PENGENALAN AKSARA JAWI TULISAN TANGAN MENGGUNAKAN FREEMAN CHAIN CODE (FCC), SUPPORT VECTOR MACHINE (SVM) DAN ATURAN PENGAMBILAN KEPUTUSAN

Safrizal*, Fitri Arnia, Rusdha Muharar

Program Studi Magister Teknik Elektro, Universitas Syiah Kuala

*Corresponding author, e-mail : rizal.el@gmail.com

Abstrak—Aksara Jawi merupakan salah satu varian dari aksara Arab yang terdiri dari 35 aksara. Dari 35 aksara Jawi tersebut terdapat beberapa aksara dengan bentuk bagian utama yang sama namun memiliki letak dan jumlah titik yang berbeda. Karena perbedaan tersebut maka proses pengenalan aksara Jawi dapat dilakukan dengan melakukan klasifikasi berdasarkan perbedaan bentuk bagian utama. Pada penelitian ini *Freeman Chain Code* (FCC) digunakan sebagai ekstraksi fitur dan *Support Vector Machine* (SVM). FCC digunakan untuk merepresentasikan garis batas (*boundary*) aksara Jawi kedalam kode rantai. Kode rantai tersebut diklasifikasi dengan menggunakan SVM kedalam 19 kelompok. Fitur letak titik dan jumlah titik digunakan sebagai aturan pengambilan keputusan terhadap 19 kelompok hasil klasifikasi SVM kedalam aksara Jawi. Aksara Jawi yang digunakan merupakan tulisan tangan dari 10 orang penulis dari berbagai latar belakang dan umur. Tingkat keberhasilan klasifikasi penelitian ini mencapai 80,00%.

Kata Kunci : aksara Jawi, tulisan tangan, FCC, SVM, aturan pengambilan keputusan

Abstract—Jawi is one variant of Arabic script consists of 35 characters. Some of Jawi characters have the same main shape, but different number of dots in different location. Thus, recognition process of Jawi characters can be done by performing a classification based on the main shape. In recognition process, feature extraction plays an important role. In this research, Freeman Chain Code (FCC) was used as feature extraction and Support Vector Machine (SVM) as classifier. Then we apply the decision rules to classify SVM result into Jawi characters. FCC is used to represent the boundary of Jawi characters into a chain code. Then the chain code is used by SVM to classify the characters into 19 groups. Feature of location and the number of dots are used by decision rules to classify the groups into Jawi characters. The Jawi characters are handwritten and generated by 10 writers from different backgrounds and ages. The recognition rate of this research was 80.00%.

Keywords : Jawi script, handwriting, FCC, SVM, decision rules

Copyright © 2016 JNTE. All rights reserved

1. PENDAHULUAN

Aksara Jawi adalah aksara yang berasal dari tulisan Arab yang digunakan untuk menuliskan bahasa Melayu. Aksara Jawi muncul bersamaan dengan masuknya Islam ke nusantara, kata nusantara merujuk pada daerah yang didiami bangsa Melayu [1]. Kata Jawi berasal dari bahasa Arab, Ibn Batutah menggunakan kata *al-jawah* yang merujuk kata nusantara. Kata *jawah* dan Jawi tidak hanya merujuk kepada kepulauan Jawa dan penduduknya saja namun digunakan untuk seluruh daerah Asia Tenggara yang didiami bangsa Melayu [2]. Aksara Jawi sekarang ini sudah tidak digunakan pada kegiatan-kegiatan resmi kenegaraan, aksara tersebut hanya digunakan pada kegiatan

pendidikan agama Islam. Padahal banyak sekali kitab-kitab dan dokumen-dokumen dalam aksara Jawi yang ditulis sebelum pertengahan abad 19 yang merupakan sejarah yang penting dijaga kelestariannya. Salah satu cara menjaga kelestarian kitab-kitab dan dokumen-dokumen tersebut adalah dengan merubah kitab-kitab dan dokumen-dokumen tersebut kedalam bentuk digital melalui aplikasi *Optical Character Recognition* (OCR).

Pada awalnya OCR digunakan untuk mengenali jenis tulisan hasil cetakan atau dokumen ketikan, namun sekarang ini terjadi perubahan kecenderungan untuk mengenali karakter dokumen tulisan tangan. Dokumen tulisan tangan lebih sulit dikenali oleh OCR karena perbedaan gaya penulisan antara satu

penulis dengan penulis lainnya. Secara umum pengenalan tulisan tangan terbagi dalam tiga tahap yaitu *pre-processing*, ekstraksi fitur dan klasifikasi. *Pre-processing* adalah tahap untuk membersihkan citra aksara dari *noise* sehingga dapat digunakan secara efektif pada tahap ekstraksi fitur. Ekstraksi fitur merupakan tahap menghilangkan data yang tidak perlu dan mengambil ciri khusus dari setiap aksara. Tahap terakhir adalah klasifikasi yang digunakan untuk mengenali aksara.

