turunan spasial yang tinggi dengan menggunakan persamaan 7.
3. Selanjutnya dengan menggunakan invers tangen dari *gradient magnitude* Y (Gy) dibagi *gradient magnitude* X (Gx) akan didapat arah tepi. Arah tepi yang didapat lalu dipetakan ke 0, 45, 90, atau 135 derajat.

$$\theta = \arctan \left(\frac{|Gy|}{|Gx|} \right) \dots\dots\dots(2)$$

4. Non Maximum Suppression diterapkan untuk menghilangkan piksel-piksel yang dianggap tidak bagian dari tepi. Sehingga hanya tepi yang tebal yang tebal saja yang akan tetap ada. Prosesnya dilakukan dengan menelusuri wilayah yang didapat pada langkah 2 dengan bantuan arah yang didapat pada langkah 3, dan menghilangkan piksel yang tidak maksimum.

Langkah terakhir adalah menerapkan *Hysteresis Tresholding* yang merupakan proses tresholding dengan menggunakan dua nilai treshold yaitu ambang atas (T1) dan ambang atas (T2). Seluruh piksel dengan gradient dibawah T2 akan ditolak, piksel dengan nilai gradient diatas T1 akan diterima, dan nilai gradient piksel di antara T1 dan T2 hanya akan diterima apabila piksel bersinggungan dengan piksel yang memiliki nilai gradient diatas T1.

2.6. Template Matching

Template matching merupakan proses mencocokkan *template* dengan suatu Citra, dimana template merupakan bagian dari gambar yang memiliki bentuk yang berusaha kita temukan. Kecocokan didapat dari berapa banyak bagian dari gambar yang memiliki kecocokan dengan *template* yang ada. Proses dilakukan pada seluruh *template* yang ada, lalu nilai kemiripan tertinggi dan berada di atas nilai ambang batas pengenalan dari objek tersebut merupakan objek yang kita cari. Bila nilai kemiripan berada pada bawah nilai ambang batas maka objek tersebut dapat dikatakan tidak dapat dikenali. *Template matching* memiliki beberapa kelebihan tersendiri, yaitu:

1. Mudah dituliskan ke dalam Bahasa program dan mudah mempersiapkan data referensinya.
2. Komputasi tidak terlalu besar karena data yang digunakan berupa matriks.

Kecocokan antar dua buah matriks Citra dapat dihitung dengan menghitung nilai korelasinya dengan rumus dalam persamaan (3).

$$r = \frac{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (x_{ij} - \bar{x})(y_{ij} - \bar{y})}{\sqrt{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (x_{ij} - \bar{x})^2 \cdot \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (y_{ij} - \bar{y})^2}} \dots(3)$$

Keterangan:

- r = Nilai korelasi antar dua matrik
 x = Matrik acuan (template)
 y = Matrik masukan
 \bar{x} = Nilai rata-rata Citra acuan
 \bar{y} = Nilai rata-rata Citra masukan
 N, M = Jumlah *piksel* dalam suatu matrik

2.7. Stroke Extraction

Stroke extraction merupakan teknik yang banyak digunakan dalam pengenalan tulisan tangan. Algoritma ini sudah banyak dikembangkan untuk membantu dalam pengenalan tulisan tangan Mandarin. Tujuan dari penggunaan Stroke extraction dalam huruf Mandarin adalah memisahkan goresan dasar yang merupakan penyusun dasar untuk setiap huruf Mandarin. Proses yang harus dilakukan sebelum melakukan ekstraksi goresan adalah menerapkan teknik seperti *line thinning* untuk mendapatkan kerangka dalam Citra huruf yang akan diproses.



Gambar 1. Contoh huruf kerangka hasil *line thinning*

Dalam melakukan ekstraksi goresan dasar huruf Mandarin terdapat tahapan yang harus dilalui yaitu [3]:

1. Ekstraksi tiap goresan dalam Citra.
2. Pemisahan goresan yang saling bersinggungan.
3. Ekstraksi goresan dasar yang sempurna.

2.8. Receiver Operating Characteristic

Receiver Operating Characteristic (ROC) pada umumnya dipergunakan untuk menampilkan informasi kinerja suatu

algoritma klasifikasi ke dalam sebuah grafik. ROC berbeda dari confusion matrix yang hanya dapat menyajikan informasi dalam bentuk angka. Kurva ROC dibentuk dari nilai yang didapat dari perhitungan confusion matrix, yaitu antara *False Positive Rate* dan *True Positive Rate*. Kualitas kinerja algoritma klasifikasi dikatakan jelek apabila kurva yang dihasilkan mendekati garis yang melintang dari titik (0,0), dan bagus apabila kurva mendekati titik (0,1). Setelah itu akurasi dihitung dengan persamaan (6).

$$Akurasi = \frac{(TP+TN)}{(TP+TN+FP+FN)} \times 100\% \dots\dots\dots(4)$$

Kemudian dari nilai akurasi dapat dicari tingkat kesalahan deteksi atau *error rate*, dengan menggunakan persamaan 5.

$$Error Rate = 100\% - Akurasi \dots\dots\dots(5)$$

Receiver Operating Characteristic (ROC) pada umumnya dipergunakan untuk menampilkan informasi kinerja suatu algoritma klasifikasi ke dalam sebuah grafik. ROC berbeda dari confusion matrix yang hanya dapat menyajikan informasi dalam bentuk angka. Kurva ROC dibentuk dari nilai yang didapat dari perhitungan confusion matrix, yaitu antara *False Positive Rate* dan *True Positive Rate*. Kualitas kinerja algoritma klasifikasi dikatakan jelek apabila kurva yang dihasilkan mendekati garis yang melintang dari titik (0,0), dan bagus apabila kurva mendekati titik (0,1). Berikut persamaan untuk mencari jarak ke titik (0,1).

