PENERAPAN METODE *LEARNING VECTOR QUANTIZATION* DAN *DIRECTIONAL ELEMENT FEATURE* UNTUK PENGENALAN TULISAN TANGAN

David Suwandi[1], Galih Hermawan[2]

[1,2] Universities Komputer Indonesia
Jalan Dipatiukur No. 112 Bandung, Jawa Barat 40132
E-mail : suwandidavid7@gmail.com[1], galih.hermawan@email.unikom.ac.id[2]

ABSTRAK

Pengenalan tulisan tangan merupakan bagian *Optical Character Recognition* (OCR) yaitu sebuah proses untuk mengenali karakter dari suatu citra digital. Citra dari tulisan tangan akan diubah kedalam bentuk digital, yang nantinya akan mampu dikenali sistem. Metode klasifikasi digunakan untuk proses *learning* dalam pengenalan tulisan tangan. Berdasarkan penelitian sebelumnya Metode *Learning Vector Quantization* (LVQ) mampu mengenali citra dengan tingkat keakuratan 61,07% dan untuk data yang belum dilatih sebesar 48,17%. Proses pengenalan tulisan tangan melalui berbagai proses diantaranya proses *grayscale, savuola threshold*, segmentasi dan ekstraksi ciri yang kemudian hasil dari ekstraksi akan masuk pada metode LVQ untuk dilakukan pelatihan dan pengujian. Proses Metode LVQ digabungkan dengan metode ekstraksi ciri *Directional Element Feature*, metode DEF digunakan untuk mendapatkan ciri dari sebuah citra yaitu dengan melihat perbedaan kontur dan tanpa proses skeletonisasi. Setelah melakukan pengujian sistem dengan metode *Black Box* dan dihitung tingkat akurasi dapat ditarik kesimpulan bahwa program berfungsi dengan benar. Pada pengenalan tulisan tangan memiliki tingkat akurasi 73.80% dan untuk citra tulisan tangan yang belum dilakukan pembelajaran sebesar 36.99%.

Kata kunci : Pengenalan Tulisan Tangan, *Learning Vector Quantization (LVQ), Directional Element Feature (DEF), Jaringan Saraf Tiruan, Ekstraksi Ciri, Kecerdasan Buatan.*

1. PENDAHULUAN

Pengenalan citra pada huruf atau sering disebut dengan *Optical Character Recognition (OCR)* merupakan konversi dari tulisan tangan ke dalam bentuk file citra yang diubah. Tulisan tangan tersebut nantinya di-scan menggunakan scanner untuk menghasilkan suatu citra [1]. Teknik OCR biasa diterapkan pada berbagai tulisan tangan, salah satunya membahas tulisan aksara latin maupun non latin.

Bahasa merupakan salah satu sarana komunikasi manusia. Oleh karena itu untuk berkomunikasi satu sama lain dibutuhkan Bahasa yang dapat dipahami sebagai alat penghubung, hal ini disebabkan oleh adanya perbedaan suku, adat, budaya dan lain sebagainya [2]. Maka untuk menjembatani perbedaan tersebut dibuatlah Bahasa yang sama dan untuk memahami suatu Bahasa maka diperlukan pemahaman terhadap aksaranya [2]. Terkadang suatu aksara sulit untuk dikenali ketika dituliskan, dikarenakan setiap Bahasa mempunyai aksara yang karakteristiknya berbeda. Maka dengan berkembang teknologi informasi dan banyak penelitian yang dilakukan untuk melakukan pengenalan suatu aksara yang ditulis tangan pada Bahasa tertentu dengan menggunakan terknologi komputer yaitu dengan pendekatan menggunakan Teknik *OCR* untuk mengukur tingkat akurasi suatu sistem agar dapat menganali suatu aksara pada tulisan tangan.

Pada penelitian sebelumnya *Learning Vector Quantization* mampu mengenali citra dengan nilai keakuratan rata-rata sebasar 61,07 dan rata-rata keakuratan hasil citra yang belum dilakukan pembelajaran sebesar 48,17% [3]. Penelitian tersebut belum memakai ekstraksi ciri sebagai tambahan dari proses *preprocessing* yang digunakan untuk melihat sebuah ciri dari sebuah karakter.

Directional Element Feature (DEF) merupakan metode ekstrasi ciri yang telah banyak digunakan dan terbukti memberikan hasil yang baik, pada penelitian sebelumnya DEF digabungkan dengan SVM dalam pengenalan tulisan tangan aksara jawa didapat hasil keakuratan sebesar 93.6% [4]. Metode DEF pada OCR yang melihat perbedaan kontur dan tanpa mengalami proses *skeletonizing* [4].

Metode Learning Vector Quantization akan dikombinasikan dengan metode ekstrasi ciri *Directional Element Feature* untuk melihat keakurasian pengenalan tulisan tangan.

