

**AUTENTIKASI TANDA TANGAN MENGGUNAKAN
METODE LVQ (*Learning Vector Quantization*)
UNTUK KEAMANAN DOKUMEN**

SKRIPSI

Oleh :

NUR JAZILAH NIAM

NIM : 0965039



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI (UIN) MAULANA MALIK
IBRAHIM MALANG
2014**

HALAMAN PENGANTAR
AUTENTIKASI TANDA TANGAN MENGGUNAKAN
METODE LVQ (*Learning Vector Quantization*)
UNTUK KEAMANAN DOKUMEN

SKRIPSI

Diajukan kepada :
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam Memperoleh Gelar Sarjana
Komputer (S. Kom)

Oleh :
NUR JAZILAH NIAM
NIM : 09650139

JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI (UIN) MAULANA MALIK IBRAHIM
MALANG
2014

HALAMAN PERSETUJUAN
AUTENTIKASI TANDA TANGAN MENGGUNAKAN
METODE LVQ (*Learning Vector Quantization*)
UNTUK KEAMANAN DOKUMEN

SKRIPSI

Oleh :

Nama : Nur Jazilah Niam
NIM : 09650139
Jurusan : Teknik Informatika
Fakultas : Sains dan Teknologi

Telah Disetujui, Juli 2014

Dosen Pembimbing I

Dosen Pembimbing II

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

Hani Nurhayati M.T
NIP. 19780625 200801 2 006

Mengetahui,

Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

HALAMAN PENGESAHAN
AUTENTIKASI TANDA TANGAN MENGGUNAKAN
METODE LVQ (*Learning Vector Quantization*)
UNTUK KEAMANAN DOKUMEN

SKRIPSI

Oleh :
Nur Jazilah Niam
NIM. 09650139

Telah Dipertahankan Di Depan Dewan Penguji Skripsi
Dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S. Kom)

Tanggal, 14 Juli 2014

Susunan Dewan Penguji :	Tanda Tangan
1. Penguji Utama : <u>Dr. Muhammad Faisal, M.T</u> 19740510 200501 1 007	()
2. Ketua Penguji : <u>Irwan Budi Santoso</u> 19770103 201101 1 004	()
3. Sekretaris : <u>Dr. Cahyo Crysdian</u> NIP. 19740424 200901 1 008	()
4. Anggota Penguji : <u>Hani Nurhayati M.T</u> NIP. 19780625 200801 2 006	()

Mengetahui,
Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

SURAT PERNYATAAN

Saya yang bertandatangan di bawah ini :

Nama : Nur Jazilah Niam
NIM : 09650139
Fakultas/Jurusan : Sains dan Teknologi / Teknik Informatika
Judul Penelitian : Autentikasi Tanda Tangan Menggunakan Metode
LVQ (*Learning Vector Quantization*) Untuk
Keamanan Dokumen

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila di kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, Juli 2014
Yang Membuat Pernyataan,

Nur Jazilah Niam

NIP. 09650139

HALAMAN MOTTO

مَنْ جَدًّا وَجَدَّ

Barang Siapa yang bersungguh-sungguh Pasti akan mendapat

Tuhan itu seperti apa yang disangkakan oleh hambanya



HALAMAN PERSEMBAHAN

الحمد لله رب العالمين

Karya ini saya persembahkan kepada :

- Kedua orang tuaku, H. Kholil Ghozali dan Rochimah yang telah memberikan saya kesempatan untuk meraih keinginan ku untuk duduk dibangku perkuliahan. Dan memberikan semangat dan waktunya agar dapat menyelesaikan penelitian ini.
- Suamiku, Lukman Jazuli. Yang sabar menantikanku meyelesaikan karya ini, dan rela mengalah serta selalu memberiku semangat agar aku melakukan yang terbaik.
- Si kecil , Ubai yang rela kehilangan waktu bermain bersama. Dan mebuatku tersenyum ketika kepenatan datang.
- Teman – temanku TI 09 E, yang memberi warna lain dalam kehidupanku. khususnya agung dan fathiyat yang selalu aku repotin dengan pertanyaan-pertanyaanku mengenai penelian ini. Terimakasih teman, jasa kalian tak akan pernah aku lupakan.
- Dan seluruh rekan yang berjasa selama ini, yang tidak dapat disebutkan satu persatu. Terima kasih...

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Puji syukur kehadiran Allah SWT, yang telah memberikan rahmad, hidayah serta taufiknya hingga penulis dapat menyelesaikan karya ilmiah ini dengan judul **“Autentikasi Tanda Tangan Menggunakan Metode LVQ (*Learning Vector Quantization*) Untuk Keamanan Dokumen”** dengan baik. Semoga rahmat Allah SWT selalu melimpahkannya pada Nabi Muhammad SAW.

Penulis sangat menyadari dalam menyelesaikan tugas akhir ini banyak mendapat bantuan, karena itu atas bantuannya penulis menyampaikan penghargaan dan terima kasihnya kepada :

1. Bapak Prof. DR. H. Mudjia Raharjo M.Si selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim.
2. Bapak Dr.Cahyo Crysdiya selaku ketua jurusan Teknik Informatika sekaligus Dosen pembimbing I dan Ibu Hani Nurhayati M.T selaku Dosen Pembimbing II, yang telah banyak memberikan bimbingan serta motifasinya kepada penulis untuk menyelesaikan tugas akhir ini.
3. Seluruh dosen Universitas Islam Negeri Maulana Malik Ibrahim Malang, khususnya dosen Teknik Informatika dan staf yang telah memberikan ilmu dan dukungannya selama penulis menempuh pendidikan.
4. Keluarga kecilku, Bapak, Ibu, suami seta si kecilku yang telah memberikan doan dan dukungannya.

5. Semua sahabat yang telah membrikan bantuannya secara tulus hingga tugas akhir ini dapat selesai.
6. Dan seluruh pihak yang mendukung dalam penulisan tugas akhir ini. Semoga karya kecilku ini bermanfaat bagi yang lain.

Penulis menyadari banyak sekali kekurangan dalam pembuatan tugas akhir ini, sehingga penulis dengan rendah hati mengharapkan kritik dan saran bagi pembaca.

Malang, Juli 2014

Penulis



DAFTAR ISI

Halaman Judul	i
Halaman pengajuan	ii
Halaman Persetujuan	iii
Halaman Pengesahan	iv
Surat Pernyataan	v
Motto	vi
Halaman Persembahan	vii
Kata Pengantar	viii
Daftar Isi	x
Daftar Gambar	xii
Daftar Tabel	xiv
Abstrak	xiv
Bab I Pendahuluan	
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	4
1.6 Sitematika Penelitian	4
Bab II Tinjauan Pustaka	
2.1 Autentikasi Tanda Tangan	5
2.2 Algoritma LVQ	8
2.3 Enkripsi Caesar Chiper	8
Bab III Perancangan dan Implementasi Aplikasi	
3.1 Desain Proses	15
3.1.1 Akuisi Citra Tanda Tangan	16
3.1.2 Preprocessing Citra	17
3.1.3 Arsitektur Algoritma LVQ pada Sistem	20
3.1.4 Pembelajaran Cira Menggunakan Algoritma LVQ	22
3.1.5 Autentikasi Citra Menggunakan Algoritma LVQ	29
3.2 Kemanan File	32
3.2.1 Enkripsi File	32
3.2.2 Dekripsi File	34
3.3 Desain GUI	36
3.4 Implementasi Aplikasi	47
Bab IV Eksperimen dan Pembahasan	
4.1 Langkah-langkah Eksperimen	52
4.2 Eksperimen atau Uji Coba	50
4.3 Pembahasan	58

