

Pengenalan Aksara Jawa dengan Menggunakan Metode *Area Based Feature Extraction* dan *Support Vector Machine*

Alfred Louis¹, Rudy Adipranata², Alvin Nathaniel Tjondrowiguno³
Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra
Jl. Siwalankerto 121-131, Surabaya 60236
Telp (031) – 2983455, Fax. (031) - 8417658
alfredlouisw8@gmail.com¹, rudya@petra.ac.id², Alvin.tjondrowiguno@gmail.com³

ABSTRAK

Huruf Jawa merupakan salah satu dari budaya Indonesia yang beragam di pulau Jawa. Namun seiring dengan perkembangan teknologi, budaya huruf Jawa ini mulai ditinggalkan. Banyak generasi muda sekarang yang lebih tertarik untuk belajar bahasa asing daripada Bahasa Jawa. Walaupun orang Jawa tinggal di pulau Jawa tetapi masih banyak yang tidak mengerti huruf Jawa. Sangat disayangkan apabila suatu budaya yaitu huruf Jawa harus hilang karena tidak ada generasi muda yang mengerti huruf Jawa. Huruf Jawa memiliki tingkat kesulitan tersendiri untuk dipelajari karena bentuknya yang mirip antara satu huruf dengan huruf lainnya. Kesulitan dalam mempelajari huruf Jawa ini juga berpengaruh pada minat generasi muda untuk mempelajari huruf Jawa. Selain itu, saat ini aplikasi mengenai huruf Jawa juga masih cenderung tidak terlalu banyak.

Berdasarkan permasalahan tersebut, maka aplikasi pengenalan huruf Jawa ini dibuat dengan harapan dapat melestarikan budaya Indonesia sekaligus mempermudah pembelajaran huruf Jawa. Adapun penelitian sebelumnya yang juga merupakan pengenalan huruf Jawa tetapi dengan metode lain seperti *Shape Based Feature Extraction* sebagai metode ekstraksi fitur dan *Probabilistic Neural Network, Elman Type, Feedforward Backpropagation* sebagai metode pengenalan namun akurasi masih relatif rendah. Proses dari pengenalan huruf Jawa ada 3 tahap, yaitu segmentasi, ekstraksi fitur, dan pengenalan. *Input* berupa gambar dokumen huruf Jawa yang akan di segmentasi. Proses segmentasi telah dibuat oleh penelitian sebelumnya dan hasilnya adalah gambar satu huruf Jawa. Proses fitur ekstraksi dilakukan dengan menggunakan metode *Area Based Feature Extraction* sehingga menghasilkan data fitur ekstraksi dari gambar satu huruf Jawa untuk digunakan dalam proses pengenalan. Proses pengenalan dilakukan dengan menggunakan metode *Support Vector Machine*. *Output* nya adalah teks digital dari huruf Jawa yang diklasifikasi.

Hasil pengujian menunjukkan bahwa akurasi pengenalan huruf Jawa menggunakan metode *Support Vector Machine* mencapai 90.84%.

Kata Kunci: Pengenalan Huruf Jawa, *Area Based Feature Extraction*, *Support Vector Machine*

ABSTRACT

Javanese is one of the diverse Indonesian cultures on the island of Java. But along with the development of technology, Javanese letter culture began to be abandoned. Many young people today are more interested in learning foreign languages than Javanese.

Although Javanese lives on the island of Java, many still do not understand Javanese characters. It is unfortunate if a culture which is a Javanese characters must be lost because there is no young generation who can understand Javanese characters. Javanese characters have their own level of difficulty to learn because their shapes are similar between each other. Difficulties in learning Javanese characters also affect the interest of the younger generation to learn Javanese characters. In addition, at the present the application about Javanese characters is still not too much. Based on these problems, this Javanese characters recognition application is made in the hope of preserving Indonesian culture while making it easier to learn Javanese characters. The previous researches which were also Javanese characters recognition but using other methods such as Shape Based Feature Extraction as a feature extraction method and Probabilistic Neural Network, Elman Type, Feedforward Backpropagation as a recognition method but their accuracy are still relatively low. The process of introducing Javanese letters has 3 stages, namely segmentation, extraction features, and recognition. Input is in the form of Javanese letter document images that will be segmented. The segmentation process has been made by previous research and the result is a single Javanese image. The extraction feature process is done by using the Area Based Feature Extraction method to produce extraction feature data from one Javanese letter image to be used in the recognition process. The recognition process is done using the Support Vector Machine method. The output is digital text from classified Javanese letters.

The test results show that the accuracy of Javanese character recognition using Area Based Feature Extraction and Support Vector Machine method reaches 90.84%.

Keywords: *Javanese Character Recognition, Area Based Feature Extraction, Support Vector Machine.*

1. PENDAHULUAN

Indonesia merupakan bangsa yang memiliki budaya yang banyak dan beragam. Salah satu dari keragaman tersebut adalah bahasa Jawa. Huruf yang digunakan dalam bahasa Jawa bukanlah huruf latin dan biasa disebut sebagai aksara Jawa. Akan tetapi seiring dengan perkembangan teknologi, bahasa Jawa semakin dilupakan. Dulu, aksara Jawa menjadi mata pelajaran dan diajarkan di sekolah-sekolah sehingga siswa-siswi Indonesia masih mendapat ilmu mengenai aksara Jawa. Tetapi sekarang mata pelajaran ini pun kian menghilang. Orang Jawa sekarang lebih tertarik untuk mempelajari bahasa asing daripada mempelajari bahasa Jawa. Walaupun orang Jawa tinggal di pulau Jawa tetapi masih banyak

yang tidak mengerti huruf Jawa. Kondisi seperti ini sangat memprihatinkan. Sangat disayangkan apabila suatu budaya yaitu huruf Jawa harus hilang karena tidak ada generasi muda yang bisa berbahasa Jawa. [9]

Huruf Jawa atau lebih sering disebut aksara Jawa ini memiliki tingkat kesulitan tersendiri untuk dipelajari. Hal ini disebabkan oleh banyak lengkungan-lengkungan, lingkaran-lingkaran yang mirip antara satu huruf dengan huruf lainnya. Kesulitan dalam mempelajari huruf Jawa ini juga berpengaruh pada minat generasi muda untuk mempelajari huruf Jawa. Selain itu, saat ini aplikasi mengenai huruf Jawa juga masih cenderung tidak terlalu banyak. Berdasarkan permasalahan di atas, aplikasi pengenalan huruf Jawa ini dibuat dengan harapan dapat melestarikan budaya Indonesia sekaligus mempermudah pembelajaran huruf Jawa.