Penelitian pengenalan karakter tulisan tangan Latin, Cina, Jepang, Arab dan India sudah banyak dilakukan namun penelitian pengenalan aksara Jawi tulisan tangan relatif lebih sedikit yang melakukannya. Penelitian tentang pengenalan karakter Jawi sudah dilakukan oleh beberapa peneliti seperti yang dilakukan oleh [3] menggunakan *Hidden Markov Model* (HMM) tanpa dilakukan segmentasi citra kedalam urutan *frame*, fitur *geometric* di ekstraksi menggunakan *sliding window* dari setiap *frame*. Penelitian yang dilakukan oleh [4] menggunakan *hybrid Artificial Neural Network* (ANN) dan *dynamic programming* untuk pengenalan karakter dan *Viterbi Dynamic Programming* (VDP) sebagai verifikator karakter Jawi. Penelitian yang dilakukan oleh [5] menggunakan *Trace transform* yang merupakan bentuk umum *Radon transform* untuk membedakan antar karakter Jawi, fitur di ekstraksi menggunakan metode *object signature* yang invarian terhadap *affine distortion*.

Dari beberapa penelitian diatas dan sejauh penelitian penulis tidak ditemui adanya penelitian pengenalan aksara Jawi menggunakan FCC sebagai metode ekstraksi fitur dan SVM sebagai *classifier* serta menerapkan aturan pengambilan keputusan terhadap hasil klasifikasi SVM. Penelitian pengenalan aksara Jawi dengan fitur FCC dan SVM sebagai *classifier* serta menerapkan aturan pengambilan keputusan terhadap hasil klasifikasi SVM menghasilkan sebuah metode alternatif untuk membangun sistem OCR aksara Jawi.

Pada makalah ini diperkenalkan sebuah metode pengenalan aksara Jawi dengan melakukan ekstraksi fitur dan klasifikasi fitur secara otomatis. Secara khusus penelitian ini bertujuan melakukan ekstraksi fitur aksara Jawi dengan metode FCC dan menghitung performansi klasifikasi dengan menggunakan SVM dan aturan pengambilan keputusan.

2. TINJAUAN PUSTAKA

2.1. Pre-Processing

Ketika sebuah citra di akuisisi maka citra tersebut akan disimpan dalam bentuk nilai kerapatan piksel, untuk citra keabuan maka nilai piksel akan bernilai dari 0-255. Citra aksara tulisan tangan akan lebih optimal untuk diproses jika dalam bentuk biner. *Thresholding* merupakan teknik utama dalam analisis similaritas yang dapat menentukan ambang batas yang tepat untuk setiap pengolahan citra sehingga suatu citra keabuan dapat diubah kedalam level biner [6]. Penerapan *threshold* dengan nilai tertentu (*fixed threshold*) sangat berisiko untuk diterapkan pada citra hasil tulisan tangan karena dapat menghasilkan citra biner yang tidak sempurna sehingga mempersulit pembentukan pola tulisan tangan. *Threshold* ada dua jenis yaitu *local thresholding* dan *global thresholding*. Perbedaan antara keduanya adalah metode penentuan nilai *threshold*.

Metode Otsu adalah salah satu bentuk *global thresholding* yang secara otomatis mereduksi citra keabuan berdasarkan histogram dengan ambang batas tertentu kedalam bentuk biner. Pendekatan yang dilakukan dengan metode Otsu adalah dengan melakukan analisis ambang batas optimum yang dapat memisahkan antara *background* dan *foreground*. Ambang batas optimum yang akan memisahkan antara *background* dan *foreground* pada metode Otsu adalah k dimana k adalah level *threshold* dan L merupakan tingkat keabuan citra [7].

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

Setelah dilakukan binerisasi maka proses selanjutnya adalah mendapatkan kerangka aksara. Untuk mendapatkan kerangka aksara maka yang diperlukan adalah proses *thinning*. *Thinning* adalah proses pengurangan data biner yang tidak perlu dengan cara mengikis suatu objek menjadi berukuran 1 piksel sehingga menghasilkan kerangka tanpa merubah pola objek asalnya. Tujuan dari *thinning* adalah menghilangkan beberapa piksel tertentu pada suatu objek dengan tetap menjaga agar tidak memutuskan koneksi antar piksel. Banyak sekali metode *thinning* dengan tingkat kompleksitas, efisiensi dan akurasi yang berbeda. Salah satu metode yang cukup populer dan sederhana

adalah algoritma Zhang-Suen. Zhang-Suen adalah algoritma *thinning fast parallel* yang diperkenalkan oleh Zhang dan Suen pada tahun 1984 [8].

Seiring perkembangan metode *thinning*, algoritma Zhang Suen juga mengalami modifikasi dan menghasilkan kualitas citra *thinning* yang lebih baik. Salah satu modifikasi algoritma Zhang Suen agar penipisan dapat membentuk skeleton citra dengan 8 konektifitas tetangga dan mempertahankan garis diagonal dan persegi ukuran 2x2 diperkenalkan oleh [9].

