

# Jumlah Transisi pada Ciri Transisi dalam Pengenalan Pola Tulisan Tangan Aksara Jawa *Nglegeno* dengan Multiclass Support Vector Machines

## Numbers of Transition Features on Basic Jawanese (*Nglegeno*) Characters Recognition System with Multiclass Support Vector Machines

Azis Wisnu Widhi Nugraha<sup>#1</sup>, Widhiatmoko H. P.<sup>\*2</sup>

asw\_te@yahoo.com

arieswhp@yahoo.com

Prodi Teknik Elektro Jurusan Teknik FST Unsoed

**Abstract**— Feature extraction is one of the most important step on characters recognition system. Transition features is one from many features used on characters recognition system. This paper report a research on handwritten basic Jawanese characters recognition system to found the proper numbers of transitions used on transition features. To recognize the characters, the Multiclass Support Vector Machines were used. The Directed Acyclic Graph (DAG) SVM were used for multiclass classification strategy and to map each input vector to a higher dimension space, the Gaussian Radial Basis Function (RBF) kernel with parameter 1 were used. It can be shown, for basic Jawanese characters recognition system, the optimal numbers of transitions used for transition features is 4 (a half of maximum numbers of transition on all patterns).

**Keywords**— Handwritten characters recognition, Jawanese characters, Transition Features, Multiclass SVM, DAG SVM.

### PENDAHULUAN

Komputerisasi aksara Jawa dapat digunakan sebagai media pembelajaran dan melestarikan keberadaan aksara Jawa yang sudah mulai jarang digunakan (Sayogo, 2006a, 2006b). Penelitian mengenai pengenalan aksara Jawa hingga saat ini masih jarang dilakukan, diantaranya adalah Widiarti (2006), Asriani, dkk (2006) dan Nugraha, A.W.W. (2009). Widiarti (2006) melakukan penelitian dengan obyek berupa aksara Jawa pada dokumen tercetak. Sedangkan Asriani, dkk (2006) dan Nugraha, A.W.W. (2009) melakukan penelitian dengan obyek tulisan tangan aksara Jawa *nlegeno* terisolasi.

Nugraha, A.W.W. (2009) melakukan pengenalan terhadap tulisan tangan aksara Jawa *nlegeno* terisolasi dengan menggunakan ciri batang (Gader et al., 1995b) dan ciri transisi (Gadel et al., 1997) serta menggunakan *Multiclass Support Vector Machines* (SVM) sebagai pengelompoknya. Nugraha, A.W.W. (2009) menyatakan ciri transisi lebih dipilih karena waktu komputasi yang jauh lebih singkat dibandingkan dengan ciri batang dengan tingkat keberhasilan yang hampir dapat dikatakan sama. Nugraha, A.W.W. (2009) menggunakan 5 transisi pertama untuk mendapatkan vektor ciri. Pemilihan penggunaan 5 transisi ini tidak didasarkan pada nilai yang paling optimal. Untuk itu

diperlukan penelitian untuk mengetahui berapa jumlah transisi yang paling optimal untuk digunakan dalam pengenalan pola tulisan tangan aksara Jawa *nlegeno* dengan menggunakan *multiclass* SVM.

### AKSARA JAWA

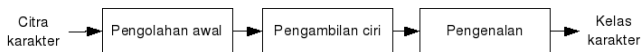
Aksara Jawa terdiri atas aksara dasar yang disebut dengan aksara *carakan* atau *nlegeno* atau aksara Jawa tanpa sandangan (terlihat pada Tabel 1), aksara *pasangan*, aksara *swara*, aksara *rekan*, aksara *murda*, *wilangan/angka*, dan *sandangan/tanda baca* (Sayogo, 2004). Penggunaan aksara Jawa pada penyusunan kata dan kalimat membutuhkan penggabungan aksara dasar dan aksara lainnya.

TABEL 1  
AKSARA JAWA NGLEGENO

Huruf	Lafat	Huruf	Lafal	Huruf	Lafal	Huruf	Lafal	Huruf	Lafal
ꦲ	Ha	ꦤ	Na	ꦕ	Ca	ꦫ	Ra	ꦏ	Ka
ꦢ	Da	ꦠ	Ta	ꦱ	Sa	ꦮ	Wa	ꦭ	La
ꦥ	Pa	ꦨ	Dha	ꦗ	Ja	ꦪ	Ya	ꦤ	Nya
ꦩ	Ma	ꦒ	Ga	ꦨ	Bha	ꦠ	Tha	ꦤ	Nga

### SISTEM PENGENALAN POLA TULISAN TANGAN

Model sistem pengenalan pola pada umumnya terdiri atas pengolahan awal, pengambilan ciri dan pengenalan karakter (Widiarti, 2006; Asriani dkk, 2006; Asriani dkk, 2007; dan Asriani dan Azis Wisnu W.N., 2008). Diagram blok dari model sistem pengenalan pola tulisan tangan dapat digambarkan secara umum seperti pada Gambar 1.



