

## Pengenalan Pola Vektor Tanda Tangan Citra Digital Menggunakan Metode Pembagian Wilayah dan *Learning Vector Quantization* (LVQ)

Cakra Adipura Wicaksana<sup>1</sup>, Dina Estining Tyas Lufianawati<sup>1</sup>

<sup>1</sup>Jurusan Teknik Elektro, Fakultas Teknik, Universitas Sultan Ageng Tirtayasa, Cilegon, Banten.

### Informasi Artikel

**Naskah Diterima :** 15 Nopember 2021

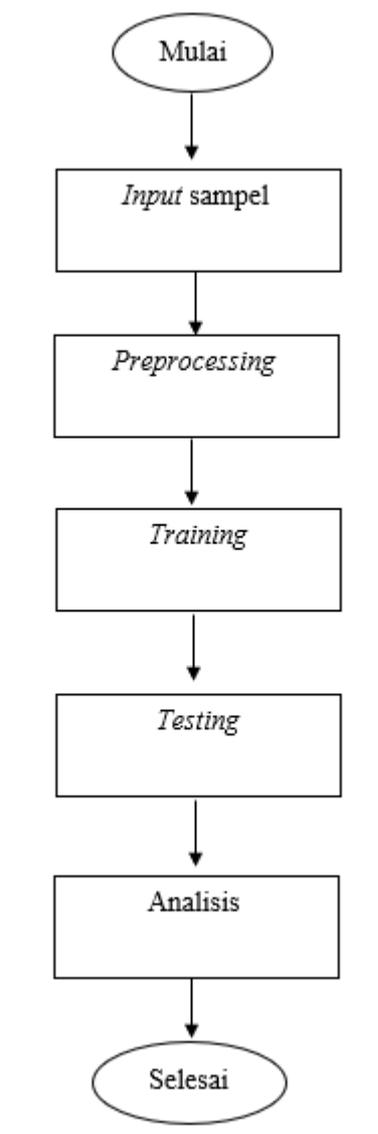
**Direvisi :** 17 Nopember 2021

**Disetujui :** 17 Nopember 2021

*doi:* 10.36055/setrum.v10i2.13054

**\*Korespondensi Penulis :**  
cakraadipura@untirta.ac.id

### Graphical abstract



### Abstract

The signature recognition process actually has many methods. In recognizing pattern of signatures i.e. using Artificial Neural Network Algorithms, Learning Vector Quantization (LVQ), Invariant Moment Method, Arc Method, and so forth. In this paper, the authors intend to try applying the Learning Vector Quantization (LVQ) method as a way for the classification process. Based on many literature and references that the author has read, the Learning Vector Quantization (LVQ) method is one method that is quite effective when compared to the methods that have been used previously to detect signatures. There are three main stages of in this paper i.e. preprocessing, training, and testing. The application has been successfully created by using Java Swing with graphics user interface. Thus, as the result show that this application can decisively recognize and classify signature vector input very well.

**Keywords:** signature, Learning Vector Quantization (LVQ), classification

### Abstrak

Proses pengenalan tanda tangan sebenarnya memiliki banyak metode. Dalam mengenali pola tanda tangan yaitu menggunakan Algoritma Jaringan Syaraf Tiruan, *Learning Vector Quantization* (LVQ), Metode *Invariant Moment*, Metode Arc, dan lain sebagainya. Dalam tulisan ini, penulis bermaksud mencoba menerapkan metode *Learning Vector Quantization* (LVQ) sebagai salah satu cara untuk proses klasifikasi. Berdasarkan banyak literatur dan referensi yang penulis baca, metode *Learning Vector Quantization* (LVQ) merupakan salah satu metode yang cukup efektif jika dibandingkan dengan metode yang telah digunakan sebelumnya untuk mendeteksi tanda tangan. Ada tiga tahapan utama dalam makalah ini yaitu preprocessing, training, dan testing. Aplikasi telah berhasil dibuat dengan menggunakan Java Swing dengan antarmuka pengguna grafis. Dengan demikian, sebagai hasilnya menunjukkan bahwa aplikasi ini dapat mengenali dan mengklasifikasikan vektor *input* tanda tangan dengan sangat baik.

