APLIKASI PENGENALAN KARAKTER PADA PLAT NOMOR KENDARAAN BERMOTOR DENGAN LEARNING VECTOR QUANTIZATION

Ng Poi Wong[1)], Hardy[2)], Ade Maulana[3)]
Program Studi S-1 Teknik Informatika, STMIK Mikroskil
Jln. Thamrin no. 140, Medan, 20212
HP : +62 81361718607
E-mail : poiwong@mikroskil.ac.id[1)], hardy@mikroskil.ac.id[2)], 091112719@students.mikroskil.ac.id[3)]

Abstrak

Sistem perparkiran saat ini, operator cenderung salah dalam menginput nomor kendaraan. Hal ini dapat mengakibatkan kendaraan bisa bebas keluar masuk meskipun nomor kendaraannya tidak sesuai dengan nomor kendaraan saat masuk serta dapat mengakibatkan waktu pelayanan yang agak lama. Aplikasi pengenalan karakter pada nomor kendaraan merupakan salah satu solusi untuk mengatasi masalah ini. Aplikasi ini akan mengenali karakter dengan membaca setiap karakter pada nomor kendaraan. Dalam mengenali karakter, dimanfaatkan deteksi tepi sobel untuk menemukan lokasi plat kendaraan. Nomor kendaraan yang telah terdeteksi akan dilanjutkan tahap segmentasi tiap karakternya, tiap karakter ini akan melalui tahap ekstraksi ciri menggunakan Directional Feature Extraction dan akan dikenali menggunakan metode Learning Vector Quantization. Untuk melakukan pengujian terhadap aplikasi ini dilakukan dengan pengujian K-Fold Cross Validation. Berdasarkan hasil pengujian, aplikasi mampu mengenali karakter dengan tingkat keberhasilan pengenalan rata-rata 87,093%, rata-rata waktu eksekusi 4,583 detik, dan rata-rata waktu pengenalan karakter 0,28 detik.

In parking system nowadays, operator tends to do mistakes in entering car's license plate number. Due to that, cars can freely enter and exit the parking lot even though the plate number do not match the parking ticket number. It also prolong service time. One solution to that is using character recognition application on license plate. It will recognize character by reading every character on plate number. In recognizing the character, Sobel edge detection is used to locate the plate number. A located plate number will go into segmentation phase for each character, each character will then go through a feature extraction phase using Directional Feature Extraction and wll be recognized using Learning Vector Quantization method. For application testing, K-Fold Cross Validation is used. Based on testing result, this application can recognize character with 87.093% average success rate, 4.583 second average execution time, and 0.28 second average character's recognition time.

Kata kunci: *nomor kendaraan, Learning Vector Quantization, deteksi tepi Sobel, Directional feature Extraction*

1. PENDAHULUAN

Sistem perparkiran saat ini khususnya di beberapa mall di Sumatera Utara telah menerapkan sistem parkir tanpa operator pada pintu masuk dimana dalam prosesnya kendaraan masuk difoto dan karcis parkir dicetak manual dengan disertai *barcode* sebagai nomor pengenal karcis. Pada pintu keluar dilakukan pemindaian karcis, pencatatan nomor kendaraan, dan proses transaksi pembayaran parkir oleh operator secara manual. Namun pada penerapannya, operator cenderung salah dalam melakukan pencatatan nomor kendaraan. Kesalahan ini tidak berdampak pada sistem yang sedang berlangsung karena nomor kendaraan tidak menjadi catatan dalam karcis parkir, akibatnya kendaraan bisa bebas keluar dari area parkir walaupun nomor kendaraan yang dicatat berbeda dari nomor kendaraan yang sebenarnya.
Dalam penelitian-penelitian sebelumnya, beberapa pendekatan telah digunakan untuk mendeteksi plat nomor kenderaan. Dalam penelitian Hou, et al. [1], pendeteksian plat menggunakan *Tophat-bothat changing* dan *Line scanning*, serta morfologi matematika. Sedangkan Nagare [2] menggunakan Back Propagation Neural Network dan Learning Vector Quantization Neural Network untuk mendeteksi plat nomor.