Gambar 1 Model sistem pengenalan pola tulisan tangan.

Pengolahan awal dimaksudkan untuk melakukan perbaikan citra masukan agar dapat dikenali citranya dengan baik. Pengolahan awal ini dapat juga melibatkan segmentasi karakter untuk memisahkan karakter yang akan dikenali dengan karakter maupun obyek yang lain. Pada bagian ini citra digital biasanya juga diubah menjadi citra biner untuk meringankan dan menyederhanakan proses perhitungan berikutnya. Derau yang ada dihilangkan, cacat-cacat yang ada dihilangkan. Selain itu jika diperlukan juga dilakukan normalisasi terhadap karakter baik normalisasi orientasi maupun ukuran.

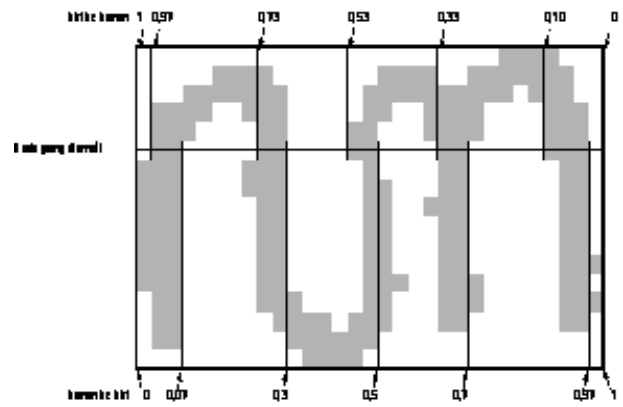
Pengambilan ciri karakter dimaksudkan untuk menciptakan vektor masukan bagi sistem pengenalan karakter. Ciri yang digunakan sebagai vektor masukan ini diharapkan dapat membedakan antar karakter yang akan dikenali. Ciri-ciri yang digunakan sebagai contoh berupa jumlah piksel obyek pada suatu wilayah (Widiarti, 2006), ciri batang (Gader *et al.*, 1995a; Gader *et al.*, 1995b; Mudjirahardjo, 2001; Asriani dkk, 2006, 2007; Asriani dan Azis Wisnu W.N., 2008; Nugraha, A.W.W., 2009), dan ciri transisi (Gader *et al.*, 1995a; Gader *et al.*, 1997; Mudjirahardjo, 2001; Nugraha, A.W.W., 2009). Untuk mendapatkan hasil pengenalan yang baik, dapat juga digunakan lebih dari satu ciri (Gader *et al.*, 1995a dan Gader *et al.*, 1997; Nugraha, A.W.W., 2009).

Pengenalan dapat dilakukan dengan berbagai macam metode klasifikasi yang telah banyak dikenal, seperti jaringan syaraf tiruan (Asriani dkk, 2006; Asriani dan Azis Wisnu W.N., 2008; Mudjirahardjo, 2001, Gader *et al.*, 1995b, 1997), jarak *euclidian* (Widiarti, 2006), logika kabur (Gader *et al.* 1995a), ANFIS (Asriani dkk, 2007) dan SVM sebagaimana yang disurvei oleh Oliveira dan Sabourin (2004), serta Byun dan Lee (2003) maupun yang telah dilakukan oleh Nugraha, A.W.W. (2009) untuk mengenali tulisan tangan aksara Jawa *nglegeno*.

#### CIRI TRANSISI

Salah satu ciri yang digunakan untuk mengenali aksara adalah ciri transisi. Menurut Gaderl *et al.* (1997) ciri transisi merupakan informasi lokasi transisi latar belakang ke obyek pada citra. Dicari pada empat arah,

kiri ke kanan, kanan ke kiri, atas ke bawah dan bawah ke atas. Transisi dicari pada setiap baris (untuk transisi kiri ke kanan dan kanan ke kiri) serta pada setiap kolom (untuk transisi atas ke bawah dan bawah ke atas). Pada setiap garis pencarian transisi, lokasi transisi dihitung menurun. Ilustrasi lokasi transisi pada arah kanan ke kiri dan kiri ke kanan dapat dilihat pada Gambar 2



Gambar 2 Ilustrasi perhitungan ciri transisi pada baris horizontal.