Aplikasi pengenalan huruf Jawa ini memiliki 3 tahap dalam mengenali huruf Jawa. Tahap yang pertama adalah segmentasi huruf Jawa. Segmentasi huruf Jawa ini akan membagi gambar dokumen huruf Jawa menjadi potongan-potongan gambar yang hanya berisi satu huruf Jawa. Setelah gambar dokumen huruf Jawa disegmentasi menjadi potongan satu huruf Jawa, langkah selanjutnya adalah ekstraksi fitur dari masing-masing gambar huruf Jawa tersebut. Langkah yang terakhir adalah pengenalan huruf Jawa berdasarkan fitur yang telah diekstrak tersebut. Pengenalan huruf Jawa ini akan dilakukan dengan menggunakan metode *Support Vector Machine*. *Support Vector Machine* adalah suatu metode yang memiliki kemampuan untuk mengklasifikasi data dengan tingkat keakuratan yang relatif tinggi. Oleh karena itu, penelitian ini akan menggunakan metode *Support Vector Machine* dalam mengenali huruf Jawa.

Penelitian ini melanjutkan penelitian sebelumnya yang dibuat oleh Singgih Mardianto dengan judul “Aplikasi Segmentasi Huruf Jawa”. Penelitian ini bertujuan mensegmentasi dokumen huruf Jawa yang utuh menjadi gambar satu huruf - satu huruf. Metode yang dipakai adalah metode *Projection Profile*. Hasil dari penelitian ini menunjukkan bahwa metode *Projection Profile* rata-rata dapat mensegmentasi tulisan pada dokumen yang memiliki jarak antar baris konsisten sebesar 77% untuk kelompok data foto dan 22% untuk kelompok data scan. Sedangkan untuk dokumen yang memiliki jarak antar baris tidak konsisten metode *Projection Profile* dapat mensegmentasi tulisan sebesar 22% untuk kelompok data foto dan 1% untuk kelompok data scan [7].

Salah satu penelitian yang telah dilakukan sebelumnya dibuat oleh Hans Christian Indrayana dengan judul “Pengenalan Huruf Jawa Menggunakan Jaringan Syaraf Tiruan *Feedforward Backpropagation* dan *Elman Type*”. Penelitian ini bertujuan mengenali huruf Jawa dengan menggunakan metode jaringan syaraf tiruan *Feedforward Backpropagation* dan *Elman Type*. Metode yang dipakai untuk ekstraksi fitur dalam penelitian ini adalah *Shape Based Feature Extraction*. Penelitian ini juga menerapkan metode Chi^2 untuk meningkatkan akurasi pengenalan. Hasilnya tanpa menggunakan metode Chi^2 adalah 33,53% untuk metode *Backpropagation* dan untuk metode *Elman Type* sebesar 38,27%. Jika menggunakan metode Chi^2 maka akurasi untuk metode *Backpropagation* meningkat menjadi 36,77% dan untuk metode *Elman Type* sebesar 39,1% [6].

Penelitian lain yang telah dilakukan sebelumnya dibuat oleh Bondan Sebastian dengan judul “Penggunaan Jaringan Syaraf Tiruan *Probabilistic Neural Network* dan *ID3 Algorithm* untuk Mengenali Dokumen Beraksara Jawa”. Penelitian ini juga bertujuan mengenali huruf Jawa tetapi menggunakan metode jaringan syaraf tiruan *Probabilistic Neural Network* dan *ID3 Algorithm*. Metode yang dipakai untuk ekstraksi fitur dalam

penelitian ini adalah *Shape Based Feature Extraction*. Untuk data yang sudah pernah di-*training*, metode PNN dapat mencapai akurasi 93,12%, sedangkan metode ID3 dapat mencapai akurasi hingga 100%. Untuk data yang belum pernah di-*training*, metode PNN dapat mencapai akurasi hingga 18,74%, sedangkan metode ID3 mencapai akurasi 14,1% [11].

Adapun penelitian yang meneliti perbandingan metode fitur ekstraksi *Shape Based Feature Extraction* dan *Area Based Feature Extraction* untuk pengenalan huruf Jawa. Penelitian ini berjudul “*Comparison between Shape-Based and Area-Based Feature Extraction for Java Character Recognition*” dan dilakukan oleh Rudy Adipranata, Gregorius Satia Budhi, Liliana, dan Bondan Sebastian. Hasil penelitian ini menunjukkan bahwa penggunaan metode *Area Based Feature Extraction* memberikan hasil yang lebih baik daripada *Shape Based Feature Extraction*. Dengan menggunakan *Probabilistic Neural Network* sebagai metode pengenalan, *Area Based Feature Extraction* menghasilkan akurasi lebih dari 60% sedangkan *Shape Based Feature Extraction* menghasilkan akurasi kurang dari 20% karena huruf Jawa banyak memiliki fitur bentuk yang mirip satu sama lain [1].