$$Jarak ke (0,1) = \sqrt{(FP Rate)^2 + (1 - TP Rate)^2} \dots\dots(6)$$

3. METODOLOGI

3.1. Input Citra tulisan huruf mandarin

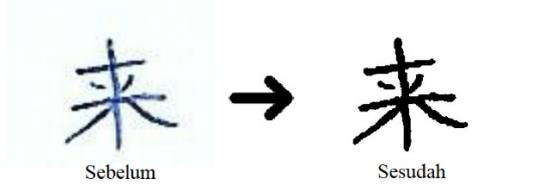
Citra tulisan huruf Mandarin diambil dengan menggunakan kamera, scanner atau diupload langsung oleh pengguna. Inputan Citra dilakukan dengan mengupload Citra melalui *file picker* yang disediakan atau dengan menggunakan webcam.

3.2. Pre-processing

Pre-processing merupakan tahapan dimana dilakukan perbaikan kualitas Citra. Tahapan pertama dalam pre-processing disini adalah proses binerisasi. Proses binerisasi merupakan proses mengubah Citra RGB menjadi Citra biner yang hanya mempunyai nilai warna 1 atau 0 yaitu warna hitam dan putih. Proses ini dimulai dengan menentukan nilai treshold yang akan dilakukan dengan Metode Iterasi. Berikut langkah yang harus dilakukan untuk menentukan nilai treshold dengan metode iterasi [10]:

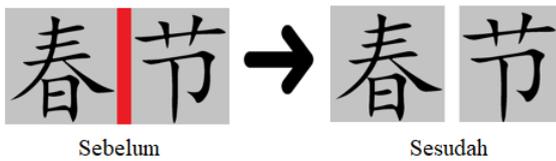
1. Pertama nilai awal treshold ditentukan dari rata rata dari nilai keseluruhan Citra.
2. Citra kemudian dibagi menjadi dua daerah (R1 dan R2) dengan menggunakan nilai awal treshold pada langkah 1.
3. Lalu hitung nilai rata-rata intensitas untuk tiap daerahnya (μ_1 dan μ_2)
4. Kemudian hitung nilai treshold baru dari penjumlahan nilai rata-rata intensitas dibagi 2.
5. Kemudian ulangi langkah 2 hingga 4 sampai nilai rata-rata intensitas (μ_1 dan μ_2) tidak lagi berubah. Kemudian nilai treshold digunakan untuk mengubah

Citra inputan menjadi Citra keabuan sesuai dengan nilai pada tiap piksel.

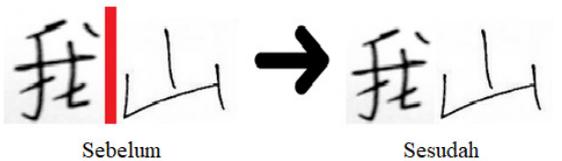


Gambar 2. Citra sebelum dan sesudah proses binerisasi

Kemudian dilakukan proses slicing guna untuk memotong dan memisahkan huruf Mandarin yang menjadi objeknya dari latarnya. Dan juga proses slicing akan dilakukan untuk citra inputan dengan lebih dari satu huruf mandarin dengan melihat jarak antara huruf secara horizontal. Huruf yang berhasil dipisahkan akan disimpan dan diproses secara bergiliran.



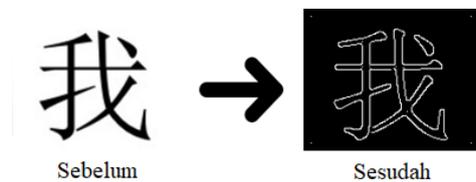
Gambar 3. Hasil pemisahan huruf mandarin hasil cetak



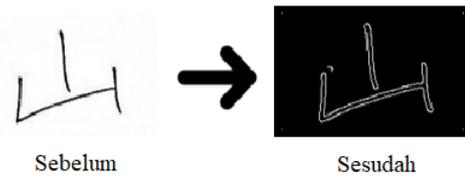
Gambar 4. Hasil pemisahan huruf mandarin tulisan tangan

3.3. Segmentasi Citra (Algoritma Canny)

Segmentasi Citra dilakukan dengan tujuan untuk mengambil daerah pokok dalam suatu Citra. Deteksi tepi dilakukan dengan algoritma Canny pada Citra yang telah melalui tahap *pre-processing*. Pada algoritma Canny tahap pertama adalah melakukan filter Gaussian agar derau yang ada berkurang. Kemudian dengan operator Sobel akan didapat *gradient magnitude* dan arah tepinya. Lalu nilai yang didapat dari operator Sobel digunakan untuk mendapatkan Non Maximum Suppression untuk menghilangkan nilai yang tidak mencapai batas atas ruang warna. Berikut contoh citra huruf mandarin hasil cetak dan tulisan tangan setelah menerapkan deteksi tepi canny.



Gambar 5. Hasil Deteksi tepi Canny huruf cetak Mandarin

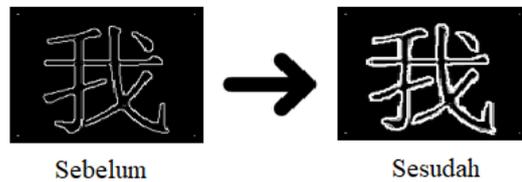


Gambar 6. Hasil deteksi tepi Canny tulisan tangan huruf Mandarin

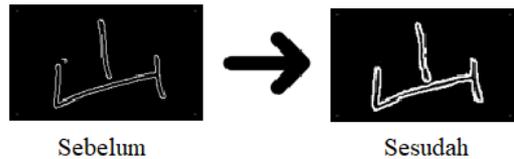
Setelah tepi didapat maka selanjutnya akan dilakukan proses segmentasi untuk pengambilan Citra huruf Mandarin berdasarkan pada tepi huruf Mandarin yang terdeteksi.