Pada setiap garis pencarian hanya akan digunakan  $M$  transisi pertama jika terdapat lebih dari  $M$  transisi. Jika pada garis terdapat kurang dari  $M$  transisi, maka hingga nilai ke  $M$  diisi dengan angka '0'. Jika ukuran citra adalah  $m \times n$ , dengan  $m$  adalah jumlah baris dan  $n$  adalah jumlah kolom, maka pada arah kiri ke kanan akan diperoleh matriks transisi dengan ukuran  $m \times M$  (demikian juga untuk arah kanan ke kiri), sedangkan untuk arah atas ke bawah akan diperoleh matriks transisi dengan ukuran  $n \times M$  (demikian juga untuk arah bawah ke atas). Selanjutnya setiap matriks transisi pada setiap arah di ubah menjadi matriks berukuran  $M \times M$  dengan membagi  $m$  baris dan  $n$  kolom menjadi  $M$  bagian dan meratakan masing-masing nilai transisi pada tiap bagainnya. Kemudian keempat matriks transisi dari tiap arah disusun menjadi vektor dan digabungkan menjadi satu. Sehingga ukuran vektor transisi menjadi  $4 \times M \times M$ .

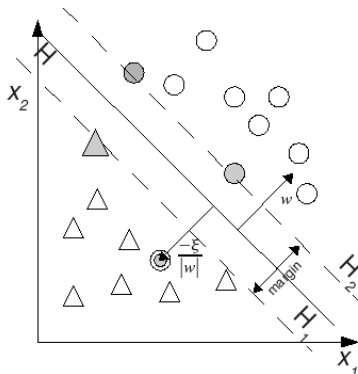
#### SUPPORT VECTOR MACHINES (SVM)

*Support Vector Machines* merupakan salah satu teknik untuk melakukan klasifikasi. Pada dasarnya SVM adalah sebuah *linear classifier*. Dengan menggunakan teknik kernel SVM mampu menyelesaikan permasalahan yang tidak linear. (Burges, 1998; Byun dan Lee, 2003; Cortes dan Vapnik, 1995; Fletcher, 2008; Fradkin dan Muchnik, 2006; Nugroho dkk, 2003; Oliveira dan Sabourin, 2004; dan Vijayakumar dan Wu, 1999). Penggunaan kernel ini pada dasarnya memetakan vektor masukan pada ruang berdimensi rendah ke ruang ciri berdimensi lebih tinggi.

Selain merupakan *linear classifier*, SVM juga merupakan *binary classifier*. Untuk menyelesaikan permasalahan *multiclass*, SVM dapat dengan mudah dikombinasikan (Cortes dan Vapnik, 1995; Burges, 1998; Byun dan Lee, 2003; Oliveira dan Sabourin, 2004; dan Fradkin dan Muchnik, 2006). Dalam menyelesaikan permasalahan *multiclass* ada dua strategi yang dapat digunakan, satu lawan yang lain (*One-vs-Rest*) dan pohon struktur (*pairwise SVM* dan *DAG SVM*) (Byun dan Lee, 2003; dan Oliveira dan Sabourin, 2004).

Menurut Nugroho dkk (2003) SVM memiliki kelebihan dalam hal kemampuan generalisasi yang lebih baik, kemampuan memecahkan masalah berdimensi tinggi dengan keterbatasan sampel, memiliki landasan teori yang dapat dianalisa secara jelas, serta relatif mudah untuk diimplementasikan dengan sudah banyak tersedianya pustaka *quadratic programming* (QP) *solver*.

Pengelompokan data dalam dua kelas dapat dinotasikan dengan  $\{x_i, y_i\}$ , dengan  $x_i$  adalah vektor data dengan N dimensi dan  $y_i \in \{-1, 1\}$  adalah target. Diantara kedua kelas data dapat dibuat sebuah bidang pemisah *hyperplane*  $H$  sebagaimana ditunjukkan pada Gambar 3.  $H_1$  dan  $H_2$  adalah bidang sejajar  $H$  yang pada keduanya terdapat vektor-vektor pola terluar dari masing-masing kelas yang disebut dengan *Support Vector* (SV). Algoritma SVM pada dasarnya berusaha mencari jarak terbesar antara  $H_1$  dan  $H_2$ .



Gambar 3 *Hyperplane* pemisah dua kelas.

Dari gambar,  $H$  adalah *hyperplane* optimal. Jika  $w$  adalah vektor bobot,  $x$  adalah vektor pola, dan  $b$  adalah bias, maka persamaan *hyperplane*  $H$  adalah sebagai berikut.

$$w \cdot x + b = 0 \tag{1}$$

Jika dari data pelatihan dibuat matriks  $H$  dengan komponen penyusun sebagai berikut.

$$H_{ij} = y_i y_j x_i \cdot x_j \tag{2}$$

Maka SVM akan mencari nilai optimasi persoalan *quadratic programming* berikut.

$$\max \left[ \sum_{i=1}^l \alpha_i - \frac{1}{2} \vec{\alpha}^T H \vec{\alpha} \right] \text{ s.t. } 0 \leq \alpha_i \leq C ; \sum_{i=1}^l \alpha_i y_i = 0 \tag{3}$$

dengan  $\alpha_i$  adalah pengali *Lagrange* untuk data pelatihan ke- $i$ , dan  $c$  adalah *variable slack*.