Gambar 1. Diagram 8-neighborhood Zhang Suen modifikasi [9]

1. Pada subiterasi pertama, hapus piksel p hanya jika kondisi $G1$, $G2$ dan $G3$ semuanya memenuhi.
2. Pada subiterasi kedua, hapus piksel p hanya jika kondisi $G1$, $G2$ dan $G3'$ semuanya memenuhi.

Kondisi $G1$:

$$X_H(p) = 1$$

dimana,

$$X_H(p) = \sum_{i=1}^4 b_i$$

$$b_i = f(x) = \begin{cases} 1, & \text{jika } x_{2i-1} \text{ and } (x_{2i} = 1 \text{ or } x_{2i+1} = 1) \\ 0, & \text{lainnya} \end{cases}$$

Kondisi $G2$:

$$2 \leq \min\{n_1(p), n_2(p)\} \leq 3$$

dimana,

$$n_1(p) = \sum_{k=1}^4 x_{2k-1} \vee x_{2k}$$

$$n_2(p) = \sum_{k=1}^4 x_{2k} \vee x_{2k+1}$$

dimana $\vee$ merupakan operator “or” Kondisi $G3$:

$$(x_2 \vee x_3 \vee \bar{x}_8) \wedge x_1 = 0$$

dimana $\wedge$ merupakan operator “and” Kondisi $G3$:

$$(x_6 \vee x_7 \vee \bar{x}_4) \wedge x_5 = 0$$

2.2. Freeman Chain Code (FCC)

Ekstraksi fitur merupakan proses mencari nilai keunikan karakteristik pada sebuah aksara. Dengan adanya nilai-nilai tersebut maka sebuah aksara dapat dibedakan dengan aksara lainnya. Nilai karakteristik yang dihasilkan dari proses ekstraksi fitur selanjutnya digunakan sebagai vektor fitur dan *support vector* oleh SVM.

Chain code atau kode rantai banyak digunakan dalam pengolahan citra untuk merepresentasikan garis, kurva atau batas tepi dari suatu daerah. Kode rantai merepresentasikan garis batas (*boundary*) yang membentuk suatu aksara, kode pada *chain* menunjukkan arah pembacaan piksel. *Chain code* pertama kali diperkenalkan oleh Freeman pada tahun 1961 yang dikenal dengan nama FCC [10]. Terdapat dua arah kode yaitu 4-neighborhood dan 8-neighborhood.

Gambar 2. FCC (a) 4-neighborhood dan (b) 8-neighborhood [11]

Penentuan arah pembacaan kode rantai dapat dimulai dari kiri atas, kiri bawah, kanan atas dan kanan bawah. Setelah ditemukan titik awal dari kode rantai dari sebuah aksara, maka arah pembacaan piksel yang terhubung akan menentukan nilai rantai dari FCC. Contohnya pada FCC 8-neighborhood jika arah pembacaan piksel ke arah timur maka nilai rantai adalah 1, ke arah tenggara nilai rantai adalah 8 dan ke arah barat nilai rantai adalah 5.

3. METODOLOGI PENELITIAN

3.1. Objek Penelitian

Pada pengenalan dokumen hasil tulisan tangan, baik secara *online* maupun *offline*, pengenalan aksara akan sangat dipengaruhi oleh gaya penulisan masing-masing penulis bahkan penulis yang sama dapat menuliskan bentuk aksara yang berbeda pada saat yang berbeda. Distorsi dan *noise* pada saat digitalisasi juga merupakan faktor utama yang dapat mengurangi akurasi pengenalan.

Pada penelitian ini dilakukan pengenalan aksara Jawi yang terdiri dari 35 aksara terpisah. Aksara yang terhubung baik diawal, ditengah maupun diakhir tidak digunakan. Untuk mendapatkan citra aksara Jawi maka 10 orang dipilih dengan berbagai latar belakang dan umur untuk menuliskan sampel aksara pada kertas ukuran A-4 dengan latar belakang putih. Kertas yang sudah dituliskan aksara Jawi tersebut kemudian di *scan* dengan format *Red Green Blue* (RGB) dan *grayscale*. Aksara tersebut kemudian di *crop* dengan ukuran berbeda-beda berkisar antara 148x148 piksel sampai dengan 179x177 piksel. Setiap citra yang diakuisisi dalam format RGB kemudian di konversi kedalam format *grayscale* sedangkan jika citra yang diakuisisi dalam format *grayscale* maka citra tersebut tidak dilakukan lagi proses konversi citra ke format *grayscale*.

Gambar 3. Proses akuisisi aksara “nga”: (1) citra RGB asli, (2) citra grayscale, (3) citra biner

Dari setiap penulis diakuisisi 35 sampel aksara dengan bentuk terpisah dan dimasukkan kedalam *database* sehingga total aksara yang disimpan kedalam *database* adalah 350 aksara. Sampel yang diambil dari 10 penulis tersebut memiliki variasi pada bentuk tulisannya, variasi penulisan tersebut disebabkan oleh beberapa aspek diantaranya profesi penulis, usia penulis dan pendidikan penulis. Penulisan aksara

tersebut memiliki ukuran tulisan dan ketebalan pena yang berbeda-beda.