**Kata kunci:** tanda tangan, *Learning Vector Quantization* (LVQ), klasifikasi

## 1. PENDAHULUAN

Tanda tangan adalah salah satu bentuk yang dibubuhkan seseorang di atas kertas untuk mengesahkan atau menyetujui suatu pernyataan. Tanda tangan setiap orang berbeda satu sama lain. Tanda tangan setiap orang tidak akan pernah sama persis ukuran atau bentuknya. Karena tanda tangan memiliki ciri khusus, ada baiknya jika tanda tangan dideteksi atau dikenali dengan bantuan aplikasi komputer.

Proses pengenalan tanda tangan sebenarnya memiliki banyak metode. Dalam mengenali sebuah tanda tangan kita bisa menggunakan Algoritma Jaringan Syaraf Tiruan, *Learning Vector Quantization* (LVQ), Metode *Invariant Moment*, Metode Arc, dan lain sebagainya. Dalam penelitian ini, penulis bermaksud mencoba menerapkan metode *Learning Vector Quantization* (LVQ) sebagai salah satu cara untuk proses klasifikasi.

Berdasarkan beberapa literatur dan referensi yang penulis baca khususnya dari [1] [2], metode *Learning Vector Quantization* (LVQ) merupakan salah satu metode yang cukup efektif jika dibandingkan dengan metode yang telah digunakan sebelumnya untuk mendeteksi tanda tangan. Dalam tulisan ini, penulis membatasi beberapa keterbatasan dalam melakukan penelitian. Batasan tersebut adalah sampel tanda tangan yang digunakan adalah citra dengan citra digital RGB dan proses pengenalan atau identifikasi pola vektor tanda tangan belum dinamis seperti menggunakan kamera.

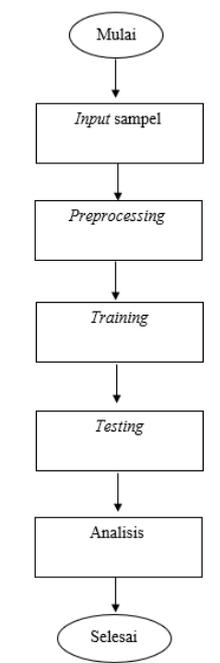
## 2. METODE PENELITIAN

### 2.1 Metode Penelitian

Secara garis besar, ada 5 tahapan pada penelitian ini antara lain tahapan persiapan, *preprocessing*, *training*, *testing*, dan tahapan analisis.

- a) Dalam tahapan persiapan dilakukan *input* sampel tandatangan
- b) *Preprocessing*
- c) *Training*
- d) *Testing*
- e) Analisis

### 2.2 Diagram Alir Penelitian



Gambar 1. Diagram Alir Penelitian

## 2.3 Perancangan Penelitian

### 2.3.1 Persiapan

Pada tahapan ini dilakukan *input* sampel tandatangan.

### 2.3.2 Preprocessing

Tahap *preprocessing* yang dilakukan pada sistem ini adalah mengubah citra menjadi *grayscale*, kemudian deteksi tepi menggunakan kernel Roberts, dan dilanjutkan dengan proses negasi. Setelah proses negasi, citra akan diolah menjadi bentuk 0 dan 1 dengan *threshold* 200. Jika nilai piksel  $> 200$  maka nilai piksel menjadi 1, sedangkan jika nilai piksel  $\leq 200$  maka nilai piksel menjadi 0. Karena data citra yang digunakan adalah  $340 \times 272$  piksel maka akan terdapat 92.840 baris bilangan biner 0 dan 1. Karena 92.840 cukup besar, maka dilakukan proses pembagian citra untuk mereduksi baris bilangan biner dengan membagi citra menjadi  $10 \times 8$  wilayah dimana setiap region berukuran  $34 \times 34$  piksel. Setiap piksel pada area berukuran  $34 \times 34$  akan ditambahkan sehingga mendapatkan nilai biner sebesar 8 bit. Penjumlahan biner dilakukan pada daerah  $10 \times 8$ . Setelah peringkasan selesai, selanjutnya mencoba menyusun deretan bilangan biner dengan panjang  $10 \times 8 \times 8$ , yaitu 640 baris bilangan biner yang merupakan vektor yang siap untuk proses pelatihan.