4.4 Integrasi Autentikasi Tanda Tangan Menggunakan Algoritma LVQ untuk Keamanan Dokumen dengan AI quran	62
--	----

Bab V Kesimpulan dan Saran

5.1 Kesimpulan	65
5.2 Saran	66

Daftar Pustaka

Lampiran 1

Lampiran 2

Lampiran 3

Lampiran 4

Lampiran 5



DAFTAR GAMBAR

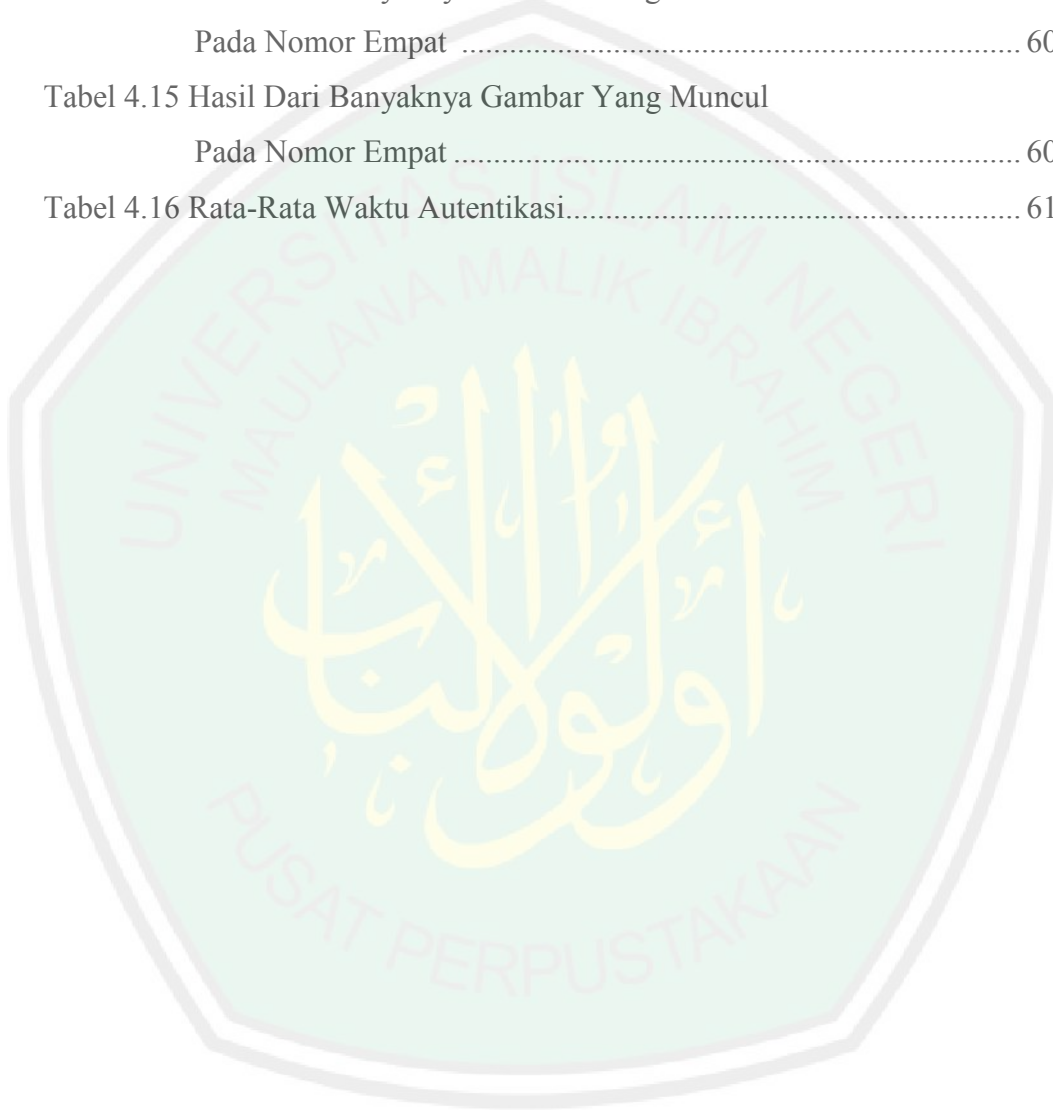
Gambar 2.1 Arsitektur Jaringan LVQ (<i>Learning Vector Quantization</i>)	8
Gambar 2.2 Enkripsi Caesar Chiper.....	12
Gambar 3.1 Diagram Blok Aplikasi Autentikasi Tanda Tangan Menggunakan Algoritma LVQ	16
Gambar 3.2 Hasil Penggambaran Tanda Tangan.....	16
Gambar 3.3 Tempat Untuk Membuat (Menggambar) Tanda Tangan	17
Gambar 3.4 Diagram Alur <i>Preprocessing</i>	17
Gambar 3.5 <i>Source Code</i> Konversi Citra Rgb Ke <i>Grayscale</i>	18
Gambar 3.6 <i>Source Code</i> Proses Tresholding.....	19
Gambar 3.7 <i>Source Code</i> Penentuan Nilai T Pada Tresholding	20
Gambar 3.8 <i>Source Code</i> Proses Binerisasi	20
Gambar 3.9 Desain Arsitektur Pembelajaran LVQ Pada Sistem.....	21
Gambar 3.10 Desain Arsitektur Autentikasi LVQ Pada Sistem	22
Gambar 3.11 Diagram Alur Algoritma LVQ.....	22
Gambar 3.12 <i>Source Code</i> Algoritma LVQ.....	24
Gambar 3.13 Hasil Dari Proses Perhitungan LVQ	24
Gambar 3.14 Contoh Gambar Tanda Tangan Satu Bobot Dan Tiga Data Latih	24
Gambar 3.15 Nilai R Pada Gambar Tanda Tangan.....	25
Gambar 3.16 Nilai G Pada Gambar Tanda Tangan	25
Gambar 3.17 Nilai B Pada Gambar Tanda Tangan.....	25
Gambar 3.18 Hasil <i>Preprocessing</i> Pada Gambar Tanda Tangan	26
Gambar 3.19 Bobot Baru Citra Tanda Tangan	27
Gambar 3.20 Alur Penyimpanan Ke Database	28
Gambar 3.21 Pengisian Informasi User	28
Gambar 3.22 <i>Source Code</i> Penyimpanan Informasi Ke Database.....	29
Gambar 3.23 Diagram Alur Proses Autentikasi TTD.....	30

Gambar 3.24 Diagram Alur Proses Autentikasi Tanda Tangan Menggunakan LVQ.	31
Gambar 3.25 Source Code Autentikasi TTD Menggunakan LVQ.	31
Gambar 3.26 Source Code Shorting	32
Gambar 3.27 Hasil Autentikasi Tanda Tangan	32
Gambar 3.28 Diagram Alur Proses Enkripsi File.	33
Gambar 3.29 Source Code Enkripsi File	33
Gambar 3.30 Diagram Alur Penyimpanan Informasi File Ke Database.....	34
Gambar 3.31 Diagram Alur Dekripsi File.....	35
Gambar 3.32 Source Code Dekripsi File	36
Gambar 3.33 Antarmuka Input Tanda Tangan	36
Gambar 3.34 Antarmuka Untuk Menggambar Tanda Tangan.....	37
Gambar 3.35 Textfield Pengisian Informasi	38
Gambar 3.36 Pesan Konfirmasi	39
Gambar 3.37 Texfield Hasil Perhitungan Bobot.....	39
Gambar 3.38 Halaman Autentikasi Tanda Tangan.....	39
Gambar 3.39 Hasil Dari Pre-Processing	41
Gambar 3.40 Hasil Dari Autentikasi.....	41
Gambar 3.41 Tampilan Tempat Untuk Menggambar Tanda Tangan	42
Gambar 3.42 Tampilan Halaman Pengaman File	43
Gambar 3.43 Tampilan Halaman Dekripsi File	45
Gambar 3.44 Halaman Muka Untuk Menggambar Tanda Tangan.....	46
Gambar 4.1 Hasil Autentikasi Tanda Tangan	50

DAFTAR TABEL

Tabel 2.1 Penelitian yang Terkait Autentikasi Tanda Tangan.....	7
Tabel 2.2 Penelitian yang Terkait Autentikasi Tanda Tangan Menggunkan LVQ.....	10
Tabel 2.3 Alphebet Awal	12
Tabel 2.4 Hasil Pergeseran Alphabet	13
Tabel 2.5 Penelitian yang Terkait Enkripsi Caesar Chiper	14
Tabel 3.1 Struktur Tabel	29
Table 3.2 Field-Field Dari Tabel “File”.....	34
Tabel 4.1 Data Yang Dugunakan Untuk Proses Testing.....	51
Tabel 4.2 Hasil Uji Coba Dengan Maksimal Epoch 10 Min Error 0.1 Dan Learning Ratenya 0.1	51
Tabel 4.3 Hasil Uji Coba Dengan Maksimal Epoch 10 Min Error 0.1 Dan Learning Ratenya 0.1	52
Tabel 4.4 Hasil Uji Coba Dengan Maksimal Epoch 100 Min Error 0.1 Dan Learning Ratenya 0.1	53
Tabel 4.5 Hasil Uji Coba Dengan Maksimal Epoch 3 Min Error 0.1 Dan Learning Ratenya 0.01	54
Tabel 4.6 Hasil Uji Coba Dengan Maksimal Epoch 10 Min Error 0.1 Dan Learning Ratenya 0.01	54
Tabel 4.7 Hasil Uji Coba Dengan Maksimal Epoch 100 Min Error 0.1 Dan Learning Ratenya 0.01	55
Tabel 4.8 Hasil Uji Coba Dengan Maksimal Epoch 3 Min Error 0.01 Dan Learning Ratenya 0.1	56
Tabel 4.9 Hasil Uji Coba Dengan Maksimal Epoch 10 Min Error 0.01 Dan Learning Ratenya 0.1	56
Tabel 4.10 Hasil Uji Coba Dengan Maksimal Epoch 100 Min Error 0.01 Dan Learning Ratenya 0.1	57
Tabel 4.11 Hasil Dari Banyaknya Gambar Yang Muncul Pada Nomor satu	58

Tabel 4.12 Hasil Dari Banyaknya Gambar Yang Muncul Pada Nomor Dua	59
Tabel 4.13 Hasil Dari Banyaknya Gambar Yang Muncul Pada Nomor Tiga	59
Tabel 4.14 Hasil Dari Banyaknya Gambar Yang Muncul Pada Nomor Empat	60
Tabel 4.15 Hasil Dari Banyaknya Gambar Yang Muncul Pada Nomor Empat	60
Tabel 4.16 Rata-Rata Waktu Autentikasi.....	61



ABSTRAK

Niam, Nur Jazilah. 2014. **AUTENTIKASI TANDA TANGAN MENGGUNAKAN METODE LVQ (LEARNING VECTOR QUANTIZATION) UNTUK KEAMANAN DOKUMEN.** Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.

