

## Verifikasi Tanda Tangan Dengan *Edge Detection* Dan Metode *Learning Vector Quantization*

Irma Amelia<sup>(1)</sup>, Asep Nana Hermana<sup>(1)</sup>, Andri Pramana<sup>(1)</sup>

<sup>(1)</sup>Jurusan Teknik Informatika, Fakultas Teknologi Industri  
Institut Teknologi Nasional Bandung  
Email: imameamel@gmail.com

### ABSTRAK

*Tanda tangan merupakan salah satu karakteristik yang dimiliki oleh setiap orang . Kasus pemalsuan tanda tangan cukup banyak terjadi<sup>[6]</sup>. Hal ini dikarenakan kurangnya sistem verifikasi dalam pengidentifikasian tanda tangan. Berdasarkan hal tersebut, dibangun aplikasi verifikasi tanda tangan. Aplikasi yang dibangun menggunakan metode edge detection dalam pengekstraksi ciri dan metode jaringan syaraf tiruan Learning Vector Quantization (LVQ) pada pencocokan pola tanda tangan. Aplikasi verifikasi tanda tangan diawali dengan pemindaian tanda tangan lalu diekstraksi ciri dan disimpan ke dalam basisdata yang digunakan sebagai data latih. Kemudian pengguna memindai tanda tangan uji dan sistem mencocokkan dengan data latih menggunakan metode LVQ. Hasil pencocokan dari aplikasi tersebut didapat persentase sebesar 70% tanda tangan asli yang terverifikasi dengan benar dan Tingkat kesalahan dalam system verifikasi tanda tangan dalam membaca tanda tangan palsu sebesar 30% sistem membaca dengan tanda tangan asli. Aplikasi ini dapat memverifikasi tanda tangan.*

**Kata kunci :** Tanda tangan, Verifikasi. Edge detection, LVQ, Jaringan syaraf tiruan

### ABSTRACT

*The signature is one of the characteristics possessed by each person. Many signature forgery case. This is because the lack of a verification system in identifying signature. Based on this, signature verification applications are built. Applications built using edge detection methods in extracting the characteristics and methods of neural network Learning Vector Quantization (LVQ) the signature pattern matching. Applications signature verification begins by scanning the signature traits extracted and stored in a basis data that is used as training data. Then users scan signatures match the system test and training data using LVQ. Matching results obtained from the application of the percentage of 70% original signatures are verified correctly and the error rate in the signature verification system in reading the signature of 30% false reading system with the original signature. This application can verify the signature.*

**Keywords:** Signature, Verification. Edge detection, LVQ, neural network.

## 1. PENDAHULUAN

Tanda tangan merupakan salah satu karakteristik yang dimiliki oleh setiap orang. Seringnya terjadi pemalsuan tandatangan, antara lain disebabkan oleh sistem verifikasi yang kurang baik. Verifikasi tandatangan kebanyakan dilakukan dengan cara manual, yaitu dengan membandingkan secara langsung dengan menggunakan mata manusia yang mempunyai banyak kelemahan pengelihatannya seseorang menjadi letih, terutama jika melakukan cukup banyak verifikasi, dapat menyebabkan kebosanan orang yang melakukannya, sehingga ketelitian dan ketepatan hasil yang diinginkan seringkali kurang memuaskan.

Pengenalan dan verifikasi tanda tangan meliputi dua bagian yang berbeda tetapi berkaitan erat satu sama lain. Yang pertama adalah identifikasi dari pemilik tanda tangan, sedangkan yang kedua adalah keputusan apakah tanda tangan tersebut asli atau dipalsukan. Selain itu, bergantung pada kebutuhannya, pengenalan tanda tangan dibagi menjadi dua kelas yang berbeda yaitu pengenalan dan verifikasi tanda tangan online dan offline.