2. KAJIAN PUSTAKA

Copyright © 2013 SESINDO

Kajian pustaka untuk aplikasi pengenalan karakter pada plat nomor kendaraan meliputi penskalaan, deteksi tepi *Sobel*, pengambangan, metode *Otsu*, algoritma *Zhang-Shuen*, *Directional Feature Extraction*, dan *Learning Vector Quantization*.

2.1 Penskalaan (*Scaling*)

Penskalaan (*Scaling*) adalah operasi geometri yang memberi efek memperbesar atau memperkecil ukuran citra masukan sesuai dengan variable penskalaan citranya. Operasi geometri ini merupakan proses perubahan hubungan spasial antara setiap pixel. Di dalam penskalaan untuk memperkecil ukuran citra dikenal metode *subsampling*. Metode *subsampling* menggunakan satu nilai acak dari sebuah region untuk mewakili region tersebut [3].

2.2 Deteksi Tepi *Sobel*

Tepian citra dapat merepresentasikan objek-objek yang terkandung dalam citra, bentuk, dan ukuran serta terkadang juga informasi tentang teksturnya. Tepian citra adalah posisi dimana intensitas pixel dari citra berubah dari nilai rendah ke nilai tinggi atau sebaliknya. Deteksi tepi adalah langkah untuk menemukan tepian citra tersebut [3].

Deteksi tepi *Sobel* merupakan teknik deteksi tepi yang menghasilkan efek *relief* dari sebuah citra. Metode *Sobel* unggul dalam mengurangi *noise* sehingga mampu menghasilkan tepi yang banyak dan halus [5]. Di dalam deteksi tepi *Sobel* terdapat dua matriks yang digunakan sebagai operator *Sobel* yang nilainya dapat dilihat pada Gambar 1.

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad M_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Gambar 1. Operator Sobel

Masing-masing matriks tersebut dikalikan ke nilai masing-masing piksel citra, lalu kemudian dijumlahkan hasilnya. Persamaan matematikanya dapat ditulis sebagai berikut:

a. f(x) = f (x, y) * Mx (x, y)
b. f(y) = f (x, y) * My (x, y)
c. magn (x, y) = | f(x) | | f(y) |

dimana f(x) adalah fungsi arah sumbu x (horizontal), f(y) adalah fungsi arah sumbu y (vertikal), dan magn (x, y) adalah fungsi ketebalan gradien.

2.3 Pengambangan (*Thresholding*)

Proses pengambangan (*Threholding*) akan menghasilkan citra biner, yaitu citra yang memiliki dua nilai tingkat keabuan yaitu hitam dan putih [3]. Secara umum proses pengambangan citra grayscale untuk menghasilkan citra biner adalah sebagai berikut.

$$g(x,y) = \begin{Bmatrix} 1 \; if \; f(x,y) \geq T \\ 0 \; if \; f(x,y) < T \end{Bmatrix} \tag{1}$$

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

2.4 Metode *Otsu*

Metode *Otsu* adalah salah satu metode yang digunakan untuk mencari nilai ambang (T) untuk proses pengambangan. Pendekatan yang digunakan metode *Otsu* adalah dengan melakukan analisis diskriminan yaitu menentukan suatu variable yang dapat membedakan antara dua atau lebih kelompok yang muncul secara alami [3]. Nilai ambang k dengan metode *Otsu* dapat ditentukan dengan memaksimumkan persamaan σ :

$$\sigma_B^2(k^*) = \max_{1 \leq k < L} \sigma_B^2(k) \tag{2}$$

Dimana:

$$\sigma_B^2(k) = \frac{[\mu_T \omega(k) - \mu(k)]^2}{\omega(k)[1 - \omega(k)]}$$

Copyright © 2013 SESINDO

$$\omega(k) = \sum_{i=1}^{k} p_i$$

$$\mu(k) = \sum_{i=1}^{k} i \cdot p_i \qquad p_i = \frac{n_i}{N}$$

$$\mu_T = \sum_{i=1}^{L} i \cdot p_i$$

Dengan ω(k) adalah nilai *zeroth cumulative moment*, μ(k) adalah nilai *first cumulative moment*, μ_T adalah total nilai mean, p_i menyatakan nilai ambang yang akan dicari dan n_i menyatakan jumlah pixel dengan tingkat keabuan *i* dan *N* menyatakan banyaknya piksel pada citra.

