

**SELF ORGANIZING MAPS (SOM)
SUATU METODE UNTUK PENGENALAN AKSARA JAWA**

Akik Hidayat¹, Rahmi Nur Shofa²

¹Prodi Teknik Informatika FMIPA Universitas Padjadjaran

²Prodi Teknik Informatika Fakultas Teknik Universitas Siliwangi

e-mail : akik@unpad.ac.id¹, rahmi.shofa@gmail.com²

Abstrak

Aksara jawa/*hanacaraka* merupakan warisan budaya masyarakat jawa kuno yang digunakan untuk menulis dalam pembuatan kitab-kitab, naskah kuno, tembang-tembang Jawa, prasasti, atau surat menyurat. Sangatlah penting untuk menjaga dan memelihara warisan ini, agar bangsa Indonesia dapat memahami nilai-nilai budaya dan seni yang tertulis. Pada tanggal 1 Oktober 2009 Aksara Jawa/*hanacaraka* telah memiliki standar dari Unicode. Oleh karena itu mencoba mengembangkan model dan perangkat lunak pengenalan aksara Jawa/*hanacaraka*. Dalam penelitian ini, model dan perangkat lunak yang dibangun menggunakan *feature extraction zoning*. Sedangkan algoritma yang digunakan adalah algoritma *Self Organizing Maps (SOM)*. SOM merupakan salah satu algoritma dari Jaringan Syaraf Tiruan (JST) dengan pembelajaran tidak terawasi (*unsupervised learning*) yang mempelajari distribusi himpunan pola-pola tanpa informasi kelas. Dari hasil pengujian, tingkat akurasi JST SOM dalam mengenali aksara jawa cukup memuaskan.

Kata Kunci: Aksara Jawa/*hanacaraka*, feature extraction, SOM.

Abstract

Javaese Script / hanacaraka a cultural heritage of ancient Javanese people used to write in the manufacture of books, manuscripts, song Java, inscriptions, or correspondence. It is important to maintain and preserve this heritage, so that the Indonesian people can understand the cultural values and art writing. On October 1, 2009 Javanese Script / hanacaraka already have a standard of Unicode. Therefore, the authors tried to develop models and Java script/hanacaraka recognition software. In this study, model and software that is built using the zoning feature extraction. While the algorithm used is the algorithm Self Organizing Maps (SOM). SOM is one of the algorithms of Artificial Neural Network (ANN) with unsupervised learning (unsupervised learning) who studied the distribution of the set of patterns without any class information. From the test results, SOM ANN accuracy rate in recognizing javanese script / hanacaraka is good enough.

Keywords: *Javanese Script / hanacaraka, feature extraction, Self Organizing Maps (SOM).*

I. LATAR BELAKANG MASALAH

Perkembangan teknologi saat ini sangatlah pesat dan mempunyai peranan penting dalam kehidupan manusia. Teknologi diciptakan untuk membantu dan mempermudah manusia dalam kehidupan ini.

Salah satu perkembangan teknologi manusia yang paling pesat adalah komputer. Komputer yang dahulunya hanya merupakan alat untuk menghitung dan mengetik, namun seiring berjalan waktu, komputer berubah menjadi alat untuk menghitung, mengetik, menyimpan data, mengolah data, sistem cerdas, sistem pakar, olah gambar (*image processing*), game, *robotic* dan lain sebagainya.

Komputer diharapkan bisa melakukan segala sesuatu yang dikerjakan oleh manusia. Supaya komputer bisa bertindak seperti manusia, diperlukan beberapa metode untuk menjadikan komputer sebagai mesin yang pintar. Salah satunya yaitu

dengan implementasi jaringan Syaraf Tiruan (JST) pada komputer.

JST merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses otak manusia pada manusia tersebut. Istilah buatan disini digunakan karena JST ini diimplementasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran. Salah satu penggunaan jaringan syaraf tiruan adalah untuk pengenalan pola.

Salah satu metode dari JST adalah *Self Organizing Maps (SOM)*. SOM pertama kali diperkenalkan oleh Teuvo Kohonen pada awal 1980-an. Metode SOM termasuk algoritma pembelajaran tidak terawasi (*unsupervised learning*) dengan mempelajari himpunan distribusi pola-pola tanpa informasi kelas.

Oleh karena itu, dalam penelitian ini, penulis mencoba mengembangkan model dan aplikasi

dengan aksara jawa/*hanacaraka* sebagai objeknya dengan menggunakan metode SOM.