Metode verifikasi tanda tangan dari berbagai penelitian yang berkembang saat ini tidak banyak yang menggunakan metode yang berbasis sinyal pada tahap ekstraksi ciri. Edge detection bisa dijadikan penelitian untuk mengenali ciri karakteristik dari tanda tangan secara offline. Deteksi tepi berfungsi untuk mengidentifikasi garis batas dari suatu objek terhadap latar belakang yang saling tumpang tindih. Sehingga apabila garis tepi pada citra dapat diidentifikasi dengan akurat, semua objek dapat ditemukan dan sifat dasar seperti area, bentuk, dan ukuran objek dapat diukur

Dalam proses verifikasi tanda tangan, proses klasifikasi sama pentingnya dengan proses ekstraksi fitur. Setelah fitur-fitur penting data atau sinyal yang dihasilkan pada proses ekstraksi fitur, fitur-fitur tersebut nantinya akan digunakan untuk proses klasifikasi. Metode klasifikasi yang digunakan adalah pengklasifikasi LVQ (*Learning Vector Quantization*) adalah suatu metode klasifikasi pola yang masing-masing unit keluaran mewakili kategori atau kelas tertentu. Suatu lapisan kompetitif akan secara otomatis belajar untuk mengklasifikasikan vektor - vektor masukan. Pada penelitian ini, dibangun aplikasi verifikasi tanda tangan untuk menguji keaslian dari sebuah tanda tangan. Dengan menggabungkan Edge Detection sebagai ekstraksi fitur dan metode LVQ (*Learning Vector Quantization*) sebagai pengklasifikasiannya. Penelitian dilakukan dengan batasan pengambilan citra tanda tangan secara *offline*, penulisan tanda tangan harus diatas kertas putih polos, penulisan menggunakan pulpen warna hitam. dan gambar di ambil dengan *scanner*. Hal ini dilakukan agar memperoleh hasil pengujian terhadap penelitian yang optimal.

## 2. METODOLOGI PENELITIAN

Metode penelitian dibagi menjadi dua tahap yaitu metode pengembangan aplikasi dan pengumpulan data.

### 2.1 Metode Pengembangan Aplikasi

Pada penelitian ini pengembangan aplikasi dilakukan menggunakan metode Waterfall. Pada tahap awal persiapan dan perencanaan dilakukan proses pengambilan data tanda tangan yang akan diproses sebagai data latih. Kemudian selanjutnya dilakukan tahap analisis terhadap metode ekstraksi ciri menggunakan metode deteksi tepi operator Prewitt, dilanjutkan dengan tahap perancangan basis data dan desain antarmuka. Selanjutnya tahap pengkodean dilakukan menggunakan MATLAB mengimplementasikan metode deteksi tepi operator Prewitt dan LVQ untuk pencocokan data latih dan data uji. Implementasi dari segi desain antarmuka juga dilakukan. Pengujian sistem dilakukan terhadap 10 pengguna dengan masing-masing 10 tanda tangan.

### 2.2 Metode Pengumpulan Data

#### 1. Studi literature

Diawal penelitian dilakukan terlebih dahulu pemahaman materi terhadap metode dan penelitian sebelumnya melalui beberapa tinjauan literature.

## 2. Pengumpulan data primer

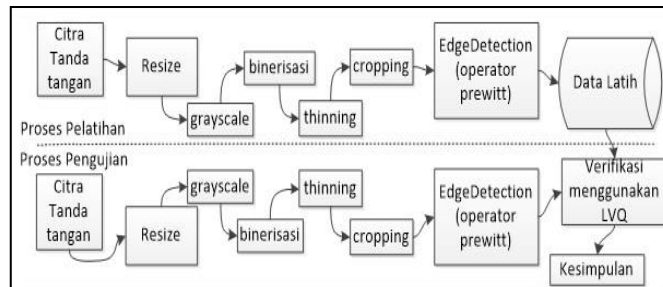
Pada penelitian ini dilakukan pengumpulan data primer dengan cara mengumpulkan sampel data tanda tangan dari teman-teman mahasiswa.

Pada tahap ini melakukan pengumpulan data tanda tangan untuk dijadikan data latih penelitian. Masing-masing tanda tangan setiap pengguna sejumlah 30 tanda tangan sebagai data latih.

## 3. ANALISIS DAN PEMBAHASAN

Pada penelitian ini dilakukan tahapan-tahapan untuk dapat melakukan identifikasi tanda tangan seseorang.