2.5 Algoritma *Zhang-Shuen*

Algoritma *Zhang-Shuen* adalah algoritma untuk proses *thinning*, dimana proses *thinning* suatu proses morphologi yang mengubah bentuk asli citra biner menjadi citra yang menampilkan batas-batas objek/*foreground* hanya setebal satu pixel [3]. Algoritma *Zhang-Shuen* menghapus semua titik *countour* pada citra kecuali titik yang merupakan *skeleton* [4]. Pada algoritma ini setiap iterasi dibagi menjadi 2 subiterasi. Pada subiterasi ke-1, titik P1 dihapus dari pola digital jika memenuhi kondisi berikut:

a. $2 < B(P_1) < 6$
b. $A(P_1) = 1$
c. $P_2 * P_4 * P_6 = 0$
d. $P_4 * P_6 * P_8 = 0$

dimana $A(P_1)$ adalah jumlah dari pola 1 dalam memerintahkan mengatur P_2, P_3, P_4, • • •, P_8, P_9 yang merupakan delapan tetangga dari P_1 (Gambar 2), dan $B(P_1)$ adalah jumlah tetangga bukan nol dari P_1, yaitu,
$B(P_1) = P_2 + P_3 + P_4 + \bullet\bullet\bullet + P_8 + P_9$.
Pada subiteration ke-2, hanya kondisi (c) dan (d) berubah sebagai berikut:

c. $P_2 * P_4 * P_8 = 0$
d. $P_2 * P_6 * P_8 = 0$

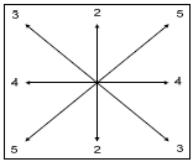
dan sisanya tetap sama.

P_9 $(i-1, j-1)$	P_2 $(i-1, j)$	P_3 $(i-1, j+1)$
P_8 $(i, j-1)$	P_1 (i, j)	P_4 $(i, j+1)$
P_7 $(i+1, j-1)$	P_6 $(i+1, j)$	P_5 $(i+1, j+1)$

Gambar 2. Penempatan posisi piksel tetangga

2.6 Directional Feature Extraction

Feature Extraction adalah cara pengenalan objek dengan melihat ciri khusus yang dimiliki oleh objek dengan tujuan melakukan perhitungan dan perbandingan untuk mengklasifikasikan ciri yang dimiliki suatu citra. Ada beberapa metode dari *Feature Extraction* antara lain *Amplitudo*, *Histogram*, *Gradient*, *Wavelet*, Arah dan lain sebagainya [3]. *Directional Feature Extraction* merupakan metode *Feature Extraction* dengan ciri arah. Pada metode ini, arah dibagi menjadi 4 arah dengan penomoran 2 untuk arah vertikal, 3 untuk diagonal kiri atas ke kanan bawah, 4 untuk arah horizontal, 5 untuk diagonal kanan atas ke kiri bawah (Gambar 3). Untuk titik awal pengecekan diberi angka 9 [6].



Gambar 3. Arah pada Directional Feature Extraction

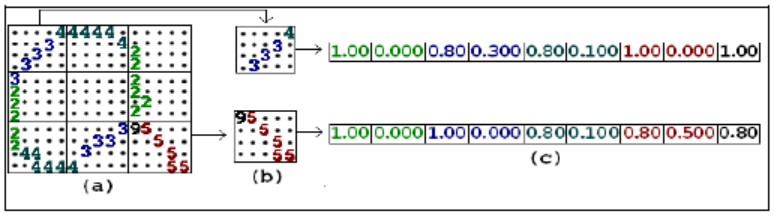
Untuk melakukan ekstraksi fitur berdasarkan arah dimulai dari piksel pertama yaitu piksel kiri paling atas hingga piksel paling kanan dan kemudian dilanjutkan lagi untuk nilai piksel di bawahnya. Setiap piksel dicek nilai tetangganya dari mulai piksel yang berada di atas piksel tersebut, pengecekan dilakukan dengan searah jarum jam. Bila piksel tetangga bernilai 0 atau berwarna hitam, set nilai sesuai ketentuan penomoran arah. Hasil proses di atas akan dikelompokkan (*zonning*) menjadi 9 wilayah (Gambar 4.a) dan masing-masing wilayah (Gambar 4.b) akan dikelompokkan menjadi 9 bagian (Gambar 4.c), masing-masing bagian akan diubah menjadi nilai vektor dengan 9 kolom dengan ketentuan masing-masing kolom sebagai berikut.