Berdasarkan pemaparan yang telah dibahas, penelitian ini dimaksudkan untuk menerapkan metode *Learning Vector Quantization* dan *Directional Element Feature* pada pengenalan tulisan tangan.

Pembahasan tersebut akan ditulis ke dalam skripsi dengan judul "**PENERAPAN METODE**

***LEARNING VECTOR QUANTIZATION* DAN *DIRECTIONAL ELEMENT FEATURE* UNTUK PENGENALAN TULISAN TANGAN"**.

2. ISI PENELITAN

2.1 Tinjauan Pustaka

2.1.1 Citra

Citra adalah gambar dua dimensi yang dihasilkan dari gambar analog dua dimensi yang kontinu menjadi gambar diskrit melalui proses sampling [5]. citra dibedakan menjadi dua jenis yaitu citra analog dan citra digital. Citra merupakan keluaran dari suatu sistem perekam data yang bersifat optik, analog maupun digital. Perekaman data citra dibagi menjadi 2 yaitu Citra Analog dan Citra Digital.

2.1.2 Citra Analog

Citra analog adalah citra yang bersifat kontinu, seperti gambar pada monitor televisi, foto sinar-X, foto yang tercetak di kertas foto, lukisan, pemandangan alam, hasil CT scan, gambar-gambar yang terekam pada pita kaset, dan lain sebagainya [5].

2.1.3 Citra Digital

Citra digital adalah gambar dua dimensi yang bisa ditampilkan pada layar komputer sebagai himpunan/diskrit nilai digital yang disebut *pixel/picture elements*. Dalam tinjauan matematis, citra merupakan fungsi kontinu dari intensitas cahata pada bidang dua dimensi [5]. Suatu citra dapat didefinisikan sebagai fungsi f(x,y) berukuran M baris dan N kolom, dengan x dan y adalah koordinat spasial, dan amplitudo f di titik koordinat (x,y) dianamakan intensitas atau tingkat keabuan dari citra pada titik tersebut. Apabila nilai x,y dan nilai amplitudo f secara keseluruhan berhingga (finite) dan bernilai diskrit maka dapat dikatakan bahwa citra tersebut adalah citra digital [6].

2.1.4 *Pattern Recognition*

Pattern Recognition (PR) atau pengenalan Pola, merupakan salah satu cabang AI (*artifical intelegence)* [6], yang mempelajari pembuatan sebuah sistem untuk dapat mengenali suatu pola tertentu. Pola dalam hal ini merujuk pada pengertian suatu komposisi stimulus penginderaan yang kompleks yang dapat dikenali oleh manusia sebagai pengamat sebagai suatu kelompok objek. *Pattern Recognition* merupakan proses pengenalan kembali terhadap pola yang pernah dikenal. Oleh karena itu, Ketika gambar yang sama pernah dilihat sebelmunya, maka proses tersebut akan mengenali sebagai sesuatu yang sebelumnya telah dialami.

2.1.5 *Preprocessing*

Citra grayscale menangani gradasi warna hitam dan putih, yang menghasilkan efek warna abu-abu, warna gambar dinyatakan dengan intensitas. Dalam hal ini, intensitas berkisar antara 0 sampai dengan 255. Nilai 0 menyatakan hitam dan nilai 255 menyatakan putih [6]. Berikut adalah rumus konversi citra berwarna (RGB) menjadi grayscale.

$$I = (0.2989 * R) + (0.5870 * G) + (0.1141 * B) \quad (1)$$

Sauvola threshold merupakan metode *threshold* yang mampu mendeteksi citra tulisan tangan dengan sangat baik, dengan cara menghitung ambang menggunakan rentang nilai dinamis dari nilai standar deviasi citra *grayscale* [7].

$$T(x,y) = m(m,y) * (1 + k * \left(\frac{s(x,y)}{R} - 1\right)) \quad (2)$$

$$m(x,y) = \frac{\sum_{i=min}^{i=max}\sum_{j=min}^{j=max} img(i,j)}{i*j} \quad (3)$$

$$s(x,y) = \frac{\sum_{i=min}^{i=max}\sum_{j=min}^{j=max}(img(i,j)-m(x,y))^2}{(i*j)-1} \quad (4)$$

Segmentasi citra adalah teknik untuk membagi citra menjadi beberapa daerah atau region dimana setiap daerah memiliki kemiripan atribut [7]. Menurut Gonzales dan Wintz bahwa segmentasi adalah proses pembagian sebuah citra ke dalam sejumlah bagian atau objek. Segmentasi merupakan suatu bagian yang sangat penting dalam analisis citra secara otomatis, sebab pada prosedur ini obyek yang diinginkan akan disadap untuk proses selanjutnya.

2.1.5 *Directional Element Feature*

Metode DEF pada OCR yang melihat perbedaan kontur dan tanpa mengalami proses *skeletonizing* [4].