3.4. Proses morfologi

Tahap selanjutnya adalah melakukan proses Dilasi dan *Filling Holes* ke Citra inputan. Dalam prosesnya dilasi akan memperbesar batas dari objek dalam Citra huruf Mandarin. Contoh penggunaan dilasi dapat dilihat pada Gambar 7 dan Gambar 8.

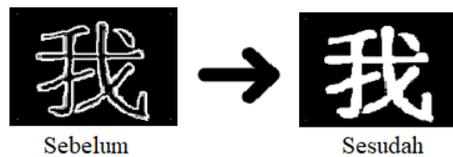


Gambar 7. Hasil proses dilasi huruf cetak Mandarin

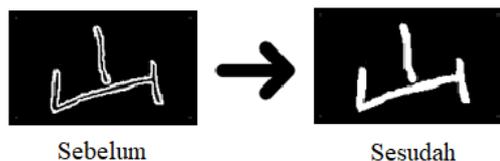


Gambar 8. Hasil proses dilasi tulisan tangan huruf Mandarin

Setelah tepi objek pada citra mandarin dipertebal maka selanjutnya bagian tengah harus diisi dengan pixel yang serupa dengan pixel tepi. Kemudian dilanjutkan dengan proses *filling holes* untuk mengisi bagian dalam dari batas objek. Contoh penerapan *filling holes* dapat dilihat pada Gambar 9 dan Gambar 10.



Gambar 9. Hasil proses Filling Holes huruf cetak Mandarin

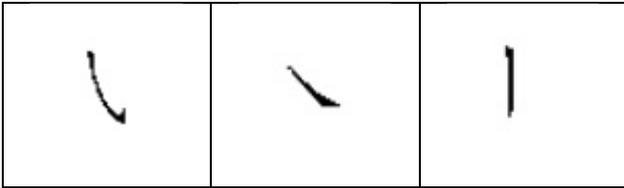


Gambar 10. Hasil proses Filling Holes tulisan tangan Mandarin

Kedua proses morfologi ini dilakukan dengan tujuan mempermudah proses selanjutnya. Setelah dilakukan penebalan maka selanjutnya adalah ekstraksi goresan.

3.5. Stroke Extraction

Stroke Extraction merupakan teknik yang dapat digunakan untuk mengekstrak goresan dasar dari Citra huruf Mandarin. Dalam proses ini goresan dasar yang didapat akan dihitung agar dapat digunakan sebagai salah satu variabel untuk mempermudah proses pengenalan pola. Adapun proses yang akan dilalui dalam melakukan ekstraksi goresan adalah melakukan proses template matching, apakah goresan yang menjadi template ada pada citra inputan.



Gambar 11. Contoh Citra goresan yang menjadi template

Goresan yang digunakan sebagai template berjumlah 13 goresan. Template matching yang dilakukan tidak membandingkan korelasi dari citra inputan dengan 13 citra goresan, tetapi mencari apakah citra goresan ada pada citra inputan. Apabila citra goresan ditemukan pada citra inputan maka jumlah goresan bertambah satu.

3.6. Template Matching

Template Matching digunakan dalam mendeteksi huruf Mandarin yang ada dalam Citra inputan dengan membandingkan Citra inputan dengan seluruh Citra pada *Knowledge Base* yang memiliki jumlah goresan yang sama dengan jumlah goresan yang didapat dari proses *Stroke Extraction*. Contoh untuk huruf wo pada Gambar 12 yang memiliki 7 goresan dasar maka hanya akan dicocokkan dengan citra template dengan 7 goresan dasar.



Gambar 12. Hanzi Wo dengan 7 goresan

Citra inputan dicocokkan dengan seluruh citra template dengan goresan yang sama kemudian dicari nilai korelasi tertinggi. Nilai korelasi tertinggi berarti citra inputan dan citra template tersebut memiliki hubungan yang paling erat, sehingga disimpulkan citra inputan merupakan huruf yang ada pada citra template.

3.7. Menterjemahkan teks ke bahasa Indonesia

Sistem akan menampilkan huruf Mandarin yang berhasil terdeteksi dalam proses sebelumnya. Proses terjemahan ini akan akan berjalan linear dengan proses sebelumnya karena *Knowledge Base* yang akan digunakan sudah berisi arti dari setiap huruf Mandarin yang menjadi templatnya. Hasil terjemahan kemudian ditampilkan ke user.

4. HASIL DAN PEMBAHASAN

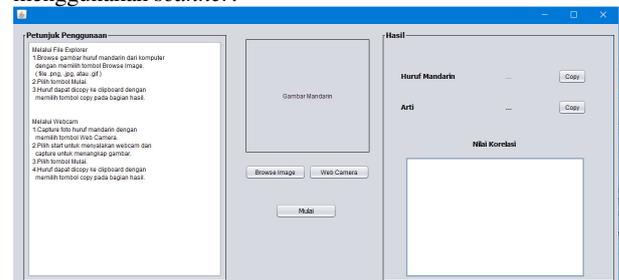
4.1. Proses Aplikasi

Aplikasi dibangun dengan bahasa pemrograman java dengan Netbeans IDE dengan bantuan library OpenCV. Citra yang digunakan pada pengujian didapat dari hasil scan tulisan tangan

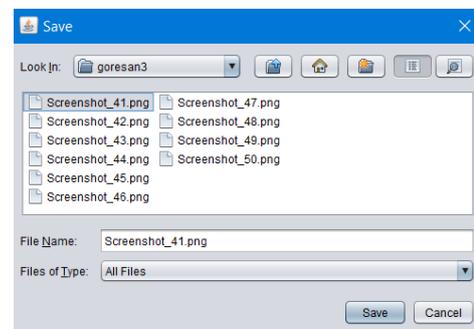
huruf Mandarin. Proses dijabarkan dari proses input citra hingga aplikasi melakukan terjemahan ke bahasa Indonesia.