Aksara jawa/*hanacaraka* merupakan warisan budaya masyarakat jawa kuno yang digunakan untuk menulis dalam pembuatan kitab-kitab, naskah kuno, tembang-tembang Jawa, prasasti, atau surat menyurat. Sangatlah penting untuk menjaga dan memelihara warisan ini, agar bangsa Indonesia dapat memahami nilai-nilai budaya dan seni yang tertulis.

Untuk itu, penelitian ini diharapkan bisa menjadi salah satu cara untuk menjaga dan memelihara aksara Jawa/*hancaraka*.



Gambar 2.1 Aksara jawa/*hanacaraka*

II. TINJAUAN PUSTAKA

2.1 Aksara Jawa

Aksara jawa atau lebih dikenal dengan *Hanacaraka* adalah turunan aksara *brahmani*(berasal dari Hindustan) yang digunakan untuk penulisan berbahasa Jawa, bahasa Makasar, bahasa Madura, bahasa Melayu, bahasa Sunda, bahasa Bali dan bahasa *Sasak*.

Di negeri Hindustan tersebut terdapat bermacam-macam aksara, salah satunya yaitu aksara *Pallawa* yang berasal dari India bagian selatan. Dinamakan aksara *Pallawa* karena berasal dari salah satu kerajaan yang ada di sana yaitu Kerajaan *Pallawa*. Aksara *Pallawa* itu digunakan sekitar abad ke-4 Masehi. Di Nusantara terdapat bukti sejarah berupa prasasti Yupa di Kutai, Kalimantan Timur, ditulis dengan menggunakan aksara *Pallawa*. Aksara *Pallawa* ini menjadi cikal bakal dari semua aksara yang ada di Nusantara, seperti aksara *Hanacaraka*, aksara *Rencong* (aksara *Kaganga*), aksara Batak, Aksara Makasar dan Aksara *Baybayin* (aksara di Filipina).

2.1.1 Pengelompokkan Aksara Jawa

Pada bentuknya yang asli aksara Jawa/*hanacaraka* ditulis menggantung (dibawah garis), seperti aksara Hindi. Namun demikian, pengajaran modern sekarang menuliskannya diatas garis.

Aksara Jawa/*hanacaraka* memiliki 20 huruf dasar (*aksara ngalegena*), 20 huruf pasangan yang berfungsi menutup bunyi vocal, 8 huruf “utama” (*aksara murda*), 8 pasangan huruf utama, 5 aksara *swara* (huruf vokal depan), 5 aksara *rekan* dan 5 pasangannya, beberapa *sandhangan* sebagai pengatur vocal, beberapa huruf khusus, beberapa tanda baca, dan beberapa tanda pengatur tata penulisan.

2.1.2 Unicode

Unicode merupakan sebuah set karakter yang menjadi dasar pembuatan *font* atau aplikasi komputer yang dapat mendukung berbagai bahasa dan aksara (Baidillah, dkk. 2008). Dengan adanya *Unicode*, setiap aksara dapat direpresentasikan untuk menulis pada komputer. *Unicode* menggunakan 16 bit untuk merepresentasikan karakter. Dengan demikian, banyaknya karakter yang dapat direpresentasikan adalah 2^{16} atau 65.536 karakter.

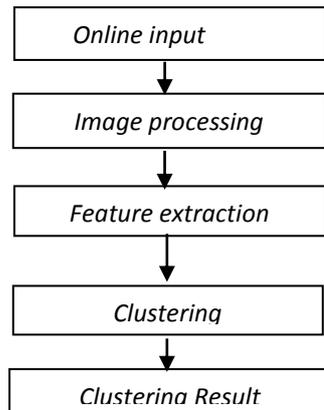
Baru sejak awal 2005 dilakukan usaha bertahap yang nyata untuk mengintegrasikan aksara Jawa ke dalam *Unicode* setelah Michael Everson membuat suatu *code table* sementara untuk didaftarkan. Kelambatan ini terjadi karena kurangnya dukungan dari masyarakat pengguna aksara ini. Baru semenjak masa ini mulai terhimpun dukungan dari masyarakat pengguna. Aksara Jawa/*Hanacaraka* saat ini telah dirilis dalam *Unicode* versi 5.2 (tergabung dalam Amandemen 6) yang keluar pada tanggal 1 Oktober 2009. Alokasi Memori Aksara Jawa (*Javanese*) pada *Unicode* 5.2.0 adalah di alamat heksadesimal A980 sampai dengan A9DF (desimal: 43392–43487).

