PENGENALAN KARAKTER JAWI TULISAN TANGAN MENGGUNAKAN FITUR SUDUT

Safrizal
Akademi Komunitas Negeri Aceh Barat
Komplek STTU Alue Peunyareng, Aceh Barat, 23615
e-mail: safrizal@aknacehbarat.ac.id

***Abstrak*— Karakter Jawi salah satu varian karakter Arab adalah karakter Jawi. Karakter tersebut mengakomodasi penulisan bahasa Melayu. Karakter Jawi banyak digunakan oleh kerajaan Islam di Nusantara sehingga banyak dokumen menggunakan karakter tersebut. Untuk menjaga kelestarian dokumen tersebut maka karakter Jawi tersebut harus di rubah kedalam format digital, salah satu caranya adalah dengan metode *Optical Character Recognition* (OCR). Salah satu metode OCR dalam mengekstraksi fitur adalah menggunakan sudut yang dibentuk dari bagian utama karakter. Fitur yang dihasilkan dari sudut tersebut digunakan oleh SVM untuk di klasifikasi kedalam karakter Jawi. Dengan menggunakan fitur tersebut, SVM mampu mengklasifikasi karakter Jawi dengan tingkat akurasi rata-rata sebesar 78,86%.**

Kata kunci: ***OCR, Jawi, sudut, SVM***

***Abstract*— The Jawi character is one variant of Arabic characters that adopts Malay language writing. The character of Jawi is widely used by the Islamic kingdom in the Archipelago. Many document written using these characters. To preserve the history and the book, the Jawi character need to be converted into digital form, one way is by Optical Character Recognition (OCR) method. One method of OCR in extracting features is to use angles formed by main part of the character. The features generated are used by SVM to be classified into Jawi characters. By using these features, SVM is able to classify Jawi characters with an average level of accuracy of 78.86%.**

Keywords: ***OCR, Jawi, angle, SVM***

I. PENDAHULUAN

Optical Character Recognition (OCR) merupakan bagian dari *computer vision* dan *machine learning* yang menerjemahkan karakter hasil cetakan komputer atau tulisan tangan kedalam format digital yang dapat diedit [1]. Pengenalan karakter dapat dilakukan secara *online* maupun *offline*. Pada pengenalan karakter secara *online*, karakter atau teks ditulis secara langsung pada perangkat elektronik, misalnya telepon genggam, tablet dan *personal data assistant*. Pada pengenalan karakter secara *offline*, karakter atau teks di tulis pada kertas dan kemudian di pindai menggunakan *scanner*. Pengenalan karakter secara *offline* lebih sulit dibandingkan dengan *online* [2].

Pada awalnya, pengenalan karakter hanya dilakukan pada karakter-karakter yang tidak terhubung, misalnya karakter latin. Seiring perkembangan metode ekstraksi fitur dan klasifikasi, pengenalan karakter mengalami peningkatan performansi serta mampu mengenali karakter-karakter dalam bentuk terhubung, misalnya karakter Arab [3]. Karakter Arab tersebut diadopsi oleh karakter-karakter bahasa lain misalnya Jawi dan Farsi untuk mengakomodasi bahasa lokal. Karakter Jawi terdiri 35 aksara, yang terdiri dari 29 aksara berdasarkan bahasa Arab dan 6 karakter tambahan untuk mengakomodasi bahasa Melayu [3].

Penelitian pengenalan karakter Jawi relatif lebih sedikit dibandingkan dengan penelitian aksara Arab, Jepang, Cina dan Latin [4]. Beberapa penelitian sebelumnya tentang pengenalan aksara Jawi dilakukan oleh [5] menggunakan *Hidden Markov Models*, [6] menggunakan *Trace Transform*, [4] menggunakan FCC dan SVM dan [7] menggunakan *Hybrid Artificial Neural Networks* dan *Dinamic Programming*.

Karakter Jawi saat ini tidak lagi digunakan pada kegiatan resmi kenegaraan. Padahal banyak sekali sejarah-sejarah kerajaan serta kitab-kitab agama Islam ditulis dalam bahasa Jawi. Untuk menjaga kelestarian sejarah dan ilmu agama Islam diperlukan sebuah metode yang mampu menghasilkan dokumen digital, salah satu metodenya adalah OCR.