Pembimbing : (1) DR. Cahyo Crysdiyan, (2) Hani Nurhayati M.T

Tanda tangan tiap-tiap orang mempunyai pola panjang dan tingkat kerumitan yang berbeda. Dari perbedaan ini maka tanda tangan dapat dikenali oleh komputer setelah melalui proses pembelajaran terlebih dahulu, sedangkan autentikasi adalah suatu sistem untuk membuktikan kebenaran identitas seseorang.

Dalam penelitian ini proses pembelajaran dan autentikasinya menggunakan metode LVQ (Learning Vector Quantization). Input tanda tangan berjumlah empat buah perorang, satu dijadikan bobot dan tiga lainnya dijadikan data latih. Tanda tangan digambar pada tempat yang sudah disediakan.

Dari penelitian serta perhitungan LVQ ini mempunyai epoch, leaning rate dan minimal error yang berbeda, dapat disimpulkan bahwa keakuratan autentikasi dari sembilan percobaan yang dilakukan tingkat akurasi dalam lima besar 80%.

Dan untuk keamanan dokumen, pada penelitian ini menggunakan metode *Caesar Chipper*, dengan *key* 3. metode yang sudah sangat lama dikenal dalam dunia enkripsi.

Kata Kunci : autentikasi, tanda tangan, lvq (Learning Vector Quantization).

ABSTRACT

Niam, Nur Jazilah. 2014. **AUTENTIKASI TANDA TANGAN MENGGUNAKAN METODE LVQ (LEARNING VECTOR QUANTIZATION) UNTUK KEAMANAN DOKUMEN.** Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
Pembimbing : (1) DR. Cahyo Crys dian, (2) Hani Nurhayati M.T

Everyone has long Signature patterns and different levels of complexity. From this the difference signature can be recognized by the computer after the first learning process, while the authentication is a system to verify the person's identity.

In this learning process and the authentication method LVQ (Learning Vector Quantization). Individual input signature has four numbered, for the first used as weight and three other and used as training data. Drawn signature to places that have been provided.

From research and calculations have LVQ this epoch, learning rate and different minimal error, it can be concluded that the authentication accuracy of the nine experiments conducted accuracy rate of 80%.

And for the security documents, in this study using the method Caesar Cipher, with three key. This Methods have been very long known in the world of encryption.

Key Word : authentication, signature, lvq (Learning Vector Quantization).

BAB I PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi digital saat ini semakin pesat, banyak penelitian dilakukan agar ditemukan penemuan-penemuan baru yang diharapkan dapat membantu manusia di segala bidang. Begitu pula di dunia citra digital, salah satunya adalah pengenalan tanda tangan manusia, dimana setiap orang mempunyai ciri dan pola yang berbeda-beda dalam menggambarinya. Dari perbedaan inilah yang nantinya akan digunakan untuk proses pengenalan tanda tangan.

Di dalam Al Qur'an Surat al Hujaarat ayat 6, Allah SWT berfirman :

يَتَأْتِيهَا الَّذِينَ ءَامَنُوا إِن جَاءَكُمْ فَاسِقٌ بِنَبَأٍ فَتَبَيَّنُوا أَن تُصِيبُوا قَوْمًا بِجَهَلَةٍ

فَتُصِيبُوا عَلَىٰ مَا فَعَلْتُمْ نَادِمِينَ ﴿٦﴾

Artinya:”Hai orang-orang yang beriman, jika datang kepadamu orang fasik membawa suatu berita, Maka periksalah dengan teliti agar kamu tidak menimpakan suatu musibah kepada suatu kaum tanpa mengetahui keadaannya yang menyebabkan kamu menyesal atas perbuatanmu itu.”

Kandungan dari ayat tersebut sangatlah jelas bahwa kita harus meneliti segala sesuatu sebelum nantinya akan merasa menyesal akibat dari ketidak telitian kita itu.

Penelitian tentang pengenalan pola saat ini berkembang pesat, karena dapat mendukung aspek keamanan suatu sistem yang dibangun. Aplikasi pengenalan pola saat ini juga sangat beragam, diantaranya adalah pengenalan pola tanda tangan, yang mempunyai ciri seperti panjang dan kerumitan yang berbeda. Salah satu metode pengenalan pola adalah Jaringan Saraf Tiruan yang salah satu diantaranya adalah LVQ (*Learning Vector Quantization*).

LVQ didesain sedemikian rupa sehingga memiliki tingkat akurasi baik dibanding algoritma yang lainnya. Begitupula performanya yang juga didesain lebih cepat sehingga tidak memakan memori yang sangat besar saat pemrosesan berlangsung.

Pengamanan file dizaman sekarang ini sangat diperlukan, agar file tersebut tidak jatuh ditangan yang salah. Kita pun tidak mau bila file yang kita punya jatuh ketangan orang yang tidak berhak dan nantinya disalah gunakan. karena itu diperlukan adanya pengamanan file.

Dalam penelitian ini diusulkan suatu model pengenalan tanda tangan menggunakan metode LVQ yang inputannya berupa file JPG . Database diigunakan untuk menyimpan hasil pembelajaran yang nantinya akan digunakan untuk proses pengenalannya. Lalu untuk pengamanan filenya digunakan enkripsi *Caesar Chiper*.

1.2 Rumusan Masalah

Adapun rumusan masalah dalam penelitian ini adalah :

- a. Apakah metode LVQ (*Learning Vector Quantization*) bisa digunakan untuk proses pembelajaran dan autentikasi tanda tangan.
- b. Berapa persenkah tingkat akurasi metode ini dalam mengenali sebuah tanda tangan.

1.3 Batasan Masalah

Diperlukan adanya batasan masalah dalam penelitian ini, agar nantinya tidak menyimpang dan melebar dari permasalahan, antara lain :

- a. Ekstensi citra yang digunakan adalah .JPG yang dihasilkan dari penggambaran tanda tangan pada tempat yang telah disediakan dan berukuran 198x168 pixel.
- b. Penelitian ini lebih dikhususkan untuk proses autentikasi tanda tangannya.
- c. *File* yang digunakan berektensi txt.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah :

- a. Mengimplementasikan algoritma LVQ (*Learning Vector Quantization*) untuk membuat aplikasi autentikasi tanda tangan.
- b. Mengukur tingkat akurasi metode LVQ dalam mengenali tanda tangan.

1.5 Manfaat Penelitian

Dengan dilakukannya penelitian ini diharapkan bisa dijadikan referensi dari penelitian yang serupa agar nantinya bisa membuat suatu aplikasi yang lebih baik dan lebih akurat dalam mengenali suatu tanda tangan.

1.6 Sistematika Penelitian

Penelitian ini tersusun dalam lima bab, yaitu sebagai berikut:

BAB I Pendahuluan

Pendahuluan, membahas tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, dan sistematika penelitian.

BAB II Landasan Teori

Landasan teori berisi tentang beberapa teori yang mendasari penelitian ini, antara lain akan membahas tentang dasar teori dari autentikasi tandatangan, algoritma LVQ dan Kriptografi *Caesar Chiper*.

BAB III Perancangan dan Implementasi Aplikasi

Berisi tentang, menganalisa kebutuhan sistem yang nantinya digunakan untuk membuat aplikasi, yaitu meliputi spesifikasi kebutuhan *software* dan langkah-langkah dalam pembuatan Aplikasi.

BAB IV Eksperimen dan Pembahasan

Menjelaskan tentang pengujian Aplikasi yang telah dibangun, serta penerapan dari algoritma LVQ tersendiri.

BAB V Penutup

Berisi tentang kesimpulan dan saran.

BAB II

TINJAUAN PUSTAKA

2.1 Autentikasi Tanda Tangan

Autentikasi adalah suatu langkah untuk menentukan atau mengonfirmasi bahwa seseorang (atau sesuatu) adalah autentik atau asli. Melakukan autentikasi terhadap sebuah objek adalah melakukan konfirmasi terhadap kebenarannya. Sedangkan melakukan autentikasi terhadap seseorang biasanya adalah untuk memverifikasi identitasnya.