2.2 Open Character Recognition (OCR)

Open Character Recognition (OCR) dikenal sebagai sistem yang dapat membaca dan mengenal huruf, baik yang dicetak dari printer, mesin tik maupun tulisan tangan. OCR dapat dipandang sebagai bagian dari pengenalan yang lebih luas, yakni pengenalan pola otomatis. Dalam pengenalan pola otomatis, sistem pengenalan pola mencoba mengenali apakah citra masukan yang diterima cocok dengan salah satu citra yang telah ditentukan.

OCR memiliki dua cara, yaitu *off-line* dan *on-line*. *Off-line* merupakan cara pengenalan tulisan dengan input berupa gambar hasil *scan*. Sedangkan *on-line* merupakan cara pengenalan tulisan dengan mengenali tulisan langsung dengan *input-an* berupa

coretan tulisan yang ditulis secara *real time* pada media penulisan digital.



Gambar 2.2 Proses OCR

2.3 Image Processing

Image processing adalah suatu pengolahan data yang *input*-nya berupa gambar dan *output*-nya juga berupa gambar dengan tujuan untuk memperbaiki informasi pada gambar sehingga mudah terbaca atau memperbaiki kualitas dari gambar itu sendiri. *Image Processing* memiliki beberapa tipe untuk memproses gambar, namun dalam penelitian ini yang digunakan dalam penulis hanya akan menggunakan proses segmentasi. Proses segmentasi bertujuan untuk men-segmen atau *capture* suatu citra kedalam beberapa segmen dengan suatu kriteria tertentu. Proses segmentasi dilakukan dengan mencari batas daerah yang akan di-*capture*, mulai dari batas atas, batas bawah, batas kanan dan batas kiri.

Proses segmentasi dimulai dengan men-*scan* citra dari atas ke bawah dan dari kiri ke kanan. Proses *scan* merupakan proses untuk mengetahui awal terdapatnya piksel hitam sampai selanjutnya bertemu piksel hitam yang terakhir. Selanjutnya akan dilakukan pengkotakan atau pembatasan daerah citra yang memiliki piksel hitam, deteksi tepi adalah operasi yang dijalankan untuk mendeteksi garis tepi yang membatasi dua wilayah citra homogen yang memiliki tingkat kecerahan yang berbeda (Pitas, 1993). Tujuan dari deteksi tepi adalah untuk meningkatkan penampakan garis batas suatu daerah atau obyek dalam suatu citra. dan *scalling* yang merupakan proses mengubah ukuran suatu citra, dengan menormalisasikan ukuran sehingga ukuran yang diperoleh selalu sama, walaupun ukuran tulisan awal atau gambar awal tidak sama. *Scalling* terdiri atas *Feature Extraction* bertujuan untuk mendapatkan karakteristik suatu karakter yang membedakannya dengan karakter lain, yang disebut *feature* (Kusumadewi, 2009). Teknik *feature extraction* dalam penelitian ini adalah *zoning* dan ekstraksi ciri titik. Cara kerja metode *zoning* adalah

membagi karakter menjadi $N \times M$ wilayah. Dari setiap wilayah, *feature* diekstraksi untuk membentuk *feature* vektor. Tujuan *zoning* adalah memperoleh karakteristik lokasi disamping karakteristik global.

2.4 Jaringan Syaraf Tiruan (JST)

Jaringan Syaraf Tiruan (JST) adalah salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia tersebut. Istilah buatan disini digunakan karena JST ini diimplementasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses penghitungan selama proses pembelajaran (Fausett, 1994).

2.4.1 Self Organizing Maps (SOM)

Teknik *self Organizing Maps* (SOM) dikenalkan pertama kali oleh Teuvo Kohonen pada awal tahun 1980-an. SOM merupakan metode JST dengan pembelajaran tidak terawasi (*unsupervised learning*) yang mempelajari distribusi himpunan pola-pola tanpa informasi kelas. SOM terdiri dari dua buah lapisan (*layer*), yaitu lapisan *input* dan lapisan *output*. Setiap *neuron* dalam lapisan *input* terhubung langsung dengan setiap *neuron* pada lapisan *output*. Setiap *neuron* dalam lapisan *output* mempresentasikan kelas dari *input* yang diberikan

2.4.2 Algoritma Pembelajaran SOM

Menurut Song dan Resman (1999) prinsip kerja algoritma SOM adalah pengurangan *node-node* tetangganya (*neighbour*), sehingga pada akhirnya hanya ada satu keluaran *node* yang terpilih (*winner node*).