Semua vektor pola yang memiliki nilai pengali *Lagrange*  $0 \leq \alpha_i \leq C$  merupakan *Support Vector*. Dengan telah diperolehnya semua *support vector* beserta pengali *Lagrange*-nya, maka nilai bobot  $w$  dan bias  $b$  pada persamaan (1) dapat dicari dengan persamaan berikut (dengan  $N_s$  adalah jumlah *support vector* dan  $s$  adalah himpunan semua *support vector*).

$$w = \sum_{i=1}^l \alpha_i y_i x_i \tag{4}$$

$$b = \frac{1}{N_s} \sum_{s \in S} \left( y_s - \sum_{m \in S} \alpha_m y_m x_m \cdot x_s \right) \tag{5}$$

Untuk melakukan klasifikasi terhadap pola baru, maka fungsi klasifikasi yang digunakan adalah sebagai berikut.

$$y_i = \text{sign}(x_i \cdot w + b)$$

$$y_i = \text{sign} \left( \sum_{m \in S} \alpha_m y_m x_m \cdot x_i + b \right) \tag{6}$$

$$y_i = \text{sign} \left( \sum_{m \in S} \alpha_m y_m x_m \cdot x_i + \frac{1}{N_s} \sum_{s \in S} \left( y_s - \sum_{m \in S} \alpha_m y_m x_m \cdot x_s \right) \right)$$

Untuk menyelesaikan permasalahan yang tidak linear, maka digunakan fungsi *kernel*. Yang pada dasarnya memetakan vektor pola berdimensi  $N$  ke dimensi yang jauh lebih tinggi. Pada persamaan (2), (5), dan (6) terlihat vektor pola masukan  $x$  selalu muncul bersama dengan vektor pola yang lain dan dikenai operasi *dot product*. Hal ini memungkinkan untuk mengganti *dot product*  $x_i \cdot x_j$  kedua vektor pola tersebut dengan menggunakan sebuah fungsi kernel  $K(x_i, x_j)$ . Adapun fungsi *kernel* yang sering digunakan pada SVM tampak seperti pada Tabel 2.

TABEL 2  
FUNGSI KERNEL YANG SERING DIGUNAKAN DALAM SVM

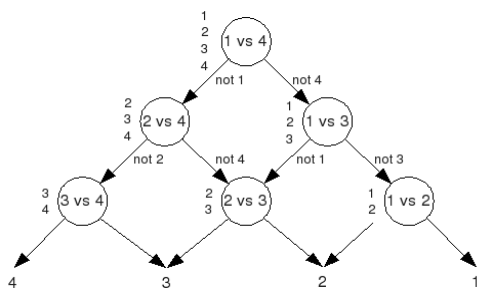
Nama Kernel	$K(x, x_i), i = 1, 2, \dots, N$
Linear	$K(x, x_i) = (x^T x_i)$
Polinomial	$K(x, x_i) = (x^T x_i + 1)^d$
Gaussian RBF	$K(x, x_i) = \exp\left(-\frac{\ x - x_i\ ^2}{2\sigma^2}\right)$
Sigmoid	$K(x, x_i) = \tanh(\beta_0 x^T x_i - \beta_1)$

SVM pada dasarnya adalah pengklasifikasi biner. Untuk melakukan klasifikasi kelas jamak SVM dapat digabungkan (Byun dan Lee, 2003; dan Oliveira dan Sbourin, 2004). Ada dua strategi utama memecahkan permasalahan  $N$  kelas dengan SVM, *one-to-others* dan struktur pohon (*pairwise SVMs* dan *Decision Directed Acyclic Graph / Decision DAG*) (Byun dan Lee, 2003).

Pada strategi *One-vs-Rest*, SVM dilatih dengan memberikan label +1 pada salah satu kelas dan label -1 pada kelas yang lainnya, sehingga akan didapatkan permasalahan dua kelas. Untuk permasalahan  $N$  kelas ( $N > 2$ ), akan terdapat  $N$  SVM yang dinyatakan dengan  $SVM_i, i = 1, 2, \dots, N$ . Untuk melakukan klasifikasi terhadap vektor masukan  $x$ , maka akan terdapat  $N$  keluaran  $d_i(x) = w_i \cdot x + b_i; i = 1, 2, \dots, N$ . Vektor masukan  $x$  termasuk dalam kelas  $j$  jika:

$$d_j(x) = \max_{i=1, 2, \dots, N} d_i(x) \quad (7)$$

Pada pendekatan struktur pohon, SVM dilatih dengan hanya membandingkan dua kelas saja, sehingga untuk permasalahan  $N$  kelas akan terdapat  $N(N-1)/2$  SVM. Pengelompok SVM disusun dalam bentuk pohon dengan masing-masing titik pada pohon menyatakan sebuah SVM. Pohon dari bawah ke atas yang mirip dengan pohon eliminasi yang digunakan dalam turnamen tenis diusulkan oleh Pontil dan Verri untuk mengenali obyek 3D dan diterapkan pada pengenalan wajah oleh Guo *et al.* (Byun dan Lee 2003).



