

PENGARUH PEMAKAIAN *EDGE DETECTION* PADA SISTEM PENGENALAN HURUF KAPITAL TULISAN TANGAN

(The Influence of Edge Detection Use on Capital Letter Hand Writing Recognition System)

Kevin Ananta¹, Iwan Setyawan², Ivanna K. Timotius³

Fakultas Teknik Elektronika dan Komputer Program Studi Teknik Elektro
Universitas Kristen Satya Wacana – Salatiga

¹kanantaka@gmail.com, ²iwan.setyawan@ieee.org, ³ivanna.timotius@staff.uksw.edu

Abstrak

Dalam tulisan ini diteliti pengaruh dari *edge detection* sebagai metode pemrosesan awal citra pada sistem pengenalan huruf kapital tulisan tangan dan pengaruhnya terhadap keakuratan sistem pengenalan dengan metode klasifikasi *nearest neighbor*. Hasil percobaan menunjukkan bahwa penggunaan *edge detection* tidak memberikan kontribusi yang lebih tinggi dalam meningkatkan akurasi pengenalan dibandingkan dengan pemilihan ukuran *structuring element* yang besar pada operasi dilasi.

Kata Kunci: *Sobel edge detection, nearest neighbor, proses morfologis dilasi, citra biner, pengenalan tulisan tangan*

Abstract

This paper investigates the influence of edge detection as a pre processing method for a capital letter handwriting recognition system on the accuracy of nearest neighbor classification system. The experimental results show that the use of edge detection did not significantly enhance the recognition accuracy in comparison to the selection of large structuring element size in the dilation operation.

Keywords: *Sobel edge detection, nearest neighbor, dilation morphology, binary image, handwriting recognition*

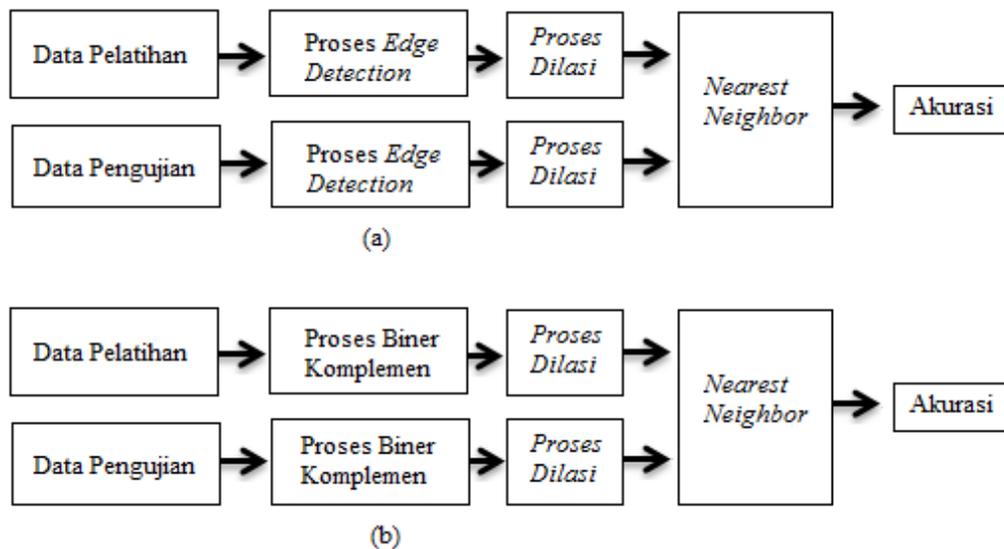
1. PENDAHULUAN

Sistem untuk pengenalan tulisan tangan banyak digunakan dalam aplikasi pengenalan karakter optik. Dengan sistem tersebut tulisan tangan pada sebuah kertas dapat dengan mudah dan cepat dikonversikan menjadi sebuah *text file* tanpa harus mengetik ulang dokumen tersebut. Selain itu pengenalan tulisan tangan banyak digunakan untuk devais layar sentuh. Beberapa penelitian sebelumnya telah membahas tentang penggunaan pengenalan tulisan tangan sebagai *input* dari *tablet PC* [1], penggunaan *Markov Model* dalam pengenalan tulisan tangan [2], dan penggunaan LS – SVM dalam pengenalan tulisan tangan [3].