Adapun penelitian lain yang telah dilakukan sebelumnya dibuat oleh Aulia, Mahmud dan Retno dengan judul “Pengenalan Aksara Jawa Tulisan Tangan Menggunakan *Directional Element Feature* dan *Multi Class Support Vector Machine*”. Penelitian ini juga bertujuan mengenali huruf Jawa dengan metode sama yaitu *Support Vector Machine* tetapi dengan metode fitur ekstraksi *Directional Element Feature* dan aksara Jawa yang dikenali hanya aksara carakan dan swara. Penelitian ini memadukan metode DEF dengan SVM dan berhasil mencapai akurasi hingga 93,6% [5].

2. LANDASAN TEORI

2.1 Huruf Jawa

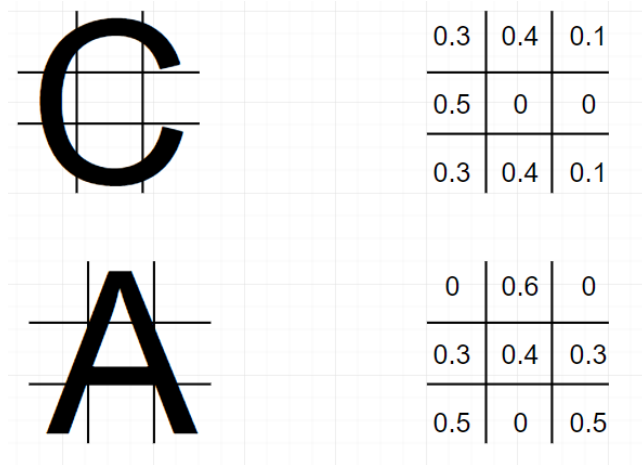
Aksara Jawa terdiri dari 20 huruf dasar, 20 huruf pasangan yang berfungsi menutup bunyi vokal, 8 huruf “utama” (aksara murda, ada yang tidak berpasangan), 8 pasangan huruf utama, lima aksara swara (huruf vokal depan), lima aksara rekan dan lima pasangannya, beberapa sandhangan sebagai pengatur vokal, beberapa huruf khusus, beberapa tanda baca, dan beberapa tanda pengatur tata penulisan [4].

2.2 Area Based Feature Extraction

Area Based Feature Extraction adalah metode untuk mengekstraksi fitur dari sebuah gambar dengan menggunakan kepadatan gambar [8]. Kepadatan dari sebuah gambar didefinisikan sebagai rasio antara jumlah *pixel* hitam dengan *area local* dari sebuah *region*. Ini dapat digunakan sebagai sebuah *feature vector* karena beberapa karakter memiliki kepadatan yang melebihi karakter lainnya.

Pengenalan karakter dengan menggunakan kepadatan gambar dapat dikembangkan dengan cara menghitung kepadatan dari bagian kecil sebuah gambar. Maksudnya adalah membagi gambar menjadi *grid* yang lebih kecil misalnya 3x3 atau 4x4 *pixel*. *Value* dari tiap *grid* ini adalah kepadatan dari *area* yang sudah dibagi tersebut dan akan berbeda untuk setiap karakter. Sebuah 3x3 *grid* akan menghasilkan 9 nilai yang bisa digunakan sebagai 9 *feature vector* untuk pengenalan karakter. Bagian ini dapat berbeda-beda ukurannya tergantung dari ukuran gambar karena tidak semua gambar dapat dibagi 3 sama rata. Angka-angka di sebelah kanan adalah angka hasil dari rasio jumlah *pixel* hitam dengan *area* dari bagian tersebut. Dari gambar di atas dapat dilihat bahwa perbedaan antara huruf A dan C cukup jelas. Huruf A tidak

memiliki *pixel* hitam di bagian kiri atas, kanan atas, dan tengah bawah. Contoh dari metode ini dapat dilihat pada gambar 1 di bawah ini:



Gambar 1. Contoh Area Based Feature Extraction

Sedangkan huruf C tidak memiliki *pixel* hitam di bagian tengah dan kanan tengah. Dengan kata lain, ini adalah metode yang mengekstraksi fitur dengan cara membagi gambar dokumen menjadi $A \times A$ bagian, lalu tiap bagian dihitung berapa persentase *pixel* hitamnya terhadap area dari bagian tersebut. Persentase inilah yang akan digunakan sebagai *feature vector*.

2.3 Support Vector Machine

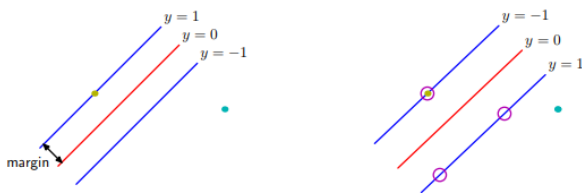
Support Vector Machine (SVM) merupakan metode yang berfungsi untuk memisahkan data menjadi dua buah kelas atau *binary classification* [2]. Rumus dasar dari *Support Vector Machine* adalah

$$y(x) = w^T \phi(x) + b \quad (1)$$

dimana $\phi(x)$ menyatakan *fixed feature-space transformation* dan b merupakan *bias*. *Training data set* terdiri dari N input vector x_1, \dots, x_n , dengan *target values* t_1, \dots, t_n dimana $t_n \in \{-1, 1\}$ dan data x yang baru akan diklasifikasi berdasarkan *sign* $y(x)$.

Training data set diasumsikan *linearly separable* sehingga ada paling tidak satu kombinasi *parameter* w dan b yang membuat fungsi (1) memenuhi $y(x_n) > 0$ untuk data yang memiliki $t_n = +1$ dan $y(x_n) < 0$ untuk data yang memiliki $t_n = -1$ sehingga $t_n y(x_n) > 0$ untuk semua *training data*.