Gambar 4 Struktur DAG SVM untuk permasalahan 4 kelas.

Platt *et al.* (2000), mengusulkan struktur pohon dari atas ke bawah yang disebut dengan *Decision Directed Acyclic Graph (Decision DAG)* yang kemudian disebut dengan DAG SVM. Ilustrasi penggunaan DAG SVM untuk menyelesaikan permasalahan 4 kelas dapat dilihat pada **Error! Reference source not found.**

Berdasarkan pada jumlah SVM yang harus dibangun selama proses pelatihan, maka strategi *One-vs-Rest* lebih disukai karena hanya dibutuhkan  $N$  SVM untuk permasalahan  $N$  kelas. Namun berdasarkan penelitian yang dilakukan oleh Hsu dan Lin dalam Byun dan Lee (2003) diindikasikan metode struktur pohon lebih cocok untuk keperluan praktis dibandingkan dengan metode *One-vs-Rest*.

Nugraha, A.W.W (2009) menyatakan untuk pengenalan pola tulisan tangan aksara Jawa, penggunaan strategi *One-vs-Rest* memerlukan waktu pelatihan yang jauh lebih lama (antara 61 sampai dengan 81 kali lebih lama) dibandingkan dengan strategi DAG-SVM. Sedangkan waktu klasifikasi strategi *One-vs-Rest* lebih cepat (sekitar 2 kali lebih cepat) dibandingkan dengan strategi DAG-SVM. Dengan demikian jika waktu pelatihan yang singkat sangat penting, maka penggunaan DAG-SVM lebih baik digunakan, namun jika waktu pelatihan tidak menjadi pertimbangan yang penting, maka strategi *One-vs-Rest* akan menjadi lebih baik.

## METODE PENELITIAN

### A. Materi penelitian

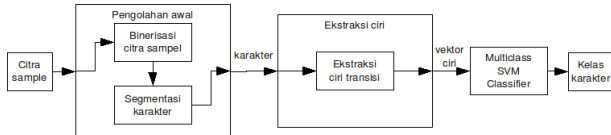
Materi penelitian adalah tulisan tangan aksara Jawa *nglegeno* yang dikumpulkan dari 16 responden. Setiap responden diminta untuk menuliskan aksara Jawa *nglegeno* sebanyak 10 set. Dengan demikian akan diperoleh 160 set tulisan aksara Jawa *nglegeno* (3200 karakter).

### B. Alat penelitian

1. Pemayar digital.
2. Seperangkat *notebook* untuk melakukan komputasi dengan prosesor Intel Core 2 Duo T5500 dan memori 2 GB.
3. Perangkat lunak *Octave* (Eaton, 2002; www.octave.org) sebagai perangkat lunak untuk melakukan perhitungan.

### C. Jalan penelitian

*Pengumpulan contoh tulisan tangan aksara Jawa nglegeno.* Dikumpulkan tulisan tangan aksara Jawa *nglegeno* dari responden. Tiap responden diminta untuk menuliskan 10 set aksara Jawa *nglegeno*. Akan diperoleh 200 contoh karakter tulisan tangan aksara Jawa dari setiap responden. Selanjutnya setiap lembar contoh akan di bayar dengan menggunakan pemayar digital untuk mendapatkan citra digitalnya.



Gambar 5 Diagram blok sistem pengenalan tulisan tangan aksara Jawa nglegeno.

Pengembangan sistem pengenalan pola tulisan tangan aksara Jawa nglegeno dengan menggunakan multiclass SVM. Sistem pengenalan pola tulisan tangan aksara Jawa nglegeno yang dibangun memiliki diagram blok seperti tampak pada Gambar 5 Ekstraksi ciri yang digunakan adalah ciri transisi, sedangkan multiclass SVM yang digunakan adalah teknik DAG SVM, sedangkan kernel yang digunakan adalah kernel RBF dengan parameter 1.