Dalam pengenalan tulisan tangan, huruf kapital umumnya lebih mudah dikenali dibandingkan dengan huruf kecil. Hal ini dikarenakan bentuknya yang lebih standar dan tidak memiliki terlalu banyak perbedaan antar penulis. Oleh karena itu, dalam tulisan ini hanya akan membahas pengenalan huruf kapital tulisan tangan saja.

Pada tulisan ini dibahas segmentasi citra dengan metode *edge detection* dan pengaruhnya terhadap sistem pengenalan huruf kapital tulisan tangan. *Edge detection* merupakan cara yang sering digunakan untuk melakukan segmentasi citra berdasarkan keberadaan perubahan tingkat intensitas yang mendadak (*abrupt*) [4]. Pra proses *edge detection* dipilih karena dapat menghasilkan garis pemisah antara objek dengan *background* dan menyatakan batasan antara objek. Jika objek sudah berhasil dipisahkan dengan *background* maka akan meningkatkan akurasi dari sebuah mesin klasifikator.

Proses pengenalan huruf besar yang dibahas pada tulisan ini adalah sebagai berikut. Pertama-tama, tulisan tangan di atas kertas diubah menjadi citra digital biner. Untuk menguji pengaruh proses segmentasi menggunakan *edge detection* terhadap tingkat akurasi pengenalan huruf, dilakukan dua pengujian. Pada pengujian pertama, citra masukan disegmentasi menggunakan *Sobel edge detector*, kemudian diproses secara morfologis menggunakan operasi dilasi. Pada pengujian kedua, citra masukan mengalami proses biner komplement kemudian diproses menggunakan operasi dilasi. Setelah citra masukan mengalami proses seperti yang telah disebutkan di atas, pengenalan huruf dilakukan dengan pengklasifikasi yang menggunakan metode *nearest neighbor*. Blok diagram sistem dalam penelitian ini dapat dilihat pada Gambar 1.



Gambar 1. Blok diagram sistem: (a) blok diagram untuk pra proses *edge detection* dan dilasi, (b) blok diagram sistem tanpa pra proses *edge detection*.

Sistematika pembahasan dalam tulisan ini adalah sebagai berikut. Bagian 2 tulisan ini membahas mengenai metode *Edge Detection*, bagian 3 membahas mengenai operasi dilasi, bagian 4 membahas klasifikasi dengan *nearest neighbor*, bagian 5 berisi eksperimen dan hasil, dan bagian 6 berisi kesimpulan.

2. EDGE DETECTION

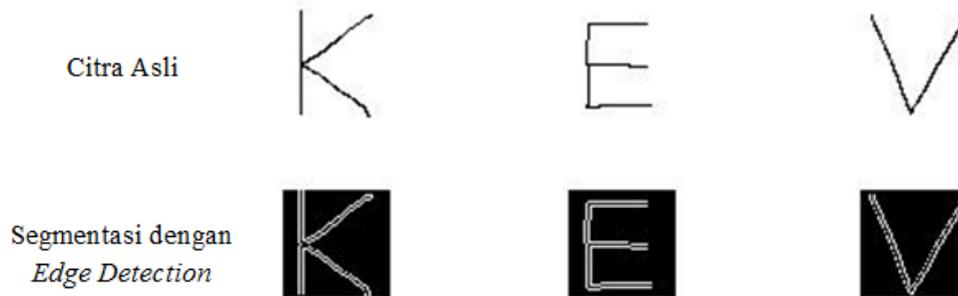
Dalam suatu citra digital, peralihan antara objek (latar depan) ke *background* (latar belakang) pada umumnya ditandai oleh adanya perubahan intensitas yang mendadak (*abrupt*). Peralihan seperti ini akan menghasilkan adanya tepian (*edge*) yang menandakan batas antara objek dan latar belakang. *Edge detection* merupakan teknik yang digunakan untuk mendeteksi tepian pada citra.