4.1.1. Input Citra

Tampilan awal aplikasi dapat dilihat pada Gambar 13. Citra tulisan mandarin yang digunakan merupakan hasil scan dengan menggunakan *scanner*.

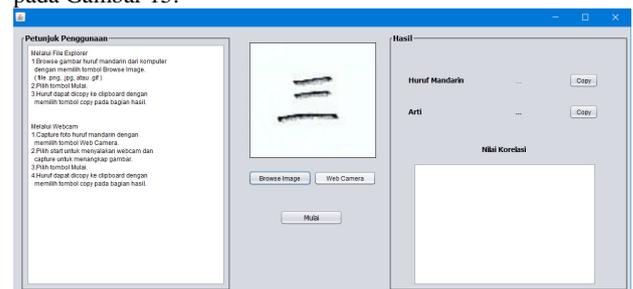


Gambar 13. Tampilan aplikasi Pengguna memilih citra tulisan mandarin yang akan dideteksi dengan memilih tombol *Browse Image*. Posisi *fileChooser* akan mengarah ke *home* pengguna saat pertama kali dibuka. Tampilan *file chooser* dapat dilihat pada Gambar 14.



Gambar 14. Memilih Gambar

Setelah pengguna memilih gambar maka gambar tersebut akan ditampilkan di aplikasi. Tampilan setelah file dipilih dapat dilihat pada Gambar 15.



Gambar 15. Tampilan gambar yang terpilih

4.1.2. Pre-processing

Tahap pre-processing terbagi menjadi 2 proses yaitu *Slicing* dan *Binerisasi*. Proses *slicing* akan dilakukan pada gambar yang lebarnya 1,6 kali lebih panjang dari tingginya menjadi 2 citra yang kemudian diproses satu-persatu, hal ini diterapkan sehingga aplikasi dapat mengolah citra yang memiliki 2 hanzi mandarin. Nilai threshold untuk lebar adalah 1,6 dan nilai ini didapatkan dari melakukan percobaan pada seluruh citra yang memiliki 2 hanzi mandarin dari treshold 1,1 hingga 2,4, dan hasilnya pada nilai 1,6 seluruh citra hanzi yang memiliki 2 hanzi mandarin berhasil dipisah. Berikut code untuk melakukan proses *slicing*. Berikut potongan code untuk proses *Slicing*.

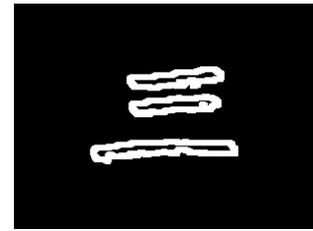
```

Mat source = Imgcodecs.imread(path,
Imgcodecs.CV_LOAD_IMAGE_GRAYSCALE);
double x , y, total;
double batas = 1.6;
x = source.height();
y = source.width();
total = y/x;
if(total>=batas){

    Rect roi = new Rect(0, 0, source.width()/2,
source.height());
    Rect roi2 = new Rect(source.width()/2, 0,
source.width()/2,
source.height());

    terbelah2[0] = new Mat(source, roi);
    terbelah2[1] = new Mat(source,roi2);
}

```



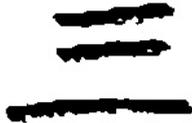
Gambar 18. Citra hasil Dilasi

Kemudian objek pada citra dilasi dipisahkan dari backgroundnya. Proses ini dilakukan dengan menemukan posisi piksel pertama yang berwarna putih dari empat arah yaitu atas, bawah, kiri, dan kanan. Kemudian dengan posisi tersebut citra diambil dalam bentuk persegi. Hasil pemisahan objek dapat dilihat pada gambar 19.



Gambar 19. Citra hasil pemisahan Objek dengan Background

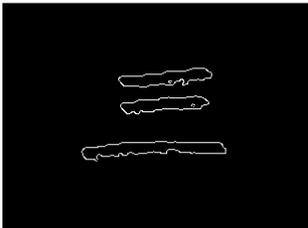
Selanjutnya citra inputan akan melalui proses binerisasi. Proses binerisasi mengubah citra inputan menjadi citra biner atau biasa disebut citra hitam putih. Nilai *threshold* yang digunakan adalah 211, sehingga seluruh nilai warna pixel yang diatas 211 akan menjadi warna hitam dan pixel dengan nilai warna dibawah 211 akan diubah menjadi warna putih. Nilai *threshold* 211 tersebut didapat setelah dilakukan percobaan dari nilai *threshold* 0 hingga 255 pada citra testing yang digunakan, dimana pada angka 211 hanzi mandarin tampak jelas.



Gambar 16. Citra hasil proses binerisasi

4.1.3. Segmentasi Citra (Algoritma Canny)

Pada tahap ini citra inputan akan melalui proses deteksi tepi dengan algoritma Canny, dengan tujuan untuk mendapatkan tepi pada hanzi mandarin pada citra inputan. Hasil deteksi tepi dapat dilihat pada Gambar 17.