Berikut adalah tahapan-tahapan yang ada dalam ekstraksi DEF [4]:

1. *Resize*

Resize digunakan untuk merubah ukuran *pixel* sesuai dengan kebutuhan dalam melakukan proses ektraksi ciri. Hal ini dilakukan agar semua citra karakter dari hasil segmentasi mempunyai ukuran yang sama dan menghasilkan ciri yang baik pada proses *dot orientation.*

2. *Dot Orientation*

Directional Element Feature adalah pencarian nilai *feature* berdasarkan label arah dari sebuah pixel. Pada metode ini setiap pixel f*oreground*(piksel karakter) pada citra digital akan diberikan label (*feature* arah) dengan cara menelusuri tetangga dari masing-masing pixel *foreground* searah perputaran jarum jam dimana arah yang digunakan terdiri dari 4 arah.

3. *Vector Construction*

Tahapan ini untuk menghitung nilai dari setiap *pixel* ketetanggaan yang sebelumnya telah melalui proses *dot orientation*, awalnya citra akan dibagi menjadi $N_1 \times N_1$ area yang saling overlap sebanyak $N_1/2$ *pixel* terhadap area yang bertetanggaan.

Selanjutnya area, setiap area dibagi kembali menjadi 4 *sub-area,* yaitu A, B, C dan D, yang eksklusif satu sama lainnya. Setelah diperoleh jumlah pixel di setiap *sub-area*, akan dikalikan masing-masing jumlah elemen ketetanggan disetiap *sub-area* dengan bobot(*w*) pada *sub-area* tersebut dan dijumlahkan dari setiap elemen di *sub-area* dengan rumus sebagai berikut[4]:

$$x_j = w_A x_j + w_B x_j + w_C x_j + w_D x_j \quad (5)$$
$$j = 1,\ldots,4$$

j merupakan *pixel* ketetanggaan

Selanjutnya vektor ciri DEF yang diperolh dengan menggabungkan jumlah elemen disetiap area menjadi satu kolom vektor *V*.

$$V = [X_1^1, X_2^1, X_3^1, X_4^1, \dots, X_1^N, X_2^N, X_3^N, X_4^N] \quad (6)$$

Dimana *N* merupakan jumlah area dari setiap karakter. Vektor *V* inilah yang nantinya jadi input untuk proses *learning* dan *testing*.

2.1.4 *Learning Vector Quantization*

Banyak metode yang bisa digunakan untuk pengenalan citra tulisan tangan, antara lain metode Artificial Neural Network, Backpropagation, Hough Transform, dan Support Vector Machine. Dari beberapa metode yang disebutkan metode Learning Vector Quantization masih belum banyak digunakan. Learning Vector Quantization(LVQ) merupakan jaringan syaraf dengan tipe arsitektur jaringan lapis-tunggal umpan-maju (Single Layer Feedforward) yang terdiri atas unit masukan dan unit keluaran. Suatu lapisan kompetitif akan secara otomatis belajar untuk mengklasifikasikan vektor-vektor masukan [2]. Kelas-kelas yang didapatkan sebagai hasil dari lapisan kompetitifini hanya tergantung pada jarak antara vektor vektor masukan. Jika 2 vektor masukan mendekati sama, maka lapisan kompetitif akan meletakkan kedua vektor masukan tersebut ke dalam kelas yang sama [2].

$$Cj = \sqrt{\sum_{i=1}^{n}(X_i - W_j)^2} \quad (7)$$

C_j = nilai jarak Euclidean
X_i = nilai vector input
W_j = nilai vector bobot

Selain itu LVQ selalu memperbaharui bobotnya sesuai jumlah maksimal epoch yang ditetapkan. Epoch yaitu satu siklus pelatihan yang melibatkan semua pola. Persamaan 2 merupakan rumus bobot LVQ yang digunakan untuk memperbaharui nilai bobot:

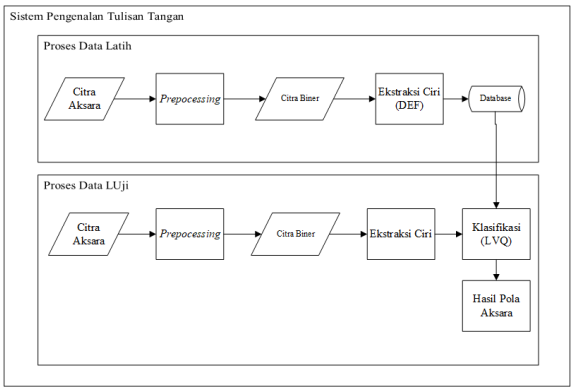
$$\overrightarrow{W_m} = \overrightarrow{W_m} + \alpha \,.\, (\overrightarrow{X} - \overrightarrow{W_m}) \quad (8)$$

W_m = bobot
α = *learning rate*
$\overrightarrow{X}$ = input

2.2 Analisa dan Implementasi

2.2.1 Analisis Proses

Proses yang dilakukan dalam aplikasi ini dibagi menjadi beberapa bagian dimana setiap proses memiliki peranan masing-masing dalam melakukan pengenalan tulisan tangan dan evaluasi jawaban ujian. Proses yang dilakukan pada gambar 1.