### 2.3.3 Training Data

*Learning* akan menyesuaikan bobot dengan pola yang dipelajari dari data. *Learning* pada metode LVQ, bobot awal menggunakan pola yang ada, kemudian bobot akan diubah (diperbarui) tergantung pada kelas vektor input sesuai dengan kelas yang dinyatakan sebagai neuron pemenang. Jika sesuai, vektor bobot diperbarui sehingga jaraknya lebih dekat dengan vektor input. Jika tidak, vektor bobot diperbarui sehingga jaraknya semakin jauh dari vektor input. Bobot awal ini diambil dari data 1 dan 2, yang harus diambil ke dalam bentuk vektor. Vektor bobot ini biasanya ditulis dengan  $w_{1j} = (w_{11}, w_{12}, w_{13}, \dots, w_{1m})$  yang merupakan vektor bobot kelas satu dengan variabel  $m$  dan  $w_{2j} = (w_{21}, w_{22}, w_{23}, \dots, w_{2m})$  yang merupakan vektor bobot kelas kedua dengan variabel  $m$ . Parameter yang digunakan dalam metode LVQ adalah sebagai berikut :

#### a. Alfa (Kecepatan belajar)

Alpha didefinisikan sebagai tingkat belajar. Jika alpha terlalu besar, maka algoritma akan menjadi tidak stabil sebaliknya jika alpha terlalu kecil, proses akan memakan waktu terlalu lama. Nilai alfa adalah  $0 < \alpha < 1$ .

#### b. DecAlfa (Penurunan Tingkat Pembelajaran)

Yaitu penurunan tingkat belajar.

#### c. MinAlfa (Tingkat Pembelajaran Minimum)

Itulah nilai minimal tingkat pembelajaran yang masih diperbolehkan.

#### d. MaxEpoch (Epoch maksimum)

Jumlah maksimum epoch atau iterasi yang dapat dilakukan selama pelatihan. Iterasi akan dihentikan jika nilai epoch melebihi epoch maksimum.

### 2.3.4 Data Testing

Setelah *training*, bobot akhir ( $W$ ) akan diperoleh. Bobot ini kemudian digunakan untuk melakukan simulasi atau pengujian. Misalnya  $p$  data akan diuji. Misalkan diambil vektor input misalnya  $(0, 1, 0, 1)$ , maka jarak input dihitung terlebih dahulu dengan vektor bobot kelas yang telah dilatih. Jarak Euclidian terpendek akan menjadi kelasnya. Contoh perhitungannya adalah sebagai berikut :

Jarak pada vektor berat pertama :

$$\sqrt{(0 - 0,3727)^2 + (1 - 0,2161)^2 + (0 - 0,6347)^2 + (1 - 0,2164)^2}$$

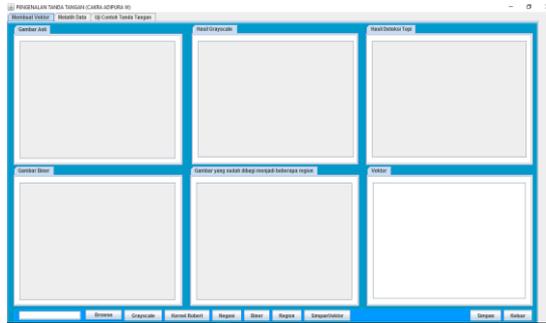
Jarak pada vektor berat kedua :

$$\sqrt{(0 - 0)^2 + (1 - 0,7969)^2 + (0 - 0,7900)^2 + (1 - 1)^2}$$

Yang memiliki jarak terkecil merupakan kandidat terbesar sebagai bagian dari kelas tersebut.

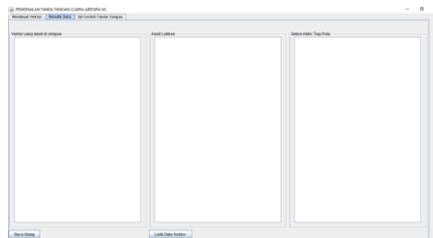
### 3. HASIL DAN PEMBAHASAN

Hasil dari aplikasi ini berupa dokumen berupa.jar. Untuk menjalankan dokumen ini, cukup klik dua kali. Sebelum menjalankan aplikasi ini, pastikan komputer telah terinstal Java JRE terlebih dahulu.