II. STUDI PUSTAKA

Agar dapat diklasifikasi oleh *classifier* diperlukan beberapa proses *pre-processing* agar menghasilkan fitur yang optimal. Proses *pre-processing* tersebut adalah binerisasi dan *thinning*.

A. Binerisasi

Pada saat dipindai, citra dari setiap karakter biasanya memiliki *noise*. Untuk menghilangkan noise tersebut dilakukan *pre-processing*. Pada tahap *pre-processing*, setelah dihilangkan *noise* maka biasanya dilakukan

binerisasi. Binerisasi merupakan proses memisahkan nilai *pixel* dari citra kedalam dua bagian yaitu bagian latar (*background*) dan bagian tulisan (*foreground*) [8]. Terdapat dua metode dalam proses binerisasi yaitu binerisasi global dan binerisasi lokal. Beberapa teknik binerisasi global yaitu *fixed thresholding method*, metode Otsu dan Metode Kittler. Pada binerisasi lokal, teknik yang digunakan biasanya metode Niblack, metode adaptif, metode Sauvola dan metode Bernsen [9].

Salah satu metode binerisasi yang cukup popular dalam pengenalan karakter adalah metode Otsu. Metode Otsu melakukan binerisasi secara otomatis pada suatu dokumen berdasarkan bentuk dari histogram [10]. Metode ini mencari sebuah nilai *threshold* optimum untuk memisahkan antara *background* dan *foreground*. Nilai *threshold* optimum tersebut adalah *k*, nilai *k* ini merupakan level *threshold* dan *L* merupakan level keabuan citra [10].

$$\sigma_B^2(k*) = \max_{1 \le k \le L} \sigma_B^2(k) \quad (1)$$

B. *Thinning*

Citra karakter tulisan tangan biasanya memiliki ketebalan lebih satu *pixel*. Ketebalan *pixel* dapat berbeda tergantung dengan jenis alat tulis yang digunakan. Perbedaan tersebut dapat mengakibat fitur tidak optimal. Untuk mengoptimalkan fitur agar dapat diklasifikasi dengan baik, maka dilakukan pengikisan ketebalan menjadi satu *pixel*. Proses pengikisan disebut dengan *thinning* yang akan menghasilkan kerangka karakter.

Salah satu metode *thinning* yang cukup popular adalah metode Zhang Suen [11]. Metode tersebut kemudian di modifikasi agar dapat menghasilkan kerangka yang lebih baik dengan mempertahankan *pixel* diagonal dan persegi dengan 8 konektivitas tetangga [12].

z4	*z3*	*z2*
z5	*p*	*z1*
z6	*z7*	*z8*

Gambar 1. 8 – konektivitas tetangga Zhang Suen hasil modifikasi [12]

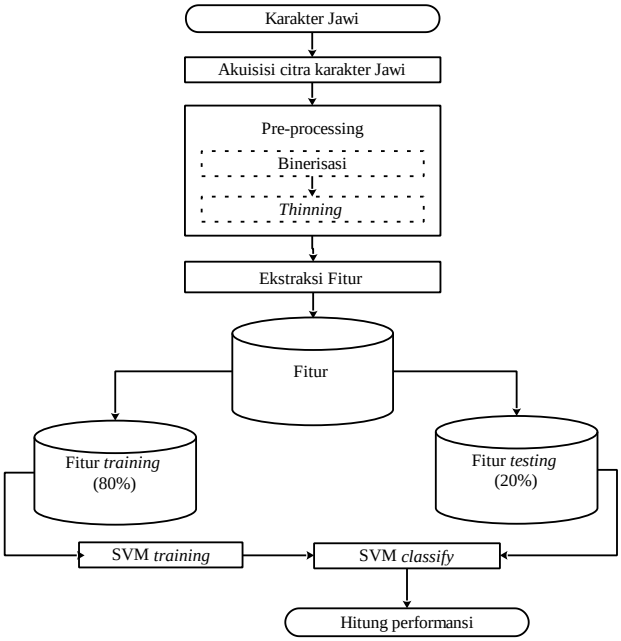
III. METODE

A. *Objek Penelitian*

Penelitian ini menggunakan karakter hasil tulisan *offline* dari 10 orang penulis dengan latar belakang pendidikan, profesi dan usia yang berbeda. Masing-masing penulis diminta untuk menuliskan karakter Jawi pada kertas HVS dengan latar belakang putih. Karakter Jawi tersebut di *scan* dan di *crop* dengan ukuran berkisar antara 148×148 px sampai dengan 179×177 px. Karakter hasil *crop* tersebut disimpan dalam sebuah basis data. Karakter hasil tulisan ini juga digunakan pada penelitian [4].