Gambar 1. Gambaran Umum Sistem

2.2.2 *Preprocessing*

1. *Grayscale*: Tahapan ini merupakan merubah citra RGB yang didapatkan dari hasil scan, kemudian akan diubah menjadi citra keabuan yang tadinya 24-*bit* menjadi 8-*bit* lebih mudah pada proses selanjutnya.
2. *Savuola Threshold:* Pada tahapan ini citra yang sebelumnya telah melalui proses *grayscale* akan diubah menjadi citra hitam putih, citra yang akan dihitung sebagai pada savuola berukuran 11x11 *pixel*, konstanta R : 128, dan K = 0,5, kemudian akan dicari *Mean*, Standar devisi, dan *Threshold*. Berikut hasil dari proses threshold.

Tabel 1. Hasil Perhitungan *Savuola Threshold*

X/Y	0	1	…	532
0	255	255	…	255
1	255	255	…	255
2	255	255	…	255
3	255	255	…	255
…	…	…	…	…
956	255	255	…	255

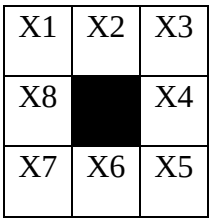
3. Binerisasi: Pada tahapan ini hasil penghitungandari proses *threshold* akan diubah menjadi 0 dan 1, dimana 0 merupakan warna putih dan 1 merupakan warna hitam.
4. Segmentasi: Tahapan citra akan dibagi menjadi per karakter sebelum dilakukan proses ekstraksi ciri, segementasi dilakukan pada setiap baris pada citra terlebih dahulu yang kemudian dilanjutkan dengan pemotongan kolom, pertama-tama ukuran citra akan dihitung kemudian menelusuri *pixel* citra sampai menemukan pixel hitam yang akan menjadi label awal pemotongan, kemudian menelusuri kembali hingga satu baris *pixel* jika tidak ditemukan akan menjadi label akhir pemotongan. Proses yang sama dilakukan pada pemotongan kolom.

2.2.3 Ekstraksi Ciri

Pada proses ekstrasi ciri *directional element feature* bekerja dengan cara mengambil ciri sebuah

citra berdasarkan kontur. Berikut adalah tahapan yang ada pada DEF:

1. *Resize:* Pada tahapan ini citra hasil segmentasi akan diubah ukurannya menjadi 15x15 *pixel*, agar semua citra hasil segmentasi mempunyai ukuran yang sama.
2. *Dot Orientation:* Tahapan ini akan menelusuri *pixel* ketetanggaan dari setiap *pixel foreground* searah dengan jarum jam, seperti pada gambar berikut:



Gambar 2. Tabel Arah DEF

Ketika pixel hitam didapatkan maka akan dilakukan pengecakan dengan cara melihat tetangga pixel tersebut. Perhatikan tabel 3., ketika pixel hitam didapatkan maka pengecekan dilakukan secara berurutan dari X1-X8. Pengecekan berhenti ketika menemukan X yang merupakan pixel hitam lainnya. Maka ubahlah nilai pixel hitam yang tadinya 1 menjadi nilai arah berdasarkan aturan dibawah ini:

- Jika posisi X1 atau X5 maka nilainya 5
- Jika posisi X2 atau X6 maka nilainya 2
- Jika posisi X3 atau X7 maka nilainya 3
- Jika posisi X4 atau X8 maka nilainya 4

3. *Vector Construction:* Setelah proses *dot orientation* akan dihitung nilai dari setiap area yang nantinya akan jadi vektor ciri. Proses yang dilakukan adalah membagi citra menjadi 2x2 area dengan ukuran masing-masing 8x8 *pixel* yang saling *overlap*. Setelah itu setiap area tadi akan dibagi kembali menjadi 4 *sub*-area yaitu A, B, C, D yang saling ekslusif satu sama lainnya, dengan besar *sub-area* A=2x2 *pixel*, B=4x4 *pixel*, C=6x6 *pixel*, dan D=8x8 *pixel*. Maka vektor ciri yang akan dihasilkan dari proses ini sebanyak 16 vektor ciri yang nantinya akan masuk pada proses klasifikasi LVQ. Setelah melakukan penghitungan setiap pixel pada setiap *sub*-area dengan persamaan 5, maka didapatkan nilai dari setiap *sub*-area, seperti berikut perhitungan pada karakter A:
   - x_1 = (0.0) + (5.2) + (22.7) + (37.11) = 0 + 10 + 154 + 407
   - x_2 = (0.0) + (5.1) + (22.5) + (40.9) = 0 + 5 + 110 + 360
   - x_3 = (9.3) + (19.5) + (26.7) + (31.8) = 27 + 95 + 182 + 248
   - x_4 = (4.2) + (11.5) + (16.7) + (18.8) = 8 + 55 + 112 + 144