Gambar 2. Tampilan GUI pada Proses *Preprocessing*

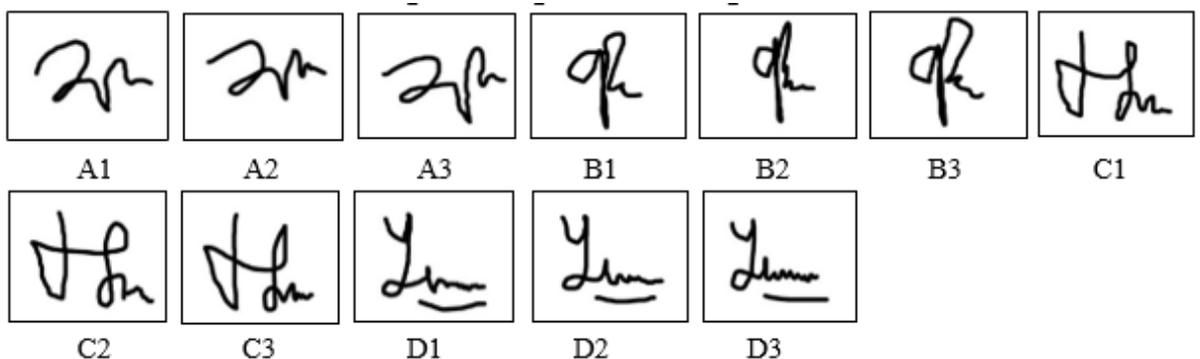
Pada gambar 2 di atas ada enam panel. Panel pertama yang menampilkan gambar asli. Panel kedua berfungsi untuk menampilkan citra *grayscale*. Cara membuat aplikasi java swing bisa merujuk ke [8].



Gambar 3. Tampilan Proses *Data Training*

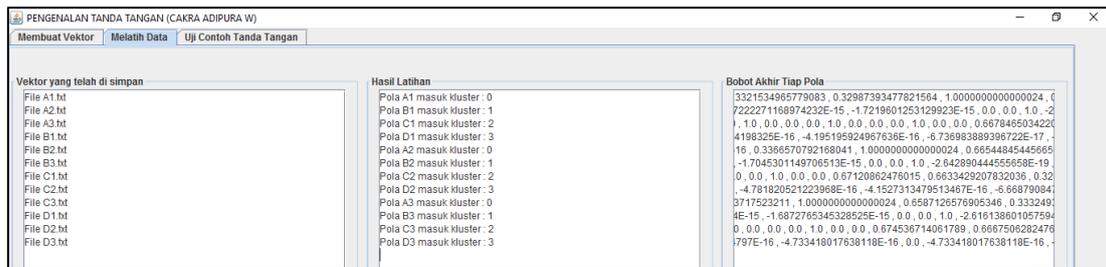
Pada gambar di atas, ada tiga panel. Panel pertama digunakan untuk membaca data atau dokumen .txt yang telah disimpan pada proses *preprocessing* sebelumnya. Panel kedua berfungsi untuk menampilkan hasil dari proses *training*, berupa hasil pengelompokan vektor input, sedangkan panel terakhir berfungsi untuk menampilkan bobot akhir dari setiap *input* berbentuk vektor.

Dalam percobaan ini, data yang digunakan adalah 12 data tanda tangan. Ke-12 data tersebut dibagi menjadi 3 kelompok yaitu tanda tangan milik A, B, C, dan D. Data lengkap yaitu A1, B1, C1, D1, A2, B2, C2, D2, A3, B3, C3, D3 . Berikut adalah gambar data tanda tangan yang akan digunakan :