3.2. Alur Penelitian

Alur penelitian menggambarkan proses pengenalan aksara Jawi menggunakan ekstraksi fitur FCC dan klasifikasi dengan SVM serta penerapan aturan pengambilan keputusan. Hasil klasifikasi SVM merupakan 19 kelompok aksara seperti yang terlihat pada Tabel 1. Untuk mendapatkan aksara Jawi maka hasil klasifikasi SVM yang merupakan bentuk bagian utama aksara, letak titik (*P1*) dan jumlah titik (*P2*) akan diuji oleh aturan pengambilan keputusan. Alur lengkap penelitian ini dapat dilihat pada Gambar 4.

Tabel 1. Kelompok aksara Jawi *training* berdasarkan bentuk bagian utama

No.	Kelompok aksara	Bentuk bagian utama aksara	Aksara Jawi			
			1	2	3	4
1	Alif	ا	ا	-	-	-
2	BaTaTsa	ب	ب	ت	ث	-
3	JimHhaKhaCa	ح	ج	خ	چ	-
4	DalDzal	د	ذ	-	-	-
5	RaZai	ر	ز	-	-	-
6	SinSyin	س	ش	-	-	-
7	ShadDhad	ص	ض	-	-	-
8	ThoZho	ط	ظ	-	-	-
9	AinGhainNga	ع	غ	-	-	-
10	FaPa	ف	ق	-	-	-
11	Qaf	ق	ك	-	-	-
12	KafGa	ك	ل	-	-	-
13	Lam	ل	م	-	-	-
14	Mim	م	ن	-	-	-
15	NunNya	ن	و	-	-	-
16	WauVa	و	ه	-	-	-
17	Ha	ه	ء	-	-	-
18	Hamzah	ء	ي	-	-	-
19	Ya	ي	-	-	-	-

3.3. Ekstraksi Fitur Letak Titik dan Jumlah Titik

Untuk melakukan ekstraksi fitur letak titik dan jumlah titik maka setiap aksara harus melewati tahap *pre-processing* terlebih dahulu. Ekstraksi fitur merupakan suatu cara pengambilan ciri dari sebuah citra. Ciri dari sebuah citra terdiri dari perbedaan bentuk, ukuran dan orientasi citra. Pada penelitian ini fitur aksara Jawi di ekstraksi dari hasil *thinning* dengan metode FCC sehingga didapatkan ciri

husus dari setiap aksara Jawi dalam bentuk kode rantai. Semua aksara yang memiliki titik, baik titik berjumlah satu, dua ataupun tiga maka di ekstraksi secara terpisah.

Fitur aksara *training* hasil ekstraksi disusun berdasarkan arah pembacaan $F = [FCC]$. Sedangkan aksara *testing* terdiri letak titik dan jumlah titik disusun berdasarkan urutan $F = [P1, P2, FCC]$, dimana $P1$ adalah letak titik dan $P2$ adalah jumlah titik.

Titik didefinisikan sebagai luas piksel tetangga yang saling terhubung dengan luas antara 1 piksel sampai dengan 30 piksel. Jika pada aksara hasil *thinning* ditemukan piksel yang bernilai 1 maka akan dilakukan *trace* konektifitas ketetangga piksel bernilai 1 tersebut dengan *mask* 3x3 sehingga diketahui luas piksel yang saling terhubung.

Pada penelitian ini, titik pada aksara Jawi dibedakan dalam tiga letak yaitu di atas, di

tengah dan di bawah. Bagian atas didefinisikan sebagai letak titik yang terletak diatas bagian utama aksara dan 6 piksel dibawah batas atas bagian utama aksara, hal ini dilakukan karena beberapa penulis menuliskan aksara yang memiliki titik diatas namun titik tersebut dituliskan agak ke tengah. Bagian bawah didefinisikan sebagai letak piksel titik dibawah bagian utama sedangkan bagian tengah adalah bagian yang terletak antara bagian atas dan bagian bawah. Pada Gambar 5 dapat dilihat pembagian letak titik, titik yang terletak di bagian atas garis kuning atau bagian atas maka $P1 = 1$, titik yang terletak di bagian bawah bagian biru atau bagian bawah maka $P1 = 2$, titik yang terletak diantara bagian kuning dan bagian biru atau bagian tengah maka $P1 = 3$ dan jika aksara yang dibaca tidak memiliki titik maka $P1 = 4$.

Gambar 4. Alur penelitian

Sama seperti letak titik, jumlah titik pada aksara Jawi juga sangat menentukan aksara yang dibaca. Sebagai contoh aksara “ف” dan “ف” memiliki bentuk utama aksara yang sama namun memiliki jumlah titik yang berbeda.

Gambar 5. pembagian letak titik: (1) aksara “nga” dan (2) aksara “ta”

Jumlah titik didefinisikan sebagai banyaknya titik pada suatu aksara yang diisolasi oleh piksel bernilai 0. Jumlah maksimal titik pada setiap aksara Jawi yang memiliki titik adalah 3. Jika pada aksara tersebut tidak ada titik maka $P2 = 1$ jika jumlah titik berjumlah satu maka $P2 = 2$, jika jumlah titik berjumlah dua maka $P2 = 3$ dan jika jumlah titik berjumlah tiga maka $P2 = 4$.