   Selanjutnya hasil dari penghitungan akan dimasukkan ke dalam persamaan 6. Berikut adalah hasil vector ciri DEF karakter A:
   - *V* = [0, 10, 154, 407, 0, 5, 110, 360, 27, 95, 182, 248, 8, 55, 112, 144]

2.2.4 *Learning Vector Quantizaton*

Proses pembelajaran dimana sistem mempelajari sejumlah data yang ada dan menghasilkan bobot. Data *training* yang digunakan adalah vektor ciri yang didapatkan dari proses sebelumnya. Berikut tabel 2 merupakan data bobot dan masukan.

Tabel 2. Inisialisasi Bobot dan Data Masukan

Huruf	Kelas	Vektor Ciri	Ket
A	1	[0, 10, 154, 407, 0, 5, 110, 360, 27, 95, 182, 248, 8, 55, 112, 144]	Bobot (W_1)
B	2	[0, 0, 182, 1782, 0, 44, 168, 840, 0, 3, 297, 1554, 36, 95, 248, 1260]	Bobot (W_2)
C	3	[0, 3, 108, 507, 0, 0, 20, 162, 0, 64, 144, 256, 18, 120, 304, 650]	Bobot (W_3)
A	1	[0, 10, 96, 560, 0, 0, 95, 480, 20, 80, 126, 360, 0, 36, 133, 444]	Masukan (X_1)

Kemudian akan ditentukan paramater untuk proses pembelajaran, pada penelitian ini ditentukan paramaternya yaitu, Maksimum perulangan sebanyak 1000, eror minimum 0,01, *learning rate* 0,05, dan pengurangan rasio 0,1. Setelah itu dihitung dengan persamaan 7, untuk melihat jarak terdekat dan berikut adalah hasil dari perhitungan tersebut.

Tabel 3. Hasil Jarak Terdekat

Vektor	Nilai
C_1	385,0233759
C_2	1944,918764
C_3	447,888379

Dari hasil tersebut dapat dilihat bahwa jarak (C) terkecil diperoleh pada perhitungan C_1, maka bisa disimpulkan bahwa target X_1 adalah kelas 1 dan kelas bobot ke-1 juga merupakan kelas 1 berarti target sesuai dengan hasil perhitungan.

Setelah perhitungan jarak terdekat dilakukan maka akan dilakukan perubahan bobot dengan persamaan 8 karena target sesuai dengan kelas bobot. Maka berikut ini merupakan bobot akhir yang nantinya digunakan pada proses pengenalan LVQ.

Tabel 4. Hasil Bobot Akhir

Huruf	Kelas	Vektor Ciri	Ket
A	1	[0. 10. 151,1. 414,65. 0. 4,75. 109,25. 366. 26,65. 94,25. 179,2. 253,6. 7,6. 54,05. 113,05. 159]	Bobot (W_1)

Huruf	Kelas	Vektor Ciri	Ket
B	2	[0. 0,5. 177,7. 1720,9. 0. 41,8. 164,35. 822. 1. 6,85. 288,45. 1494,3. 34,2. 92,05. 242,25. 1219,2]	Bobot (W_2)
C	3	[0. 3,35. 107,4. 509,65. 0. 0. 23,75. 177,9. 1. 64,8. 143,1. 261,2. 17,1. 115,8. 295,45. 639,7]	Bobot (W_3)

Proses pengenalan akan menggunakan bobot akhir yang didapat pada proses pengenalan LVQ.

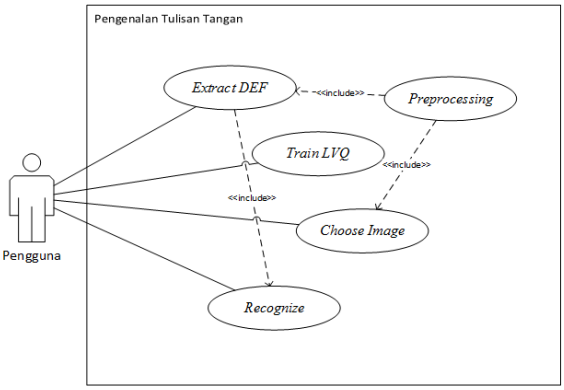
Tabel 5. Hasil Perhitungan Jarak Terdekat

Vektor	Nilai
C_1	365,7722071
C_2	1847,672826
C_3	425.49396

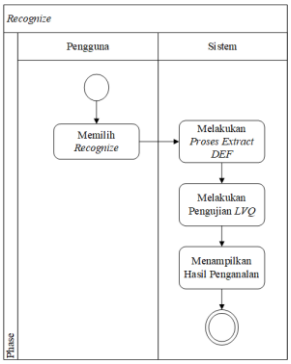
Dilihat dari hasil perhitungan maka jarak terkecil dari proses pengenalan adalah bobot ke-1(C_1) dan merupakan kelas 1 yaitu citra huruf A.