B. *Alur Penelitian*

Alur penelitian dimulai ekstraksi fitur sudut dan di klasifikasi menggunakan SVM. Gambar 2 berikut menggambarkan alur lengkap penelitian.

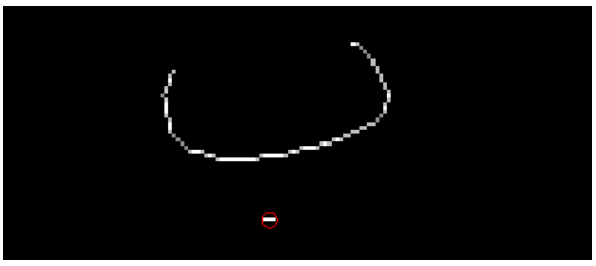


Gambar 2. Alur penelitian

C. *Ekstraksi Fitur Jumlah Titik*

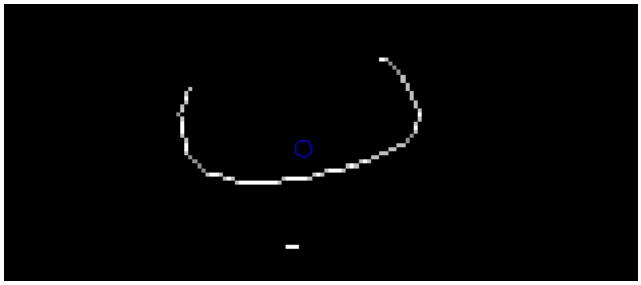
Seperti halnya pada karakter Arab, karakter Jawi juga memiliki titik. Terdapat beberapa karakter Jawi yang memiliki bentuk bagian utama yang sama, misalnya karakter "*ba*", "*ta*" dan "*tsa*" namun memiliki jumlah dan letak titik yang berbeda.

Pada penelitian ini, titik didefinisikan sebagai luas *pixel* yang terhubung antara 1 sampai dengan 30 *pixel*. Jika ditemukan nilai *pixel* 1 pada karakter biner, maka akan dilakukan penulusuran konektivitas dengan *mask* 3x3. Jika konektivitas lebih dari 30 *pixel* maka dianggap sebagai bagian utama karakter. Jika jumlah titik satu maka fitur jumlah titik (*JT*) adalah satu [*JT*=1], jika ditemukan dua titik maka [*JT*=2] dan jika ditemukan tiga titik maka [*JT*=3].



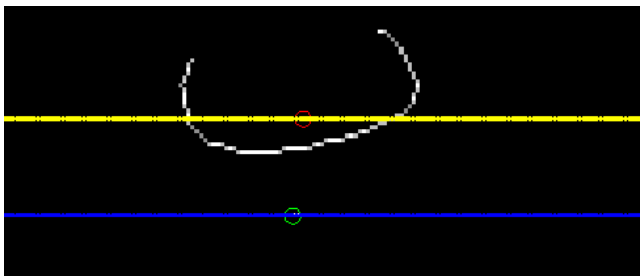
Gambar 3. Ekstraksi fitur jumlah titik

D. *Ekstraksi Fitur Letak Titik*



Gambar 4. Pusat massa

Proses ekstraksi letak titik dimulai dengan menentukan pusat massa dari setiap karakter. Pusat massa merupakan titik sentral dari sebuah karakter termasuk titik. Gambar 4 berikut menunjukkan pusat massa dari karakter "*ba*". Perbedaan baris pusat massa dengan baris letak titik dijadikan fitur letak titik (*LT*) seperti yang terlihat pada gambar 5. Karena maksimum jumlah titik pada karakter Jawi berjumlah tiga, maka panjang fitur *LT* adalah 3. Fitur *LT* disusun dengan format [*LT*= (*Letak Titik 1*) (*Letak Titik 2*) (*Letak Titik 3*)]



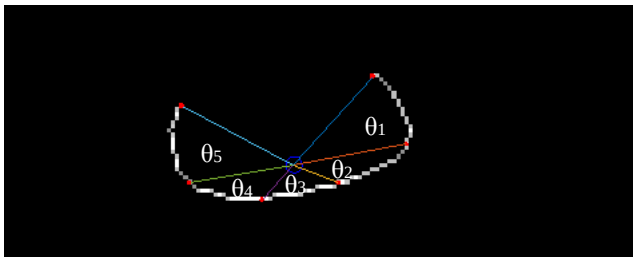
Gambar 5. Selisih baris pusat massa dan letak titik

E. Ekstraksi Fitur Sudut

Perbedaan kurva menjadi ciri khas bentuk bagian utama karakter Jawi. Perbedaan tersebut membentuk sudut yang berbeda-beda. Dengan mengekstraksi sudut bagian utama aksara maka didapatkan fitur berupa sudut.