Gambar 17. Citra hasil deteksi tepi Canny

4.1.4. Proses morfologi

Pada proses deteksi tepi ada beberapa hanzi mandarin yang memiliki cacat setelah didapat tepinya, dimana ada lubang pada tepi yang tergambar. Pada proses morfologi citra inputan akan melalui 2 tahapan untuk memperbaiki lubang hasil deteksi tepi, yaitu:

1. Dilasi

Proses dilasi akan menebalkan tepi dari objek untuk menutup lubang-lubang yang terdapat pada citra hasil deteksi tepi. Hasil proses dilasi dapat dilihat pada Gambar 18.

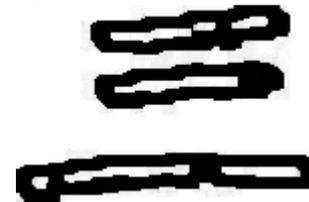
2. Fill Holes

Pada tahap ini piksel yang berada di dalam garis tepi dilasi akan diisi dengan piksel putih. Pada proses ini aplikasi akan mencari lubang dengan lebar dan tinggi tertentu untuk diubah menjadi piksel putih. Hasil proses *fill holes* dapat dilihat pada Gambar 20.



Gambar 20. Citra hasil proses Fill Holes

Hasil *fill holes* pada gambar diatas tidak tertutup secara sempurna karena menggunakan ukuran 3x3 pixel sebagai acuan apakah lubang harus ditutup, dan apabila ukuran diperbesar lagi akan berakibat pada hanzi mandarin menjadi menyambung satu sama lain antar goresan. Kemudian dilakukan proses inverse untuk membalikan nilai warna, proses ini dilakukan dikarenakan citra template memiliki objek dengan warna hitam. Hasil proses *inverse* dapat dilihat pada Gambar 21.



Gambar 21. Citra hasil Proses Inverse

4.1.5. Stroke Extraction

Pada tahapan ini jumlah goresan yang terdeteksi akan dihitung untuk membantu proses *template matching*. Pertama citra hasil

inverse akan melalui proses *resize* menjadi ukuran 100x100 piksel, kemudian citra inputan akan melalui proses template matching untuk menemukan citra goresan yang terdapat pada citra inputan. Jumlah goresan yang terdeteksi akan melalui proses pembulatan. Proses pembulatan dilakukan dengan tujuan mengurangi jumlah kesalahan ekstraksi goresan yang dapat terjadi, sehingga apabila goresan yang terdeteksi lebih dari 7 maka citra tersebut disimpulkan memiliki 7 goresan. Berikut merupakan code untuk melakukan proses pembulatan jumlah goresan.

```
int goresanbulat;

if(gores2>=7){
    goresanbulat = 7;
}else if(gores2<=3){
    goresanbulat = 3;
}else{
    goresanbulat = 5;
}
gores2 = 0;
return goresanbulat;
```

4.1.6. Deteksi Citra

Proses deteksi citra akan dilakukan dengan bantuan template matching korelasi. Citra inputan akan dibandingkan dengan citra template yang memiliki jumlah goresan yang sama dengan goresan yang terdeteksi. Jumlah goresan yang terdeteksi pada citra inputan adalah tiga sehingga citra hanya akan dibandingkan dengan citra template yang memiliki 3 goresan. Proses perbandingan ini dilakukan dengan menggunakan persamaan (3). Berikut tabel hasil perhitungan korelasi antara citra inputan dengan citra template. Contoh hasil perhitungan korelasi citra inputan dengan citra template dapat dilihat pada tabel 1.

Tabel 1. Hasil perhitungan korelasi terkecil hingga terbesar

Nama huruf	Nilai korelasi
Kou	0.4453
Men	0.4693
Ma	0.474175
Kou	0.49075
Cai	0.51725
Cai	0.549825
Men	0.55065
Gan	0.57275
Qian	0.575475
Wan	0.5775
Wan	0.58885
Qian	0.6023
Wen	0.607975
Wen	0.610275
Ma	0.61465
Gan	0.6308
Tu	0.63865
San	0.645775
Tu	0.6513
San	0.7281

Dari tabel diatas, aplikasi mendeteksi citra inputan merupakan citra huruf San yang memiliki nilai korelasi yang terbesar saat dibandingkan dengan citra template San.

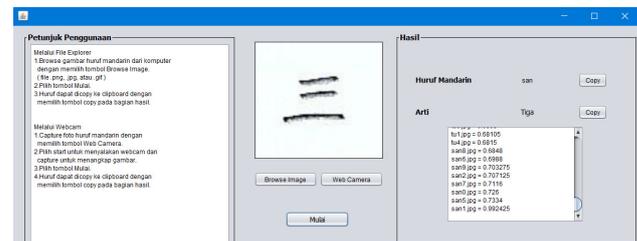
4.1.7. Terjemahan ke Bahasa Indonesia

Setelah proses pendeteksian citra maka didapatkan hasil berupa nama huruf mandarin yang telah terdeteksi, sistem kemudian

mengambil terjemahan Bahasa Indonesia dari huruf yang terdeteksi dari basis data. Berikut potongan code untuk mengambil hasil terjemahan bahasa Indonesia.

```
public static String ambildata(String kode,int gores) throws
SQLException{
    try{
        Connection conn = ConnectDb();
        PreparedStatement sta = conn.prepareStatement(
        "SELECT huruf,nama_huruf,arti FROM template");
        ResultSet rs = sta.executeQuery();
        String str = "";
        while(rs.next()){
            str = rs.getString("nama_huruf");
            if(kode.contains(str)){
                hasilapp[0] = rs.getString("huruf");
                hurufbesar = rs.getString("arti");
                hasilapp[1] = Character.toUpperCase(
                hurufbesar.charAt(0)) + hurufbesar.substring(1);
            }
        }
    }catch(Exception e){
        JOptionPane.showMessageDialog(null,"Periksa
        koneksi ke database!","Koneksi
        Gagal",JOptionPane.ERROR_MESSAGE);
    }
    return hasilapp[0];
}
public String getArti(){
    return hasilapp[1];
}
```

Aplikasi kemudian menampilkan hasil terjemahan dan nama huruf mandarin pada kotak hasil, dan tombol *copy* dapat digunakan untuk memudahkan mengambil hasil deteksi.