Metode *edge detection* yang digunakan dalam penelitian ini adalah *Sobel edge detection*. Metode ini digunakan karena mampu menghasilkan hasil ekstraksi paling optimal dan kinerja yang lebih baik untuk citra *input* berupa tulisan tangan dibandingkan dengan metode Prewitt maupun Roberts [5], [6]. Metode ini dilakukan dengan cara mengkonvolusikan citra *input* dengan operator-operator Sobel, yang ditunjukkan pada Gambar 2.

-1	-2	-1	-1	0	1	0	1	2	-2	-1	0
0	0	0	-2	0	2	-1	0	1	-1	0	1
1	2	1	-1	0	1	-2	-1	0	0	1	2

Gambar 2. Operator-operator Sobel

Contoh hasil pengolahan citra masukan menggunakan operator Sobel ditunjukkan pada Gambar 3. Pada gambar ini, baris pertama menunjukkan citra asli dan baris kedua menunjukkan hasil pengolahan citra tersebut menggunakan operator Sobel.



Gambar 3. Citra asli dan hasil segmentasi dengan *edge detection*

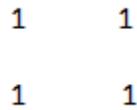
3. OPERASI DILASI

Operasi dilasi adalah suatu operasi citra yang termasuk dalam operasi morfologis. Operasi dilasi dilakukan untuk menebalkan garis-garis yang ada pada suatu citra. Operasi ini didefinisikan sebagai berikut. Misalkan terdapat dua buah himpunan, A dan B , yang merupakan himpunan dalam ruang dua dimensi. Operasi dilasi A oleh B , dilambangkan dengan $A \oplus B$, dan didefinisikan sebagai berikut:

$$A \oplus B = \left\{ z \mid (\hat{B})_z \cap A \neq \emptyset \right\} \dots\dots\dots (1)$$

Persamaan di atas menunjukkan bahwa B dicerminkan terhadap titik asalnya kemudian ditranslasikan (digeser) dengan nilai pergeseran z . Jadi dilasi A oleh B adalah sebuah himpunan yang berisi semua nilai pergeseran z , sedemikian sehingga setidaknya satu elemen \hat{B} dan A saling *overlap*, dengan \hat{B} adalah himpunan B yang telah dicerminkan terhadap titik asal [5], [6]. Operasi dilasi akan menghasilkan objek yang mirip dengan objek aslinya tetapi lebih tebal, karena garis-garis yang ada pada citra tersebut telah mengalami penebalan. Pada operasi dilasi seperti pada contoh di atas, himpunan A adalah citra yang akan diolah sedangkan himpunan B disebut sebuah *structuring element*.

Pada tulisan ini digunakan *structuring element* berbentuk bujursangkar berukuran 2×2 seperti ditunjukkan pada Gambar 4.



Gambar 4. *Structuring elements* 2×2 yang digunakan untuk segmentasi

Contoh hasil operasi dilasi yang diterapkan pada citra yang telah diproses oleh operator Sobel ditunjukkan pada Gambar 5. Dapat dilihat bahwa operasi dilasi menghasilkan sebuah objek yang memiliki bentuk yang sama tetapi terlihat lebih tebal dari objek pada citra aslinya.



Gambar 5. Hasil *edge detection* yang dilasi dengan SE 2×2

Citra masukan yang tidak mengalami proses *edge detection* juga diolah menggunakan operasi dilasi dengan *structuring element* yang sama. Sebelum dilakukan operasi dilasi, citra masukan pertama-tama dikomplemen terlebih dahulu untuk menghasilkan objek berwarna putih dan latar belakang berwarna hitam. Hasil operasi komplemen dan dilasi ditunjukkan pada Gambar 6.

Citra Asli yang di Komplemen



setelah di dilasi dengan SE 2×2



Gambar 6. Citra asli yang di komplemen (atas), citra asli yang di komplemen dan kemudian dilasi dengan SE 2×2 (bawah)

4. KLASIFIKASI DENGAN *NEAREST NEIGHBOR*

Algoritma *nearest neighbor* merupakan metode pengklasifikasian objek berdasarkan *training samples* yang terdekat dalam ruang fitur [5], [7]. Pada tulisan ini, jarak antar-*sample* dalam algoritma *nearest neighbor* ditentukan berdasarkan kriteria *Euclidean distance*.