Untuk mendapatkan sudut maka bentuk bagian utama karakter dibagi kedalam lima bagian, seperti yang terlihat pada gambar 6. Masing-masing bagian memiliki sudut yang berbeda-beda sehingga menghasilkan fitur yang unik. Fitur sudut (*S*) disusun kedalam vektor [$S = \theta_1, \theta_1, \theta_1, \theta_1, \theta_1$] Sudut tersebut dihasilkan dari perkalian titik (*dot product*) dengan menggunakan persamaan 2 berikut.

$$\theta = arc\ cos\ \frac{a\ .\ b}{|a|\ .\ |b|} \qquad (2)$$



Gambar 6. Pembagian sudut

F. Klasifikasi

Kemampuan SVM dalam mengklasifikasi *multiclass* menjadikan klasifikasi seluruh karakter Jawi cukup dilakukan satu kali. Metode yang digunakan untuk klasifikasi *multiclass* SVM dapat berupa *one-against-all* dan *one-against-one*. Penelitian ini menggunakan metode *one-against-one*. Dengan menggunakan metode tersebut maka model klasifikasi biner yang dibangun adalah 35*(35-1)/2=595 buah. Dari setiap model yang sudah dibangun tersebut diterapkan hasil voting sehingga menghasilkan satu karakter hasil klasifikasi.

G. Prosedur Pengujian

Seluruh fitur (*F*) disusun dengan format [*F=JT LT S*]. Fitur tersebut digunakan SVM untuk diklasifikasikan kedalam karakter Jawi. Akurasi klasifikasi dihitung dengan persamaan 3 berikut:

$$Akurasi = \frac{Benar\ klasifikasi}{Total\ klasifikasi} \times 100\% \qquad (3)$$

IV. HASIL DAN PEMBAHASAN

A. Implementasi

Implementasi pengenalan karakter Jawi menggunakan sudut sebagai fitur membutuhkan beberapa komponen pendukung yaitu aplikasi *Matlab* dan komputer yang mampu menjalankan aplikasi *Matlab* tersebut. Implementasi ini secara umum dapat dibagi kedalam beberapa modul, yaitu akuisisi citra karakter, binerisasi, *thinning*, ekstraksi fitur dan klasifikasi dengan SVM.

B. Hasil Pengujian

Pada pengujian ini, karakter hasil akusisi melalui *scanner* yang berjumlah 350, dibagi kedalam dua bagian yaitu karakter *training* dan karakter testing. 80% × 350 = 280 karakter dijadikan sebagai fitur *training*, sedangkan sisanya 20% × 350 = 70 karakter dijadikan fitur testing.

Seperti yang terlihat pada gambar 2, pembagian fitur *training* dan fitur testing dilakukan secara acak. Untuk meningkatkan keakurasian, maka dilakukan pengulangan klasifikasi sebanyak 10 kali, dengan asumsi setiap karakter memiliki probabilitas terpilih sebanyak 1 kali. Hasil pengujian dapat dilihat pada tabel 1 dibawah ini.

Tabel 1. Akurasi hasil pengujian

Pengujian	Persentase Akurasi (%)
1	80
2	78,57
3	84,29
4	72,86
5	81,42
6	74,29
7	75,71
8	78,57
9	82,86
10	80
rata-rata	**78,86**

Akurasi paling tinggi yang didapat adalah sebesar 84,29% sedangkan yang paling rendah adalah sebesar 72,86%. Rata-rata keberhasilan klasifikasi adalah sebesar 78,86%. Karakter "*alif*", "*ba*", "*ta*" mampu diklasifikasi secara sempurna, sedangkan karakter "*tsa*" hanya mampu diklasifikasi "90%". Hal ini disebabkan kemiripan karakter "*tsa*" dan "*nya*" (ت dan ڽ). Karakter yang paling rendah mampu diklasifikasi adalah karakter "*dhad*" (ض) dan

"*hamzah*" (ء). Kedua karakter tersebut hanya mampu diklasifikasi sebesar 45%. Hasil lengkap klasifikasi dapat dilihat pada tabel 2.

