

Pengenalan Tanda Tangan Menggunakan Learning Vector Quantization dan Ekstraksi Fitur Local Binary Pattern

Signature Recognition Using Learning Vector Quantization and Local Binary Pattern

Erwin Yudi Hidayat¹, Muhammad Farhan Radiffananda²

^{1,2}Fakultas Ilmu Komputer, Universitas Dian Nuswantoro, Semarang
e-mail: ¹erwin@dsn.dinus.ac.id, ²mfarhanradiffananda@gmail.com

Abstrak

Tanda tangan merupakan salah satu biometrik pada karakteristik perilaku yang digunakan untuk mengenali seseorang sebagai sistem identifikasi. Fungsi tanda tangan adalah untuk menentukan kebenaran ciri-ciri dari penandatanganan atau diperlukan untuk memastikan identifikasi seseorang yang valid. Meskipun unik, banyak terjadi kasus tanda tangan yang disalahgunakan dengan cara dipalsukan. Tidak mudah mengenali tanda tangan yang palsu dengan tanda tangan asli. Untuk itu, diperlukan mekanisme pengenalan tanda tangan ini dengan suatu algoritma. Proses pengenalan tanda tangan bisa dilakukan secara statik (offline) maupun dinamik (online). Penelitian ini dilakukan berdasarkan pengenalan secara offline, yang dikenal lebih sulit daripada pengenalan secara online. Algoritma yang melibatkan meliputi Learning Vector Quantization, deteksi tepi Sobel, dan ekstraksi fitur Local Binary Pattern untuk mengidentifikasi tanda tangan. Hasil penelitian menunjukkan, jumlah data citra, iterasi, dan learning rate mempengaruhi akurasi dan waktu proses identifikasi. Dari percobaan yang dilakukan pada parameter yang berbeda-beda, akurasi yang didapat adalah 68% pada data latih dan pada data uji sebesar 54,6%.

Kata kunci—identifikasi, Learning Vector Quantization, tanda tangan, pengenalan pola

Abstract

A signature is one of the biometrics on behavioral characteristics used to recognize a person as an identification system. The function of the signature is to determine the correctness of the characteristics of the signatory or is needed to ensure valid identification of a person. Although unique, there are many cases of signatures that are misused by falsification. It is not easy to recognize a fake signature with an original signature. For this reason, this signature recognition mechanism is needed with an algorithm. The signature recognition process can be done either statically (offline) or dynamically (online). This research is based on offline recognition, which is known to be more difficult than online recognition. The algorithms involved include Learning Vector Quantization, Sobel edge detection, and Local Binary Pattern feature extraction to identify signatures. The results showed, the amount of image data, iteration, and learning rate affect the accuracy and time of the identification process. From experiments conducted on different parameters, the accuracy obtained was 68% in the training data and in the test data at 54.6%.

Keywords—identification, Learning Vector Quantization, signatures, pattern recognition

1. PENDAHULUAN

Perkembangan ilmu pada dunia komputer sangatlah cepat. Hal yang menandai ini adalah biometrik yang merambah pada dunia komputer. Biometrik dapat digunakan untuk mengenali seseorang berdasarkan fitur tertentu yang dimiliki seseorang [1]. Biometrik dikenal sebagai identifikasi dengan menerapkan karakteristik alami dari manusia, yang meliputi karakteristik perilaku dan juga karakteristik fisiologis. Karakteristik perilaku seperti halnya dengan pola ucapan, pola mengetik, tanda tangan [2], telapak tangan [3] [4], dan intonasi suara [5].

Tanda tangan adalah salah satu karakteristik perilaku yang digunakan untuk mengenali seseorang sebagai sistem identifikasi [6]. Tanda tangan atau dikenal dengan paraf yaitu tulisan tangan yang diberikan gaya tulisan tertentu dari nama orang lain sebagai sebuah bukti dari suatu identitas yang ditulis pada suatu dokumen. Fungsinya adalah untuk menentukan kebenaran ciri-ciri dari penandatanganan atau dapat memastikan identifikasi. Pemakaian paraf tersebut bersifat penting yang mengakibatkan kemungkinan terjadinya tindak kejahatan seperti halnya pemalsuan paraf pada seseorang yang tidak bertanggung jawab [7]. Pada banyak kasus, seringkali ada pemalsuan tanda tangan untuk sebuah dokumen penting. Melakukan verifikasi apakah sebuah tanda tangan adalah asli atau tiruan merupakan sebuah tantangan [8].

Terdapat dua metode utama yang terkait dalam proses penelitian ini, yaitu pendekatan secara statik (*offline*) dan dinamik (*online*). Pendekatan statik atau dalam jaringan (*offline*) yaitu pendekatan yang menangkap suatu citra tanda tangan pada kertas yang kemudian diakuisi dengan *scanner* atau kamera digital hingga setelah itu data diubah menjadi format dalam bentuk digital yang akan diproses lebih lanjut. Sedangkan pendekatan secara dinamik atau luar jaringan (*online*) menggunakan alat elektronik atau penandatanganan modern dalam proses penandatanganan. Dalam verifikasi tanda tangan *offline* gambar tanda tangan dianggap sebagai fungsi 2D diskrit. Pengenalan tanda tangan *offline* lebih sulit daripada *online* karena informasi dinamis tidak tersedia dan sulit untuk memulihkannya dari gambar *offline* [9]. Metode luar jaringan atau luring (*offline*) memiliki kelebihan, yaitu dalam hal biaya dan presisi [10].