Euclidean distance antara dua titik \mathbf{p} dan \mathbf{q} pada sebuah ruang fitur adalah panjang dari segmen garis yang menghubungkan antara \mathbf{p} dan \mathbf{q} . Dalam sebuah koordinat kartesian, jika $\mathbf{p} = (p_1, p_2, \dots, p_n)^T$ dan $\mathbf{q} = (q_1, q_2, \dots, q_n)^T$ adalah dua buah titik dalam ruang *Euclidean* dimensi n , maka jarak dari \mathbf{p} ke \mathbf{q} atau dari \mathbf{q} ke \mathbf{p} diberikan oleh:

$$D(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^n |p_i - q_i|^2} \dots\dots\dots (2)$$

5. EKSPERIMEN DAN HASIL

Langkah-langkah yang dilakukan untuk mengimplementasikan sistem pengenalan huruf adalah sebagai berikut. Pertama-tama, *training samples* (berukuran $m \times n$) diproses menggunakan tahap *preprocessing* seperti yang telah dijelaskan pada bagian 2. Kemudian, masing-masing sampel diubah menjadi sebuah matriks kolom dengan panjang mn . Matriks-matriks kolom ini disusun menjadi sebuah matriks pelatihan (*training matrix*).

Setelah sistem menyelesaikan tahap pelatihan, kita dapat masuk ke dalam tahap pengujian. Pengujian dilakukan menggunakan data uji yang juga telah menjalani tahap *preprocessing* yang sama dengan citra pada dataset pelatihan. Pada percobaan yang dilakukan dalam tulisan ini, digunakan *database* yang berisi 2.600 citra biner. Masing-masing citra pada *database* berisi sebuah huruf besar yang ditulis dengan tangan. Jadi *database* ini terdiri atas 26 huruf (A – Z) yang masing-masing memiliki 100 citra. Citra pada *database* ini diperoleh dari tulisan tangan 8 orang responden. Masing-masing responden diminta untuk menulis huruf besar A – Z di atas kertas yang kemudian di-*scan* menjadi citra digital. Seorang responden diminta menuliskan minimal 10 set huruf dan beberapa responden memberikan 15 set huruf. Citra-citra pada *database* ini disimpan dalam format *bitmap* (.bmp) dengan ukuran 50×50 *pixel*. Citra-citra pada *database* ini telah diatur sedemikian rupa sehingga huruf pada citra tersebut berada di tengah-tengah citra dan tidak mengalami rotasi.



Gambar 7. Contoh data dalam *database* citra yang digunakan

Kinerja metode klasifikasi *nearest neighbor* diuji dengan menggunakan metode validasi silang (*cross validation*). Pada percobaan ini, digunakan *5-fold cross validation* dengan pengujian dilakukan sebanyak 10 kali (10 *run*). Tabel 1 menyajikan hasil pengujian sistem dimana citra masukan mengalami proses *edge detection* dan dilasi sedangkan Tabel 2 menyajikan hasil pengujian sistem dimana citra masukan hanya mengalami proses dilasi.

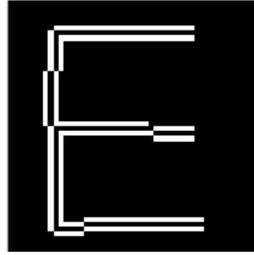
Tabel 1. Hasil eksperimen dengan pemrosesan awal *Sobel edge detection* dilanjutkan dilasi menggunakan *structuring elemen 2x2*

RUN	FOLD NO					RATA-RATA
	1	2	3	4	5	
1	97.30	99.42	97.88	98.65	98.65	98.38
2	97.88	98.65	97.50	98.65	98.84	98.30
3	99.03	98.84	99.42	98.46	97.88	98.73
4	98.46	97.88	99.23	99.42	98.46	98.69
5	99.23	99.23	97.69	98.84	98.84	98.76
6	99.03	97.88	98.46	98.26	98.84	98.50
7	98.26	99.03	98.65	98.65	98.26	98.57
8	98.84	99.23	98.46	98.84	98.26	98.73
9	98.46	99.03	98.65	99.61	97.88	98.73
10	98.26	98.65	98.65	97.69	99.61	98.57
Rata-Rata 10 Run						98.60

Tabel 2. Hasil eksperimen dengan pemrosesan dilasi menggunakan *structuring element 2x2* tanpa *edge detection*