Berikut ini adalah tahapan/algoritma dari metode SOM :

Langkah 0 : Inisialisasi pembobotan w_{ij} dengan nilai random. Menyet parameter *learning rate* (α), pengurangan *learning rate* (β) dan *MaxEpoch*.

Langkah 1 : Apabila kondisi selesai belum terpenuhi, lakukan langkah berikut

a. Untuk tiap j ($j=1, \dots, n$), hitung:

$$d_j = \sum_{i=0}^{n-1} (x_i - w_{ij})^2 \quad (2.1)$$

b. Cari indeks j yang membuat d_j minimum

c. Lakukan perbaikan nilai w_{ij} dengan nilai tertentu, yaitu:

$$w_{ij} = w_{ij}(\text{lama}) + \alpha (x_i - w_{ij}(\text{lama})) \quad (2.2)$$

d. Lakukan update *learning rate*

$$\alpha = \alpha \cdot \beta \quad (2.3)$$

e. Test kondisi stop-nya

Keterangan: x : input
 j : indeks node output
 d_j : Jarak Eucliden
 α : Learning rate ; $0 < \alpha < 1$
 w : bobot
 β : Pengurangan learning rate

III. ANALISIS DAN PERANCANGAN

3.1 Analisis Kebutuhan Aplikasi

Untuk membangun aplikasi ini, kita harus menganalisis semua kebutuhan yang diperlukan. Kebutuhan yang diperlukan dalam penelitian ini adalah sebagai berikut:

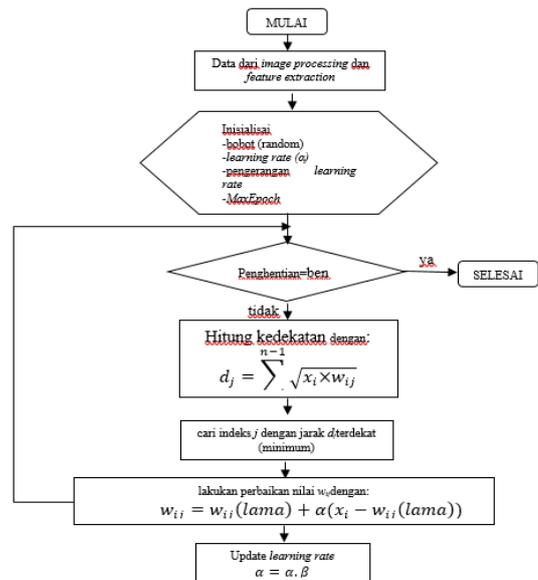
- Input handwriting**
 User meng-input-kan tulisan tangan berupa aksara jawa/hanacaraka kedalam sistem sesuai keinginannya.
- Ekstraksi ciri**
 Aplikasi ini mengekstraksi *feature* secara otomatis setelah tulisan tangan aksara jawa/hanacaraka di-input-kan.
- Training**
 Aplikasi ini akan men-*training* data yang telah di-input-kan dan tersimpan dalam dataset untuk melatih atau meng-*cluster* data tersebut.
- Recognize**
 User melakukan *recognize* terhadap tulisan tangan yang telah di-input-kan oleh user dengan data hasil *training*.

3.2 Model Pengembangan Self Organizing Maps (SOM)

Self Organizing Maps (SOM) akan dimodifikasi dengan tujuan untuk mengenali tulisan tangan aksara jawa/hanacaraka. Untuk menghitung kedekatan antar *node*, rumus dotproduk digunakan untuk menggantikan *euclidean distance* dengan persamaan berikut:

$$d_j = \sum_{i=0}^{n-1} \sqrt{x_i \times w_{ij}} \quad (3.1)$$

Keterangan: x_i : input w_{ij} : bobot
 d_j : Jarak



Gambar 2.3 Modifikasi JST OM

3.3 Rancangan Aplikasi

Untuk membuat aplikasi pengenalan aksara jawa/hanacaraka diperlukan beberapa proses atau tahapan. Dalam penelitian ini, proses yang akan dilakukan meliputi, *image processing*, *feature extraction*, *training* dan *recognize*.