Autentikasi bertujuan untuk membuktikan siapa anda sebenarnya, apakah anda benar-benar orang yang anda klaim sebagai dia (*who you claim to be*). Metode autentikasi bisa dilihat dalam 4 kategori metode:

a. *Something you know*

Ini adalah metode autentikasi yang paling umum. Cara ini mengandalkan kerahasiaan informasi, contohnya adalah *password* dan *PIN*. Cara ini berasumsi bahwa tidak ada seorangpun yang mengetahui rahasia itu kecuali anda seorang.

b. *Something you have*

Cara ini biasanya merupakan faktor tambahan untuk membuat autentikasi menjadi lebih aman. Cara ini mengandalkan barang yang sifatnya unik, contohnya adalah kartu magnetic/*smartcard*, *hardware token*, USB token dan sebagainya. Cara ini berasumsi bahwa tidak ada seorangpun yang memiliki barang tersebut kecuali anda seorang.

c. *Something you are*

Ini adalah metode yang paling jarang dipakai karena faktor teknologi dan manusia juga. Cara ini menghandalkan keunikan bagian-bagian tubuh anda yang tidak mungkin ada pada orang lain seperti sidik jari, suara atau sidik retina. Cara ini berasumsi bahwa bagian tubuh anda seperti sidik jari dan sidik retina, tidak mungkin sama dengan orang lain.

d. *Something you do*

Melibatkan bahwa setiap *user* dalam melakukan sesuatu dengan cara yang berbeda. Contoh : Penggunaan analisis suara (*voice recognition*), dan analisis tulisan tangan.

Sedangkan tanda tangan menurut Kamus Besar Bahasa Indonesia adalah tanda sebagai lambang nama yang dituliskan dengan tangan oleh orang itu sendiri sebagai penanda pribadi.

Jadi, bisa disimpulkan autentikasi tanda tangan adalah proses untuk menginformasikan apakah seseorang itu asli dengan menggunakan sebuah lambang nama yang dituliskan sendiri oleh orang yang bersangkutan.

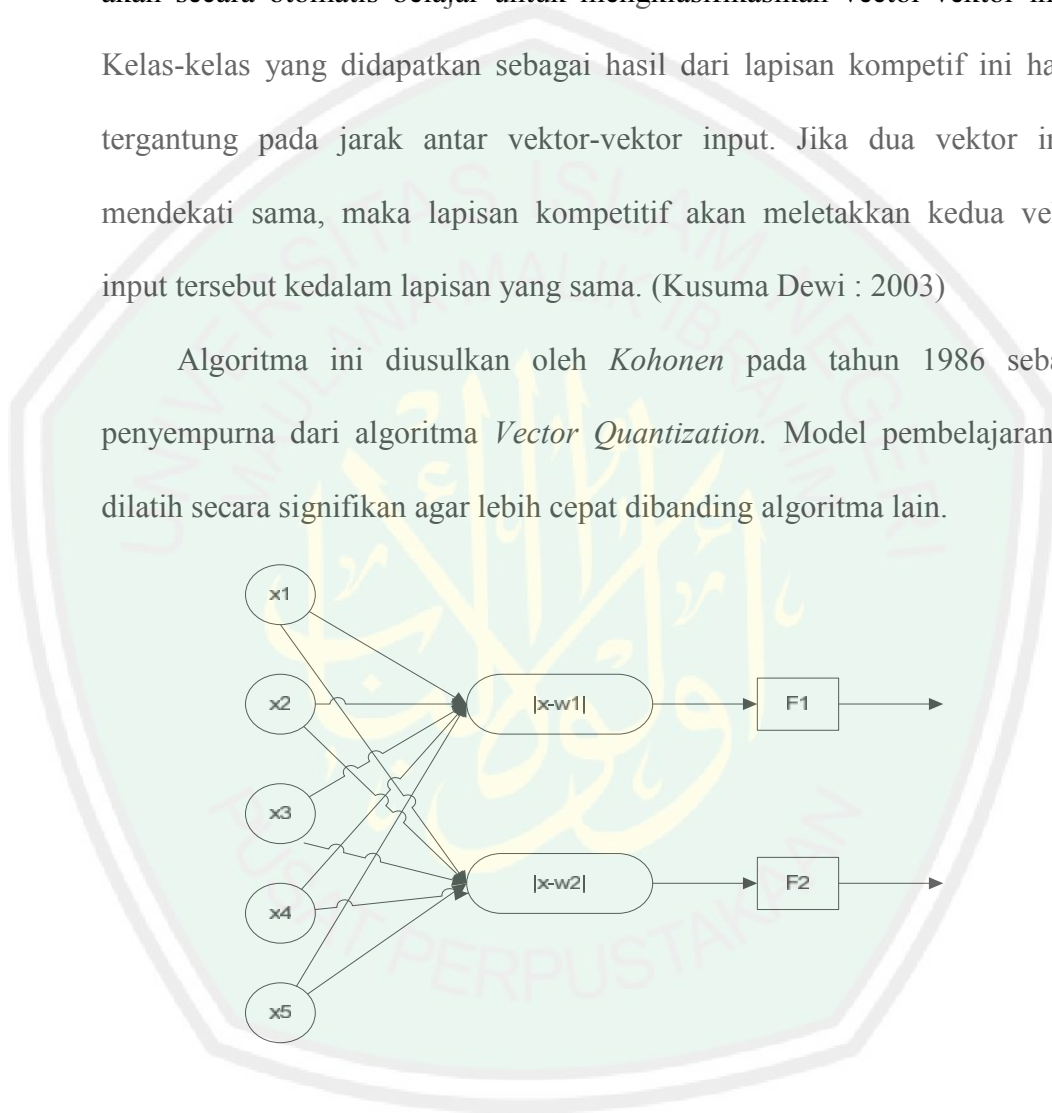
Tabel 2.1 Penelitian yang Terkait Autentikasi Tanda Tangan

No	Penulis	Tahun	Judul	Metode	Hasil/Kesimpulan
1	Frebiansyah dkk	2007	Autentikasi Tanda Tangan Menggunakan Metode C-Means	C-Means	Akurasi <ul style="list-style-type: none"> - 20 fitur hasilnya 77,5% - 26 fitur hasilnya 88,75% - 34 fitur hasilnya 92,5%
2	Andam Zainal dkk	2002	Aplikasi Neural Network pada Pengenalan Pola Tanda Tangan	Neural Network	<ul style="list-style-type: none"> - Banyaknya input yang dimasukkan akan mempengaruhi proses kerja jaringan - Keakuratan sistem untuk mengenali pola training data set mencapai 100% - Keakuratan sistem untuk pengenalan pola blind data set mencapai 0%
3	Ignatius Ricardo	-	Pengenalan Tanda Tangan Melalui Pengolahan Citra Digital dan Jaringan Saraf Tiruan RBF	Jaringan Saraf Tiruan RBF	<ul style="list-style-type: none"> - Tingkat akurasi 88% - Pengenalan dipengaruhi oleh kemiripan citra

2.2 Algoritma LVQ (*Learning Vector Quantization*)

Learning Vector Quantization adalah suatu metode untuk melakukan pembelajaran pada lapisan kompetitif yang terawasi. Suatu lapisan kompetitif akan secara otomatis belajar untuk mengklasifikasikan vector-vector input. Kelas-kelas yang didapatkan sebagai hasil dari lapisan kompetitif ini hanya tergantung pada jarak antar vektor-vektor input. Jika dua vektor input mendekati sama, maka lapisan kompetitif akan meletakkan kedua vektor input tersebut kedalam lapisan yang sama. (Kusuma Dewi : 2003)

Algoritma ini diusulkan oleh *Kohonen* pada tahun 1986 sebagai penyempurna dari algoritma *Vector Quantization*. Model pembelajaran ini dilatih secara signifikan agar lebih cepat dibanding algoritma lain.



Gambar 2.1 Arsitektur Jaringan LVQ (*Learning Vector Quantization*)

Secara garis besar algoritma LVQ sebagai berikut (Darma Putra: 2010):

- a. Menentukan masing-masing kelas *output*, bobot, dan *learning rate*.
- b. Bandingkan masing-masing input dan masing-masing bobot yang telah ditetapkan dengan melakukan pengukuran jarak antara masing-masing bobot dan input
- c. Nilai Min dari hasil perbandingan itu akan menentukan kelas dari *vector* input dan perubahan bobot dari kelas tersebut. perubahan bobot baru dapat dihitung dengan persamaan berikut :
 - Untuk input dan bobot yang memiliki kelas yang sama :

$$w_o' = w_o + \alpha(x - w_o)$$
 - Untuk input dan bobot yang memiliki kelas yang berbeda :

$$w_o' = w_o - \alpha(x - w_o)$$

Pada dasarnya perhitungan diatas akan dilakukan terus-menerus sampai nilai bobot tidak berubah jika ada input baru, hal ini tentu saja memerlukan memori yang besar, untuk itu, dalam melakukan perhitungan LVQ diperlukan adanya *epoch* atau disebut juga perulangan maksimal yang ditempatkan diawal.

LVQ termasuk *Single Layer Feedforward* karena itu tidak ada layer tersembunyi didalamnya. Hanya ada satu layer dengan bobot-bobot terhubung. Layer ini menerima input kemudian secara langsung diolah menjadi lapisan output.

Didalam pengenalan tanda tangan, kumpulan tanda tangan atau yang disebut template dipelajari terlebih dahulu sebelum dilakukan pencocokan.