Copyright © 2013 SESINDO

1. *value* dari garis horizontal,
2. *length* dari garis horizontal,
3. *value* dari garis diagonal kanan,
4. *length* dari garis diagonal kanan,
5. *value* dari garis vertikal,
6. *length* dari garis vertikal,
7. *value* dari garis diagonal kiri,
8. *length* dari garis diagonal kiri
9. *value* dari intersection points

Berikut adalah rumus perhitungan *value* dan *length* :

a. *value* = 1 - ((jumlah garis / 10) * 2)
b. *length* = (jumlah piksel) / (lebar atau tinggi window * 2)



Gambar 4. (a) Gambar yang telah di proses (b) Zoned windows (c) Nilai vector

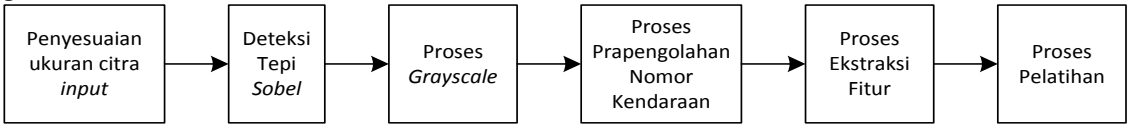
2.7 Learning Vector Quantization

Learning Vector Quantization (LVQ) adalah salah satu jenis jaringan saraf tiruan untuk melakukan pembelajaran pada lapisan kompetitif yang terawasi. Suatu lapisan kompetitif akan secara otomatis belajar untuk mengklasifikasikan vektor input. Kelas yang didapat sebagai hasil dari lapisan kompetitif ini hanya tergantung pada jarak antara vektor input. Jika vektor input mendekati sama maka lapisan kompetitif akan meletakkan kedua vektor input tersebut ke dalam kelas yang sama [3]. Algoritma LVQ dapat ditulis sebagai berikut.

1. Tentukan masing-masing kelas *output*, menetukan bobot, dan menetapkan *learning rate* α.
2. Bandingkan masing-masing *input* dengan masing-masing bobot yang telah ditetapkan dengan melakukan pengukuran jarak antara masing-masing bobot W_0 dan input X_p. persamaannya adalah sebagai berikut :
   $\|X_p - W_0\|$ (3)
3. Nilai minimum dari perbandingan itu akan menentukan kelas dari vektor *input* dan perubahan bobot dari kelas tersebut. Perubahan untuk bobot baru (W_0') dapat dihitung dengan persamaan berikut.
   a. Untuk input dan bobot yang memiliki kelas yang sama:
      $W_0' = W_0 + \alpha (X - W_0)$ (4)
   b. Untuk input dan bobot yang memiliki kelas yang berbeda:
      $W_0' = W_0 - \alpha (X - W_0)$ (5)

3. PENGENALAN KARAKTER PADA PLAT NOMOR KENDARAAN BERMOTOR

Pada aplikasi pengenalan karakter pada nomor kendaraan bermotor dilakukan dengan beberapa tahapan proses, yang bisa dilihat di Gambar 5.