Gambar 22. Hasil Akhir Aplikasi

4.2. Pengujian

Pengujian pertama dilakukan pada 450 citra hanzi mandarin yang ditulis oleh 15 orang dengan tujuan untuk mendapatkan data training yang baik. Data training yang digunakan adalah 30 huruf yang ditulis langsung oleh seorang ahli mandarin, dan dari hasil pengujian diambil 300 citra hanzi mandarin yang memiliki akurasi yang baik untuk dijadikan citra template pada pengujian selanjutnya.

Tabel 2. Pengujian pertama

Nama penulis	Terdeteksi	Akurasi (%)
Linda	19	63,33333333
Sinria	19	63,33333333
Winnie	18	60
Leony	17	56,66666667
Devi	16	53,33333333
Sally	15	50
Yumico	15	50

Daysi	14	46,66666667
Vivian	13	43,33333333
Cindy	12	40
Fandi	12	40
Wandi	12	40
Delvin	11	36,66666667
Raymond	11	36,66666667
Irvandi	10	33,33333333

Lalu kuisioner disebar ke tempat les mandarin cia yin mandarin dan mandarin channel, sehingga total data yang ada adalah 930 data. Lalu dicari huruf mandarin tulisan cetak sebanyak 90 data dimana 30 data akan digunakan sebagai citra training untuk huruf cetak dan 60 data digunakan sebagai data testing. Total seluruh data yang akan diuji adalah 1020 citra hanzi mandarin.

Pengujian dilakukan dengan menggunakan 1020 citra testing yang akan melalui tahap pengujian, dimana 330 merupakan data training dan 690 merupakan data testing. Pengujian dilakukan pada data testing maupun data training sesuai dengan prinsip pengujian ROC. Aplikasi memiliki 330 citra template huruf mandarin untuk proses deteksi citra dan 11 citra template goresan mandarin untuk proses *stroke extraction*.

Pertama untuk mendapat jarak ke titik(0,1) dibutuhkan nilai TP rate dan FP rate, dimana TP rate merupakan tingkat deteksi aplikasi untuk citra yang merupakan data training (jumlah citra terdeteksi dibagi jumlah citra training), dan FP rate merupakan tingkat deteksi aplikasi untuk citra yang bukan data training (jumlah citra terdeteksi dibagi jumlah citra testing yang bukan data training).

Tabel 1. Pengujian ROC

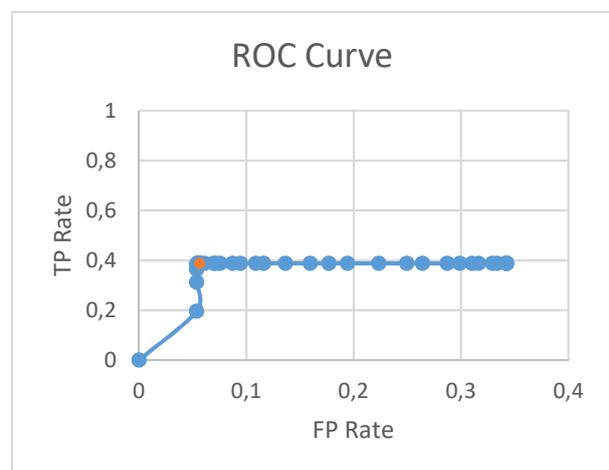
Treshold	TP Rate	FP Rate	Jarak Ke (0,1)
0.99	0	0	1
0.98	0,196969697	0,053623188	0,804818684
0.97	0,312121212	0,053623188	0,689965704
0.96	0,363636364	0,053623188	0,638618919
0.95	0,384848485	0,053623188	0,617484277
0.94	0,387878788	0,055072464	0,61459365
0.93	0,387878788	0,055072464	0,61459365
0.92	0,387878788	0,055072464	0,61459365
0.91	0,387878788	0,055072464	0,61459365
0.90	0,387878788	0,055072464	0,61459365
0.89	0,387878788	0,055072464	0,61459365
0.88	0,387878788	0,055072464	0,61459365
0.87	0,387878788	0,055072464	0,61459365
0.86	0,387878788	0,056521739	0,614725211
0.85	0,387878788	0,057971014	0,61486016
0.84	0,387878788	0,060869565	0,615140214
0.83	0,387878788	0,069565217	0,61606144
0.82	0,387878788	0,071014493	0,616226774
0.81	0,387878788	0,075362319	0,616742943
0.80	0,387878788	0,086956522	0,618266783
0.79	0,387878788	0,094202899	0,61932751

0.78	0,387878788	0,108695652	0,621696971
0.77	0,387878788	0,115942029	0,623004761
0.76	0,387878788	0,136231884	0,627097683
0.75	0,387878788	0,15942029	0,632540281
0.74	0,387878788	0,176811594	0,637145759
0.73	0,387878788	0,194202899	0,642189337
0.72	0,387878788	0,223188406	0,651540822
0.71	0,387878788	0,249275362	0,660931604
0.70	0,387878788	0,263768116	0,666532818
0.69	0,387878788	0,286956522	0,676044691
0.68	0,387878788	0,298550725	0,681046925
0.67	0,387878788	0,310144928	0,686208609
0.66	0,387878788	0,315942029	0,688848128
0.65	0,387878788	0,328985507	0,694927221
0.64	0,387878788	0,333333333	0,696996047
0.63	0,387878788	0,342028986	0,701196267
0.62	0,387878788	0,342028986	0,701196267
0.61	0,387878788	0,342028986	0,701196267
0.60	0,387878788	0,342028986	0,701196267

*Nilai threshold berwarna hijau merupakan threshold optimal.