Pengujian sistem dengan melakukan variasi jumlah transisi yang digunakan pada vektor ciri. Sistem pengenalan pola tulisan tangan aksara Jawa nglegeno pada Gambar 5 di atas diuji dengan menggunakan variasi jumlah transisi yang digunakan (dari 3 sampai dengan 10). Dicari tingkat keberhasilan yang paling tinggi untuk menentukan jumlah transisi yang paling optimal yang dapat digunakan. Pemilihan variasi transisi dari 3 sampai dengan 10 ini didasari pada kenyataan bahwa aksara Jawa memiliki garis tegak tersedikit adalah 3 pada aksara 'ra' (ᮊ) dan terbanyak adalah 7 pada aksara 'nya' (ᮊᮒ). Variasi hingga 10 transisi diharapkan dapat mengakomodasi variasi penulisan. Adapun langkah-langkah pengujian sistem adalah sebagai berikut.

1. Bagi karakter dalam kelompok pelatihan dan kelompok pengujian.
2. Cari vektor ciri transisi setiap karakter untuk setiap variasi jumlah transisi yang digunakan (3 hingga 10 transisi).
3. Latih model multiclass SVM untuk setiap variasi jumlah transisi dengan kelompok pelatihan.
4. Untuk setiap model hitung tingkat keberhasilan pengenalan pada kelompok pengujian.
5. Cari variasi jumlah transisi yang memberikan tingkat keberhasilan pengenalan pada kelompok pengujian yang tertinggi. Jumlah transisi tersebut merupakan jumlah transisi yang paling optimal.

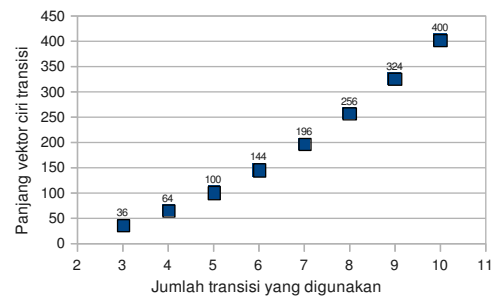
Keseluruhan sistem yang digunakan pada penelitian ini dibangun pada perangkat lunak Octave (www.octave.org).

HASIL DAN PEMBAHASAN

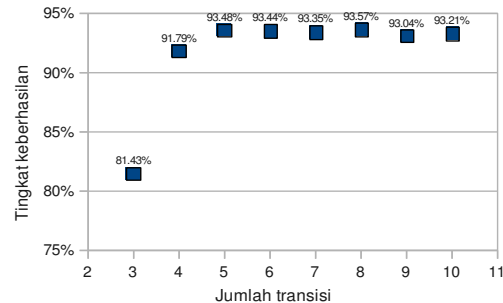
Panjang vektor ciri transisi adalah  $4 \times M \times M$ , dengan  $M$  adalah jumlah transisi. Variasi panjang vektor ciri

pada penelitian ini dapat dilihat pada Gambar 6. Terlihat panjang vektor ciri berubah secara kuadratis terhadap jumlah transisi yang digunakan. Sedangkan tingkat keberhasilan pengenalan untuk setiap variasi jumlah transisi dapat dilihat pada Gambar 7.

Dari Gambar 7 terlihat bahwa pengenalan tertinggi (93,57%) diperoleh pada jumlah transisi 8. Tingkat keberhasilan diatas 90% telah dapat dicapai dengan menggunakan ciri transisi sebanyak 4 (91,79%). Rentang keberhasilan pengenalan dengan transisi antara 4 dan 10 adalah antara 91,79% (4 transisi) hingga 93,57% (8 transisi). Selisih tingkat keberhasilan pengenalan pada transisi diantara 4 hingga 10 terlihat tidak terlalu signifikan. Berbeda dengan penggunaan 3 transisi yang hanya memberikan tingkat keberhasilan pengenalan sebesar 81,43%.



Gambar 6 Variasi panjang vektor ciri terhadap jumlah transisi yang digunakan.



Gambar 7 Tingkat keberhasilan pengenalan dengan variasi jumlah transisi.

Dari Gambar 6 panjang komponen vektor ciri pada penggunaan 4 transisi adalah 64 komponen, sedangkan pada penggunaan 8 transisi adalah 256 komponen. Dengan perbedaan tingkat keberhasilan pengenalan yang tidak terlalu signifikan (91,79% dan 93,57% atau hanya 1,78%) penggunaan vektor ciri yang lebih pendek lebih dipilih. Hal ini disebabkan karena penggunaan vektor ciri yang lebih pendek akan mengurangi beban komputasi dari sistem yang dibangun. Perbedaan yang tidak signifikan ini juga menunjukkan bahwa terdapat

*redundancy* informasi pada vektor ciri dengan transisi di atas 4.

Ciri transisi mengandung informasi mengenai transisi latar belakang ke obyek pada citra pada arah kiri ke kanan dan kanan ke kiri (horisontal) serta arah atas ke bawah dan bawah ke atas (vertikal). Pada arah horisontal, informasi yang dikandung pada arah kiri ke kanan dan kanan ke kiri, sesungguhnya sama, hanya berbeda arah pandangnya saja. Demikian pula untuk arah vertikal.

