

c

Cakra Adipura Wicaksana¹, Dina Estining Tyas Lufianawati¹

¹Jurusan Teknik Elektro, Fakultas Teknik, Universitas Sultan Ageng Tirtayasa, Cilegon, Banten.

Informasi Artikel

Naskah Diterima : 15 Nopember 2021

Direvisi : 17 Nopember 2021

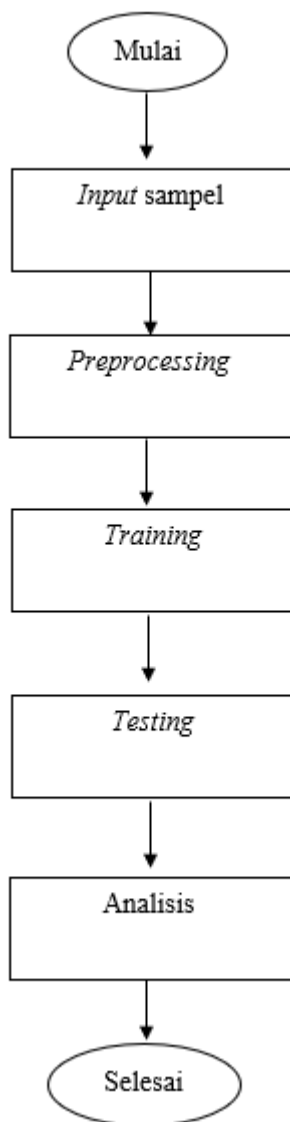
Disetujui : 17 Nopember 2021

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 dapat dilakukan melalui berbagai metode, seperti, Learning Vector Quantization, Metode Arc, Metode Invariant Moment, Metode Jaringan Syaraf Tiruan, dan lainnya. Dalam mengenali pola tanda tangan, penulis memilih menerapkan algoritma Learning Vector Quantization (LVQ) sebagai salah satu pendekatan untuk proses pengklasifikasi. Menurut sumber dari literatur ilmiah dan referensi jurnal atau buku, algoritma LVQ terbukti cukup efektif dibandingkan metode-metode lain yang telah diaplikasikan sebelumnya untuk mengenali pola tanda tangan. Tulisan ini mencakup tiga tahapan utama: preprocessing, training, dan testing. Pada tahap preprocessing, data tanda tangan dipersiapkan sebagai input yang sesuai untuk algoritma. Pada tahap training, sistem dilatih menggunakan data tanda tangan yang telah diproses. Tahap testing dilakukan untuk mengevaluasi kinerja sistem dalam mengenali dan mengklasifikasikan tanda tangan. Aplikasi ini dikembangkan memanfaatkan Java Swing dengan tampilan yang memudahkan pengguna. Hasil percobaan memberikan informasi bahwa pendekatan ini mampu merekognisi dan mengklasifikasikan masukan berupa vektor pola tanda tangan dengan performa yang bagus. Metode LVQ yang diterapkan dalam penelitian ini menunjukkan potensi besar dalam meningkatkan keandalan proses pengenalan pola tanda tangan.

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

1. PENDAHULUAN

Tanda tangan merupakan pola atau 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 memiliki berbagai metode yang bisa diterapkan. Beberapa metode yang digunakan untuk mengenali tanda tangan termasuk Algoritma Artificial Neural Network, Learning Vector Quantization, Invariant Moment Method, Arc Metode, dan lainnya. Fokus penelitian ini, yaitu menggunakan algoritma Learning Vector Quantization (LVQ) sebagai metode pendekatan untuk pengklasifikasian atau pengenalan objek.

Menurut beberapa referensi dan literatur ilmiah yang telah didalami khususnya dari [1] [2], metode *Learning Vector Quantization* (LVQ) adalah teknik yang relatif efektif apabila dibandingkan dengan teknik yang telah diutilasi sebelumnya untuk mendeteksi pola dari 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.

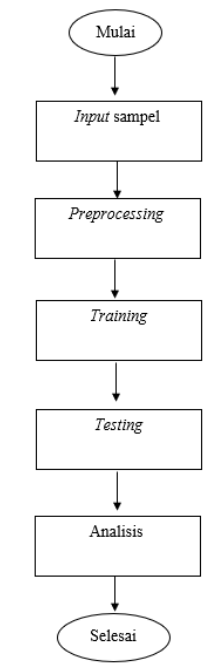
2. METODE PENELITIAN

2.1 Metode Penelitian

Secara keseluruhan, ada lima 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 Alur Proses Penelitian



Gambar 1. Gambar Alur Proses Penelitian

2.3 Perancangan Penelitian

2.3.1 Persiapan

Pada tahapan ini dilakukan *input* sampel tandatangan.