Kemudian dicari nilai *threshold* optimal dari rentang nilai 0,99 hingga 0,60, dan dihitung *TP rate* dan *FP rate* pada tiap *threshold*. Pada pengujian ini angka *threshold* berhenti diuji pada titik 0,6 karena dari titik 0,6 hingga 0 tidak lagi terdapat perubahan nilai *TP rate* dan *FP rate*. Lalu dicari jarak ke titik (0,1) dengan menggunakan persamaan (6) sehingga dapat diketahui angka *threshold* optimal. Nilai *threshold* ini akan digunakan untuk perhitungan akurasi dimana nilai korelasi hasil deteksi akan dibandingkan dengan nilai *threshold*, dan jika nilai korelasi lebih kecil maka data sistem akan menyimpulkan bahwa citra inputan bukanlah huruf mandarin yang pernah di training. Hasil pengujian dapat dilihat pada tabel 3.

Nilai *threshold* yang optimal memiliki jarak (0,1) yang paling kecil atau terdekat, dan nilai *threshold* 0,87 memiliki jarak terpendek ke titik (0,1). Kemudian dari tabel diatas dapat digambarkan kurva ROC dengan bantuan dari nilai *TP rate* dan *FP rate* yang didapat pada *threshold* 0,87 seperti pada gambar 23.



Gambar 23. Kurva ROC

Kurva ROC diatas digambarkan dengan nilai FP rate sebagai sumbu x dan nilai TP rate sebagai sumbu y. Titik berwarna oranye merupakan titik yang memiliki jarak (0,1) terdekat. Nilai FP rate dan TP rate dipengaruhi oleh besarnya nilai *threshold*. Titik (0,1) juga biasa disebut *perfect classification* yang tidak mengandung nilai *False Positive* maupun *False Negative*, sehingga semakin dekat jaraknya akan mengurangi tingkat *False Positive* dan *False Negative*.

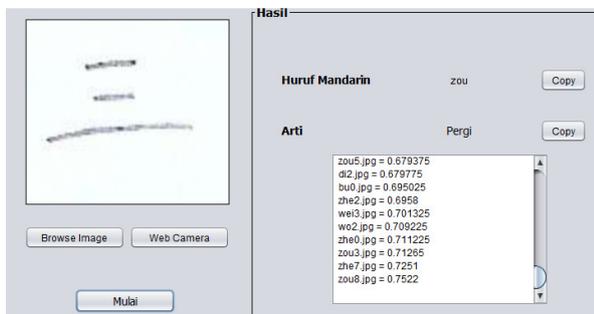
4.3. Analisis

4.3.1. Proses Stroke Extraction

Ekstraksi goresan dilakukan dengan algoritma template matching dimana sistem akan mencoba mendeteksi apakah terdapat goresan tertentu pada citra inputan. Akan tetapi berdasarkan dari hasil pengujian proses ekstraksi dan menghitung goresan ini dapat disimpulkan mengganggu keakurasian dari proses deteksi citra yang diakibatkan oleh kurang optimalnya ekstraksi yang dilakukan. Proses ekstraksi goresan yang tidak optimal ini merupakan penyebab kurva ROC hasil pengujian menjadi abnormal.

Berdasarkan hasil pengujian terdapat beberapa data yang terdeteksi salah akibat dari salahnya jumlah goresan yang terdeteksi. Banyaknya kesalahan pada pendeteksian data training ini menyebabkan nilai *True Positive Rate* dan *False Positive Rate* tidak lagi berubah karena data sudah dianggap salah terdeteksi bahkan sebelum mulai dihitung nilai korelasinya.

Pada gambar 24 pada huruf san menjadi gagal terdeteksi karena jumlah goresan yang terdeteksi adalah 7 goresan, sehingga citra inputan hanya dibandingkan dengan citra template yang memiliki jumlah goresan 7 yang berarti inputan tidak akan terdeteksi sebagai huruf san walau tanpa melalui proses pendeteksian citra.



Gambar 24. Kesalahan akibat salahnya deteksi goresan

Pada proses ekstraksi goresan yang dilakukan aplikasi kesulitan dalam mendeteksi citra yang memiliki 5 goresan hanzi mandarin sehingga menyebabkan hanzi mandarin yang memiliki 5 goresan memiliki tingkat deteksi paling rendah, disisi lain citra dengan hanzi mandarin 7 goresan memiliki akurasi yang paling tinggi apabila dibandingkan dengan keberhasilan citra hanzi 3 goresan dengan citra hanzi 5 goresan, hal ini disebabkan karena ketika ada *noise* pada citra inputan yang gagal dibersihkan aplikasi sering kali akan menghitung titik maupun garis yang bukan bagian dari huruf sebagai goresan sehingga menyebabkan tingginya jumlah goresan yang terdeteksi.