3.4. Ekstraksi Fitur FCC

Fitur FCC dipilih karena pernah digunakan [11] untuk mengesktrak fitur aksara *English*, pada penelitian tersebut ekstraksi fitur berhasil dilakukan. Selain itu fitur FCC dipilih karena mampu membentuk pola dengan tingkat ketelitian yang sangat baik dimana setiap arah pembacaan per piksel dapat menghasilkan satu kode rantai.

Gambar 6. FCC aksara “nga” tanpa titik

Penentuan arah baca FCC aksara Jawi dilakukan dengan menentukan *endpoint* dari suatu aksara. Penulisan aksara Jawi dari kanan ke kiri sehingga membentuk aksara yang lebih dominan condong ke kiri. Karena dominan

condong ke kiri maka pemilihan kolom paling kecil sebagai piksel awal pembacaan FCC merupakan sebuah pilihan yang baik.

Untuk menentukan kode rantai maka piksel ke-2 dikurangi dengan piksel ke-1 berdasarkan baris (x) dan kolom (y), piksel ke-3 dikurangi piksel ke-2 dan seterusnya. Uraian penentuan nilai kode rantai berdasarkan aturan pada Tabel 2.

Misalnya aksara “nga” tanpa titik seperti pada Gambar 6 didapatkan kode rantai 455677881155667778882126564443332244332118, Untuk menghilangkan kode rantai yang tidak penting maka nilai kode rantai yang frekuensinya berjumlah 1 maka dihilangkan sehingga menghasilkan kode rantai 55778811556677788844433322442211.

Tabel 2. Penentuan Nilai FCC

Baris (x)	Kolom (y)	Nilai FCC
0	+1	1
-1	+1	2
-1	0	3
-1	-1	4
0	-1	5
+1	-1	6
+1	0	7
+1	+1	8

Untuk membentuk kode rantai suatu aksara agar efisien digunakan *classifier* maka dilakukan normalisasi panjang kode rantai tersebut, panjang kode rantai dinormalisasi agar memiliki dimensi yang sama untuk setiap aksara. Frekuensi kemunculan kode rantai dijumlahkan agar dapat dilakukan normalisasi panjangnya, panjang kode rantai hasil normalisasi adalah n . Frekuensi ternormalisasi dihitung dengan persamaan 2, dimana ΣF_i adalah total frekuensi kode rantai dan F_i adalah frekuensi tiap kode rantai.

$$FCC_i^n = \frac{F_i}{\Sigma F_i} \times n \quad (2)$$

Fitur FCC ini yang digunakan SVM untuk melakukan klasifikasi.

3.5. Klasifikasi

Tahapan ini merupakan tahapan yang sangat penting karena hasil ekstraksi fitur akan diklasifikasi dan dianalisa tingkat kecocokan antara fitur data *training* dan fitur data *testing*. Pada tahapan ini dapat diukur keefektifan SVM sebagai *classifier*. SVM melakukan klasifikasi

kedalam 19 kelompok berdasarkan perbedaan bentuk bagian utama. Hasil klasifikasi SVM kemudian diterapkan klasifikasi berdasarkan aturan pengambilan keputusan yang akan menghasilkan *output* berupa aksara Jawi.

Pada penelitian ini digunakan metode *one-against-one* karena model yang dibangun lebih sedikit daripada *one-against-all*. Dengan menggunakan metode tersebut maka dibangun $19 \times (19-1)/2 = 171$ model klasifikasi biner (19 adalah jumlah kelas berdasarkan kelompok pada tabel 1). Terdapat beberapa metode untuk melakukan pengujian salah satunya adalah metode *voting*. Metode *voting* diterapkan terhadap input aksara *testing* setelah keseluruhan model klasifikasi selesai dibangun.

Hasil klasifikasi SVM kemudian diterapkan aturan-aturan dalam pengambilan keputusan agar keakuratan pengenalan aksara menjadi lebih besar. Aturan-aturan dalam pengambilan keputusan dapat dilihat pada Tabel 3.

3.6. Prosedur Pengujian

Untuk mendapatkan persentase keberhasilan maka fitur citra akan diklasifikasi oleh SVM. Hasil klasifikasi SVM merupakan 19 kelompok aksara yang dibedakan berdasarkan bentuk. Untuk mendefinisikan aksara Jawi maka dibuat aturan pengambilan keputusan berdasarkan letak titik (*P1*) dan jumlah titik (*P2*) seperti terlihat pada Tabel 3. Untuk mengukur akurasi tulisan maka akan digunakan persamaan berikut:

$$Akurasi = \frac{\text{Jumlah terklasifikasi benar}}{\text{Total jumlah data test}} \times 100\% \quad (3)$$