### 3.1. Analisa Sistem

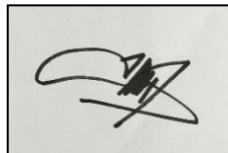


Gambar 1. Gambar Blok diagram

Tahapannya dibagi menjadi 2 proses yaitu, proses pelatihan dimana sistem akan mendata contoh-contoh tanda tangan yang menjadi acuan saat proses identifikasi. Berikutnya adalah proses pengujian dimana sistem melakukan proses identifikasi dari tanda tangan pengguna yang disesuaikan dengan tanda tangan yang sudah tersimpan di basisdata. Proses pelatihan dilakukan dengan beberapa tahap diantaranya:

#### 1. Mendata citra tanda tangan

Citra tanda tangan sebagai masukan dari langkah awal dari kerja sistem. Citra tanda tangan yang berupa citra digital.

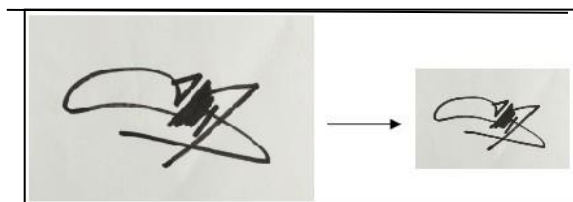


Gambar 2. citra tanda tangan

#### 2. Tahap *Resize*

Pada tahap *resize* ukuran gambar akan diubah sesuai ukuran yg ditentukan. Di tentukan dengan ukuran citra 100x100pixel. agar semua gambar yang diuji seragam ukurannya sehingga menghasilkan keluaran yg sesuai. pada proses *resize* menggunakan MATLAB. MATLAB menyediakan fungsi untuk melakukan proses *Resize* yaitu *imresize*. Contoh *resize* gambar ke 100x100 pada MATLAB.

```
I2=imresize(I,[100 ,100]);
```



Gambar 3. proses *resize*

#### 3. Tahap *Grayscale*

Tahap prapemrosesan bertujuan untuk memperbaiki citra dengan cara memanipulasi parameter citra agar menjadi citra dengan kualitas yang lebih baik. Dalam penelitian ini, prapemrosesan terdiri dari konversi RGB ke *grayscale*. Rumus untuk proses *grayscale*:

$$I(x,y) = \alpha * R + \beta * G + \gamma * B \dots \dots \dots (1)$$

Keterangan :

$I(x,y)$  = level keabuan pada suatu koordinat

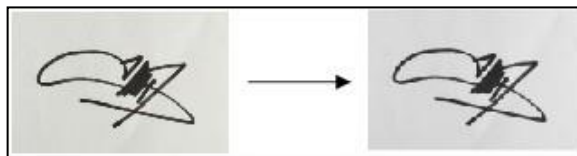
R = nilai warna merah

G = nilai warna hijau

B = nilai warna biru

Dengan konstanta nilai  $\alpha=0.2989$ ,  $\beta=0.5870$ ,  $\gamma=0.1140$

MATLAB menyediakan fungsi untuk melakukan proses grayscale dengan *coding* `rgb2gray`. Pada penerapannya dengan cara `I3=rgb2gray(I2)`;

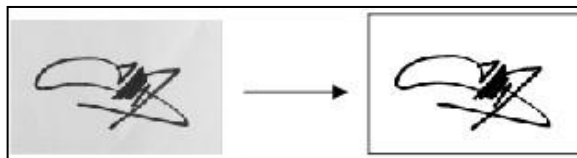


Gambar 4. Proses *grayscale*

#### 4. Tahap Binerisasi

Proses binerisasi melibatkan pengujian nilai tingkat keabu-abuan dari tiap piksel yang dinilai nilai biner satu, warna dari hasil proses biner akan menjadi 2 warna saja, yaitu warna hitam dan warna putih. MATLAB menyediakan fungsi untuk melakukan proses binerisasi yaitu `im2bw`. Contoh penerapan pada MATLAB:

`I3=im2bw(I3)`;