Solusi yang dapat memisahkan kelas dengan tepat tentu ada banyak. Jika ada 2 atau lebih solusi yang mengklasifikasi *training data set* dengan tepat, maka solusi yang memberikan *generalization error* terkecil lah yang akan dipilih. *Support Vector Machine* menyelesaikan masalah ini dengan konsep *margin* yang didefinisikan sebagai jarak terkecil antara *decision boundary* dan sebuah sampel apapun, seperti digambarkan pada gambar 2:



Gambar 2. Margin dan Support Vector

Margin adalah jarak tegak lurus antara *decision boundary* dengan *data point* terdekat, seperti di gambar sebelah kiri. *Support Vector Machine* menggunakan *decision boundary* dengan *margin* maksimum. Memaksimumkan *margin* akan mengarah ke *decision boundary* tertentu seperti di gambar sebelah kanan. Lokasi dari *boundary* ini ditentukan oleh *subset* dari *data points* atau disebut sebagai *support vectors* yang disimbolkan oleh lingkaran.

Jarak tegak lurus dari titik x ke *hyperplane* yang didefinisikan oleh $y(x)$ dimana $y(x)$ merupakan fungsi (1) adalah $|y(x)| / \|w\|$. Solusi yang dipilih adalah solusi dimana semua *data points* terklasifikasi dengan tepat sehingga $t_n y(x_n) > 0$ untuk semua n . Jarak point x_n dengan *decision boundary* adalah

$$\frac{t_n y(x_n)}{\|w\|} = \frac{t_n (w^T \phi(x_n) + b)}{\|w\|} \quad (2)$$

Parameter w dan b akan optimisasi untuk memaksimumkan *margin* sehingga solusi untuk maksimum *margin* dapat ditemukan dengan menyelesaikan persamaan

$$\arg \max_{w,b} \left\{ \frac{1}{\|w\|} \min_n [t_n (w^T \phi(x_n) + b)] \right\} \quad (3)$$

dimana *factor* $1/\|w\|$ telah dikeluarkan dari optimisasi n karena w tidak tergantung pada n . Solusi untuk optimisasi ini akan sangat kompleks sehingga akan diubah menjadi persamaan yang lebih mudah untuk diselesaikan. Untuk melakukan ini jika dilakukan *rescaling* $w \rightarrow \kappa w$ dan $b \rightarrow \kappa b$, maka jarak antara *point* x_n ke *decision surface* yang diberikan $t_n y(x_n) / \|w\|$ tidak berubah. Kebebasan ini dapat digunakan untuk set

$$t_n (w^T \phi(x_n) + b) = 1 \quad (4)$$

untuk *point* yang paling dekat dengan *decision surface*. Dalam kasus ini, semua *data points* akan memenuhi *constraints*

$$t_n (w^T \phi(x_n) + b) \geq 1 \quad n = 1, \dots, N. \quad (5)$$

Ini dapat disebut sebagai *canonical representation* dari *decision hyperplane*. Pada dasarnya, akan selalu ada paling tidak 1 *active constraints* karena akan selalu ada titik terdekat dan ketika *margin* sudah maksimum akan ada paling tidak 2 *active constraints*. Setelah itu masalah optimisasi hanya membutuhkan memaksimumkan $\|w\|^{-1}$ atau ekuivalen dengan meminimalkan $\|w\|^2$, sehingga harus menyelesaikan masalah optimisasi

$$\arg \min_{w,b} \frac{1}{2} \|w\|^2 \quad (6)$$

dengan *constraints* (5). *Bias parameter* b sepertinya hilang dari optimisasi tetapi sebenarnya ditentukan secara implisit *via constraints* karena perubahan pada $\|w\|$ akan dikompensasi oleh perubahan b . Untuk menyelesaikan *constrained optimization problem* ini, maka digunakan *Lagrange multiplier* $a_n \geq 0$ dengan satu *multiplier* a_n untuk setiap *constraints* (5), sehingga menjadi *Lagrangian function*

$$L(w, b, a) = \frac{1}{2} \|w\|^2 - \sum_{n=1}^N a_n \{ (w^T \phi(x_n) + b) - 1 \} \quad (7)$$

dimana $a = (a_1, \dots, a_N)^T$. Tanda *minus* di depan *Lagrange multiplier* karena akan meminimalkan berdasarkan w dan b dan memaksimumkan berdasarkan a . *Set derivatives* dari $L(w, b, a)$ berdasarkan w dan b sama dengan 0, didapatkan 2 kondisi

$$w = \sum_{n=1}^N a_n t_n \phi(x_n) \quad (8)$$

$$0 = \sum_{n=1}^N a_n \quad (9)$$

menghilangkan w dan b dari $L(w, b, a)$ menggunakan kondisi di atas akan memberikan *dual representation* dari *maximum margin problem* yang memaksimalkan

$$\tilde{L}(a) = \sum_{n=1}^N a_n - \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N a_n a_m t_n t_m k(x_n, x_m) \quad (10)$$

dengan *constraints*

$$a_n \geq 0, \quad n = 1, \dots, N, \quad (11)$$

$$\sum_{n=1}^N a_n = 0 \quad (12)$$

disini *kernel function* didefinisikan oleh $k(x, x') = \phi(x)^T \phi(x')$. *Kernel* yang dipakai dalam skripsi ini adalah *RBF/Gaussian Kernel*.

Untuk mengklasifikasi data baru menggunakan *trained model*, digunakan *sign* dari $y(x)$ yang didefinisikan oleh persamaan (1).

Ini dapat diekspresikan oleh *parameter* $\{a_n\}$ dan *kernel function* dengan cara mensubstitusi w menggunakan persamaan (8) menjadi

$$y(x) = \sum_{n=1}^N a_n t_n k(x, x_n) + b \quad (13)$$

Pada awalnya, SVM hanya bisa digunakan untuk mengklasifikasi dua buah kelas saja. Tetapi seiring dengan bertambahnya tingkat kerumitan masalah yang ingin diselesaikan, peneliti mengembangkan SVM sehingga bisa digunakan untuk mengklasifikasi data dengan lebih dari dua kelas.