2.3.2 Preprocessing

Tahap *preprocessing* yang diutilasi pada sistem ini adalah mengubah citra atau gambar 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 *pixel* > 200 maka nilai *pixel* menjadi berubah menjadi 1, sedangkan jika nilai *pixel* ≤ 200 maka nilai *pixel* menjadi 0. Karena data citra yang digunakan adalah 340 x 272 *pixel* 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 x 8 wilayah dimana setiap region berukuran 34 x 34 piksel. Setiap piksel pada area berukuran 34 x 34 akan ditambahkan sehingga mendapatkan nilai biner sebesar 8 bit. Penjumlahan biner dilakukan pada daerah 10 x 8. Setelah peringkasan selesai, selanjutnya mencoba menyusun deretan bilangan biner dengan panjang 10 x 8 x 8, yaitu 640 baris bilangan biner yang merupakan vektor yang siap untuk proses pelatihan.

2.3.3 Training Data

Proses learning dalam metode LVQ menyesuaikan bobot berdasarkan pola yang dipelajari dari data. Bobot awal didasarkan pada pola yang ada, kemudian bobot ini akan diperbarui berdasarkan kelas vektor input sesuai dengan kelas neuron pemenang. Jika kelasnya sesuai, vektor bobot diperbarui agar lebih dekat dengan vektor input. Jika tidak sesuai, vektor bobot diperbarui agar menjauh dari vektor input. Nilai bobot pertama diperoleh dari data ke-1 dan ke-2, yang harus diubah menjadi bentuk masukkan vektor. Nilai masukkan bobot ini dinyatakan sebagai $w_{1j} = (w_{11}, w_{12}, w_{13}, \dots, w_{1m})$ untuk vektor bobot kelas satu dengan variabel m , dan $w_{2j} = (w_{21}, w_{22}, w_{23}, \dots, w_{2m})$ untuk vektor bobot kelas dua dengan variabel m . Nilai atau parameter yang akan digunakan dalam teknik LVQ ini, yaitu pada penjelasan berikut:

a. Alfa (*Kecepatan belajar*)

Alpha digunakan sebagai *learning rate*. Jika alpha terlalu besar, maka metode atau proses akan berubah ke arah tidak stabil, dan sebaliknya jika alpha terlampaui kecil, proses akan memakan waktu terlampaui lama. Nilai simbol alfa, yaitu jika $0 < \alpha < 1$.

b. DecAlfa atau (*Tingkat Pembelajaran yang menurun*)

Yaitu penurunan tingkat belajar.

c. MinAlfa (*Tingkat Pembelajaran Minimum*)

Itulah nilai minimal tingkat pembelajaran masih dalam batas yang ditentukan.

d. MaxEpoch (*Epoch maksimum*)

Jumlah maksimum pengulangan atau iterasi yang dapat dilakukan selama proses training. Pengulangan akan dihentikan apabila nilai iterasi telah mencapai nilai iterasi maksimum.

2.3.4 Data Testing

Setelah *training*, bobot akhir (W) akan diperoleh. Nilai ini kemudian dimanfaatkan untuk percobaan pengujian atau simulasi. Misalnya p data akan dicoba. 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 :

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

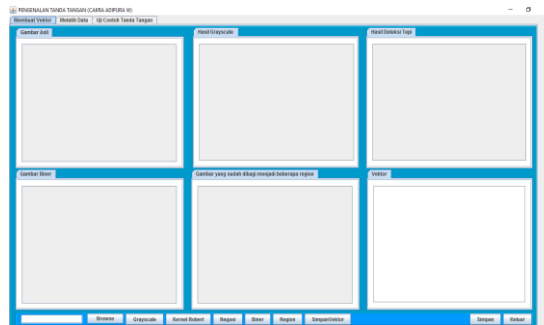
Jarak pada vektor berat kedua :