3.3.1 Image processing

Image processing adalah proses awal dalam pembuatan aplikasi pengenalan aksara jawa/hanacaraka. *Image processing* memiliki peranan penting sebagai pemroses gambar hasil guratan dengan *mouse*. Dengan *image processing*, gambar akan mudah digunakan untuk dilakukan proses ekstraksi ciri. Dalam penelitian ini, metode *imageprocessing* yang digunakan adalah segmentasi, deteksi tepi dan *scalling*

3.3.2 Feature Extraction

Fungsi dari *feature extraction* adalah ekstraksi untuk mendapatkan ciri dari sebuah karakter. Pada penelitian ini ekstraksi ciri yang digunakan adalah *zoning* dan ekstraksi ciri titik.

1) Zoning

Zoning adalah ekstraksi ciri dengan membagi atau mempartisi gambar (hasil dari *image processing*) ke dalam $N \times M$ wilayah. Dari setiap wilayah, *feature* diekstraksi untuk membentuk *feature* vektor. Setiap *pixel* yang dilalui oleh citra (*pixel* berwarna hitam) akan diberi nilai 1, sedangkan *pixel* yang tidak dilalui oleh citra (*pixel* berwarna putih) akan diberi nilai 0.

3.3.3 Training

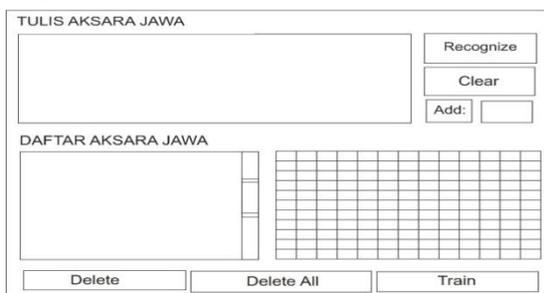
Training merupakan proses pembelajaran dengan tujuan untuk mencari nilai bobot akhir yang akan digunakan untuk meng-*cluster* data sampel.

3.3.4 Recognize

Recognize merupakan proses mengenali pola dari tulisan tangan aksara jawa/hanacaraka yang di-*input*-kan. Proses *recognize* ini melibatkan proses *training*.

3.4 Desain Interface

Desain *interface* (antar-muka) memiliki peran penting dalam tahapan perancangan pembuatan suatu program. Perancangan *interface* ini diharapkan dapat membantu user dalam pengimplementasian ke dalam pembuatan program yang sebenarnya.

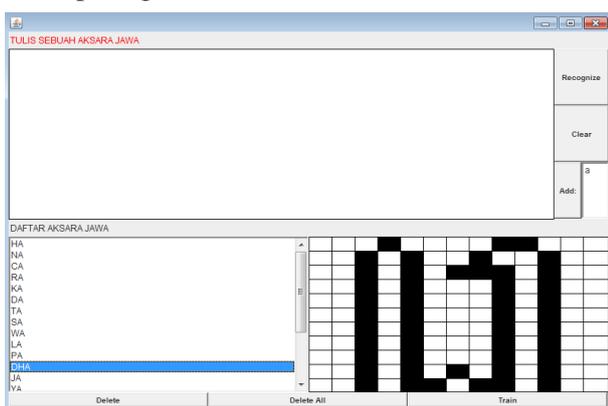


Gambar 3.7 Desain Interface

IV. IMPLEMENTASI DAN PEMBAHASAN

4.1 Implementasi Sistem

Hasil implementasi *interface* aplikasi dapat dilihat pada gambar dibawah ini:



Gambar 4.1 Implementasi Interface

4.1.1. Deskripsi Umum Aplikasi

Aplikasi yang digunakan sebagai implementasi penelitian ini adalah aplikasi pengenalan tulisan tangan aksara jawa/hanacaraka. Aplikasi ini memiliki fungsi sebagai berikut:

- 1) Menerima *input*-an berupa tulisan tangan aksara jawa/hanacaraka yang tunggal secara *online*
- 2) Melakukan proses *feature extraction* dengan metode *zoning* ukuran matriks 13 x 11.
- 3) Melakukan proses *training* yang akan digunakan untuk proses *recognize*.
- 4) Melakukan proses *recognize* (pengenalan) aksara jawa/hanacaraka.