Pada deteksi tepi, algoritma Sobel sangatlah banyak digunakan [11]. Algoritma ini secara ilmiah terbukti dapat mempertahankan kehalusan kualitas sebuah citra, yang tidak hanya mangkus membuang derau, tapi juga sekaligus mampu menyajikan detail informasi secara simultan [12].

Penelitian terkait pengenalan pola tanda tangan pernah dilakukan oleh beberapa peneliti. Misalnya komparasi kecepatan pengenalan citra tanda tangan metode 2D-PCA dan 2D-LDA [13], yang menunjukkan bahwa 2D-PCA lebih cepat dan akurat dibanding 2D-LDA. *Euclidean Distance* dan *PCA* untuk identifikasi tanda tangan seseorang berfungsi sangat baik dan mendapatkan hasil akurasi tinggi tetapi untuk penggunaan *Euclidean Distance* dan *PCA* dalam proses identifikasi tanda tangan tiruan belum cukup baik [14]. Penelitian oleh [15] tentang identifikasi tanda tangan menggunakan *Back Propagation* dan Wavelet Haar menyimpulkan bahwa dari percobaan yang dilakukan, akurasi yang diperoleh mencapai hampir 96% dan 100%.

Berkenaan dengan metode *Learning Vector Quantization* (LVQ) sendiri, [16] membuktikan bahwa metode ini mampu membaca tanda tangan palsu dengan tingkat kesalahan sebesar 30%. Sedangkan pada [17], LVQ terbukti dapat mempermudah pendeteksian keaslian tanda tangan.

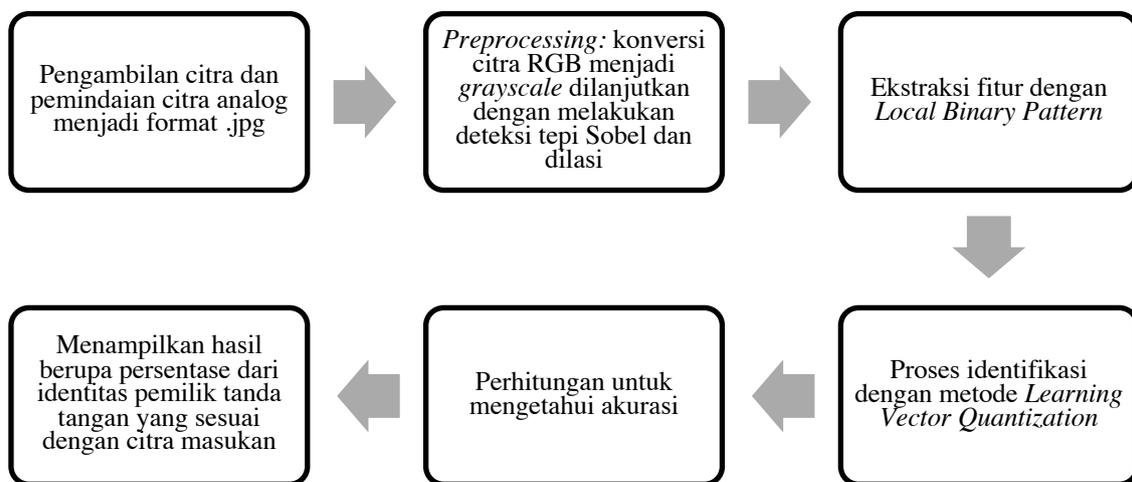
Dalam tahap pengenalan pola, tanda tangan misalnya, diperlukan ekstraksi fitur untuk mengambil ciri objek, sebagai gambaran dari objek yang ditinjau. Beberapa metode yang dapat digunakan untuk mengekstraksi fitur antara lain *Local Ternary Pattern* (LTP) [18], *MIFT: A Moment-Based Local Feature Extraction Algorithm* [19], dan *Local Binary Pattern* (LBP) [20]. Di antara metode-metode tersebut, LBP termasuk yang paling efisien dalam pengenalan suatu pola obyek [21], termasuk varian dari LBP [22] tersebut.

2. METODE PENELITIAN

2.1 Pengambilan data citra

Pengambilan data citra diambil dengan cara meminta sampel tanda tangan secara langsung kepada responden pada kertas. Jumlah sampel yang digunakan yaitu 200 citra dimana tiap responden diminta memberikan sampel tanda tangan sebanyak 8 kali yang terdiri dari 5 tanda tangan asli yang nantinya akan digunakan sebagai *training set* dan 3 tanda tangan yang akan menjadi *testing set*.

Tanda tangan analog yang diminta diubah menjadi citra digital dalam format .jpg dengan pemindaian secara teratur. Proses ini diperlihatkan pada Gambar 2. Adapun secara keseluruhan, metode yang dikembangkan dalam penelitian ini disajikan pada Gambar 1, yang dijelaskan dengan beberapa langkah di bawahnya.