Gambar 5. Tahapan pengenalan karakter pada nomor kendaraan bermotor

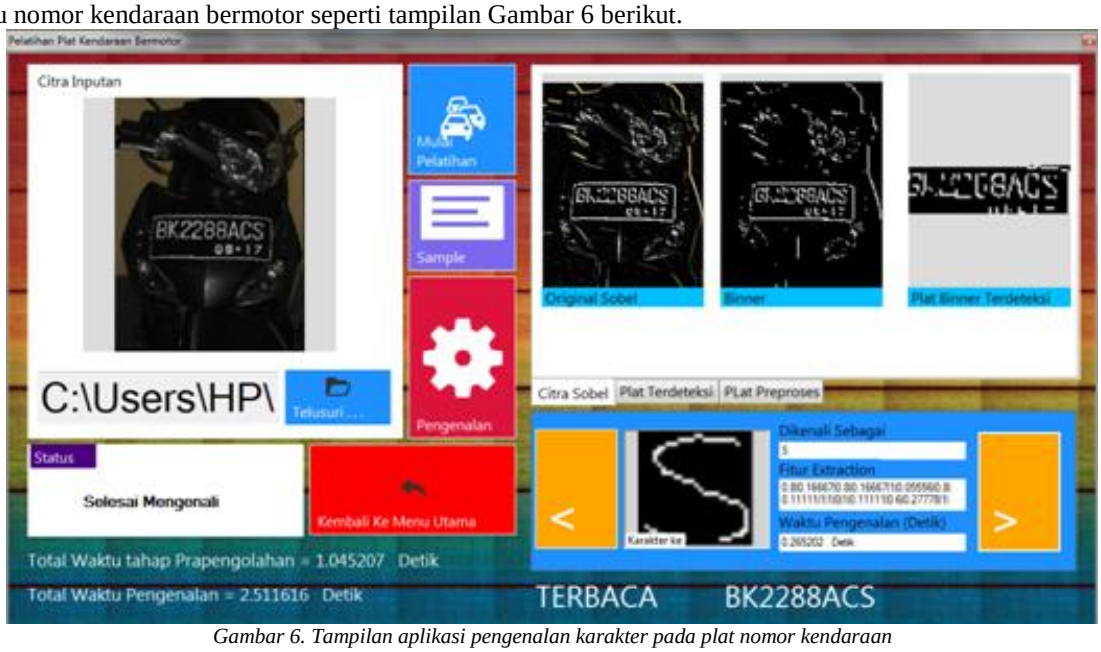
Pada penyesuaian ukuran citra masukan, dilakukan penyesuaian ukuran plat kendaraan inputan dengan ukuran plat yang akan diproses sistem dan mempercepat proses pengolahan citra dengan proses pengecilan resolusi dengan skala pengecilan = tinggi citra asli / 320. Pada deteksi tepi, digunakan metode *Sobel* untuk menemukan pinggiran dari sebuah citra, sehingga memudahkan sistem dalam melacak kandidat pixel yang merupakan sebuah plat kendaraan bermotor. Pada proses *Grayscale*, hasil dari deteksi tepi *Sobel* sebelumnya akan dicari derajat keabuannya masing-masing pikselnya yang nantinya akan diubah menjadi citra biner. Pada proses pra-pengolahan nomor kendaraan, citra plat kendaraan telah didapatkan akan diproses thresholding terlebih dahulu untuk memisahkan karakter dengan *background* plat dengan menghitung nilai ambang (T) digunakan metode *Otsu*. Hasil thresholding tersebut akan dilakukan proses negasi untuk membalikkan nilai *foreground* dan

Copyright © 2013 SESINDO

background untuk membedakan warna plat. Dilanjutkan dengan proses *thinning* dengan algoritma *Zhang-Shuen* untuk menipiskan citra dengan tujuan mengambil kerangka tulisan sehingga dapat lebih mudah di ekstraksi fiturnya dan hasil pengenalan karakternya lebih akurat. Pada proses ekstraksi fitur, akan dikenali karakter dengan mengekstraksi ciri dengan mengeluarkan ciri-ciri khusus dari karakter tersebut dengan algoritma *Directional Feature Extraction*. Pada proses pelatihan, ciri khusus yang telah diperoleh akan dilatih dengan algoritma *Learning Vector Quantization* dengan menghitung jarak tiap bobot sampai nilai bobot tidak berubah atau sebanyak maksimal *epoch* dan dilakukan pembentukan bobot baru dari nilai minimumnya.

4. HASIL

Aplikasi pengenalan karakter pada plat nomor kendaraan yang dibangun memiliki fitur untuk menampilkan waktu yang diperlukan untuk tahap prapengolahan dan waktu pengenalan yang diperlukan untuk mengenali suatu nomor kendaraan bermotor seperti tampilan Gambar 6 berikut.