Transisi horisontal maksimal pada aksara Jawa adalah 7 (terdapat pada aksara NYA). Dari Gambar 7 terlihat bahwa penggunaan 4 transisi telah memberikan tingkat keberhasilan pengenalan di atas 90%. 4 transisi telah menghitung setengah dari jumlah transisi maksimal (7 transisi). Dengan menggabungkan informasi 4 transisi dari arah kiri ke kanan dan kanan ke kiri pada ciri transisi, maka seluruh informasi transisi (7 transisi) pada sebuah garis dapat diperoleh. Dengan menggunakan lebih dari 4 transisi, maka akan terdapat *redundancy* informasi yang diperoleh. Sehingga akan mengakibatkan vektor ciri menjadi tidak optimal.

Dengan demikian dapat disimpulkan jumlah ciri transisi optimal yang dapat digunakan dalam pengenalan pola aksara Jawa *nglegeno* adalah  $N/2$ , dengan  $N$  adalah jumlah transisi maksimal yang terdapat pada aksara Jawa *nglegeno*.

#### KESIMPULAN DAN SARAN

##### A. Kesimpulan

1. Hasil pengenalan aksara Jawa *nglegeno* terbaik diberikan oleh vektor ciri transisi dengan menggunakan 8 transisi sebesar 93,57%.
2. Hasil pengenalan aksara Jawa *nglegeno* menggunakan vektor ciri transisi dengan 4 transisi telah memberikan tingkat pengenalan di atas 90% (91,79%).
3. Perbedaan tingkat pengenalan dengan jumlah transisi yang digunakan antara 4 sampai dengan 10 yang tidak terlalu signifikan menunjukkan adanya *redundancy* informasi pada vektor ciri yang lebih panjang. Dengan demikian penggunaan 4 transisi untuk menyusun vektor ciri pada pengenalan pola tulisan tangan aksara Jawa *nglegeno* dapat dikatakan telah cukup optimal.
4. Empat transisi merupakan  $\frac{1}{2}$  dari transisi maksimal yang terdapat pada aksara Jawa *nglegeno*. Dengan demikian secara umum dapat dituliskan bahwa ciri transisi optimal yang dapat digunakan dalam pengenalan pola tulisan tangan aksara Jawa *nglegeno* adalah sebagai berikut:  
 $M_{optimal} = M_{maksimal} / 2$ , dengan  $M_{optimal}$  adalah jumlah transisi optimal yang dapat digunakan dan  $M_{maksimal}$  adalah jumlah transisi maksimal yang ada pada semua pola.

##### B. Saran

1. Penelitian ini hanya dilakukan pada tulisan tangan aksara Jawa *nglegeno*, sehingga perlu dilakukan penelitian lanjutan dengan obyek aksara Jawa yang lain.
2. Ciri transisi yang digunakan pada penelitian ini menggunakan jumlah transisi yang sama pada arah horisontal dan vertikal. Berdasarkan hasil di atas, ditunjukkan adanya *redundancy* pada arah horisontal. Hal ini memungkinkan munculnya *redundancy* juga pada arah vertikal. Perlu dilakukan penelitian lanjutan untuk mengetahui efek penggunaan jumlah transisi yang berbeda pada arah horisontal dan vertikal.