Gambar 1 Metode yang dikembangkan



Gambar 2 Pengumpulan data dan pemindaian citra analog menjadi digital

2.2 Preprocessing Data

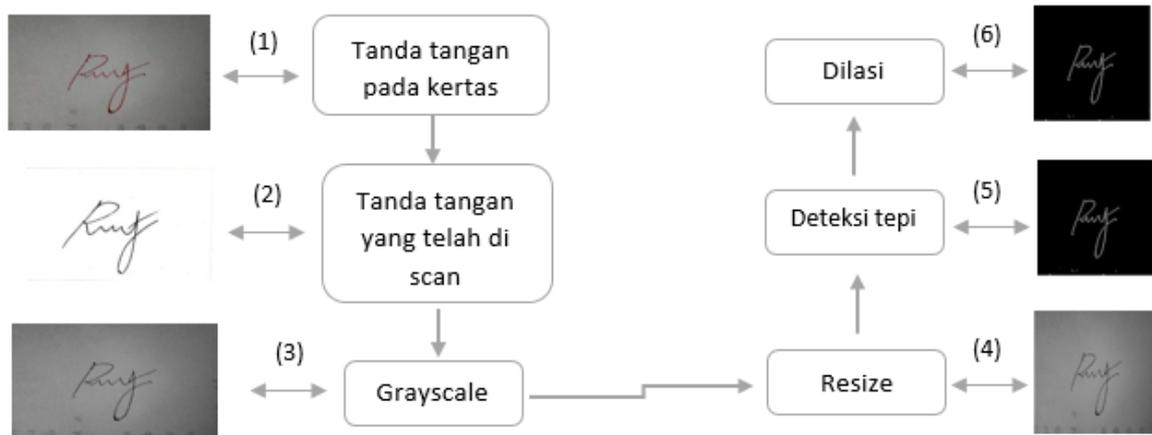
Tahap *preprocessing* secara lengkap disajikan pada Gambar 4, antara lain terdiri dari:

- Melakukan proses konversi citra hasil RGB hasil pemindaian menjadi *grayscale* agar derau yang terdapat pada citra latih berkurang. Pengubahan RGB menjadi *grayscale* dilakukan agar citra yang berwarna menjadi citra dengan aras keabuan menggunakan persamaan:

$$\text{Grayscale} = 0,21 * R + 0,72 * G + 0,07 * B \tag{1}$$



Gambar 3 Konversi citra RGB menjadi *grayscale*



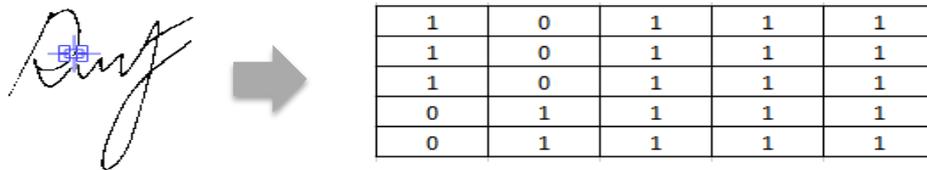
Gambar 4 Tahapan *preprocessing* citra

- b) *Resize* atau pemotongan pada bagian yang tidak diperlukan dengan tujuan menyederhanakan ukuran menjadi 200x200 agar objek yang akan diproses lebih fokus dan agar semua citra mempunyai ukuran yang sama.
- c) *Thresholding* dilakukan dengan menggunakan nilai *threshold (T)*. *Threshold* sendiri didapatkan dengan menetapkan kuantisasi 256 menjadi 2 nilai jika $x < 128 = 0$, dan jika tidak maka $x = 1$. Proses ini nantinya akan mengubah seluruh piksel pada citra bila diatas nilai *threshold* maka akan berubah menjadi 1 dan sebaliknya akan menjadi 0.
- d) *Edge Detection* dengan menggunakan metode Sobel. Tujuan dilakukan deteksi tepi adalah untuk mengubah citra *grayscale* menjadi biner dan meningkatkan garis batas suatu daerah atau objek dengan jelas. Kelebihan dari metode Sobel adalah kemampuan untuk mengurangi derau sebelum melakukan perhitungan deteksi tepi.



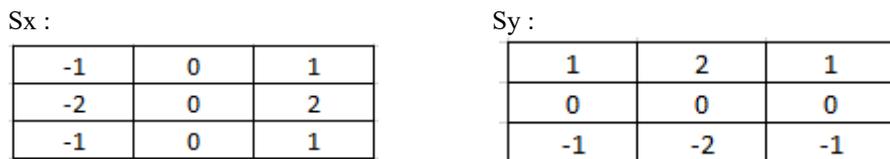
Gambar 5 Proses deteksi tepi citra

Dari contoh di atas, diambil matriks 5x5 untuk perhitungan Sobel, misalnya:



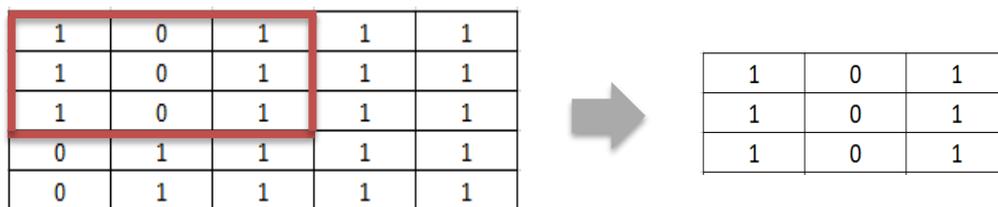
Gambar 6 Proses deteksi tepi citra menggunakan Sobel

Kernel yang digunakan pada Sobel dapat digambarkan sebagai berikut:



Gambar 7 Kernel pada metode Sobel

Berhubung kernel Sobel merupakan matriks 3x3 maka diambil matriks 3x3 bagian kiri atas terlebih dahulu.