2.2.5 Analisis Kebutuhan Fungsional

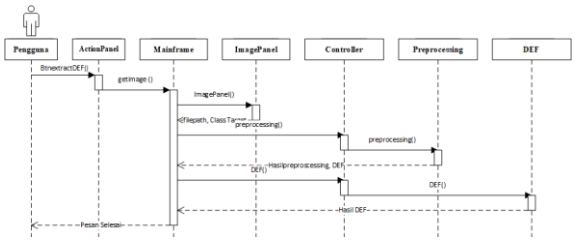
Analisis kebutuhan fungsional yang dibahas berupa usecase, activity, class dan squence diagram dapat dilihat pada Gambar 3, 4 dan gambar 5.



Gambar 3. Usecase Diagram Sistem Pengenalan Tulisan Tangan



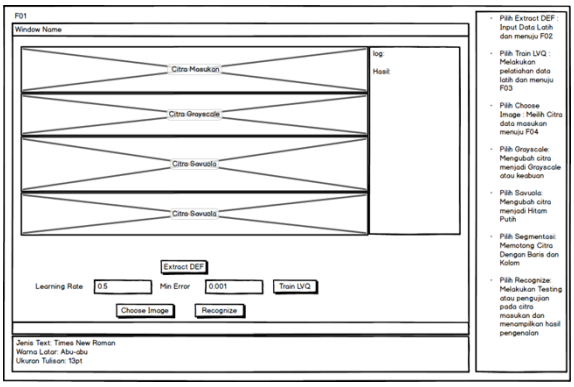
Gambar 4. Activity Diagram Proses *Recognize*



Gambar 5. Class Diagram Proses *Extract* DEF

2.2.6 Perancangan Sistem

Adapun tahapan yang dilakukan dalam perancangan sistem ini membahas mengenai tujuan perancangan menu serta antarmuka yang dibuat pada sistem pengenalan tulisan tangan ini. Perancangan sistem dapat dilihat pada Gambar 6, 7, 8, 9, dan Gambar 10.



Gambar 6. Perancangan Antarmuka Program

2.2.7 Implementasi

Tahap implementasi sistem merupakan tahap selanjutnya dari tahapan antarmuka. Pada tahapan ini sistem akan diuji coba serta akan memperlihatkan hasil. Sehingga dapat mengahasilakan suatu solusi. Pada tahapan ini yang akan dibahas yaitu implementasi antarmuka dan hasil pengujian dari perangkat keras pada sistem.

2.2.8 Implementasi Perangkat Keras

Implementasi perangkat keras menjelaskan perangkat keras yang digunakan untuk mengimplementasikan sistem.

Tabel 6. Implementasi Perangkat Keras

Komponen	Spesifikasi
Processsor	2.4 GHz Intel Core i5
Memory	8 GB DDR3
VGA	Intel Iris 1536 MB
Harddisk	250 GB
Layar Monitor	Resolusi 1366x768 *pixel*

2.2.9 Implementasi Perangkat Lunak

Perangkat lunak yang digunakan untuk sistem pengenalan tulisan tangan adalalah sebagai berikut.

Tabel 7. Implementasi Perangkat Lunak

No	Perangkat Keras	Spesifikasi
1	Sistem Operasi	OS X Yosemite – 10.10.5
2	Bahasa Pemrograman	Java
3	Perangkat Pendukung	Java Development Kit
4	Aplikasi Tools	Netbeans IDE 1.8.0_31
5	Permodelan	Microsoft Visio 2013

2.2.10 Implementasi Antarmuka

Berikut adalah implementasi antarmuka dari sistem pengenalan tulisan

1. Tampilan Menu Utama

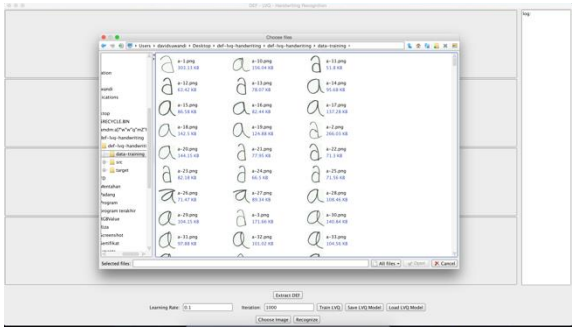
Pada gambar 7 merupakan Tampilan yang akan muncul pertama kali pada saat program dijalankan didalamnya terdapat beberapa tombol.