4.3.2. Proses Deteksi Citra

Setelah proses ekstraksi goresan, kemudian jumlah goresan yang terdeteksi akan digunakan untuk memilah data template yang akan digunakan pada proses deteksi citra dengan menggunakan *Template matching* berdasarkan korelasi. Pengujian dilakukan pada 1020 citra testing dengan menggunakan *threshold* optimal yang didapat dari pengujian ROC sebelumnya. Hasil deteksi dapat dilihat pada tabel 4.

Tabel 4. Hasil Pengujian

	Hasil Deteksi		
	Total Population	Predicted Condition positive	Predicted Condition negative
Huruf pada basis data	Condition positive	128	38
	Condition negative	202	652

Pada tabel hasil pengujian di atas 1020 citra yang digunakan dalam pengujian didapatkan kuadran dengan nilai 128 TP dan 652 TN yang hasil deteksi dinyatakan valid, sedangkan 38 FN dan 202 FP menyatakan hasil deteksi tidak valid. Kemudian dengan persamaan(4) dapat dihitung nilai akurasi aplikasi dalam deteksi citra huruf mandarin.

$$Akurasi = \frac{(652 + 128)}{(652 + 128 + 202 + 38)} \times 100\%$$

$$Akurasi = 76,47\%$$

Dari hasil perhitungan tersebut didapatkan akurasi deteksi citra huruf mandarin sebesar 76,47% pada *threshold* optimal yaitu 0,87. Dari nilai akurasi tersebut dapat dicari tingkat kesalahan atau *error rate* dengan persamaan(5).

$$Error Rate = 100\% - 76,47\% = 23,53\%$$

Didapatlah tingkat kesalahan dalam aplikasi dalam mendeteksi huruf mandarin yaitu 23,53%. Penelitian tingkat efisiensi suatu aplikasi [11] menyimpulkan bahwa pada tingkat error 23,53% aplikasi dikatakan cukup efisien untuk digunakan.

5. KESIMPULAN

Kesimpulan yang diperoleh dari penelitian ini adalah *stroke extraction* untuk menghitung jumlah goresan dan kemudian digunakan untuk filter template pada algoritma Template matching, telah berhasil diimplementasikan dalam membangun aplikasi pada penelitian ini. Aplikasi yang dibangun cukup efisien dalam mendeteksi dan menterjemahkan hanzi mandarin.

DAFTAR PUSTAKA

- [1] M. N. Ayyaz, I. Javed, and W. Mahmood, "Handwritten Character Recognition Using Classification with Hybrid Feature Extraction," *Pakistan J. Eng. Appl. Sci.*, vol. 10, pp. 57–67, 2012.
- [2] B. Habibi, "Sistem Penerjemah Mobile Berbasis Citra Digital Dengan Teknologi Web Service," *Fak. Tek. Inform. Progr. Stud. Tek. Inform.*, 2013.
- [3] R. Cao and C. L. Tan, "A model of stroke extraction from Chinese character images," *Proc. 15th Int. Conf. Pattern Recognition. ICPR-2000*, vol. 4, pp. 368–371, 2000.
- [4] R. A. Silaen, "Sistem Pengenalan Karakter Pada Plat Kendaraan Bermotor Menggunakan Profile Projection Dan Algoritma Korelasi," Pekanbaru, 2015.
- [5] D. N. Putri, F. Rafmadhanty, I. P. Megantara, I. N. Jayanti, and K. D. Hapsari, "Klasifikasi Huruf Korea (Hangul) dengan Metode Template Matching Correlation," Malang, 2014.
- [6] S. A. Li, C. N., & Thompson, *Mandarin Chinese: A functional reference grammar*. Univ of California Press, 1989.

- [7] D. Putra, *Pengolahan Citra Digital*. Yogyakarta: Penerbit ANDI, 2010.
- [8] S. D. Cahyo, "Analisis Perbandingan Beberapa Metode Deteksi Tepi Menggunakan Delphi 7," *Skripsi Progr. Stud. Tek. Inform.*, 2011.
- [9] R. Munir, *Pengolahan Citra Digital dengan Pendekatan Algoritmik*. Bandung: Penerbit Informatika Bandung, 2004.
- [10] A. Setia, "Binerisasi Otomatis Pada Citra Bergradasi Dengan Metode Variabel dan Metode Iterasi," 2010, vol. 9, no. 3, pp. 53–65.
- [11] P. D. W. Ayu, "Analisis Pengukuran Tingkat Efektivitas dan Efisiensi Sistem Informasi Manajemen Surat STIKOM Bali," *J. Sist. Dan Inform.*, vol. 11, no. 2, pp. 99–109, 2017.

BIODATA PENULIS



Robert

Lahir di Pekanbaru, 20 Oktober 1996. Saat ini kuliah di Politeknik Caltex Riau pada program studi Teknik Informatika, Jurusan Teknologi Informasi.



Ananda S.Kom, M.T

Lahir di Padang, 9 Februari 1985, Saat ini berkarir sebagai dosen di program studi teknik informatika, jurusan teknologi informasi, politeknik Caltex riau. Lulusan magister dari Teknik Elektronika, konsentrasi Telematika, Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia. Pendidikan sarjana juga di dapatkan dari ITS pada program studi Teknik Informatika. Penelitian berkisar pada bidang pengolahan citra digital, dan image analysis.



Ibnu Surya, S.T., M.T

Lahir di Pekanbaru, 28 Oktober 1983. Saat ini berkarir sebagai dosen muda di kampus Politeknik Caltex Riau pada Program studi Teknik Informatika, jurusan teknologi informasi. Pendidikan tertinggi Strata 2 dengan konsentrasi pada bidang Informatikan – Data dan Rekayasa Perangkat Lunak. Pernah terlibat pada beberapa penelitian dibidang system informasi dan jaringan multimedia.