Tabel 2.2 Penelitian yang Terkait Autentikasi Tanda Tangan Menggunakan LVQ

No	Penulis	Tahun	Judul	Metode	Hasil/Kesimpulan
1	Afif Rakhmanullah	2010	Autentifikasi Pengenalan Pola Tanda Tangan Manual Menggunakan Jaringan Saraf Tiruan LVQ (Learning Vector Quantization) dan Tanda Tangan Digital Menggunakan Algoritma RSA (Riset Shamir Adleman)	Jaringan Saraf Tiruan LVQ (Learning Vector Quantization) dan Algoritma RSA (Riset Shamir Adleman)	<ul style="list-style-type: none"> - Tingkat keberhasilan LVQ dipengaruhi oleh 3 inputan: maksimal epoch, learning rate, target error - Algoritma RSA menghasilkan tingkat keberhasilan sebesar 100 %, dan 85 % untuk tanda tangan manual
2	S.Sutikno	-	Aplikasi Pengenalan Tanda Tangan dengan Metode Learning Vektor Quantization dan Backpropagation	Learning Vektor Quantization dan Backpropagation	<ul style="list-style-type: none"> - Proses pelatihan pada backpropagation membutuhkan epoch lebih banyak sehingga membutuhkan waktu yang lebih lama dari LVQ - Untuk pengenalan tanda tangan LVQ bekerja lebih baik dari Backpropagation
3	Prasetya, Reynatha Yudha	2001	Pengenalan Tanda Tangan Menggunakan Jaringan Saraf Tiruan LVQ	Jaringan Saraf Tiruan LVQ	<ul style="list-style-type: none"> - Akurasi dilakukan pembelajaran 73,91% - Belum dilakukan pembelajaran 69,56%

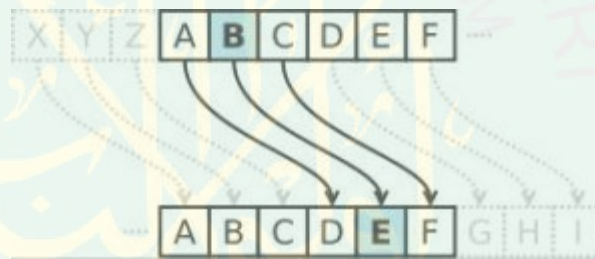
2.3 Enkripsi *Caesar Chiper*

Enkripsi yaitu suatu proses pengaman suatu data yang disembunyikan atau proses konversi data (*plaintext*) menjadi bentuk yang tidak dapat dibaca atau dimengerti. Di pertengahan tahun 1970-an enkripsi kuat dimanfaatkan untuk pengamanan oleh sekretariat agen pemerintah *Amerika Serikat* pada domain publik. Informasi yang asli disebut sebagai *plaintext*, dan bentuk yang sudah dienkripsi disebut sebagai *chiphertext*. Pesan *chiphertext* berisi seluruh informasi dari pesan *plaintext*, tetapi tidak dalam format yang didapat dibaca manusia ataupun komputer tanpa menggunakan mekanisme yang tepat untuk melakukan dekripsi. Sedangkan Dekripsi yaitu kebalikan dari proses enkripsi yaitu proses konversi data yang sudah dienkripsi (*ciphertext*) kembali menjadi data aslinya (*Original Plaintext*) sehingga dapat dibaca atau dimengerti kembali.

Algoritma enkripsi digunakan pada saat melakukan proses enkripsi terhadap suatu *plaintext* dan algoritma dekripsi digunakan pada saat melakukan proses dekripsi terhadap suatu *ciphertext*. Sedangkan dalam penerapannya algoritma enkripsi dan algoritma dekripsi harus menggunakan kunci untuk membuka dan menutup sandinya, hal ini untuk menjaga keamanan data atau informasi tersebut. Kunci yang dimaksud dapat dilambangkan dengan K . Kunci yang digunakan dapat berupa sebuah angka bernilai kecil atau besar sesuai dengan angka-angka yang telah ditentukan untuk sebagai nilai transformasi matematis yang memetakan *plaintext* ke *ciphertext* dan sebaliknya. *Ciphertext* sangat dipengaruhi oleh keberadaan

plaintext dan kuncinya, jadi nilai dari suatu kunci akan mempengaruhi fungsi enkripsi dan dekripsi,

Sandi *Caesar*, atau sandi geser, kode *Caesar* atau Geseran *Caesar* adalah salah satu teknik enkripsi paling sederhana dan paling terkenal. Sandi ini termasuk sandi substitusi dimana setiap huruf pada teks terang (*plaintext*) digantikan oleh huruf lain yang memiliki selisih posisi tertentu dalam alfabet. Misalnya, jika menggunakan geseran 3, W akan menjadi Z, I menjadi L, dan K menjadi N. Nama *Caesar* diambil dari *Julius Caesar*, jenderal, konsul, dan diktator Romawi yang menggunakan sandi ini untuk berkomunikasi dengan para panglimanya.



Gambar 2.2 Enkripsi *Caesar Chiper*

Contoh secara detail pergeseran alphabet dicontohkan pada tabel 2.1 dan 2.2.

Tabel 2.3 Alphebet Awal

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Jika digeser tiga angka maka akan menjadi :

Tabel 2.4 Hasil Pergeseran Alphebet

D	E	F	G	H	I	J	K	L	M	N	O	P
0	1	2	3	4	5	6	7	8	9	10	11	12
Q	R	S	T	U	V	W	X	Y	Z	A	B	C
13	14	15	16	17	18	19	20	21	22	23	24	25



Tabel 2.5 Penelitian yang Terkait Enkripsi Caesar Chiper

No	Penulis	Tahun	Judul	Metode	Hasil/Kesimpulan
1	Andy Nugroho	2012	Implementasi Algoritma Caesar Cipher Rot13 Dan Base64 Untuk Enkripsi Dan Dekripsi Pesan Sms Pada Handphone Berbasis Android	Caesar Cipher Rot13 Dan Base64	<ul style="list-style-type: none"> - Gabungan enkripsi ini bisa diaplikasikan di mobile - Metode enkripsi harus disesuaikan dengan kebutuhan tingkat keamanan
2	Rifky Chandra U	2013	Implementasi Algoritma Enkripsi Abacusian Dan Caesar Cipher Pada Sms Dan Koordinat Lokasi Berbasis Android Menggunakan App Inventor	Enkripsi Abacusian Dan Caesar Cipher	Dengan dua pilihan cipher, maka pengguna memiliki pilihan untuk menentukan kekuatan sistem enkripsi, bila menggunakan kombinasi keduanya tentu bisa lebih baik.
3	Donny Seftianto dkk	2012	Peran Algoritma Caesar Cipher Dalam Membangun Karakter Akan Kesadaran Keamanan Informasi	Caesar Cipher	Kriptografi yang mudah di implementasikan pada kehidupan sehari-hari secara nyata dengan rumus matematika yang tidak sulit untuk di mengerti dan untuk mewujudkan hal tersebut, diperlukan adanya pengenalan kriptografi khususnya Caesar cipher.



BAB III

PERANCANGAN DAN IMPLEMENTASI APLIKASI

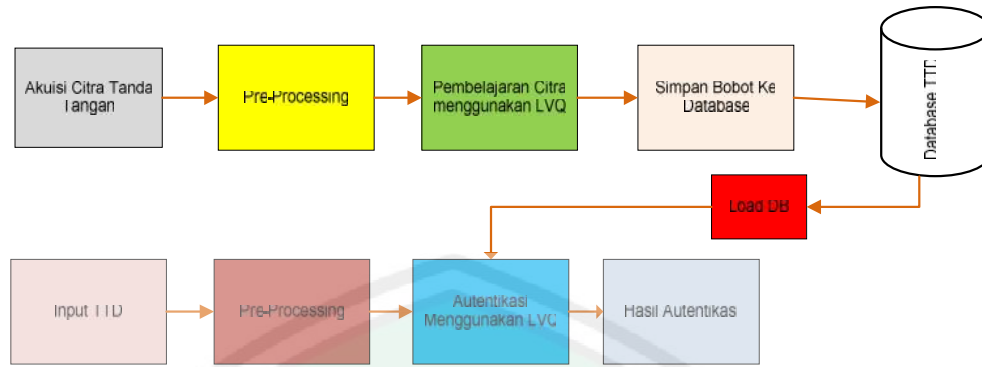
3.1 Desain Proses

Input dalam penelitian ini berupa citra tanda tangan berekstensi JPG yang berukuran 198x168. Setelah itu dilakukan preprocessing yang hasil akhirnya berupa matrix biner yang nantinya akan digunakan untuk proses pembelajaran citra menggunakan algoritma LVQ. *Preprocessing* sendiri terdiri dari beberapa proses, yang pertama adalah merubah citra *RGB* menjadi *grayscale*, *tresholding* lalu merubah citra menjadi binari agar nantinya bisa diproses.

Setelah *preprocessing* selesai dilakukan, barulah proses pembelajaran menggunakan metode LVQ dikerjakan. Dari delapan input gambar, dua input dijadikan bobot awal dan tiga lainnya digunakan untuk data latih. Perhitungan ini menghasilkan sebuah bobot baru yang disimpan dalam database yang nantinya akan digunakan saat proses autentikasi tanda tangan.