Gambar 6. Tampilan aplikasi pengenalan karakter pada plat nomor kendaraan

Dengan aplikasi pengenalan karakter pada plat nomor kendaraan yang telah dihasilkan, dilakukan pengujian dengan menggunakan metode *K-Fold Cross Validation* dengan K = 3 terhadap 99 citra plat nomot kendaraan bermotor yang terdiri dari 15 citra berplat putih, 15 citra berplat merah, dan 69 citra berplat hitam. Untuk citra berplat hitam dibedakan atas 50 plat dengan jarak dekat dan 19 plat dengan jarak jauh. Pengujian dilakukan dengan menggunakan 3-*fold*, dimana pembagian data untuk setiap *fold* terdiri dari 23 citra berplat hitam, 5 citra berplat putih, dan 5 citra berplat merah. Adapun hasil pengujian dapat dilihat di Tabel 1.

Tabel 1. Tabel tingkat keberhasilan dan waktu eksekusi pengenalan karakter pada plat nomor kendaraan

Keterangan	**Fold 1**	**Fold 2**	**Fold 3**	**Rata-rata**
Jumlah benar (karakter)	220	228	220	222.667
Jumlah salah (karakter)	39	24	36	33.000
Jumlah Penemuan Karakter bukan bagian plat (karakter)	42	57	38	45.667
Persentase Kebenaran Pengenalan Karakter (%)	84.942	90.476	85.938	87.093
Rerata waktu prapengolahan (Detik)	1.513	1.553	1.582	1.549
Rerata waktu pengenalan karakter (Detik)	0.264	0.254	0.321	0.28
Rerata waktu pengenalan (Detik)	2.694	2.835	3.575	3.034

Dari data di atas dapat diketahui bahwa tingkat keberhasilan pengenalan karakter pada plat kendaraan adalah 87,093%, rata-rata total waktu eksekusi adalah 4,583 detik dan rata-rata waktu pengenalan pengenalan karakter 0,28 Detik.

5. KESIMPULAN DAN SARAN

Copyright © 2013 SESINDO

Berdasarkan aplikasi pengenalan karakter pada plat nomor kendaraan yang telah dihasilkan dan pengujian yang telah dilakukan, maka dapat diambil kesimpulan dan saran.

5.1 Kesimpulan

Berdasarkan hasil pengujian yang telah dilakukan terhadap aplikasi pengenalan karakter pada plat nomor kendaraan, maka kesimpulan yang dapat ditarik adalah sebagai berikut.

1. Tingkat keberhasilan pengenalan karakter pada plat kendaraan bermotor cukup baik yakni 87,093%, rata-rata total waktu eksekusi 4,583 detik dan rata-rata waktu pengenalan pengenalan karakter 0,28 detik.
2. Tingkat kemunculan karakter yang bukan karakter pada plat cukup banyak yakni 14,9%.

5.2 Saran

Untuk penelitian dan pengembangan terhadap aplikasi mengenalan karakter pada plat nomor kendaraan ke depannya, maka dapat disarankan sebagai berikut :

1. Mengatasi masalah plat kendaraan bermotor yang miring menggunakan transformasi *hough*, sehingga mudah unutk mengenali plat kendaraan bermotor dalam posisi miring.
2. Menggunakan algoritma *thinning* yang lain serta menggunakan teknik morfologi lain seperti *opening* dan *closing* untuk menormalisasi karakter.
3. Menambahkan data kecerdasan untuk mengenali kode wilayah nomor kendaraan bermotor, sehingga pengenalan plat kendaraan bermotor jadi lebih terarah.

5. DAFTAR RUJUKAN

[1] Hou, P. G., Zhao, J. dan Liu, M., "A License Plate Locating Method Based on Tophat-bothat Changing and Line Scanning", In Proc. of International Symposium on Instrumentation Science and Technology, 2006, pp. 431–436.

[2] Nagare, A., P., 2011. License Plate Character Recognition System using Neural Network. *IJCA*, 25 (10), pp. 36-39.

[3] Putra, D., 2010. *Pengolahan Citra Digital*. Yogyakarta: Andi Offset.

[4] Zhang, T. Y. dan Suen, C. Y., 1984. A Fast Parallel Algorithm for Thinning Digital Patterns. *CACM*, 27 (3), pp. 236-239.

[5] Gonzales, R. C. dan Woods, R. E., 2007. Digital Image Processing. 3rd ed. USA: Prentice-Hall.

[6] Blumenstein, M., Verma, B., dan Basli, H., 2003, A Novel Feature Extraction Technique for the Recognition of Segmented Handwritten Characters. *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, 1, pp. 137-141.

Copyright © 2013 SESINDO