RUN	FOLD NO					RATA-RATA
	1	2	3	4	5	
1	94.61	95.96	95.76	95.38	95.38	95.42
2	95.57	95.19	95.19	95.00	95.38	95.26
3	94.03	95.00	95.76	95.57	96.34	95.34
4	93.84	95.38	96.92	95.38	93.84	95.07
5	96.15	95.96	97.30	94.80	94.23	95.69
6	94.23	96.15	95.57	96.53	96.73	95.84
7	95.19	95.38	96.92	95.57	94.42	95.50
8	96.73	94.80	95.57	94.80	95.96	95.57
9	97.30	94.80	96.53	94.03	95.57	95.65
10	95.57	95.00	96.73	95.38	94.03	95.34
Rata-Rata 10 Run						95.47

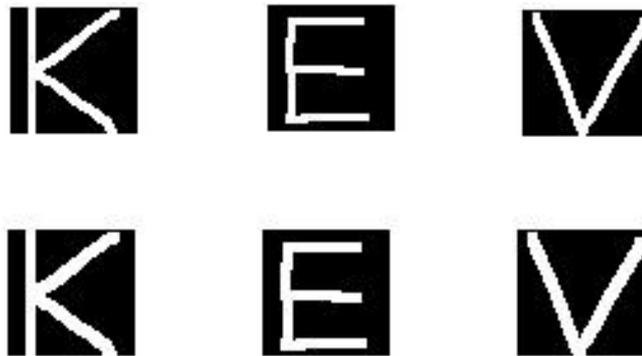
Dari hasil percobaan yang disajikan pada Tabel 1 dan 2, terlihat bahwa akurasi pengenalan huruf jika digunakan tahap *preprocessing* berupa *edge detection* lebih tinggi dibandingkan dengan jika tahap ini tidak dilakukan. Hal ini dapat dijelaskan sebagai berikut. Gambar 8 menunjukkan citra hasil *edge detection* yang sudah diperbesar. Dari Gambar 8, terlihat bahwa citra hasil *edge detection* memiliki garis ganda yang berjarak kurang dari 2 *pixel* (sebaliknya, citra yang tidak mengalami tahap *edge detection* hanya memiliki garis tunggal). Pada saat citra ini mengalami proses dilasi, maka kedua garis tersebut akan menebal. Sebagai akibatnya, hasil proses dilasi 2×2 pada citra hasil *edge detection* akan lebih tebal dibandingkan hasil dilasi citra tanpa *edge detection*. Dengan kata lain, hasil proses dilasi citra dengan *edge detection* seolah-olah dihasilkan menggunakan *structuring element* yang lebih besar dari 2×2 . Garis yang lebih tebal inilah yang menghasilkan tingkat akurasi pengenalan yang lebih tinggi.



Gambar 8. Citra huruf setelah pra proses *edge detection*

Untuk menguji hipotesis ini, dilakukan percobaan sebagai berikut. Citra masukan, tanpa *edge detection*, dikomplemen dan didilasi menggunakan *structuring element* berukuran 3×3 dan 4×4 . Hasil dilasi dengan *structuring element* tersebut dapat dilihat pada Gambar 9. Terlihat bahwa citra yang dihasilkan lebih tebal daripada hasil dilasi menggunakan *structuring element* 2×2 . Tingkat akurasi pengenalan sistem diuji menggunakan *10 run 5-fold cross validation*. Hasil percobaan ini disajikan pada Tabel 3 dan Tabel 4. Dari data-data yang diperoleh, terlihat bahwa rata-rata akurasi meningkat berturut-turut menjadi 98,46 % dan 99,20 %.

Hasil ini menunjukkan bahwa proses *edge detection* sebenarnya tidak diperlukan untuk meningkatkan kinerja sistem pengenalan huruf ini, asalkan proses dilasi dilakukan dengan *structuring element* dengan ukuran yang cukup besar.