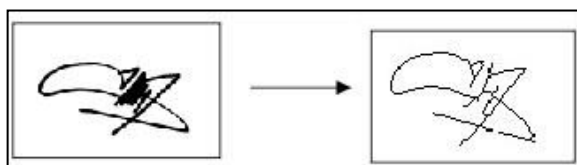


Gambar 5. citra tanda tangan

#### 5. Tahap *Thining*

*Thinning* adalah operasi morfologi yang digunakan untuk mengikis piksel latar depan sampai menjadi pixel yang tipis. Proses *thining* membuat garis pada citra menjadi 1 *pixel*. MATLAB menyediakan fungsi untuk melakukan proses *thining* yaitu `bwmorph('thin', inf)`. Contoh penerapan *thining* pada MATLAB

`I3 = bwmorph(~I3, 'thin', inf)`;



Gambar 6. citra tanda tangan

#### 6. Tahap *cropping*

*Cropping* adalah memotong satu bagian dari citra sehingga memperoleh citra yang berukuran lebih kecil. *Cropping* juga berarti pemotongan dari bagian image untuk meningkatkan konsentrasi gambar yang diutamakan.

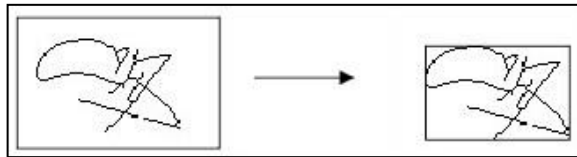
$X' = X - X_L$ , untuk  $X = X_L$  sampai  $X_R$

$Y' = Y - Y_T$ , untuk  $Y = Y_T$  sampai  $Y_B$  ..... (2)

Keterangan:

$X_L$  = nilai sisi kiri

$X_R$  = nilai sisi kanan  
 $Y_T$  = nilai sisi atas  
 $Y_B$  = nilai sisi bawah



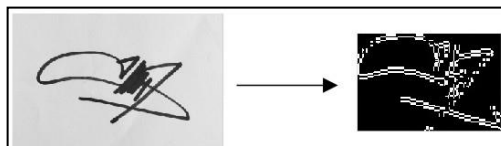
Gambar 7. Citra yang mengalami *cropping*

### 7. Tahap ekstraksi ciri dengan *edge detection*

*Edge detection* merupakan langkah pertama untuk melingkupi informasi didalam citra. Tepi berguna untuk segmentasi dan identifikasi objek di dalam citra. Pada aplikasi verifikasi tanda tangan digunakan deteksi tepi dengan operator gradien pertama yaitu operator prewitt. Metode Prewitt merupakan pengembangan metode robert dengan menggunakan filter PF yang diberi satu angka nol penyangga. Metode ini mengambil prinsip dari fungsi laplacian

MATLAB menyediakan fungsi untuk melakukan proses prewitt yaitu `edge('prewitt')`. Contoh penerapan *thining* pada MATLAB.

PR = `edge(im,'prewitt');`



Gambar 8. hasil ekstraksi ciri

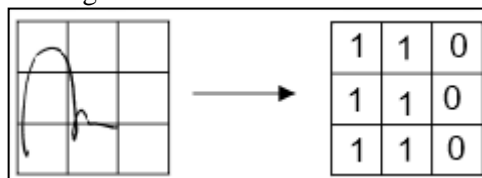
Hasil dari ekstraksi ciri berupa nilai vektor yang akan disimpan ke dalam basis data tanda tangan, dimana akan digunakan pada proses pencocokan.

Tahapan proses pengujian memiliki langkah tahapan yang sama dengan proses pelatihan sampai dari mulai proses *resize* sampai dengan tahap ekstraksi ciri. Pada tahap pengujian terdapat tahapan pencocokan data dengan data latih yang sudah tersedia di basis data. Tahap verifikasi menggunakan metode LVQ (*Learning Vector Quantization*)

### 8. Tahap Verifikasi dengan LVQ

Data latih ketika gambar data uji yang telah ditemukannya vektor dari hasil deteksi tepi maka diverifikasi dengan LVQ untuk mengklasifikasi bobotnya dengan yang ada didalam data latih. Bobot citra uji yang mendekati bobot target, maka menyatakan sebagai asli atau cocok. jika bobotnya terlalu jauh dengan bobot target maka dinyatakan palsu / tidak cocok. Metode LVQ ini hanya mencari bobot terkecil dari data uji.