Tabel 2. Akurasi klasifikasi per karakter

Karakter		Keberhasilan		Persentase	
Latin	Jawi	Berhasil	Gagal	Berhasil (%)	Gagal (%)
alif	ا	20	0	100	0
ba	ب	20	0	100	0
ta	ت	20	0	100	0
tsa	ث	18	2	90	10
jim	ج	19	1	95	5
hha	ح	14	6	70	30
kha	خ	14	6	70	30
dal	د	15	5	75	25
dzal	ذ	17	3	85	15
ra	ر	15	5	75	25
zai	ز	20	0	100	0
sin	س	20	0	100	0
syin	ش	17	3	85	15
shad	ص	17	3	85	15
dhad	ض	9	11	45	55
tho	ط	10	10	50	50
zho	ظ	16	4	80	20
ain	ع	13	7	65	35
ghain	غ	16	4	80	20
fa	ف	16	4	80	20
qaf	ق	20	0	100	0
kaf	ک	19	1	95	5
lam	ل	19	1	95	5
mim	م	13	7	65	35
nun	ن	17	3	85	15
wau	و	13	7	65	35
ha	ه	11	9	55	45
hamzah	ء	9	11	45	55
ya	ي	20	0	100	0
nya	ڽ	17	3	85	15
ca	چ	20	0	100	0
nga	ڠ	20	0	100	0
pa	ڤ	11	9	55	45
ga	ڬ	7	13	35	65
va	ۏ	10	10	50	50

V. KESIMPULAN

Pengenalan karakter tulisan tangan Jawi pada penelitian ini menggunakan fitur sudut yang dihasilkan dari kurva bagian utama karakter. Penggunaan fitur sudut mampu di klasifikasi SVM kedalam karakter Jawi dengan rata-rata keberhasilan sebesar 78,86%. Beberapa kegagalan klasifikasi terjadi karena sudut yang dibentuk oleh kurva memiliki kesamaan dengan karakter lain.

REFERENSI

[1] H. Modi and M. C., "A Review on Optical Character Recognition Techniques," Int. J. Comput. Appl., vol. 160, no. 6, pp. 20–24, 2017.

[2] L. M. Lorigo and V. Govindaraju, "Offline Arabic Handwriting Recognition :," IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 5, pp. 712–724, 2006.

[3] M. F. Nasrudin, K. Omar, M. S. Zakaria, and L. C. Yeun, "Handwritten Cursive Jawi Character Recognition : A Survey," pp. 247–256, 2008.

[4] . S., F. Arnia, and R. Muharar, "Pengenalan Aksara Jawi Tulisan Tangan Menggunakan Freemen Chain Code (FCC), Support Vector Machine (SVM) dan Aturan Pengambilan Keputusan," J. Nas. Tek. Elektro, vol. 5, no. 1, p. 45, 2016.

[5] R. Redika, F. Teknologi, K. Omar, F. Teknologi, and F. Teknologi, "Handwritten Jawi Words Recognition Using Hidden Markov Models," pp. 4–8, 2008.

[6] M. F. Nasrudin, M. Petrou, and L. Kotoulas, "Jawi Character Recognition using the Trace Transform," 2010.

[7] A. Heryanto, F. Teknologi, S. Maklumat, F. Teknologi, K. Omar, and F. Teknologi, "Offline Jawi Handwritten Recognizer Using Hybrid Artificial Neural Networks and Dynamic Programming," 2008.

[8] T. R. Singh, S. Roy, O. I. Singh, T. Sinam, and K. M. Singh, "A New Local Adaptive Thresholding Technique in Binarization," vol. 8, no. 6, pp. 271–277, 2012.

[9] N. K. Garg, "Binarization Techniques used for Grey Scale Images," vol. 71, no. 1, pp. 8–11, 2013.

[10] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," IEEE Trans. Syst. Man. Cybern., 2008.

[11] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," Commun. ACM, 1984.

[12] L. Lam and S. W. Lee, "Thinning methodologies—a comprehensive survey," IEEE Trans. Pattern Anal. Mach. Intell., 1992.