Gambar 8 Ilustrasi operasi Sobel iterasi pertama

$$Sx_1 : 1(-1) + 1(-2) + 1(-1) + 1(1) + 1(2) + 1(1) = 0$$

$$Sy_1 : 1(1) + 0(2) + 1(1) + 1(-1) + 0(-2) + 1(-1) = 0$$

$$M1 = \sqrt{(0)^2 + (0)^2} = 0 + 0 = 0$$

Hasil dari operasi Sobel sampai iterasi ke sembilan adalah:

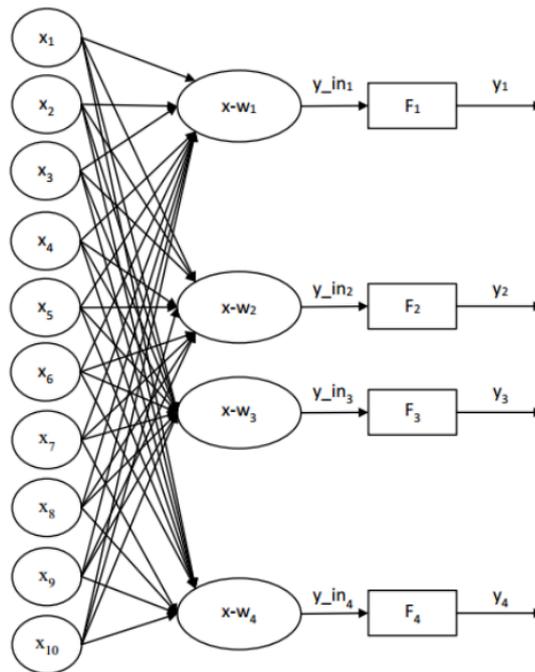
1	0	1	1	1
1	0	4	0	1
1	2	4	0	1
0	4	2	0	1
0	1	1	1	1

Gambar 9 Hasil perhitungan operasi Sobel

- e) Dilasi dengan tujuan menebalkan citra pada biner dan memperbaiki objek yang terputus dikarenakan hasil dari pengambilan citra yang terganggu oleh derau, kerusakan objek fisik yang dijadikan citra digital, atau karena jeleknya resolusi. Penebalan itu sendiri dikontrol oleh bentuk strel yang digunakan.

Learning Vector Quantization terdiri dari input, lapisan Kohonen (lapisan tersembunyi atau *hidden layer*) dan output seperti yang terlihat pada tiap output mempunyai kelas yang telah diketahui. Pemrosesan yang terjadi pada tiap neuron adalah mencari jarak antara suatu vektor input terhadap bobot yang bersangkutan (w_1 dan w_2). Bobot w_1 yaitu vektor yang menghubungkan tiap neuron pada lapisan input terhadap neuron pertama pada lapisan output. Pada w_2 yaitu vektor bobot yang menghubungkan tiap neuron pada lapisan input pada neuron kedua dalam lapisan output. Kemudian pada F_1 akan memetakan y_{in1} ke $y = 1$ bila $|x-w_1| < |x-w_2|$ dan $y = 0$ bila $|x-w_1| > |x-w_2|$.

Algoritma *LVQ* bergantung pada hasil jarak antara vektor inputannya, maksimal iterasi (epoh), error minimum (Eps), dan learning rate (α). Perhitungan jarak antara vektor inputan diambil tiap fitur citra uji yang kemudian akan dibandingkan dengan 125 citra latih. Data latih yang digunakan pada penelitian ini disajikan pada Tabel 1.



Gambar 12 Arsitektur jaringan LVQ [23]

Tabel 1 Data latih hasil perhitungan LBP

No	vektor input	Target
1	0,0012 ; 0,0001 ; 0 ; 0,0003	1
2	0,0027 ; 0,0005 ; 0 ; 0,0007	1
3	0,0063 ; 0,0013 ; 0 ; 0,0009	1
4	0,00000384 ; 0,00000267 ; 0 ; 0,0000018	1
5	0,000796 ; 0,0001 ; 0 ; 0,0000719	1
....
125	0,00011 ; 0,000058 ; 0 ; 0,00019	25

Pada Tabel 1 terdapat data latih sebanyak 125 citra, yang mana terdapat 25 target dikarenakan tiap 5 citra akan di inialisasikan kedalam 1 target. Tiap citra diambil 4 sampel nilai yang telah melewati proses ekstraksi fitur dengan LBP.

Metode pembelajaran LVQ dikerjakan dengan algoritma berikut:

- a) Menetapkan masing-masing kelas output, menentukan bobot (w), maksimum epoh (MaxEpoh), error minimum yang diharapkan (Eps), dan learning rate α .