Contoh kasus saat pengujian dengan LVQ sebagai berikut. Terdapat sebuah tanda tangan. Tanda tangan yang dibuat  $3 \times 3$  *pixel* bertujuan untuk mengetahui pixel mana saja yang mengenai garis tanda tangan. Pixel yang mengenai garis tanda tangan diberi nilai 1 dan yang tidak mengenai garis tanda tangan diberi nilai 0.



Gambar 9. tanda tangan dalam bentuk matriks

Didapat vektor dari data uji (1, 1, 0, 1, 1, 0, 1, 1, 0) . dan di hitung bobot terendah dari data latih. Rumus untuk memperbarui bobot sebagai berikut

$$\vec{W}_m \leftarrow \vec{W}_m + \eta \cdot (\vec{X} - \vec{W}_m) \dots\dots\dots(3)$$

Keterangan:

$\vec{W}_m$  = bobot

$\eta$  = Learning rate

$\vec{X}$  = masukan

$\vec{W}_m$ (baru) sama dengan  $\vec{W}_m$  lama ditambah  $\eta$  yang merupakan *learning rate* yang dikali dengan hasil dari masukan dikurangi  $\vec{W}_m$  lama.

Data latih berupa bobot  $w_1$  dan  $w_2$  karena ada 2 kelas.  $w_1$  untuk kelas 1 dan  $w_2$  untuk kelas 2 dengan bobot vektor akhir  $W_1 = (1, 0.9901, 0, 1, 1, 0.9902, 1, 0.0099, 0.0099)$  dan  $W_2 = (0, 1, 1, 0, 1, 1, 0.0099, 1, 0)$ . Berikutnya menghitung bobot terdekat dengan menguji data uji dengan mencari bobot terdekat dari data latih, dapat dilakukan sebagai berikut.

Jarak pada bobot ke - 1 =

$$(1, 0.9901, 0, 1, 1, 0.9902, 1, 0.0099, 0.0099)$$

$$= \frac{((1-1)^2 + (1-0.9901)^2 + (0-0)^2 + (1-1)^2 + (1-1)^2 + (0-0.9902)^2 + (1-1)^2 + (1-0.0099)^2 + (0-0.0099)^2)^{1/2}}{1} = 1.40$$

Jarak Pada bobot ke - 2 =  $(0, 1, 1, 0, 1, 1, 0.0099, 1, 0)$

$$= \frac{((1-0)^2 + (1-1)^2 + (0-1)^2 + (1-0)^2 + (1-1)^2 + (0-1)^2 + (1-0.01)^2 + (1-1)^2 + (0-0)^2)^{1/2}}{1} = 2.23$$

Dari hasil perhitungan kedua bobot, bobot ke-1 yang memiliki nilai terkecil. Maka hasil data uji tersebut masuk ke kelas 1 karena data uji mendekati bobot ke-1.

### 3.2. Implementasi dan Pengujian Sistem

Setelah melakukan analisis dan perancangan tahap berikutnya adalah implementasi dan pengujian. Pada tahap implementasi dilakukan implementasi desain antarmuka

#### 3.2.1 Implementasi Sistem

##### 1. Menu Verifikasi

Menu verifikasi merupakan menu utama dari aplikasi verifikasi tanda tangan. Pada menu ini terdapat 4 tombol, 2 kotak tulisan, dan 1 *axes* untuk menampilkan gambar, yang ditampilkan pada Gambar.10.



Gambar 10. tampilan aplikasi verifikasi tanda tangan

Menu verifikasi dapat langsung memverifikasi sebuah tanda tangan. bila sudah ada data latih yang disimpan dengan mengambil tanda citra tanda tangan lalu jika tanda tangan sudah tampil maka isi pada kolom masukan nama pemilik tanda tangan sebai pemilik tanda tangan tersebut.. Terdapat contoh verifikasi yang ditampilkan pada Gambar.11.



Gambar 11. tahap verifikasi berhasil

percobaan dilakukan dengan tanda tangan palsu. Dengan tanda tangan orang yang sama seperti pada Gambar 7, tetapi dibuat oleh orang yang berbeda maka didapat hasil seperti pada Gambar.12.