Dalam autentikasi prosesnya hampir sama dengan proses pembelajaran, input harus melalui *preprocessing*, setelah itu dilakukan pencocokan bobot yang sudah ada didalam *database*, hasil dari perhitungan ini adalah nilai jarak Min dari setiap bobot. nilai tersebut lalu diranking menjadi lima besar teratas yang menghasilkan lima gambar teratas yang paling cocok dengan gambar yang diinputkan.

Berikut ini adalah diagram blok dari aplikasi yang dibangun :



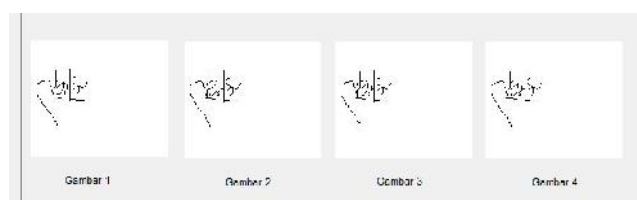
Gambar 3.1 Diagram Blok Aplikasi Autentikasi Tanda Tangan Menggunakan Algoritma LVQ

Penjelasan :

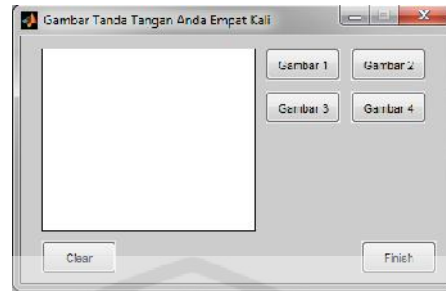
Proses pertama adalah input citra dilanjutkan dengan dengan *preprocessing* lalu pembelajaran menggunakan LVQ, hasil dari pembelajaran tersebut adalah bobot baru yang akan disimpan di database. Sedangkan untuk proses autentikasi citra tanda tangan dirubah menjadi citra biner melalui *preprocessing* lalu dicocokkan dengan menggunakan LVQ.

3.1.1 Akuisi Citra Tanda Tangan

Proses ini merupakan proses paling awal untuk mendapatkan citra tanda tangan. Terdapat dua macam cara yang disediakan untuk mendapatkan gambar tanda tangan, yang pertama dengan menggambar sendiri ke delapan tanda tangan di tempat yang sudah disediakan, dan yang ke dua memilih gambar tanda tangan yang sudah disediakan. Nantinya citra yang dihasilkan akan berektensi *.JPG dan berukuran 198x168 pixel.



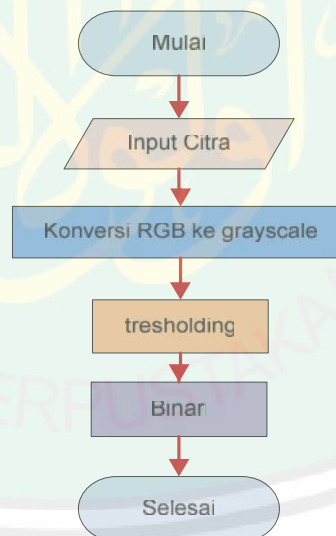
Gambar 3.2 Hasil Penggambaran Tanda Tangan



Gambar 3.3 Tempat Untuk Membuat (Menggambar) Tanda Tangan

3.1.2 Preprocessing Citra

Preprocessing dilakukan untuk mempermudah pemrosesan berikutnya dan menyiapkan citra untuk menghasilkan cirri yang lebih baik. Dalam aplikasi ini pencari/ seleksi fitur dimasukkan kedalam proses ini, yang bertujuan untuk menemukan karakteristik dari sebuah gambar. Gambar 3.4 adalah diagram alur *preprocessing* pada aplikasi ini.



Gambar 3.4 Diagram Alur *Preprocessing*

Dalam *preprocessing* dibagi menjadi beberapa proses, yaitu citra *RGB* dirubah menjadi *grayscale* agar bisa dilakukan *thresholding*, setelah itu citra hasil *thresholding* dirubah menjadi citra binari.

a. *RGB ke Grayscale*

Citra berwarna (*true color*) merupakan kombinasi dari tiga warna dasar, yaitu *Red* (R), *Green* (G) dan *Blue* (B). Citra *RGB* ini lalu dikonversikan menjadi citra *grayscale* atau citra keabuan agar citra lebih sederhana, Citra *grayscale* mempunyai tiga warna, yaitu hitam, abu-abu dan putih, tingkat keabuan suatu pixel bervariasi yang letaknya diantara warna hitam dan putih. Tiap pixel gambar hanya mempunyai satu nilai. Untuk mendapatkan nilai *grayscale*, kita dapat mengambil nilai rata-rata dari nilai R G B, sehingga dapat dirumuskan menjadi :

$$S=R+G+B$$

Berikut ini adalah *source code* konversi citra dari *RGB* ke *grayscale*.

```
data_proses1=uint8(zeros(size(image_input1,1),size(image_input1,2)));
for i=1:size(image_input1,1)
    for j=1:size(image_input1,2)
        data_proses1(i,j)=0.2989*image_input1(i,j,1)+0.5870*image_input1(i,j,2)+0.1140*image_input1(i,j,3);
    end
end
```

Gambar 3.5 *Source Code* Konversi Citra *RGB* Ke *Grayscale*

b. *Thresholding*

Thresholding termasuk salah satu teknik segmentasi citra, yaitu suatu teknik untuk membagi suatu citra menjadi beberapa daerah yang memiliki kemiripan atribut.

Thresholding memisahkan citra ke dalam daerah intensitasnya masing-masing sehingga bisa dibedakan antara objek dan *background*.

Thresholding berfungsi untuk mengatur derajat keabuan pada citra. Secara umum proses *tresholding* sebagai berikut (Darma Putra : 2010):

$$g(x,y)=\begin{cases} 1 & \text{if } f(x,y) \geq T \\ 0 & \text{if } f(x,y) < T \end{cases}$$

Dengan $g(x,y)$ adalah citra dari citra *grayscale* $f(x,y)$. dan T menyatakan nilai ambang, Nilai T memegang peranan penting dalam proses ini, karena kualitas dari hasil citra tergantung pada nilai T yang digunakan.

Ada dua jenis *tresholding*, yaitu pertama *global tresholding* yaitu seluruh pixel pada citra dikonversikan menjadi 1 atau 0 dengan satu nilai ambang T , dan yang kedua *locally tresholding* yaitu suatu citra dibagi menjadi blok-blok kecil dan kemudian dilakukan *tresholding* pada setiap blok-blok dengan nilai T yang berbeda-beda. Dalam penelitian ini *tresholding* yang digunakan adalah *global tresholding*.

Berikut ini adalah *source code* dari proses *tresholding*.

```
threshold = thres(data_proses1);
baris = size(data_proses1,1);
kolom = size(data_proses1,2);
for i=1:baris
    for j=1:kolom
        if data_proses1(i,j) > threshold
            data_proses1(i,j)=255;
        else
            data_proses1(i,j)=0;
        end
    end
end
```

Gambar 3.6 *Source code* proses *tresholding*

```

function T= thres(data_proses1)

T= 0.5* (double(min(data_proses1(:)) +
double(max(data_proses1(:))));
done= false;
while ~done
    g= data_proses1 >= T;
    Tnext = 0.5 * (mean(data_proses1(g)) + mean
(data_proses1(~g)));
    done = abs (T-Tnext) < 0.5;
    T = Tnext;
end

```

Gambar 3.7 Source Code Penentuan Nilai T Pada Thresholding

c. Citra Biner

Citra biner adalah citra digital yang hanya memiliki dua kemungkinan nilai pixel yaitu hitam dan putih. Citra biner juga sering disebut citra B&W (*black and white*), citra ini hanya membutuhkan 1 bit untuk mewakili setiap nilai pixel. (Darma Putra : 2010)