Tabel 3. Aturan pengambilan keputusan

No	Aksara Jawi	Aksara hasil klasifikasi	Validasi aturan pengambilan keputusan		
			Kelompok	P1	P2
1	alif	ا	Alif	4	1
2	ba	ب	BaTaTsa	2	2
3	ta	ت	BaTaTsa	1	3
4	tsha	ث	BaTaTsa	1	4
5	jim	ج	JimHhaKhaCa	3	2
6	hha	ح	JimHhaKhaCa	4	1
7	kha	خ	JimHhaKhaCa	1	2
8	dal	د	DalDzal	4	1
9	dzal	ذ	DalDzal	1	2
10	ra	ر	RaZai	4	1
11	zai	ز	RaZai	1	2
12	sin	س	SinSyin	4	1
13	syin	ش	SinSyin	1	4
14	shad	ص	ShadDhad	4	1
15	dhad	ض	ShadDhad	1	2
16	tho	ط	ThoZho	4	1
17	zho	ظ	ThoZho	3	2
18	ain	ع	AinGhainNga	4	1
19	ghain	غ	AinGhainNga	1	2
20	fa	ف	FaPa	1	2
21	qaf	ق	Qaf	1	3
22	kaf	ك	KafGa	4	1
23	lam	ل	Lam	4	1
24	mim	م	Mim	4	1
25	nun	ن	NunNya	1	2
26	wau	و	WauVa	4	1
27	ha	ه	Ha	4	1
28	hamza	ء	Hamzah	4	1
29	ya	ي	Ya	2	3
30	nya	ئ	NunNya	1	4
31	ca	چ	JimHhaKhaCa	3	4
32	nga	غ	AinGhainNga	1	4
33	pa	پ	FaPa	1	4
34	ga	گ	KafGa	1	2
35	va	ف	WauVa	1	2

Gambar 7. Metode *one-against-one* pada klasifikasi SVM

4. HASIL DAN PEMBAHASAN

4.1. Implementasi

Untuk mengimplementasikan pengenalan aksara Jawi tulisan tangan ini diperlukan beberapa komponen pendukung yang terdiri dari perangkat keras dan perangkat lunak. Perangkat lunak yang digunakan adalah *Matlab R2014b*, sedangkan perangkat keras yang digunakan adalah komputer yang mampu menjalankan aplikasi *Matlab R2014b* tersebut dengan sistem operasi *windows*.

Implementasi secara garis besar dapat dibagi kedalam 5 modul utama, yaitu:

1. Modul akuisisi citra
Modul ini merupakan proses mendapatkan citra aksara Jawi, proses ini meliputi penyiapan *form* untuk dituliskan oleh penulis sampai dengan proses segmentasi.
2. Modul *pre-processing* citra
Modul ini merupakan proses menghasilkan citra untuk fitur. Modul ini terdiri dari proses binerisasi dan *thinning* agar didapatkan skeleton dari citra yang akan dijadikan fitur.
3. Modul ekstraksi fitur
Modul ini merupakan proses ekstraksi fitur yang meliputi fitur letak titik, jumlah titik dan FCC yang digunakan oleh *classifier*.
4. Modul klasifikasi SVM
Modul ini merupakan proses klasifikasi kedalam 19 kelompok aksara yang dibedakan berdasarkan bentuk bagian utama dengan cara membangun SVM model untuk fitur *training* dan mengujinya dengan fitur *testing*.
5. Modul klasifikasi aturan pengambilan keputusan
Modul ini merupakan proses melakukan klasifikasi berdasarkan hasil klasifikasi SVM dengan menerapkan aturan letak titik dan jumlah titik.

4.2. Hasil Pengujian Fitur FCC tanpa Aturan Pengambilan Keputusan

Pada pengujian ini hasil ekstraksi fitur dari 350 aksara tidak dipisahkan kedalam 19 kelompok seperti yang terlihat pada Tabel 1. Semua fitur hasil ekstraksi tersebut disimpan dalam basis data. Fitur tersebut kemudian dibagi menjadi dua bagian yaitu sebagai fitur *training* 80% (280 fitur) dan 20% (70 fitur) untuk fitur *testing*. Untuk melakukan klasifikasi, maka model SVM yang dibangun adalah sebanyak $35 \times (35-1)/2 = 595$ model klasifikasi biner (35

adalah jumlah kelas berdasarkan jumlah aksara Jawi). Pengujian ini dilakukan untuk mengetahui kemampuan SVM untuk mengenali hasil ekstraksi fitur dengan FCC tanpa dilakukan ekstraksi letak titik dan jumlah titik. Permasalahan yang muncul pada pengujian ini adalah SVM tidak dapat membedakan aksara-aksara yang memiliki kesamaan bentuk, contohnya huruf “ب”, “ت” dan “ث” yang memiliki kesamaan bentuk bagian utama namun memiliki letak dan jumlah titik yang berbeda. Hasil lengkap pengujian dapat dilihat pada Tabel 4.