Gambar 12. Tahap verifikasi gagal

Terjadi ketidakcocokan dengan nama dan citra gambar, maka aplikasi akan mendeteksi bahwa tanda tangan tersebut palsu.

### 3.2.2 Pengujian Sistem

Pengujian dilakukan oleh 10 orang yang berbeda dan terdiri dari 30 tanda tangan setiap orangnya untuk dijadikan data latih dan 100 tanda tangan asli sebagai data uji pada Tabel 1.

Tabel 1 pengujian tanda tangan asli

No	Nama	Jumlah citra uji	hasil kecocokan
1	P1	10	10
2	P2	10	10
3	P3	10	10
4	P4	10	10
5	P5	10	10
6	P6	10	0
7	P7	10	10
8	P8	10	0
9	P9	10	10
10	P10	10	0

Tingkat kesuksesan (%) untuk 10 orang dan 10 citra tanda tangan

$$\frac{\text{hasil yang mendapat kecocokan}}{\text{Total jumlah citra data uji}} \times 100$$

$$= \frac{70}{100} \times 100 = 70\%$$

Percobaan dilakukan juga dengan tanda tangan palsu yang dibuat. Dari hasil 100 tanda tangan palsu masih terdapat kesalahan. bisa di lihat pada Tabel.2.

Dari hasil pengujian tanda tangan palsu. untuk tanda tangan koresponden 4 dan koresponden 5 yang di palsukan terdapat 3 tanda tangan yang terverifikasi. untuk tanda tangan Koresponden 9 yang di palsukan terdapat 4 tanda tangan yang terverifikasi. Untuk tanda tangan koresponden 2 dan koresponden 7 yang dipalsukan, terdapat 5 tanda tangan yang terverifikasi. untuk tanda tangan koresponden yang di palsukan terdapat 10 tanda tangan yang terverifikasi. Dan untuk koresponden 1, koresponden 6, koresponden 8 dan koresponden 10 tidak ada tanda tangan yang terverifikasi satupun.

Tabel.2 pengujian tanda tangan palsu

No	Nama	Hasil Kecocoka tanda tangan
1	P1	0 dari 10 tanda tangan palsu
2	P2	5 dari 10 tanda tangan palsu
3	P3	10 dari 10 tanda tangan palsu
4	P4	3 dari 10 tanda tangan palsu
5	P5	3 dari 10 tanda tangan palsu
6	P6	0 dari 10 tanda tangan palsu
7	P7	5 dari 10 tanda tangan palsu
8	P8	0 dari 10 tanda tangan palsu
9	P9	4 dari 10 tanda tangan palsu
10	P10	0 dari 10 tanda tangan palsu

Pada pengujian tanda tangan palsu tingkat keberhasilan mencapai  
 $= \frac{30}{100} \times 100 = 30\%$

#### 4. KESIMPULAN

Berdasarkan hasil penelitian yang diperoleh dari pengujian sistem yang menerapkan metode *edge detection* dan metode *Learning vector Quantization*, diperoleh kesimpulan bahwa Metode Learning Vector Quantization dapat mengenali pola tanda tangan dengan Tingkat akurasi untuk proses verifikasi mencapai 70% untuk mengenali tanda tangan asli.

#### DAFTAR RUJUKAN

- [1] Kusumadewi, S. 2003, *Artificial Intellegence (Teknik dan Aplikasinya)*, Graha Ilmu, Yogyakarta
- [2] *Kasus pemalsuan tanda tangan* (<http://surabaya.tribunnews.com/2015/08/02/kasus-pemalsuan-tanda-tangan-tahun>)
- [3] Muntasa arif , 2010, Konsep pengolahan citra digital dan ekstraksi fitur, Graha Ilmu, Yogyakarta
- [4] Putra Darma, 2009. Sistem Biometrika Konsep Dasar, Teknis analisis Citra, dan Tahapan Membangun aplikasi sistem Biometrika
- [5] Santosa Budi, 2007, Data Mining Terapan dengan Matlab, Graha Ilmu, Yogyakarta.