- b) Memasukkan data input (x) dan target berupa kelas (T).
- c) Menetapkan kondisi awal :

$$\text{Epoh} = 0$$

$$\text{Err} = 1$$
- d) Lakukan selama : ($\text{epoh} < \text{MaxEpoh}$) dan ($\alpha > \text{Eps}$)
- e) $\text{Epoh} = \text{epoh} + 1$
 Ubah bobot w_j dengan cara :
 Jika $T = c_j$, maka $w_j(\text{baru}) = w_j(\text{lama}) + \alpha (x - w_j(\text{lama}))$ (3)
 Jika $T \neq c_j$, maka $w_j(\text{baru}) = w_j(\text{lama}) - \alpha (x - w_j(\text{lama}))$ (4)
- f) Kurangi learning rate α

Keterangan :

x = Vektor pelatihan atau vektor masukan

T = Kategori atau kelas tempat untuk vektor pembelajaran

w_j = Vektor bobot untuk *output* ke- j ($w_{1j}, \dots, w_{ij}, \dots, w_{nj}$)

c_j = Kategori atau kelas yang ditunjukkan oleh output ke- j

$|x - w_j|$ = Jarak Ecludian antara vektor input dan (vektor bobot untuk keluaran ke- j).

Berikut contoh perhitungan dengan menggunakan 3 *data training*:

Tabel 2 Contoh *data training* yang akan dilatih

No	vektor input	class
1	0,0055 ; 0,0020; 0 ; 0,0052	1
2	0,0038 ; 0,0042; 0 ; 0,0024	2
3	0,0059 ; 0,0012; 0 ; 0,0062	train

Diambil beberapa citra latih untuk dijadikan sebagai inisialisasi bobot (w) dan digunakan sebagai data yang akan dilatih. Untuk nilai-nilai yang ditentukan diawali dengan pelatihan nilai *learning rate* 0,007 dan dengan pengurangan α sebesar 0,1. Berikut adalah contoh perhitungan dari metode *LVQ* dengan iterasi (epoh) = 10.

- Iterasi ke1

Data ke 3 (baris ke-3 Tabel 2) sebagai input latih:

Jarak pada :

Bobot ke 1 =

$$\sqrt{(0,0055 - 0,0059)^2 + (0,0020 - 0,0012)^2 + (0 - 0)^2 + (0,0052 - 0,0062)^2}$$

$$= 0,0004 + 0,0008 + 0 + 0,001 = 0,0022$$

Bobot ke 2 =

$$\sqrt{(0,0038 - 0,0059)^2 + (0,0042 - 0,0012)^2 + (0 - 0)^2 + (0,0024 - 0,0062)^2}$$

$$= 0,0021 + 0,003 + 0 + 0,0038 = 0,0089$$

Jarak terkecil pada bobot ke 1 ($J=1$)

Target data ke 3 = 1 ($T=1$)

Karena $T = J$, maka perbaikan bobot : $w_j(\text{baru}) = w_j(\text{lama}) + \alpha (x - w_j(\text{lama}))$

Bobot ke 1 baru :

$$w_{11} = 0,0055 + 0,007 * (0,0059 - 0,0055) = 0,005528$$

$$w_{12} = 0,0055 + 0,007 * (0,0012 - 0,0020) = 0,001944$$

$$w_{13} = 0 + 0,007 * (0 - 0) = 0$$

$$w_{14} = 0,0052 + 0,007 * (0,0062 - 0,0052) = 0,005207$$

Setelah melakukan 1 iterasi maka nilai α berkurang sebagai berikut :

$$\alpha = \alpha - 0,1 * \alpha$$

$$= 0,007 - 0,1 * 0,007$$

$$= 0,0063$$

- Iterasi ke-2
Perhitungan dilakukan dengan cara yang sama, sehingga jarak terkecil ada pada bobot iterasi ke 10 dan input tersebut termasuk kedalam kelas 1.

2.5 Pengujian

Penelitian ini dilakukan pada perangkat dengan spesifikasi memori 4059MB dan Processor Intel Core i5-6200 CPU. Untuk mengukur persentase akurasi proses pelatihan menggunakan rumus:

$$\text{akurasi} = \frac{\text{jumlah data latih} - \text{jumlah data latih tidak dikenali}}{\text{jumlah data latih}} \times 100\% \quad (5)$$

dengan jumlah data menyatakan banyaknya data pada proses pelatihan atau pengujian, dan data tak dikenali menyatakan banyaknya data yang salah.

3. HASIL DAN PEMBAHASAN

3.1 Data Penelitian

Dalam penelitian ini, peneliti mengambil sampel data mentah yang berupa sampel tanda tangan dari 25 orang hingga total keseluruhan terkumpul 200 sampel tanda tangan. Citra sampel tanda tangan dipindaimenggunakan *scanner* agar data tersebut berubah menjadi citra. Setelah itu gambar dipisahkan ke dalam data *training* yang terdiri dari 5 gambar dan data *testing* yang terdiri dari 3 gambar. Dengan total data latih sebanyak 125 citra dan data uji sebanyak 75 citra. Hal ini bertujuan untuk memudahkan pengelompokkan data citra tanda tangan berdasarkan jenis data.



Gambar 13 Sampel tanda tangan sebagai dataset

3.2 Identifikasi *Learning Vector Quantization*

Identifikasi LVQ dilakukan dengan 2 data antara lain :

a) Data latih (*training*)

Dari total 200 citra yang digunakan 125 citra digunakan sebagai data latih. Pada pengujian dengan menggunakan data latih sebanyak 125 citra dan iterasi yang sama yaitu 100, diperoleh hasil seperti pada Tabel 3.