Berikut ini adalah *source code* proses binerisasi.

```

sz1=size(data_proses1);
bina1=zeros(size(data_proses1));

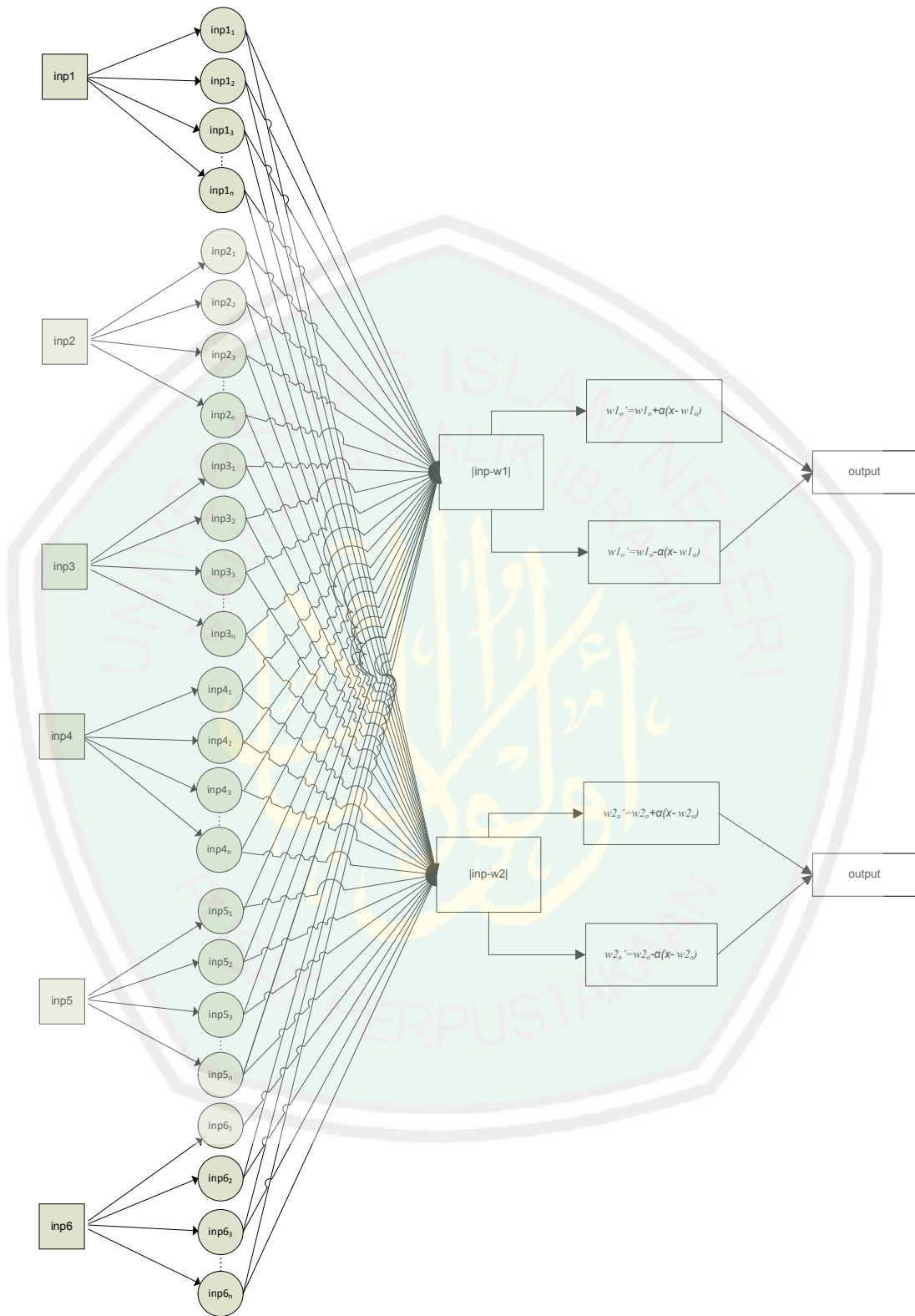
for i2=1:sz1(1)
    for j2=1:sz1(2)
        if(data_proses1 (i2,j2)<threshold)
            bina1(i2,j2)=1;
        end
    end
end
end

```

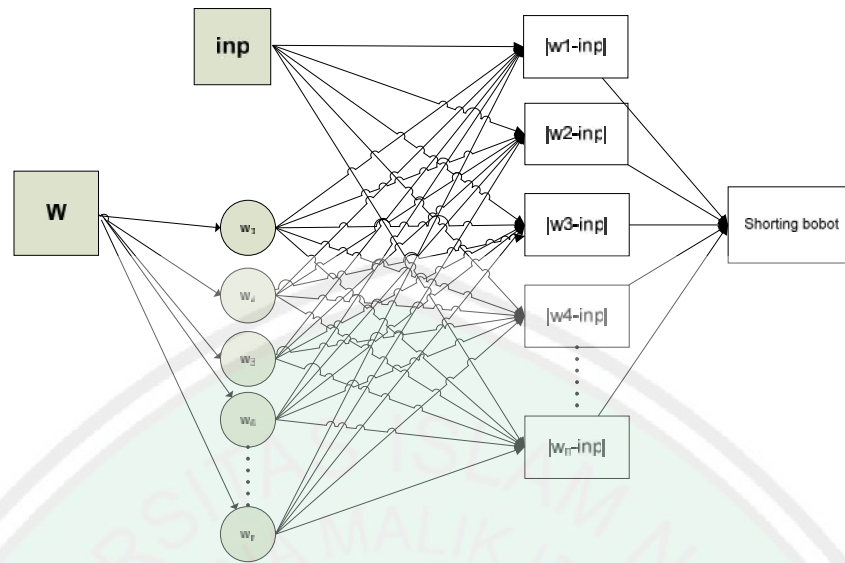
Gambar 3.8 Source code proses binerisasi

3.1.3. Arsitektur Algoritma LVQ Pada Sistem

Pada penelitian ini arsitektur aplikasi dibagi menjadi dua, yaitu arsitektur pembelajaran citra tanda tangan dan autentikasi citra tanda tangan. Berikut ini adalah arsitektur sitem dari aplikasi ini :

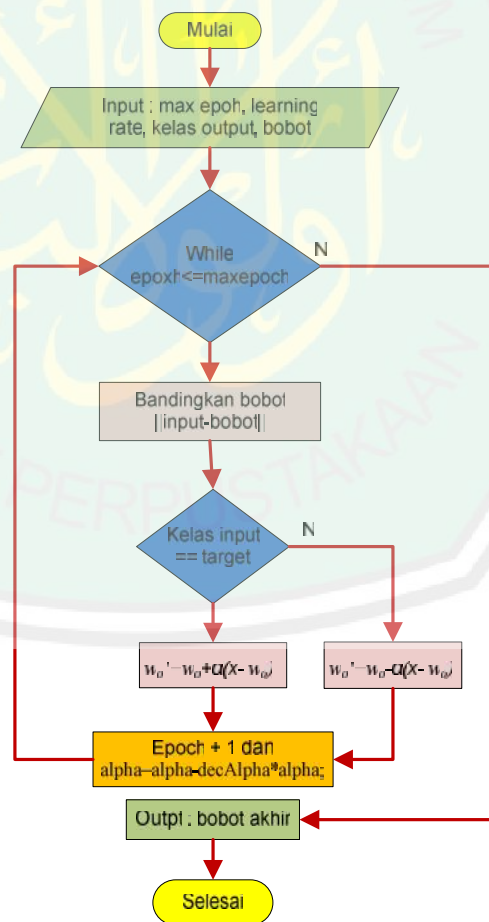


Gambar 3.9 Desain Arsitektur Pembelajaran LVQ Pada Sistem



Gambar 3.10 Desain Arsitektur Autentikasi LVQ Pada Sistem

3.1.4 Pembelajaran Citra Menggunakan Metode LVQ



Gambar 3.11 Diagram Alur Algoritma LVQ

Setelah didapatkan citra biner proses selanjutnya adalah pembelajaran citra yang sudah diinputkan.

Penjelasan dari alur Algoritma LVQ adalah sebagai berikut :

- Gambar tanda tangan berektensi .JPG dan berukuran 198x168.
- Gambar lalu di *preprocessing* hingga menghasilkan citra biner.
- Tentukan citra tanda tangan sebagai inisialisasi bobot dan lainnya sebagai data pelatih.
- Tentukan kelas, *learning rate*, *max epoch*
- Bandingkan bobot dengan data pelatih, dengan persamaan $|\text{input}-\text{bobot}|$
- Hitung nilai Min dari perbandingan sebelumnya, bila data latih dan bobot mempunyai kelas yang sama, maka untuk membuat bobot baru digunakan persamaan $w_o' = w_o + \alpha(x - w_o)$ namun bila tidak persamaannya adalah $w_o' = w_o - \alpha(x - w_o)$
- Perhitungan dilakukan terus menerus sampai *epoch* lebih kecil sama dengan *max epoch*.
- Bila *epoch* \geq *maxepoch* maka perulangan selesai dan bobot akhir akan disimpan di *database*.