4.1.2 Input-an Aplikasi

Agar Aplikasi dapat mengenali aksara jawa/hanacaraka dengan baik, diperlukan data yang sesuai dengan syarat berikut:

- 1) Data *input*-an berupa tulisan tangan aksara jawa/hanacaraka *recognition* yang ditulis langsung secara *online* pada *panel drawing* (tulisan tangan harus tunggal dan sesuai dengan kaidah penulisan aksara jawa/hanacaraka)
- 2) Untuk proses *recognize*, cukup dengan meng-*input*-kan tulisan tangan aksara jawa/hanacaraka.

4.1.3 Output Aplikasi

Output dari aplikasi ini adalah sebagai berikut:

- 1) Menampilkan huruf aksara jawa/hancaraka hasil dari *input*an dan *recognize*
- 2) Menampilkan hasil *feature extraction* metode *zoning* dengan ukuran 11x11.

4.2 Cara Kerja Aplikasi

Cara kerja aplikasi dibagi menjadi dua, yaitu *training* dan *recognition*. Proses *training* adalah proses dimana aplikasi melatih semua data yang akan digunakan pada proses *recognize*. Langkah-langkah proses *training* adalah sebagai berikut:

- 1) Meng-*input*-kan aksara jawa/hanacaraka tunggal secara *online* pada *panel drawing*
- 2) Aplikasi akan melakukan proses *image processing* yang terdiri dari segmentasi, deteksi tepi dan skaling.
- 3) Aplikasi akan melakukan proses *feature extraction* dengan menggunakan metode *zoning* dengan ukuran 13 x 11.
- 4) Aplikasi akan melakukan proses *training* dengan melibatkan data hasil *feature extraction*.

Sedangkan proses *recognize* adalah proses untuk mengenali aksara jawa/hanacaraka. Proses *recognize* melibatkan proses *training*, karena tanpa proses *training*, *recognize* tidak dapat dilakukan dengan baik. Langkah-langkah proses *recognition* hampir sama dengan proses *training*, yaitu:

- 1) Meng-*input*-kan aksara jawa/*hanacaraka* tunggal secara *online* pada *panel drawing*.
- 2) Aplikasi akan melakukan proses *image processing* yang terdiri dari segmentasi, deteksi tepi dan skaling.
- 3) Aplikasi akan melakukan proses *feature extraction* dengan menggunakan metode *zoning* dengan ukuran 13 x 11.
- 4) Aplikasi akan melakukan proses *recognize* dengan melibatkan proses *training*.

4.3. Pengujian

Setelah melakukan analisis, perancangan dan implementasi, selanjutnya aplikasi ini akan dilakukan pengujian. Tujuan dari pengujian ini adalah untuk mengetahui tingkat keakuratan aplikasi dalam mengenali aksara jawa/*hanacaraka*.

Pada proses pengujian akan dilakukan 2 skenario pengujian, yaitu:

- 1) Pengujian untuk memeriksa kesesuaian antara hasil implementasi perangkat lunak dengan hasil analisis dan perancangan aplikasi sebelumnya.
- 2) Pengujian untuk mengetahui akurasi pengenalan aksara jawa/*hanacaraka* menggunakan *Self Organizing Maps* (SOM).

4.3.1 Pengujian Implementasi Aplikasi

Pada proses pengujian ini, akan dilakukan evaluasi terhadap aplikasi “*HanacarakRecognition*”, apakah aplikasi sudah sesuai dengan analisis kebutuhan dan perancangannya.

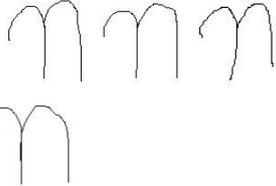
4.3.2. Pengujian Akurasi Aplikasi

Untuk mengetahui tingkat kemampuan aplikasi ini dalam mengenali aksara jawa/*hanacaraka*, maka akan dilakukan proses pengujian. Dalam pengujian ini, masing huruf akan diuji sebanyak 7 kali. Pengujian akan dilakukan dengan men-*training* 100 data (masing-masing aksara akan ditraining sebanyak 5 kali).

Tabel 4.2 Citra aksara untuk *training*

Citra	Arti Aksara	Variasi yang diberikan	Hasil Uji
	HA	tulisan putus	Salah
	HA	Utuh	Benar
	HA	Tidak sesuai dengan kaidah penulisan aksara jawa	Salah

Tabel 4.3 Contoh hasil pengujian aplikasi

Aksara	Citra
GA	
RA	
PA	

Dari 140 data yang diujikan (hasil pengujian selengkapnya terdapat pada lampiran), terlihat bahwa aksara yang dapat dikenali sebanyak 103 buah, hal ini berarti secara umum aplikasi ini telah mampu mengenali aksara jawa/*hanacaraka* dengan presentase 73,57 %.