Gambar 7. Tampilan Utama

2. Tampilan *Extract* DEF

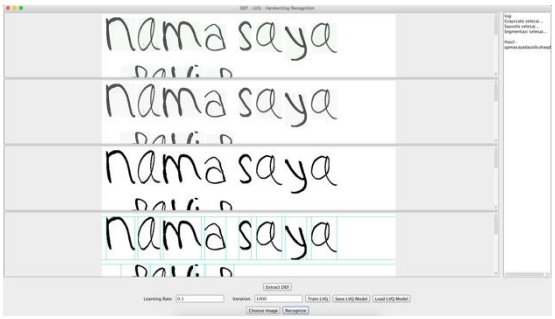
Gambar 8 merupakan tampilan pada saat pengguna melakukan proses *Extract* DEF yaitu melakukan proses *preprocessing* dan ekstraksi ciri pada citra latih.



Gambar 8. Tampilan Menu *Extract* DEF

3. Tampilan *Recognize*

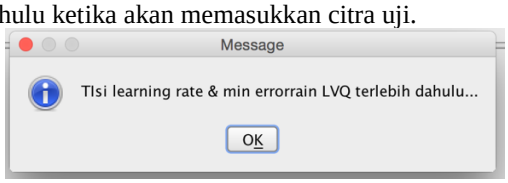
Menu *Recognize* digunakan untuk melakukan proses pengennalan pada citra uji yang nantinya akan menampilkan hasil dari pengenalan.



Gambar 9. Implementasi Antarmuka Pengenalan

4. Tampilan Pesan Kesalahan

Pesan kesalahan akan muncul apabila pengguna tidak melakukan proses pelatihan terlebih dahulu ketika akan memasukkan citra uji.



Gambar 10. Tampilan Pesan Kesalahan

2.2.11 Pengujian Akurasi

Pengujian akurasi akan dihitung nilai ketepatan atau kecocokan dari hasil aplikasi pengenalan tulisan tangan. Pengujian dilakukan setelah data training dilakukan proses pelatihan. Pengujian akurasi akan dilakukan terhadap citra yang sebelumnya telah dilatih dan citra yang belum dilatih. Berikut adalah rumus untuk menghitung nilai akurasi.

$$Akurasi = \frac{jumlah\ karakter\ sama}{jumlah\ seluruh\ karakter}\ X\ 100\% \qquad (2.10)$$

Tabel 8. Hasil Pangujian ke-1

No	Nama Citra	Akurasi Pengenalan
1	Uji-1.png	78.57%
2	Uji-2.png	82.85%
3	Uji-3.png	85.00%
4	Uji-4.png	65.90%
5	Uji-5.png	50.00%
6	Uji-6.png	65.21%
7	Uji-7.png	64.28%
8	Uji-8.png	60.52%
9	Uji-9.png	65.21%
10	Uji-10.png	73.33%
Hasil Akurasi		69.08%

Tabel 9. Hasil Pangujian ke-2

No	Nama Citra	Akurasi Pengenalan
1	Uji-1.png	78.57%
2	Uji-2.png	74.28%
3	Uji-3.png	100%
4	Uji-4.png	70.45%

No	Nama Citra	Akurasi Pengenalan
5	Uji-5.png	46.66%
6	Uji-6.png	73.91%
7	Uji-7.png	75.00%
8	Uji-8.png	68.42%
9	Uji-9.png	43.47%
10	Uji-10.png	80.00%
Hasil Akurasi		63.25%

Tabel 10. Hasil Pangujian ke-3

No	Nama Citra	Akurasi Pengenalan
1	Uji-1.png	82.14%
2	Uji-2.png	74.28%
3	Uji-3.png	80.00%
4	Uji-4.png	84.09%
5	Uji-5.png	53.33%
6	Uji-6.png	69.56%
7	Uji-7.png	82.14%
8	Uji-8.png	73.68%
9	Uji-9.png	52.17%
10	Uji-10.png	86.66%
Hasil Akurasi		73.80%

Pada pengujian diatas didapatkan hasil akurasi yang terbaik yaitu pada pengujian ke-3 dengan akurasi sebesar 73.80% dengan parameter *learning rate* 0,010 dan maksimum perulangan 2000. Untuk hasil yang memiliki tingkat akurasi yang terendah yaitu pada pengujian ke-2 dengan akurasi 63.25% pada parameter *learning rate* 0,05 dan maksimum perulangan 2000. Hasil tersebut didapatkan dari data uji yang sebelumnya sudah dilatih.

Berikut ini adalah hasil pengujian akurasi dengan data yang sebelumnya belum pernah diuji.