#### DAFTAR PUSTAKA

- Asriani, F., Azis Wisnu Widhi Nugraha, dan Elok Nuraeni, 2006, *Pengenalan Pola Aksara Jawa Dengan Jaringan Syaraf Tiruan Perambatan-Balik Sebagai Dasar Sistem Pengenalan Dokumen dengan Huruf Jawa*, Laporan Penelitian DIPA II tahun 2006, Program Studi Teknik Elektro, Universitas Jenderal Soedirman, Purwokerto.
- Asriani, F., Hesti Susilawati, dan Azis Wisnu Widhi Nugraha, 2007, *Perancangan Adaptif Neuro Fuzzy Inference System untuk Pengenalan Pola Tulisan Tangan*, Laporan Penelitian Dosen Muda tahun 2007, Program Studi Teknik Elektro, Program Sarjana Teknik, Universitas Jenderal Soedirman, Purwokerto.
- Asriani, F. dan Azis Wisnu Widhi Nugraha, 2008, *Pengenalan Otomatis Nomer Rekening Tulisan Tangan Pada Slip Setoran Bank dengan Jaringan Syaraf Tiruan*, 2008, Laporan Penelitian Dosen Muda tahun 2008, Program Studi Teknik Elektro, Fakultas Sains dan Teknik, Universitas Jenderal Soedirman, Purwokerto.
- Burges, Christopher J.C., 1998, *A Tutorial on Support Vector Machines for Pattern Recognition*, Data Mining and Knowledge Discovery, 2, pp. 121 – 167.
- Byun, H. and Lee, S.W., 2003, *A Survey on Pattern Recognition Applications of Support Vector Machines*, International Journal of Pattern Recognition and Artificial Intelligence, Vol 17, No 3, pp. 459 – 486.
- Cortes, C. and Vapnik, V., 1995, *Support-Vector Networks*, Machine Learning, Vol 20, No 3, pp 273 – 297.
- Fletcher, T., 2008, *Support Vector Machines Explained*, available: <http://www.tristanfletcher.co.uk/SVM%20Explained.pdf>, [2009, 18 Februari].
- Fradkin, D. and Muchnik, I., 2006, *Support Vector Machines for Classification*, in Abello, J. and Carmode, G. (Eds), *Discrete Methods in Epidemiology*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 70, pp. 13-20, available: <http://paul.rutgers.edu/~dfradkin/papers/svm.pdf>, [2009, 18 Februari].
- Gader, P.D., Keller, J.M., and Cai, J., 1995a, *A Fuzzy Logic System for the Detection and Recognition of Handwritten Street Number*, IEEE Transactions on Fuzzy Systems, Vol. 3. No. 1, pp. 83 – 94.
- Gader, P.D., Mohammed, M., and Chiang, J.H., 1995b, *Comparison of Crisp and Fuzzy Character Neural Networks in Handwritten Word Recognition*, IEEE Transactions on Fuzzy Systems, Vol 3, No. 3, pp. 357 - 363.
- Gader, P.D., Mohammed, M., and Chiang, J.H., 1997, *Handwritten Word Recognition With Character and Inter-Character Neural Networks*, IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics, Vol 27, No 1, pp. 158 – 164.
- Mudjirahardjo, P., 2001, *Penerapan Jaringan Perambatan-Balik untuk Pengenalan Kode Pos Tulisan Tangan*, Tesis S-2 Program Studi Teknik Elektro Jurusan Ilmu-Ilmu Teknik Program Pascasarjana UGM, Yogyakarta.

- Nugraha, A.W.W., 2009, *Pengenalan Pola Tulisan Tangan Aksara Jawa Nglegeno dengan Multiclass SVM*, Tesis S-2 Teknik Elektro Program Pascasarjana UGM, Yogyakarta.
- Nugroho, A.S., Witarto, A.B., dan Handoko, D., 2003, *Support Vector Machine – Teori dan Aplikasinya dalam Bioinformatika*, Kuliah Umum IlmuKomputer.com, available: **Error! Hyperlink reference not valid.**
- Oliveira, L.S. and Sabourin R., 2004, *Support Vector Machines for Handwritten Numerical String Recognition*, Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition (IWFHR-9), available: <http://www.livia.etsmtl.ca/people/soares/download/iwfhr-svm.pdf>, [2009, 25 Februari].
- Platt, J., Cristianini, N., and Shawe-Taylor, J., 2000, *Large Margin DAGs for Multiclass Classification*, Advances in Neural Information Processing Systems, 12, pp. 547 – 553.
- Sayogo, T. B., 2004, *Panduan Pengetikan Hanacaraka Font Untuk Pengetikan Aksara Jawa Pada Perangkat Lunak Komputer Berbasis Sistem Operasi Windows*, available: <http://hanacaraka.fateback.com>, [2006, 26 September].
- Sayogo, T. B., 2006a, *Komputerisasi Aksara Jawa Sebagai Media Pembelajaran Membaca dan Menulis Aksara Jawa*, disusun sebagai sumbangan pemikiran untuk peserta Konggres Bahasa Jawa IV Semarang 5-10 Juli 2006, Purwokerto, available: <http://hanacaraka.fateback.com>, [2006, 26 September].
- Sayogo, T. B., 2006b, *Registrasi Unicode Aksara Jawa Kepentingan dan Keuntungannya : Sebuah wacana untuk persiapan pengajuan proposal resmi registrasi Unicode Aksara Jawa*, disusun sebagai sumbangan pemikiran untuk peserta Konggres Bahasa Jawa IV Semarang 5-10 Juli 2006, Purwokerto, available: <http://hanacaraka.fateback.com>, [2006, 26 September].
- Vijayakumar, S. and Wu, S., 1999, *Sequential Support Vector Classifiers and Regression*, Proc. International Conference on Soft Computing (SOCO '99), Genoa, Italy, pp. 610 – 619.
- Widiarti, A.R., 2006, *Pengenalan Citra Dokumen Sastra Jawa Konsep dan Implementasinya*, Tesis S-2 Ilmu Komputer Program Pascasarjana UGM, Yogyakarta.