Gambar 9. Dilasi menggunakan *structuring element* 3×3 (atas) dan 4×4 (bawah)

Tabel 3. Hasil eksperimen dengan pemrosesan dilasi menggunakan *structuring element* 3×3 tanpa *edge detection*

RUN	FOLD NO					RATA-RATA
	1	2	3	4	5	
1	98.46	98.65	98.46	98.85	98.85	98.65
2	98.65	99.04	98.27	98.08	98.65	98.54
3	98.08	98.46	98.08	98.65	98.27	98.31
4	97.88	99.42	98.85	98.46	98.27	98.58
5	98.85	99.23	98.65	98.27	97.12	98.42
6	98.46	98.08	98.46	98.27	99.23	98.50
7	97.88	98.08	98.46	98.85	98.27	98.31
8	97.69	97.69	99.23	97.88	99.23	98.35
9	98.27	98.65	98.08	99.23	98.27	98.50
10	97.69	99.42	99.04	97.88	98.08	98.42
Rata-Rata 10 Run						98.46

Tabel 4. Hasil eksperimen dengan pemrosesan dilasi menggunakan *structuring element* 4×4 tanpa *edge detection*

RUN	FOLD NO					RATA-RATA
	1	2	3	4	5	
1	99.23	99.42	99.42	99.42	99.04	99.31
2	99.81	99.04	99.62	98.46	99.62	99.31
3	99.23	98.85	98.65	99.42	99.23	99.08
4	98.65	99.62	99.23	99.23	99.23	99.19
5	99.23	100.00	99.62	98.65	98.27	99.15
6	99.42	99.23	99.23	99.04	99.42	99.15
7	99.23	99.04	98.85	99.62	99.04	99.15
8	98.85	98.65	99.62	99.42	99.81	99.27
9	99.42	99.42	98.85	99.62	98.65	99.19
10	98.65	99.62	99.62	99.42	98.65	99.19
Rata-Rata 10 Run						99,20

6. KESIMPULAN

Pada tulisan ini telah diteliti pengaruh penggunaan *edge detection* pada sebuah sistem pengenalan huruf kapital tulisan tangan. Untuk *structuring element* berukuran kecil, penggunaan *edge detection* menyebabkan peningkatan akurasi dari klasifikator. Hal ini dikarenakan hasil pra proses *edge detection* pada citra tulisan tangan akan menimbulkan garis ganda yang kemudian akan menghasilkan garis yang lebih tebal setelah dilasi. Penggunaan proses dilasi dengan menggunakan *structuring element* yang lebih besar akan menghasilkan hasil yang tidak jauh berbeda. Hal ini tampak dari hasil eksperimen dimana persentasi keberhasilan klasifikator untuk penggunaan pra proses *edge detection* adalah sebesar 98,60 % sedangkan untuk pra proses dilasi dengan *structuring elements* 3×3 dan 4×4 adalah berturut-turut 98,46 % dan 99,20 %. Dari fakta tersebut, dapat disimpulkan bahwa penggunaan tahap pra proses *edge detection* untuk sistem klasifikasi huruf kapital tulisan tangan tidak memberikan kontribusi yang signifikan dibandingkan dengan penggunaan proses dilasi menggunakan *structuring element* yang cukup besar.

REFERENSI

- [1]. J. A. Pittman, “*Handwriting Recognition: Tablet PC input*”, *IEEE Journal and Magazines*, Vol. 40, No. 9, September 2007.
- [2]. P. Thomas, G. A. Fink, “*Markov models for offline handwriting recognition: a survey*”, Springerlink, Oktober 2009.
- [3]. M. M. Adankon, M. Cheriet, *Model selection for the LS-SVM, “Application to handwriting recognition, Synchronmedia Laboratory for Multimedia Communication in Telepresence”*, ÉTS, 1100 Notre Dame-Ouest, Montréal, Canada, November 2008.
- [4]. R. O Duda, P. E. Hart, dan D. G. Stork, “*Pattern Classification*”, Wiley, New York, 2001.
- [5]. R.C. Gonzales, R. E Woods, “*Digital Image Processing*”, Edisi Ketiga, Pearson Education International, New Jersey, 2010.
- [6]. I. Setyawan, “*Dasar-Dasar Machine Vision*”, Salatiga, Widya Sari Press, 2012.
- [7]. I. K. Timotius, “*Dasar-Dasar Pengenalan Pola*”, Salatiga, Widya Sari Press, 2012.