Presentasi akurasi tersebut mengidentifikasi bahwa algoritma *Self Organizing Maps* (SOM) masih kurang stabil untuk mengenali berbagai jenis variasi aksara jawa/*hanacaraka* (khususnya kemiringan, coretan putus dan aksara yang tidak sesuai dengan kaidah penulisan). Tetapi untuk variasi coretan dengan bentuk yang sama, algoritma SOM dianggap cukup stabil.

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan pembahasan diatas, maka dapat diambil kesimpulan sebagai berikut:

- 1) Algoritma *Self Organizing Maps* (SOM) dapat digunakan dalam proses pengenalan pola, dalam penelitian ini, digunakan untuk pengenalan pola aksara jawa/*hanacaraka*.
- 2) Presentase keberhasilan hasil pengujian pada aplikasi ini adalah 73,57 %, nilai ini didapat dari hasil pengujian 140 aksara jawa/*hanacaraka*.

5.2 Saran

Hasil penelitian ini masih memiliki banyak keterbatasan dan belum menunjukkan hasil yang diharapkan. Oleh karena itu, untuk meningkatkan kualitas hasil yang diperoleh, disarankan untuk melakukan langkah-langkah baru yang lebih baik untuk penelitian selanjutnya. Beberapa saran yang dapat diberikan penulis adalah sebagai berikut:

- 1) Proses pengenalan yang dilakukan hanya sampai pada aksara tunggal. Untuk pengembangan selanjutnya akan lebih baik lagi jika proses pengenalan sampai pada kata atau kalimat.
- 2) Agar hasil pengenalan lebih baik lagi, data training yang digunakan harus lebih banyak dan bervariasi.
- 3) Agar pengenalan tulisan berhasil dengan baik, diharapkan menggunakan layar sentuh (*touchscreen*) atau dengan *pen tool* (PDA)
- 4) Karena metode JST SOM kurang akurat dalam mengenali sebuah pola aksara jawa/*hanacaraka*, maka sebaiknya dicoba metode JST lain yang lebih akurat.

pengenal-pola-otomatis-dan-cara-kerjanya.html
(diakses pada 18 Oktober 2014)

DAFTAR PUSTAKA

- [1]. Heuton, Jeff. *Handwriting Recognition* (online) <http://www.heatonresearch.com/articles/42/page1.html> (diakses Pada 10 November 2013 sampai 2 Januari 2014)
- [2]. Kusumadewi, Sri dan Hartati, Sri. 2010. *Neuro-fuzzy Integrasi Sistem Fuzzy dan Jaringan Syaraf*. Bandung
- [3]. Mubarok. *Pengenalan Tulisan Tangan Aksara Sunda Menggunakan Kohonen Neural Network* (online) http://abstrak.digilib.upi.edu/Direktori/SKRIPSI/FPMIPA/ILMU_KOMPUTER/PENGENALAN_TULISAN_TANGAN_AKSARA_SUNDA_MENGGUNAKAN_KOHONEN_NEURAL_NETWORK.pdf (diakses pada 10 Oktober 2014)
- [4]. Roziqien, 2009. *Sejarah Aksara Jawa Legenda Hanacaraka*. (online) <http://roziqien.blogspot.com/2009/04/sejarah-aksara-jawa-legenda-hanacaraka.html> (diakses pada 10 Oktober 2014)
- [5]. Siong, Ang wie. Resmana. 1999. *Pengenalan Citra Objek Sederhana Dengan Menggunakan Metode Jaringan Saraf Tiruan som*. Prosiding Seminar Nasional 1 Kecerdasan Komputasional Universitas Indonesia, 20-21 Juli 1999. Surabaya: Universitas Kristen Petra. (online), http://fportfolio.petra.ac.id/user_files/91-024/som-image-recog.doc (diakses pada 20 November 2014)
- [6]. Vincent Nathanail, Stephen. 2011. *OCR (Automatic Pattern Recognition) Sebagai Pengenal Pola Otomatis dan Cara Kerjanya*. (online) <http://intracom solutions.com/blog/ocr-automatic-pattern-recognition-sebagai->