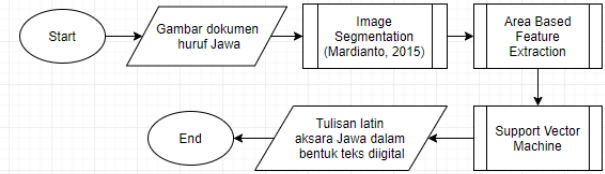
Salah satu pendekatan yang digunakan adalah metode *one-versus-the-rest*. Dengan metode ini, *hyperplane* yang akan dibuat akan sebanyak dengan jumlah kelas yang ingin dibagi. Setiap *hyperplane* bertugas membedakan apakah setiap data masuk ke suatu kelas tertentu atau lainnya.

Pendekatan lain yang dapat dilakukan adalah metode *one-versus-one*. Dengan metode ini, *hyperplane* yang akan dibuat untuk semua kombinasi dari kelas yang ingin dibagi. Setiap *hyperplane* akan membedakan misalnya kelas A dengan kelas B, kelas A dengan kelas C, kelas A dengan kelas D, kelas B dengan kelas C, dan seterusnya.

3. ANALISA dan DESAIN SISTEM

3.1 Garis Besar Sistem

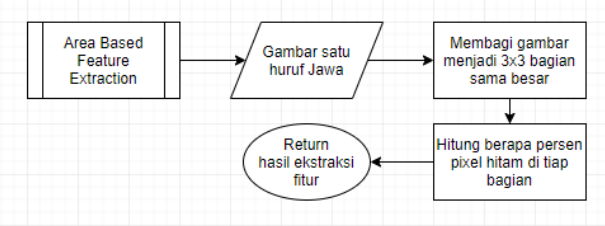
Pengenalan huruf Jawa dapat dibagi ke dalam 3 langkah yaitu *Image Segmentation* untuk memotong *image* dokumen menjadi *image* per huruf Jawa, *Area Based Feature Extraction* untuk mengekstraksi fitur dari tiap huruf Jawa, dan *Support Vector Machine* sebagai metode untuk mengenali huruf Jawa tersebut. Langkah yang pertama yaitu *Image Segmentation* sudah dikerjakan oleh Singgih Mardianto, 2015, skripsi ini hanya mengerjakan langkah ke 2 dan 3. Secara garis besar, desain yang akan digunakan dalam skripsi ini bisa dilihat pada Gambar 3



Gambar 3. Flowchart Garis Besar Sistem

3.2 Area Based Feature Extraction

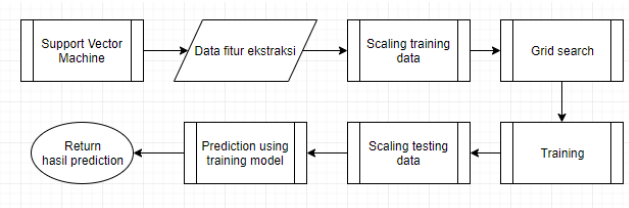
Metode yang digunakan untuk proses fitur ekstraksi adalah *Area Based Feature Extraction*. Metode ini akan membagi gambar satu huruf menjadi 3x3 bagian, lalu tiap bagian akan dihitung berapa persen *pixel* hitamnya. Akan tetapi, banyak gambar yang berhasil disegmentasi terdapat *space* putih seperti pada gambar 3.4. Ini akan mempengaruhi keakuratan dalam mengekstraksi fitur dengan menggunakan metode *Area Based* sehingga dilakukan *crop* untuk menghilangkan *space* putih tersebut. Hasil *crop* inilah yang akan diekstraksi fiturnya sehingga mendapat 9 fitur untuk pengenalan. *Flowchart* untuk *Area Based Feature Extraction* dapat dilihat pada gambar 4 di bawah ini:



Gambar 4. Flowchart Area Based Feature Extraction

3.3 Support Vector Machine

Support Vector Machine adalah metode yang akan dipakai untuk proses klasifikasi. Secara garis besar, proses ini dapat dilihat pada gambar 5 di bawah ini:



Gambar 5. Flowchart Support Vector Machine

3.3.1 Scaling Training

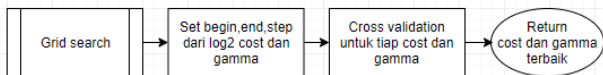
Scaling training data ini bertujuan agar fitur dari *training* memiliki *range* yang sama. Misalnya suatu fitur bernilai 0.1 dan fitur lainnya bernilai 400, jika data tidak di-*scale* terlebih dahulu maka fitur yang bernilai 0.1 akan menjadi tidak signifikan karena fitur kedua memiliki nilai yang jauh lebih tinggi. Hal ini tentu akan mempengaruhi akurasi dari klasifikasi. Setelah proses *scaling* selesai akan dihasilkan *file training scale* yang akan digunakan untuk proses selanjutnya. Secara garis besar, proses *scaling training data* dapat dilihat pada gambar 6 di bawah ini:



Gambar 6. Flowchart Scaling Training

3.3.2 Grid Search

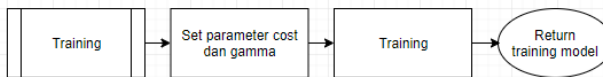
Grid search bertujuan untuk mencari parameter cost dan gamma terbaik untuk training. Parameter dari grid search sendiri adalah begin,end, dan step dari log 2 cost dan gamma. Log 2 digunakan agar angka yang digunakan tidak terlalu besar. Grid search akan mencari akurasi terbaik dari setiap kemungkinan cost dan gamma, lalu cost dan gamma tersebut akan digunakan sebagai parameter proses training. Cost adalah parameter yang menentukan apakah hyperplane yang dihasilkan ketat atau fleksibel [3]. Cost rendah akan menghasilkan hyperplane dengan margin tinggi tetapi akan membiarkan terjadi beberapa misklasifikasi. Cost yang tinggi akan menghasilkan hyperplane yang sangat ketat dalam klasifikasi walaupun margin rendah. Gamma adalah parameter dari RBF/Gaussian Kernel yang menentukan varians dari kernel [10]. Gamma yang tinggi hanya akan menganggap 2 titik mirip jika jaraknya dekat. Gamma yang rendah bisa membuat 2 titik yang jaraknya jauh dianggap mirip. Secara garis besar, proses grid search dapat dilihat pada gambar 7 di bawah ini:



Gambar 7. Flowchart Grid Search

3.3.3 Training

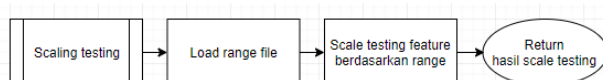
Proses training ini adalah proses untuk menentukan support vector yang digunakan untuk membuat hyperplane pemisah antar kelas. Training ini menggunakan parameter cost dan gamma terbaik dari hasil grid search. Setelah training selesai akan dihasilkan training model file untuk proses testing. Secara garis besar, proses training dapat dilihat pada gambar 8 di bawah ini:



Gambar 8. Flowchart Training

3.3.4 Scaling Testing

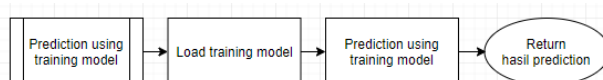
Scaling testing data juga perlu dilakukan agar data fitur training dan data fitur testing memiliki range yang sama. Proses ini akan menggunakan range file dari hasil scaling training data sebagai acuan. Setelah proses scaling selesai maka akan dihasilkan file scale testing yang akan digunakan untuk proses prediction. Secara garis besar, proses scaling testing data dapat dilihat pada gambar 9 di bawah ini.



Gambar 9. Flowchart Scaling Testing

3.3.5 Prediction using Training Model

Proses ini adalah proses dimana training model akan digunakan untuk klasifikasi data testing yang sudah di scale. Secara garis besar, proses ini dapat dilihat pada gambar 10 di bawah ini:



Gambar 10. Flowchart Prediction

4. PENGUJIAN SISTEM

4.1 Pengujian Area Based Feature Extraction

Sebelum melakukan training ataupun menjalankan testing dengan metode Support Vector Machine diperlukan proses fitur ekstraksi untuk mendapatkan data fitur gambar yang digunakan untuk input. Data tersebut berupa angka yang diperoleh dari gambar yang dibagi 3x3 sama rata dan disesuaikan dengan format input dari library LIBSVM. Sebagian dari hasil fitur ekstraksi untuk data training dapat dilihat pada gambar 11.

1	1:0.01098901098901099	2:0.14285714285714285
3:0.027472527472527472	4:0.17032967032967034	
5:0.1043956043956044	6:0.06593406593406594	
7:0.005494505494505495	8:0.13186813186813187	
9:0.02197802197802198		
1	1:0.01098901098901099	2:0.14285714285714285
3:0.027472527472527472	4:0.17032967032967034	
5:0.1043956043956044	6:0.06593406593406594	
7:0.005494505494505495	8:0.13186813186813187	
9:0.02197802197802198		

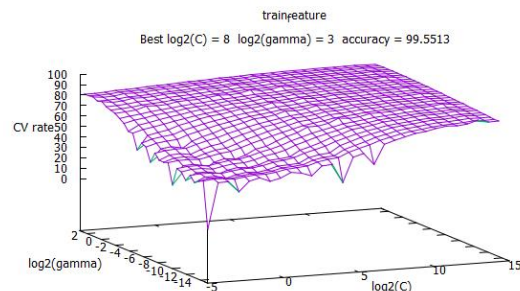
Gambar 11. Hasil Fitur Ekstraksi

Angka di paling depan melambangkan kelas dari huruf Jawa dan akan berubah sesuai dengan kelasnya, misalnya 1 adalah untuk angka 0, 2 adalah untuk angka 1, dan seterusnya. Sedangkan angka berikutnya melambangkan jumlah persen pixel hitam di area yang sudah dibagi 3x3 sama rata, misalnya 1 adalah untuk area kiri atas, 2 adalah untuk area kiri tengah, 3 adalah untuk area kanan atas, dan seterusnya.

4.2 Pengujian Support Vector Machine

4.2.1 Grid Search dengan Cost -5,15,1 dan Gamma -15,3,1

Pengujian ini adalah pengujian yang menggunakan grid search dengan log2c begin,end,step nya adalah -5,15,1 dan log2g begin,end,step nya adalah -15,3,1. Hasil dari pengujian ini dapat dilihat pada Gambar 12.

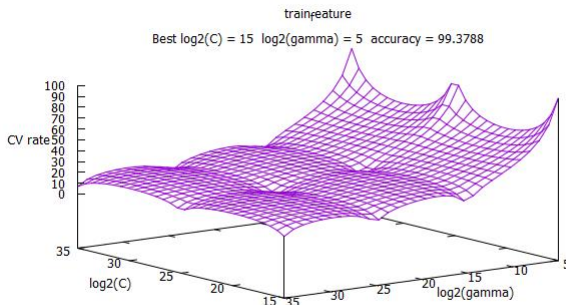


Gambar 12. Hasil Grid Search Pengujian 4.2.1

Pengujian di atas memakan waktu 1 hari 2 jam 44 menit. Pengujian ini mencakup parameter yang cukup luas agar mengetahui efek dari cost dan gamma secara general. Accuracy dalam grafik adalah akurasi dari 5 fold cross validation. Grid search akan memilih parameter dengan cross validation rate tertinggi yaitu log2c = 8 dan log2g = 3 dengan cross validation rate 99.5513 dan akurasi terhadap data testing 87.91%.