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

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

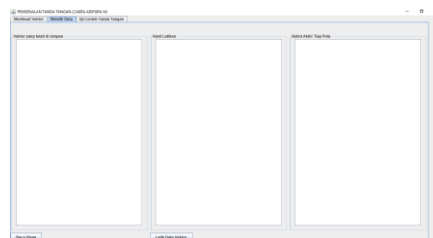
3. HASIL DAN PEMBAHASAN

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*

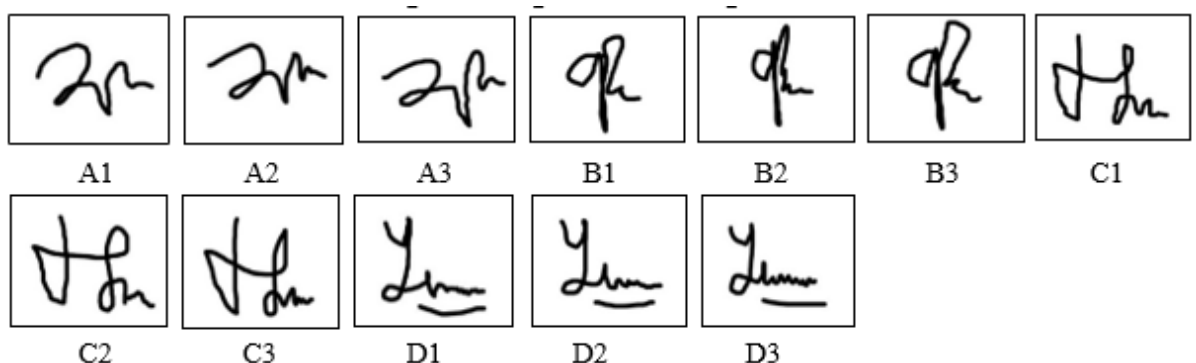
Pada gambar 2 di atas ada enam panel. Panel pertama yang menampilkan gambar asli. Panel kedua berfungsi untuk menampilkan citra *grayscale*. Cara membuat aplikasi java swing bisa merujuk ke [8].



Gambar 3. Tampilan Proses *Data Training*

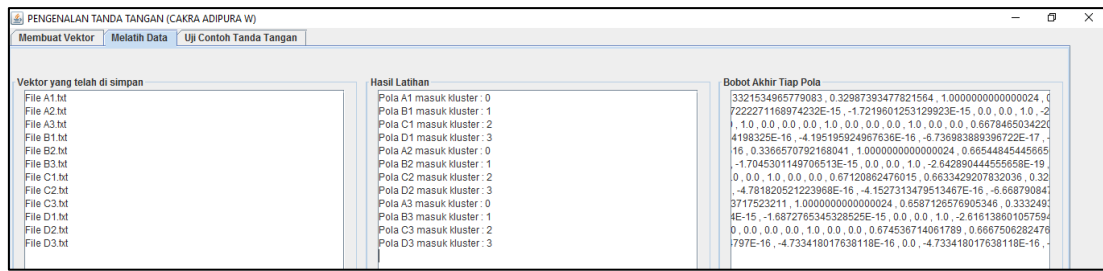
Pada gambar di atas, ada tiga panel. Panel pertama digunakan untuk membaca data atau dokumen .txt yang telah disimpan pada proses *preprocessing* sebelumnya. Panel kedua berfungsi untuk menampilkan hasil dari proses *training*, berupa hasil pengelompokan vektor input, sedangkan panel terakhir berfungsi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



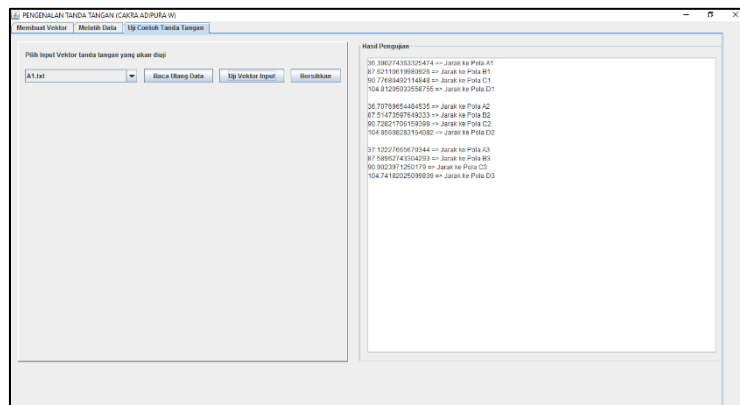
Gambar 4. Contoh Tanda Tangan

Jika semua data sudah berbentuk vektor, sekarang saatnya untuk melakukan *training* terhadap 12 data tersebut. Prosesnya akan tampil seperti tangkapan layar berikut 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 percobaan 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

Masukkan untuk penelitian berikutnya adalah membuat teknik 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. in Computer 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 Techniques for 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))