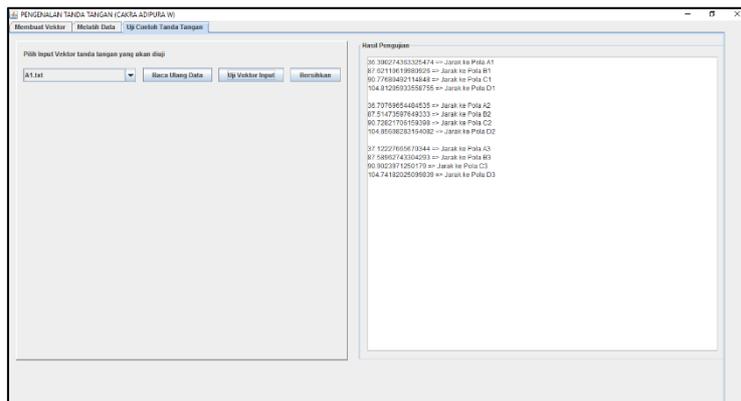
Gambar 4. Contoh Tanda Tangan

Jika semua data sudah berbentuk vektor, sekarang saatnya untuk melakukan *training* terhadap 12 data tersebut. Prosesnya akan seperti gambar di bawah ini :



Gambar 5. Hasil Data Training

Sekarang memasuki tahap *testing*. Pada *testing* di bawah ini menggunakan data A1 sebagai vektor inputnya.



Gambar 6. Pengenalan *Testing* untuk Tanda Tangan Vektor A1

Hasil pengujian menunjukkan bahwa jarak Euclidian antara vektor A1 dengan 12 vektor terkecil lainnya berada pada pola A1 dengan nilai besar 36,39. Hal ini menunjukkan bahwa vektor A1 termasuk dalam ke A. Dari dua belas lembar data semua data telah berhasil dilatih dan diuji dengan baik, 80,1 % benar. Bisa lebih akurat jika ada perbaikan pada ekstraksi ciri.

#### 4. KESIMPULAN DAN SARAN

##### 4.1 Kesimpulan

Hasil dari penelitian yang telah dilakukan, berikut adalah beberapa kesimpulan yang diperoleh yaitu dengan menggunakan metode LVQ, aplikasi mampu mengklasifikasikan dan mengenali vektor input dengan baik; LVQ dalam mengenali pola vektor tanda tangan adalah dengan melakukan *learning* sehingga diperoleh bobot akhir dari masing-masing kelas yang berfungsi untuk melakukan pengujian; Kekurangan dari metode LVQ adalah kita harus menentukan kelas target terlebih dahulu.

##### 4.2 Saran

Saran untuk penelitian selanjutnya adalah membuat metode atau memperbaiki proses ekstraksi ciri agar proses pengenalan tanda tangan dengan mengubah posisi tanda dapat dikenalkan. Saat ini jika posisi tanda tangan banyak bergeser, proses pengenalannya tidak maksimal. Semua kode sumber tersedia di github: <https://github.com/adipurapunya/vector-pattern-recognition>

## REFERENSI

- [1] Sinha G, Rani A, Dhir P R and Rani M R 2012 Zone-Based Feature Extraction Techniques and SVM for Handwritten Gurmukhi Character Recognition international J. Adv. Res. inComputer Sci. Softw. Eng. 2 106–11
- [2] Attigeri S 2018 Neural Network based Handwritten Character Recognition system Int. J. Eng. Comput. Sci. 7 23761–8
- [3] Hidayati N and Warsito B 2010 PREDIKSI TERJANGKITNYA PENYAKIT JANTUNG DENGAN METODE LEARNING VECTOR QUANTIZATION Media Stat. 3 21–30
- [4] Sutarno and Sara Putri Fauliah 2019 Implementation of Learning Vector Quantization (LVQ) Algorithm for Durian Fruit Classification Using Gray Level Co-occurrence Matrix (GLCM) Parameters IOP Conf. Series: Journal of Physics: Conf. Series p 1196
- [5] Goel R, Kumar V, Srivastava S and Sinha A K 2017 A Review of Feature Extraction Techniquesfor Image Analysis Int. J. Adv. Res. Comput. Commun. Eng. 6 153–5
- [6] Prasad V and Jayanta Y 2013 A study on method of feature extraction for Handwritten Character Recognition Indian J. Sci. Technol.
- [7] P E P, Puspitaningrum D and Mirfen A 2015 IDENTIFICATION OF SIGNATURE WITH APPROACH SUPPORT VECTOR MACHINE J. Sains, Teknol. dan Ind. 12 225–31
- [8] Conrod P and Tylee L 2017 Learn Java GUI Applications (Kidware Software; 8th edition (April 8, 2017))