Tabel 3 Hasil pengujian data latih dengan perbedaan *learning rate*

Data latih	Iterasi	<i>Learning rate</i> (α)	Waktu	Akurasi (%)
125	100	0,001	1 menit 21 detik	52
125	100	0,003	1 menit 13 detik	60
125	100	0,005	1 menit 12 detik	58,40
125	100	0,007	1 menit 9 detik	64
125	100	0,01	1 menit 8 detik	63,20
125	100	0,03	1 menit 9 detik	60
125	100	0,05	1 menit 9 detik	52
125	100	0,07	5 menit 9 detik	8
125	100	0,1	1 menit 10 detik	4
125	100	0,5	1 menit 9 detik	4

Learning rate adalah salah satu parameter yang berpengaruh terhadap akurasi pengenalan tanda tangan menggunakan LVQ. Terbukti dengan perbedaan *learning rate* ini, akurasi yang diperoleh tidak selalu sama. Akurasi tertinggi didapatkan dengan menggunakan *learning rate* 0,007 yaitu sebesar 64% dengan waktu proses selama 1 menit 9 detik. Angka tersebut diperoleh dari:

$$\text{akurasi} = \frac{125 - 45}{125} \times 100\% = 64\%$$

Adapun akurasi terendah tercatat 4%, ketika *learning rate* berada pada nilai 0,1 dan 0,5. Selain berpengaruh terhadap akurasi, *learning rate* juga ikut serta mempengaruhi waktu yang diperlukan untuk LVQ mengenali tanda tangan, sekalipun data dan jumlah iterasi yang digunakan adalah sama.

Tabel 4 Hasil pengujian data latih dengan perbedaan iterasi yang digunakan

Data latih	Iterasi	<i>Learning rate</i> (α)	Waktu	Akurasi (%)
125	10	0,007	6 detik	26
125	20	0,007	13 detik	46
125	30	0,007	20 detik	50,4
125	40	0,007	27 detik	53
125	50	0,007	34 detik	56,8
125	60	0,007	52 detik	56,8
125	70	0,007	51 detik	61
125	80	0,007	1 menit 7 detik	59
125	90	0,007	1 menit 12 detik	59
125	100	0,007	1 menit 19 detik	62

Berdasarkan hasil Tabel 4 di atas bahwa dengan menggunakan data latih yang sama dengan jumlah 125 citra dan *learning rate* yang sama yaitu 0,007 dan iterasi yang berbeda-beda menghasilkan hasil akurasi dan juga waktu proses yang berbeda-beda. Akurasi tertinggi didapatkan dengan menggunakan iterasi sebanyak 100 kali dan menghasilkan akurasi sebesar 62% dengan waktu proses selama 1 menit 19 detik. Akurasi pada data latih dengan data latih sebanyak 125, *learning rate* 0,007, dan iterasi 100:

$$\text{akurasi} = \frac{125 - 47}{125} \times 100\% = 62\%$$

Sedangkan akurasi terendah diperoleh dengan iterasi hanya 10, dengan waktu proses 6 detik. Hal ini menunjukkan bahwa, semakin tinggi jumlah iterasi, kecenderungan akurasi akan meningkat. Imbasnya adalah waktu yang diperlukan untuk menyelesaikan proses ini menjadi sebanding, yaitu semakin banyak iterasi, waktu eksekusi semakin besar.

Tabel 5 menyajikan hasil pengujian data latih dengan perbedaan data yang digunakan. Dengan menggunakan data latih yang berbeda-beda dan menggunakan *learning rate* yang sama yaitu 0,007 dan iterasi yang sama sebanyak 100 kali, menghasilkan akurasi dan juga waktu proses yang berbeda-beda. Akurasi tertinggi didapatkan dengan menggunakan data latih sebanyak 125 citra dan menghasilkan akurasi sebesar 68% dengan waktu proses selama 1 menit 39 detik. Akurasi pada data latih dengan data latih sebanyak 125, *learning rate* 0,007, dan iterasi 100:

$$\text{akurasi} = \frac{125 - 40}{125} \times 100\% = 68\%$$

Di sisi lain, dengan pengaturan yang sama, akurasi terkecil dihasilkan dari jumlah data latih 100, yaitu 60%. Sedangkan waktu tercepat sebesar 1 menit 5 detik ditunjukkan oleh data latih 50, sementara waktu terlama ada pada kisaran 2 menit 8 detik, dengan data latih 200. Hal ini menunjukkan, semakin banyak data latih, maka semakin besar pula waktu yang diperlukan untuk menyelesaikan proses kerja.

Tabel 5 Hasil pengujian data latih dengan perbedaan data yang digunakan

Data latih	Iterasi	<i>Learning rate</i> (α)	Waktu	Akurasi (%)
50	100	0,007	1 menit 5 detik	62
75	100	0,007	1 menit 30	61
100	100	0,007	1 menit 26	60
125	100	0,007	1 menit 39 detik	68
150	100	0,007	2 menit 8 detik	64,6
175	100	0,007	2 menit 17 detik	62,9
200	100	0,007	2 menit 30 detik	64

b) Data uji (*testing*)

Dari beberapa hasil yang diperoleh pada implementasi LVQ menggunakan data latih di atas, dilakukan pula pengukuran akurasi menggunakan data uji (*testing*). Tabel 6 merangkum ketentuan pengujian yang dilaksanakan.