Tabel 11. Hasil Pangujian ke-1

No	Nama Citra	Akurasi Pengenalan
1	Uji-1.png	32.14%
2	Uji-2.png	20.00%
3	Uji-3.png	40.00%
4	Uji-4.png	20.45%
5	Uji-5.png	30.00%
6	Uji-6.png	26.08%
7	Uji-7.png	21.42%
8	Uji-8.png	34.21%
9	Uji-9.png	34.78%
10	Uji-10.png	50.00%
Hasil Akurasi		30.90%

Tabel 12. Hasil Pangujian ke-2

No	Nama Citra	Akurasi Pengenalan
1	Uji-1.png	39.28%
2	Uji-2.png	28.57%
3	Uji-3.png	45.00%
4	Uji-4.png	31.81%
5	Uji-5.png	30.00%
6	Uji-6.png	30.43%
7	Uji-7.png	35.71%
8	Uji-8.png	50.00%
9	Uji-9.png	39.13%
10	Uji-10.png	40.00%
Hasil Akurasi		36.99%

Tabel 13. Hasil Pangujian ke-3

No	Nama Citra	Akurasi Pengenalan
1	Uji-1.png	42.85%
2	Uji-2.png	34.28%
3	Uji-3.png	15.00%
4	Uji-4.png	31.81%
5	Uji-5.png	26.66%
6	Uji-6.png	26.08%
7	Uji-7.png	28.57%
8	Uji-8.png	23.68%
9	Uji-9.png	30.43%
10	Uji-10.png	46.66%
Hasil Akurasi		30.60%

Pada pengujian diatas didapatkan hasil akurasi yang terbaik yaitu pada pengujian ke-2 dengan akurasi sebesar 36.99% dengan parameter *learning rate* 0,05 dan maksimum perulangan 2000. Untuk hasil yang memiliki tingkat akurasi yang terendah yaitu pada pengujian ke-3 dengan akurasi 30.60% pada parameter *learning rate* 0,010 dan maksimum perulangan 2000. Hasil tersebut didapatkan dari data uji yang belum pernah dilatih.

3. PENUTUP

3.1 Kesimpulan

Berdasarkan hasil hasil analisis, perancangan, dan pengujian perangkat lunak dan perangkat keras yang telah dibuat sebagai sistem pengenalan tulisan tangan dengan metode *Learning Vector Quantization* dan *Directional Element Feature* maka dapat disimpulkan bahwa akurasi terbaik untuk citra yang sebelumnya telah dilatih sebesar 73.80% dan untuk pengenalan pada citra yang belum dilatih sebesar 36.99%. Akurasi ini didapatkan dan dipengaruhi oleh parameter pelatihannya, data training dan juga data uji yang digunakan dalam keselurahan proses.

3.1 Saran

Aplikasi Pengenalan tulisan tangan masih mempunyai banyak kekurangan, maka dari itu perlu untuk dilakukan pengembangan untuk bisa menghasilkan akurasi yang lebih baik, berikut adalah beberapa saran yang perlu dipertimbangkan:

1. Dibutuhkan metode segmentasi citra tulisan tangan untuk tulisan yang saling berdempet atau menempel.
2. Dibutuhkan metode ekstraksi fitur yang lebih baik untuk gambar seperti *Modify Directional Feature* (MDF), *Wavelet Transform.*

DAFTAR PUSTAKA

[1] Krishnapriya, M, dkk, *Optical Character Recognition For Handwritten Text, International Journal of Contemporary Research in Computer*

Science and Technology, India, Vol.1, Issue.2, 2015.

[2] Rohwana Ulir, Isa Irawan M, "Pengenalan Tulisan Tangan Huruf Latin Bersambung secara *Real Time* Menggunakan Algoritma Learning Vector Quantization", Jurnal Sains dan Seni Pomits, Vol. 2, No.1, 2013.

[3] Fachrul Kurniawan, Hani Nurhayati, "Simulasi Pengenalan Tulisan Menggunakan LVQ (*Learning Vector Quantization*)", Jurnal Ilmu Komputer dan Teknologi Informasi, Vol. 4 No. 4, 2011.

[4] Auli Husni Nurul A.I., Mahmud Dwi Sulistiyo, Retno Novi Dayawati. "Pengenalan Akasara Jawa Tulisan Tangan Menggunakan *Directional Element Feature* dan *Multi Class Support Vector Machine*". Dalam: Konferensi Nasional Teknologi Informasi dan Aplikasinya, Palembang, 13 September. 2014.

[5] Muhammad Ferderico S. Implementasi Metode *Support Vector Machine (SVM)* dan *Zoning* pada pengolahan citra untuk pengenalan citra tulisan tangan, Skripsi. Jurusan Teknik Informatika. Universitas Komputer Indonesia. 2017.

[6] R. P. Rajashekararadhya SV, "*Novel Zone Based Feature Extraction Algorithm For Handwritten Numeral Recognition Of Four Indian Scripts*," *Digital Technology Journal,* vol. 2, pp. 41-51, 2009.

[7] K Max R. Kumaseh, Luther Latumakulita, Nelson Nainggolan, "Segmentasi Citra Digital Ikan Menggunakan Metode *Thresholding*", Jurnal Ilmiah Sains, Vol. 13, No. 1, 2013.