4.2.2 Grid Search dengan Cost 15,35,10 dan Gamma 5,35,10

Pengujian ini adalah pengujian yang menggunakan *grid search* dengan $\log_2 c$ *begin.end,step* nya adalah 15,35,10 dan $\log_2 g$ *begin.end,step* nya adalah -5,35,10. Hasil dari pengujian ini dapat dilihat pada Gambar 13.

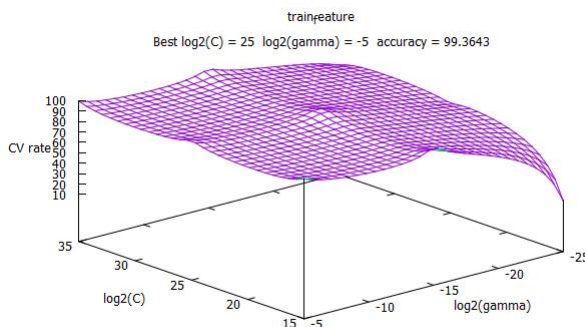


Gambar 13. Hasil Grid Search Pengujian 4.2.2

Pengujian di atas memakan waktu 10 jam 21 menit. Pengujian ini dilakukan untuk mengetahui efek ketika *cost* dan *gamma* bernilai positif. *Accuracy* dalam grafik adalah akurasi dari 5 *fold cross validation*. *Grid search* akan memilih *parameter* dengan *cross validation rate* tertinggi yaitu $\log_2 c = 15$ dan $\log_2 g = 5$ dengan *cross validation rate* 99.3788 dan akurasi terhadap *data testing* 84.5%.

4.2.3 Grid Search dengan Cost 5,35,10 dan Gamma -5,-25,-10

Pengujian ini adalah pengujian yang menggunakan *grid search* dengan $\log_2 c$ *begin.end,step* nya adalah 5,25,10 dan $\log_2 g$ *begin.end,step* nya adalah -5,-25,-10. Hasil dari pengujian ini dapat dilihat pada Gambar 14.



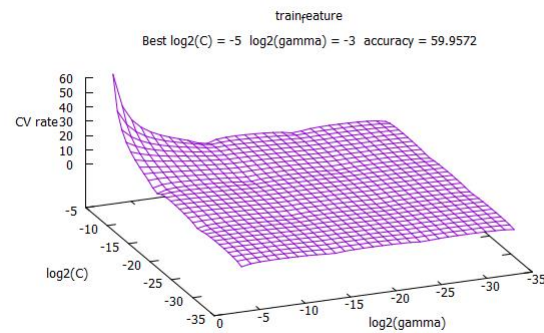
Gambar 14. Hasil Grid Search Pengujian 4.2.3

Pengujian di atas memakan waktu 26 menit. Pengujian ini dilakukan untuk mengetahui efek ketika *cost* bernilai positif dan *gamma* bernilai negatif. *Accuracy* dalam grafik adalah akurasi dari 5 *fold cross validation*. *Grid search* akan memilih *parameter* dengan *cross validation rate* tertinggi yaitu $\log_2 c = 25$ dan $\log_2 g = -5$ dengan *cross validation rate* 99.3643 dan akurasi terhadap *data testing* 90.84%.

4.2.4 Grid Search dengan Cost -5,-35,-10 dan Gamma -3,-33,-10

Pengujian ini adalah pengujian yang menggunakan *grid search* dengan $\log_2 c$ *begin.end,step* nya adalah -5,-35,-10 dan $\log_2 g$

begin.end,step nya adalah -3,-33,-10. Hasil dari pengujian ini dapat dilihat pada Gambar 15.

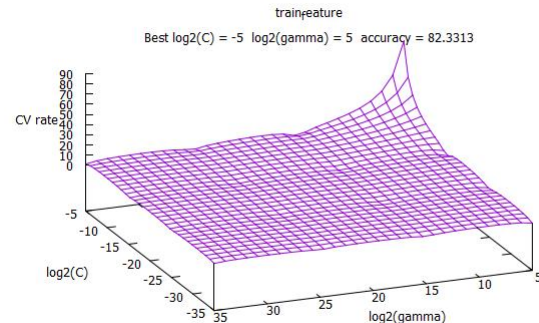


Gambar 15. Hasil Grid Search Pengujian 4.2.4

Pengujian di atas memakan waktu 43 menit. Pengujian ini dilakukan untuk mengetahui efek ketika *cost* dan *gamma* bernilai negatif. *Accuracy* dalam grafik adalah akurasi dari 5 *fold cross validation*. *Grid search* akan memilih *parameter* dengan *cross validation rate* tertinggi yaitu $\log_2 c = -5$ dan $\log_2 g = -3$ dengan *cross validation rate* 59.9572 dan akurasi terhadap *data testing* 62.8%.

4.2.5 Grid Search dengan Cost -5,-35,-10 dan Gamma 5,35,10

Pengujian ini adalah pengujian yang menggunakan *grid search* dengan $\log_2 c$ *begin.end,step* nya adalah -5,-35,-10 dan $\log_2 g$ *begin.end,step* nya adalah -5,35,10. Hasil dari pengujian ini dapat dilihat pada Gambar 16.



Gambar 16. Hasil Grid Search Pengujian 4.2.5

Pengujian di atas memakan waktu 3 jam 28 menit. Pengujian ini dilakukan untuk mengetahui efek ketika *cost* bernilai positif dan *gamma* bernilai negatif. *Accuracy* dalam grafik adalah akurasi dari 5 *fold cross validation*. *Grid search* akan memilih *parameter* dengan *cross validation rate* tertinggi yaitu $\log_2 c = -5$ dan $\log_2 g = 5$ dengan *cross validation rate* 82.3313 dan akurasi terhadap *data testing* 57.24%.