Tabel 6 Hasil pengujian data uji

Data latih	Data uji	Maksimal iterasi	<i>Learning rate</i> (α)	Akurasi (%)
125	25	100	0,007	60
125	50	100	0,007	56

125	75	100	0,007	54,6
-----	----	-----	-------	------

Berdasarkan hasil Tabel 6, ditunjukkan bahwa dengan menggunakan data latih sebanyak 125 citra dan menggunakan *learning rate* yaitu 0,007 dan iterasi yang sama sebanyak 100 kali dari proses data latih sebelumnya, menghasilkan hasil akurasi yang berbeda-beda. Didapatkan akurasi tertinggi dengan menggunakan data uji sebanyak 25 citra dan menghasilkan akurasi sebesar 60%. Sedangkan akurasi paling rendah pada angka 54,6% dihasilkan dengan data uji sebanyak 75.

$$\text{akurasi} = \frac{25 - 10}{25} \times 100\% = 60\%$$

Dari pengujian di atas dapat disimpulkan bahwa terdapat hal-hal yang dapat mempengaruhi hasil akurasi yaitu banyaknya iterasi. Banyaknya iterasi mempengaruhi hasil pada proses pelatihan. Hasil terbaik pada tabel tersebut dicapai dengan iterasi 100. Iterasi juga mempengaruhi waktu proses, semakin banyak iterasi maka akan semakin membutuhkan waktu yang lama dalam melakukan proses begitu juga jika semakin sedikit iterasinya maka waktu yang dibutuhkan saat proses semakin cepat.

4. KESIMPULAN

Tanda tangan adalah salah satu karakteristik perilaku yang digunakan untuk mengenali seseorang sebagai sistem identifikasi. Pengenalan tanda tangan menjadi bahasan menarik, karena maraknya pemalsuan tanda tangan untuk keperluan tertentu. Menggunakan algoritma LVQ, deteksi tepi Sobel, dan ekstraksi fitur LBP, pengenalan tanda tangan berhasil diterapkan. Hasil pengukuran menunjukkan, dengan data latih 125, iterasi 100, dan rentang *learning rate* 0,001-0,5, akurasi tertinggi didapatkan dengan menggunakan *learning rate* 0,007 yaitu sebesar 64% dengan waktu proses selama 1 menit 9 detik. Dengan data latih 125, *learning rate* 0,0007, dan iterasi antara 10-100, akurasi tertinggi 62% didapatkan dengan menggunakan iterasi sebanyak 100 dengan waktu proses selama 1 menit 19 detik. Adapun pada pengujian menggunakan iterasi 100, *learning rate* 0,007, dan data latih antara 50-200, akurasi tertinggi sebesar 68% didapatkan dengan menggunakan data latih sebanyak 125 citra dalam waktu 1 menit 39 detik. Sedangkan menggunakan data latih sebanyak 125 citra dan menggunakan *learning rate* yaitu 0,007 dan iterasi yang sama sebanyak 100 kali dari proses data latih sebelumnya, dihasilkan akurasi tertinggi dengan menggunakan data uji sebanyak 25 citra dan menghasilkan akurasi sebesar 60%.

Penelitian ini menunjukkan bahwa, banyaknya data latih dan data uji yang digunakan, jumlah iterasi, dan *learning rate*, mempengaruhi besar kecilnya akurasi dan waktu proses dalam pengenalan tanda tangan. Dari semua percobaan dan pengujian, akurasi yang diperoleh masih di bawah 70%. Hal ini diperoleh karena menentukan parameter-parameter dalam algoritma LVQ bukan persoalan yang mudah.

Meskipun hasil pengukuran semua akurasi tidak lebih dari 70%, bukan berarti eksperimen ini tidak berhasil. Justru eksperimen yang diperoleh dapat memberi masukan kepada peneliti lain yang tertarik untuk menggunakan algoritma LVQ, bahwa perlu studi lebih lanjut untuk menerapkannya. Selain itu, algoritma ini masih perlu pembuktian dengan penggunaan jumlah data, iterasi, dan *learning rate* yang lebih bervariasi.

5. SARAN

Untuk penelitian lebih lanjut pada identifikasi tanda tangan dengan LVQ, terdapat beberapa saran yaitu:

- a) Melakukan optimasi atau *parameter tuning* terhadap jumlah data latih dan uji, jumlah iterasi, serta *learning rate*, agar akurasi yang diperoleh bisa lebih tinggi

- b) Menambah data latih atau *training* agar semakin banyak data atau pola yang dilatih dengan berbagai bentuk maka akan meminimalisir kesalahan pengenalan pola saat pengujian.
- c) Mengombinasikan atau menggabungkan algoritma LVQ dan sejenisnya, dengan algoritma deteksi tepi dan ekstraksi fitur lainnya.