Pada pengujian panjang vektor 350 semua hasil klasifikasi SVM menunjukkan bahwa aksara “ب”, “ت” dan “ث” dikenali sebagai aksara dengan bentuk bagian “ب”, sedangkan pada panjang vektor lain beberapa aksara “ب”, “ت” dan “ث” dikenali sebagai aksara dengan bentuk bagian utama “ب”, bentuk bagian utama “ب” digunakan untuk menulis aksara “ن” dan “ئ”.

Tabel 4. Hasil pengujian klasifikasi fitur FCC tanpa aturan pengambilan keputusan

No	Panjang Vektor FCC	Keberhasilan
1	100	32,86
2	150	37,14
3	200	38,57
4	250	34,29
5	300	44,29
6	350	47,14
7	400	45,71
8	450	45,71
9	500	44,29

Dari Tabel 4 dapat dilihat bahwa pengenalan aksara Jawi paling tinggi didapat pada panjang vektor 350 yaitu sebesar 47,14%. Secara keseluruhan pengenalan aksara Jawi ini masih rendah karena FCC tidak dapat mengekstraksi fitur letak titik dan jumlah titik sehingga menyebabkan SVM tidak dapat mengklasifikasi fitur tersebut. SVM hanya menggunakan FCC dari hasil ekstraksi bentuk bagian utama aksara untuk melakukan klasifikasi kedalam aksara Jawi. Grafik hasil pengujian klasifikasi fitur FCC tanpa aturan pengambilan keputusan dengan panjang vektor fitur FCC berbeda dapat dilihat pada Gambar 8.

Gambar 8. Persentase keberhasilan hasil pengujian fitur FCC tanpa ekstraksi letak titik dan jumlah titik

4.3. Hasil Pengujian Klasifikasi SVM

Pada pengujian klasifikasi fitur FCC tanpa ekstraksi letak titik dan jumlah titik didapatkan hasil yang tidak memuaskan karena persentase pengenalan aksara yang sangat kecil. Untuk mengatasi masalah ini maka aksara dikelompokkan berdasarkan perbedaan bentuk bagian utama tanpa melihat letak dan banyaknya titik. Berdasarkan perbedaan tersebut maka aksara Jawi dapat dikelompokkan kedalam 19 kelompok seperti yang terlihat pada Tabel 1. Pada pengujian ini SVM melakukan klasifikasi fitur-fitur aksara *testing* kedalam 19 kelompok tersebut. Hasil pengujian tersebut dapat dilihat pada Tabel 5.

Tabel 5. Hasil pengujian klasifikasi SVM berdasarkan kelompok aksara

No	Panjang Vektor FCC	Keberhasilan
1	100	68,57
2	150	70,00
3	200	72,86
4	250	72,86
5	300	77,14
6	350	75,71
7	400	80,00
8	450	74,29
9	500	78,57

Dari Tabel 5 diatas dapat dilihat bahwa pengenalan aksara Jawi yang dibedakan berdasarkan perbedaan bentuk bagian utama aksara paling tinggi didapat pada panjang vektor 400 yaitu sebesar 80,00%. Terjadi peningkatan pengenalan yang sangat signifikan dari pengujian pengenalan sebelumnya, yang artinya SVM dapat melakukan klasifikasi fitur FCC berdasarkan bentuk bagian utama aksara. Pada Gambar 9 dapat dilihat grafik persentase

keberhasilan klasifikasi SVM dengan panjang vektor fitur FCC yang berbeda.

Gambar 9. Persentase keberhasilan hasil pengujian klasifikasi SVM berdasarkan kelompok aksara

4.4. Hasil Penerapan Aturan Pengambilan Keputusan

Hasil klasifikasi SVM merupakan kelompok aksara yang dibedakan berdasarkan bentuk bagian utama aksara. Agar kelompok tersebut dapat didefinisikan kedalam aksara Jawi maka diperlukan sebuah aturan pengambilan keputusan. Aturan tersebut dapat dilihat pada Tabel 3. Sebuah aksara dapat didefinisikan jika kelompok hasil klasifikasi SVM, letak titik dan jumlah titik sesuai dengan Tabel 3 tersebut. Jika kelompok hasil klasifikasi SVM, letak titik dan jumlah titik tidak ada yang sesuai maka aksara tersebut tidak dapat diklasifikasi oleh aturan pengambilan keputusan sehingga pengenalan aksara tersebut dianggap gagal. Hasil pengujian penerapan aturan pengambilan keputusan dapat dilihat pada Tabel 6. Dari tabel tersebut dapat dilihat bahwa pengenalan aksara paling baik dengan menerapkan aturan pengambilan keputusan adalah pada panjang vektor FCC 400 yaitu sebesar 80,00%.

Tabel 6. Hasil pengujian penerapan aturan pengambilan keputusan

No	Panjang Vektor FCC	Keberhasilan
1	100	68,12
2	150	70,00
3	200	72,86
4	250	72,86
5	300	77,14
6	350	75,71
7	400	80,00
8	450	74,29
9	500	78,57

Pada Gambar 10 dapat dilihat grafik persentase keberhasilan klasifikasi penerapan aturan pengambilan keputusan pengenalan aksara Jawi dengan panjang vektor fitur FCC yang berbeda.