Dari pengujian di atas dapat dilihat bahwa *gamma* yang besar akan membuat *training time* lebih lama. Pengujian di atas juga menunjukkan bahwa *gamma* yang semakin jauh dari angka 0 baik itu positif maupun negatif akan mengurangi akurasi dalam *testing*. Sedangkan *cost* tidak ada pola tertentu dan jika *cost* bernilai negatif akurasi dalam *testing* akan cenderung menurun. Pengujian ini menunjukkan bahwa *parameter* yang *optimal* adalah $\log_2 c$ bernilai 25 dan $\log_2 g$ bernilai -5 dengan akurasi 90.84%

4.3 Pengujian Model Terbaik terhadap Macam Aksara Jawa

Pengujian ini adalah pengujian dengan menggunakan *training model* terbaik terhadap *class* aksara Jawa yang berbeda. *Class* yang akurasi paling tinggi adalah *class* Pratantha sedangkan *class* yang akurasi paling rendah adalah *class* Swara dan *class* Pasangan. Pengujian ini dapat dilihat pada tabel 1 di bawah ini:






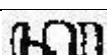


Tabel 1. Hasil Pengujian Tiap Macam Aksara Jawa

Class	Accuracy	Total Correct	Total Data
Carakan	89.8574	1134	1262
Murda	88.7728	340	383
Pasangan	82.4007	913	1108
Pratantha	99.6845	632	634
Rekan	97.5045	547	561
Sandhangan	98.9552	663	670
Swara	82.3529	210	255
Wilangan	88.4615	460	520
	90.83997775	4899	5393

4.3.1 Pengujian Aksara Carakan

Pengujian ini hanya contoh dari pengujian aksara carakan. Dari pengujian ini dapat dilihat bahwa terjadi beberapa misklasifikasi aksara carakan la sebagai ga karena bentuk dan ketebalan kedua aksara tersebut mirip. Tabel hasil pengujian aksara carakan dapat dilihat pada tabel 2 di bawah ini:

Tabel 2. Contoh Hasil Pengujian Aksara Carakan

Gambar Aksara	Nama Aksara	Pengenalan
	BA	BA
	CA	CA
	DA	DA
	DHA	DHA
	GA	GA
	KA	KA
	LA	GA
	MA	MA

5. KESIMPULAN

Berdasarkan hasil pengujian dapat disimpulkan beberapa hal berikut:

1. Berdasarkan hasil pengujian, akurasi metode *Area Based Feature Extraction* dan *Support Vector Machine* untuk pengenalan aksara Jawa adalah 90.84%.

2. *Parameter cost* dan *gamma* yang optimal untuk *Support Vector Machine* dalam pengenalan aksara Jawa adalah $\log_2 cost$ bernilai 25 dan $\log_2 gamma$ bernilai -5 atau *cost* bernilai 33.554.432 dan *gamma* bernilai 0.03125.
3. Aksara pasangan memiliki akurasi yang lebih rendah dari aksara lainnya karena banyak aksara pasangan yang bentuknya sama dengan aksara carakan tetapi letaknya berbeda sedangkan metode *Area Based Feature Extraction* tidak memiliki informasi letak aksara.

6. DAFTAR PUSTAKA

- [1] Adipranata R, et al. 2016. *Comparison between Shape Based and Area Based Features Extraction for Java Character*. ITMSOC.
- [2] Bishop, C. M. 2006. *Pattern Recognition and Machine Learning*. Singapore: Springer.
- [3] Caspersen, K.M. 2015. *What is the influence of C in SVMs with linear kernel?* Retrieved November 30, 2018 from <https://stats.stackexchange.com/q/159051>
- [4] Hastuti, D. 2011. *Mari belajar (lagi) menulis Aksara Jawa :*. Retrieved June 4, 2018, from <https://dhenokhastuti.wordpress.com/2011/04/11/mari-belajar-lagi-menulis-aksara-jawa/>
- [5] Husni, A., Dwi, M., & Novi, R. 2014. *Pengenalan Aksara Jawa Tulisan Tangan Menggunakan Directional Element Feature dan Multi Class Support Vector Machine*. Palembang: Konferensi Nasional Teknologi Informasi dan Aplikasinya.
- [6] Indrayana, H.C., 2016. *Pengenalan Aksara Jawa menggunakan Jaringan Saraf Tiruan Feedforward Backpropagation dan Elman Type*. Teknik Informatika - Universitas Kristen Petra, Surabaya.
- [7] Mardianto, S., 2015. *Aplikasi Segmentasi Aksara Jawa*. Teknik Informatika - Universitas Kristen Petra, Surabaya.
- [8] Parker, J. R., 1993. *Practical computer vision using C*. New York: Wiley.
- [9] *Penggunaan Bahasa Jawa untuk Melestarikan Warisan Budaya Indonesia dalam Lingkup Pemuda Jawa*. 2015. Retrieved from <https://www.kompasiana.com/isyaoakta/54f7563ca3331184358b45e6/penggunaan-bahasa-jawa-untuk-melestarikan-warisan-budaya-indonesia-dalam-lingkup-pemuda-jawa?page=all>
- [10] Pinto, A. 2016. *Does anyone know what is the Gamma parameter (about RBF kernel function)?*. Retrieved November 30, 2018 from https://www.researchgate.net/post/Does_anyone_know_what_is_the_Gamma_parameter_about_RBF_kernel_function
- [11] Sebastian, B., 2015. *Penggunaan Jaringan Syaraf Tiruan Probabilistic Neural Network dan ID3 Algorithm untuk Mengenali Dokumen Beraksara Jawa*. Teknik Informatika - Universitas Kristen Petra, Surabaya.