DAFTAR PUSTAKA

- [1] I. Kaur, N. Mann, Bhusan, B. Verma, and Gurbaj, "Biometric Authentication in Computer Security," *Int. J. Comput. Sci. Technol.*, vol. 7, no. 4, pp. 58–62, 2016.
- [2] J. Arifin and M. Z. Naf'an, "Verifikasi Tanda Tangan Asli Atau Palsu Berdasarkan Sifat Keacakan (Entropi)," *J. Infotel*, vol. 9, no. 1, pp. 130–135, 2017.
- [3] A. Stephen, R. R. Isnanto, and A. A. Zahra, "Ekstraksi Ciri Citra Telapak Tangan Menggunakan Gelombang Singkat Ortogonal pada Sistem Pengenalan Biometrik," *Transient - J. Ilm. Tek. Elektro*, vol. 4, no. 4, pp. 2–5, 2015.
- [4] A. M. Riandy, R. R. Isnanto, and A. A. Zahra, "Ekstraksi Ciri Citra Telapak Tangan dengan Alihragam Gelombang Singkat Haar Menggunakan Pengenalan Jarak Euclidean Pada Sistem Presensi," *Transient - J. Ilm. Tek. Elektro*, vol. 4, no. 1, pp. 12–16, 2015.
- [5] I. N. K. Wardana and I. G. Harsemadi, "Identifikasi Biometrik Intonasi Suara untuk Sistem Keamanan Berbasis Mikrokomputer," *J. Sist. Dan Inform.*, vol. 9, no. 1, pp. 29–39, 2014.
- [6] W. Fitriani, M. Z. Naf'an, and E. Usada, "Ekstraksi Fitur pada Citra Tanda Tangan Sebagai Ciri Identitas Pemiliknya Menggunakan Discrete Fourier Transform," in *Seminar Nasional Multi Disiplin Ilmu*, 2018, pp. 978–979.
- [7] G. Novandra, M. Z. Naf'an, and T. G. Laksana, "Perancangan Aplikasi Android Identifikasi Tanda Tangan Menggunakan Multi Layer Perceptron," *J. Ilm. Penelit. dan Pembelajaran Inform.*, vol. 3, no. 3, pp. 76–83, 2018.
- [8] O. A. Rosso, R. Ospina, and A. C. Frery, "Classification and Verification of Handwritten Signatures with Time Causal Information Theory Quantifiers," *PLoS One*, vol. 11, no. 12, pp. 1–19, 2016.
- [9] M. Arathi and A. Govardhan, "An Efficient Offline Signature Verification System," *Int. J. Mach. Learn. Comput.*, vol. 4, no. 6, pp. 533–537, 2014.
- [10] A. V. Bharadwaja, "Control of Brushless DC Motor with Direct Torque and Indirect Flux using SVPWM Technique," *Indian J. Sci. Technol.*, vol. 8, no. 20, pp. 507–515, 2015.
- [11] K. Zhang, Y. Zhang, P. Wang, Y. Tian, and J. Yang, "An improved sobel edge algorithm and FPGA implementation," in *International Congress of Information and Communication Technology (ICICT-2018)*, 2018, pp. 243–248.
- [12] Suyanto and J. Munte, "Implementation of Sobel Method to Detect The Seed Rubber Plant Leaves," in *International Conference on Mathematics, Science and Education*, 2017, pp. 1–5.
- [13] D. P. Pamungkas, E. Utami, and A. Amborowati, "Komparasi Pengenalan Citra Tanda Tangan dengan Metode 2D-PCA dan 2D-LDA," *Creat. Inf. Technol. J.*, vol. 2, no. 4, pp. 341–354, 2015.
- [14] R. Wulanningrum and R. K. Niswatin, "Rancang Bangun Aplikasi Identifikasi Tanda Tangan Menggunakan Ekstraksi Ciri PCA," in *Seminar Nasional Teknologi Informasi, Komunikasi dan Aplikasinya*, 2017, pp. 103–107.

- [15] R. A. Kumalasanti, Ernawati, and B. Y. Dwiandiyanta, "Identifikasi Tanda Tangan Statik Menggunakan Jaringan Syaraf Tiruan Backpropagation dan Wavelet Haar," in *Prosiding Simposium Nasional Rekayasa Aplikasi Perancangan dan Industri*, 2013, pp. 93–100.
- [16] I. Amelia, A. N. Hermana, and A. Pramana, "Verifikasi Tanda Tangan Dengan Edge Detection Dan Metode Learning Vector Quantization," *Multimed. Artif. Intell. Networkind Database J.*, vol. 1, no. 1, pp. 49–56, 2016.
- [17] A. Sefta and S. Hidayatulloh, "Verifikasi Citra Tanda Tangan Menggunakan Metode Prewitt dan Learning Vector Quantization," *J. Inform.*, vol. 5, no. 2, pp. 202–210, 2018.
- [18] S. A. Hassan and A. Naaz, "Face Recognition using Local Ternary Pattern," *Int. J. Sci. Res.*, vol. 4, no. 12, pp. 2115–2120, 2016.
- [19] H.-Z. Zhang, D.-W. Kim, T.-K. Kang, and M.-T. Lim, "MIFT: A Moment-Based Local Feature Extraction Algorithm," in *Applied Sciences*, 2019, vol. 9.
- [20] E. Prakasa, "Texture Feature Extraction by Using Local Binary Pattern," *INKOM J. Informatics, Control Syst. Comput.*, vol. 9, no. 2, pp. 45–48, 2016.
- [21] S. Yasmin and M. M. Rana, "Performance Study of Soft Local Binary Pattern over Local Binary Pattern under Noisy Images," *Int. J. Electr. Comput. Eng.*, vol. 6, no. 3, pp. 1161–1167, 2016.
- [22] C. Singh, E. Walia, and K. P. Kaur, "Color Texture Description with Novel Local Binary Patterns for Effective Image Retrieval," *Pattern Recognit.*, vol. 76, pp. 50–68, 2018.
- [23] M. Setya, A. Fauzi, B. Rahayudi, and C. Dewi, "Perbandingan Jaringan Saraf Tiruan LVQ dengan Backpropagation dalam Deteksi Dini Penyakit Jantung Koroner," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 2, pp. 1952–1960, 2019.