Gambar 10. Persentase keberhasilan hasil pengujian penerapan aturan pengambilan keputusan

5. KESIMPULAN

Hasil penelitian ini menunjukkan bahwa ekstraksi fitur menggunakan FCC dapat diklasifikasi dengan baik oleh SVM serta panjang vektor fitur FCC menentukan persentase keberhasilan. Hasil lain yang didapatkan adalah penerapan aturan pengambilan keputusan berdasarkan kelompok aksara hasil klasifikasi SVM, letak titik dan jumlah titik dapat meningkatkan akurasi pengenalan aksara Jawi dengan tingkat keberhasilan mencapai 80,00%.

6. SARAN

Dari penelitian diatas masih memungkinkan untuk dilakukan eksplorasi klasifikasi SVM dengan menerapkan kernel-kernel seperti *gaussian*, *polynomial*, *sigmoid* dan beberapa kernel lainnya. Pada penelitian ini juga tidak dilakukan eksplorasi berbagai fitur lainnya seperti fitur lubang, sudut dan jarak. Penerapan fitur-fitur tersebut memungkinkan persentase pengenalan aksara Jawi menjadi lebih tinggi.

DAFTAR PUSTAKA

- [1] H. H. Musa, "Peranan tulisan Jawi dalam perkembangan Islam di Malaysia," *Jurnal pengajian Melayu*, jilid 16, 2005.
- [2] A. J. B. Borham, "Tulisan Jawi: tulisan serantau," in *Ucuptama Seminar Tulisan Jawi dan Teknologi Peringkat Kebangsaan 2012 di Dewan Astaka, Universiti Malaysia Pahang.*, Oct. 18, 2012.
- [3] R. Redika, K. Omar, and M. F. Nasrudin, "Handwritten Jawi words recognition using Hidden Markov Models," in *International Symposium on Information Technology, ITSIm 2008.*, pp 1-5, Aug. 26-28, 2008.
- [4] A. Heryanto, M. F. Nasrudin, and K. Omar, "Offline Jawi handwritten recognizer using hybrid artificial neural networks and dynamic programming," in *International Symposium on Information Technology, ITSIm 2008.*, pp 1-5, Aug. 26-28, 2008.
- [5] M. F. Nasruddin, M. Petrou, and L. Kotoulas, "Jawi character recognition using the Trace transform," in *Seventh International Conference on Computer Graphics, Imaging and Visualization.*, pp 151-156, Aug. 7-10, 2010.
- [6] R. Rahmadewi dan R. Kurnia, "Klasifikasi Penyakit Paru Berdasarkan Citra Rontgen dengan Metoda Segmentasi Sobel". *Jurnal Nasional Teknik Elektro (JNTE)*. vol: 5, no. 1, pp.7-12, Mar 2016.
- [7] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62-66, Jan. 1979.
- [8] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Comm. ACM*, vol. 27, no. 3, pp. 236-239, 1984.
- [9] L. Lam, S. W. Lee, and C. Y. Suen, "Thinning methodologies-a comprehensive survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 14, no. 9, September 1992, page 879, bottom of first column through top of second column.
- [10] H. Freeman, "Computer processing of line-drawing images," *ACM Computing Survey*, ACM New York, vol. 6, issue 1, pp.57-97, March 1974.
- [11] D. Nasien, H. Habibollah, and S. S. Yuhaniz, "Support Vector Machine (SVM) for English handwritten character recognition," in *Second International Conference on Computer Engineering and*

Applications (ICCEA)., pp 249- 252, May. 19-21, 2010.

Biodata Penulis

Safrizal, dilahirkan di Aceh Utara. Pendidikan SD sampai SMA dilalui di Aceh Utara. Lulus SMA melanjutkan kuliah dan memperoleh gelar sarjana di jurusan teknik elektro Universitas Syiah Kuala pada tahun 2006. Saat ini masih menempuh pendidikan magister teknik elektro di Universitas Syiah Kuala.

Fitri Arnia memperoleh gelar Sarjana Teknik dari Universitas Sumatera Utara (USU), Medan pada tahun 1997. Pendidikan tingkat master dan doktoral diselesaikannya pada University of New South Wales (UNSW), Sydney, Australia dan Tokyo Metropolitan University, Jepang berturut-turut pada tahun 2004 dan 2008. Beliau bekerja sebagai staf pengajar pada Jurusan Teknik Elektro, Fakultas Teknik, Universitas Syiah Kuala sejak tahun 1999. Beliau juga anggota IEEE dan IAENG. Bidang penelitiannya meliputi pengolahan sinyal, citra dan multimedia digital.

Rusdha Muharar memperoleh gelar Sarjana Teknik dari Universitas Gadjah Mada (UGM) Yogyakarta pada tahun 1999. Pendidikan tingkat master dan doktoral diselesaikannya pada Delft University Of Technology, Belanda dan University Of Melbourne, Australia berturut-turut pada tahun 2004 dan 2013. Beliau bekerja sebagai staf pengajar pada Jurusan Teknik Elektro, Fakultas Teknik, Universitas Syiah Kuala.