Berikut ini adalah *source code* proses pembelajaran dari algoritma LVQ:

```
function w=pembelajaran_lvq(P,T)

Platih=[P(2,:);P(3,:);P(4,:);P(5,:);P(6,:);P(7,:)];%input
Tlatih=T;%target
w=[P(1,:);P(8,:)];
[x,y]=size(Platih);
z=size(w,1);

%nilai awal
alpha=0.01;
decAlpha=0.1;
minAlpha=0.001;
maksEpo=200;
epoh =1;
```

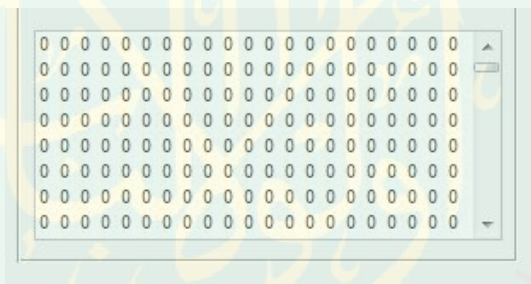
```

while (epoch<=maksEpoch) || (alpha>=minAlpha)
    for ii=1:x % inputan
        for jj=1:z % bobot awal
            J(jj)=0;
            for k=1:y % jumlah angka pada input
                J(jj)=J(jj)+(Platih(ii,k)-w(jj,k))^2;
            End
            J(jj)=sqrt(J(jj));
        end
        [Jmin, idx]= min(J);

        if idx == Tlatih(ii);
            w(idx,:)=w(idx,:)+alpha*(Platih(ii,:)-
w(idx,:));
        else
            w(idx,:)=w(idx,:)- alpha*(Platih(ii,:)-
w(idx,:));
        end
        end
        alpha=alpha-decAlpha*alpha;
        epoch=epoch+1;
    end

```

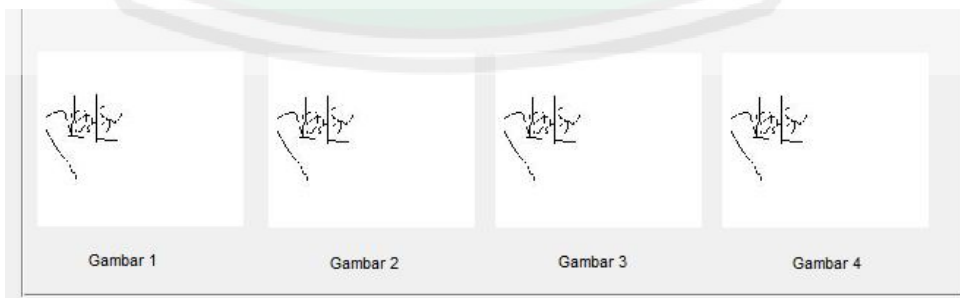
Gambar 3.12 Source code Algoritma LVQ



Gambar 3.13 Hasil Dari Proses Perhitungan LVQ

Contoh Perhitungan Manual

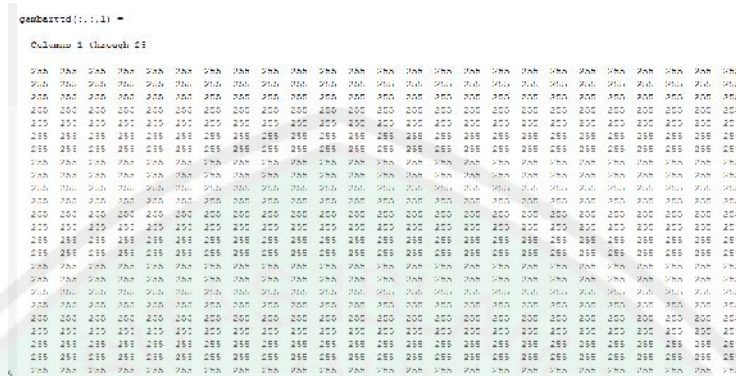
Input :



Gambar 3.14 Contoh Gambar Tanda Tangan Satu Bobot Dan Tiga Data Latih

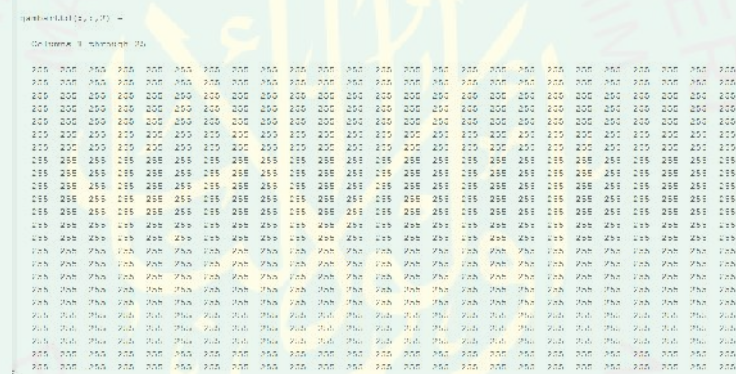
Contoh nilai pixel dari gambar nomer satu :

Nilai R



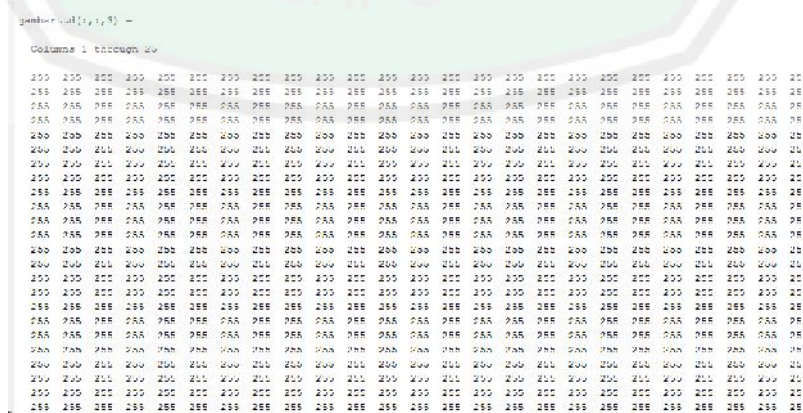
Gambar 3.15 Nilai R Pada Gambar Tanda Tangan

Nilai G



Gambar 3.16 Nilai G Pada Gambar Tanda Tangan

Nilai B



Gambar 3.17 Nilai B Pada Gambar Tanda Tangan

Hasil Preprocessing

Dataset: 1 through 21

Gambar 3.18 Hasil *Preprocessing* Pada Gambar Tanda Tangan

Hasil dari Perhitungan LVQ

Sebagai Nilai awal dipilih *Learning Rate* (α) = 0.01, dengan pengurangan sebesar $0.1 * \alpha$, dan maksimum epoch (*MaxEpoch*) = 5

Pembelajaran

Rumus : $\sqrt{(\text{Data}-\text{Target})^2}$

Data ke-1 Hasil Perhitungannya adalah :

- Epoch ke-1 = 27.9821.
- Epoch ke-2 = 31.7648.
- Epoch ke-3 = 27.9821.
- Epoch ke-4 = 31.7648.
- Epoch ke-5 = 31.7648.
- Epoch ke-6 = 31.7648.
- Epoch ke-7 = 31.7648.

Data ke-2 Hasil Perhitungannya adalah :

- Epoch ke-1 = 32.5730.
- Epoch ke-2 = 32.1714.
- Epoch ke-3 = 32.5730.

- Epoch ke-4 = 32.1714.
- Epoch ke-5 = 32.1714.
- Epoch ke-6 = 32.1714.
- Epoch ke-7 = 31.7648.

Data ke-3 Hasil Perhitungannya adalah :

- Epoch ke-1 = 36.0416.
- Epoch ke-2 = 32.
- Epoch ke-3 = 36.0416.
- Epoch ke-4 = 32.
- Epoch ke-5 = 32.
- Epoch ke-6 = 32.
- Epoch ke-7 = 32.

Min Bobotnya adalah 32.

Bobot baru gambar :

Columns 7941 through 7980	Columns 7981 through 7990	Columns 7991 through 8000	Columns 8001 through 8010	Columns 8011 through 8040	Columns 8041 through 8060	Columns 8061 through 8080
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0

Gambar 3.19 Bobot Baru Citra Tanda Tangan

Hasil dari perhitungan LVQ akan disimpan didalam database, berikut ini adalah diagram alur dari proses penyimpanan bobot ke *database* :



Gambar 3.20 Alur Penyimpanan Ke *Database*

Ada enam *field* dalam tabel, yaitu id user, nama user, jenis kelamin, target kelas, nama gambar, dan bobot akhir dari perhitungan lvq. *Textfield* satu sampai lima diisi oleh user, sedangkan *textfield* ke enam tersisi dari perhitungan lvq. Saat data sudah tersimpan dalam database maka akan muncul pesan konfirmasi bahwa data sudah tersimpan.

Informasi Tanda Tangan	
Id Pengguna	51
Nama	Nur Jazilah Niam
Jenis Kelamin	PR
Target Kelas Gambar	51
Nama Gambar	zila

Gambar 3.21 Pengisian Informasi *User*

Sever database yang digunakan di dalam aplikasi ini adalah phpmyadmin, databasnya secara lengkap bisa dilihat di dalam lampiran. Berikut ini adalah *source code* proses penyimpanan informasi *user* ke dalam *database*.

```

javaaddpath('F:\skripsiZila\programbaru\mysql-connector-java-
5.1.6-bin.jar')
%# connection parameteres
host = 'localhost';
user = 'root';
password = 'root';
dbName = 'skripsi';
%# JDBC parameters
jdbcString = sprintf('jdbc:mysql://%s/%s', host, dbName);
jdbcDriver = 'com.mysql.jdbc.Driver';

%# Create the database connection object
conn = database(dbName, user , password, jdbcDriver, jdbcString);

colnames = {'id', 'nama_u', 'jk', 'kelas', 'nama_g', 'bobot'};
exdata = {id, nama_u, jk, kelas, nama_l, bobt_bi};

insert(conn, 'ttd', colnames, exdata);

```

Gambar 3.22 Source Code Penyimpanan Informasi Ke Database

Pada aplikasi ini database diberi nama “db_skripsi”. Di dalam tabel berisi tentang informasi pemilik dan tanda tangan. *Field* pada tabel seperti pada tabel 3.1 dibawah ini.

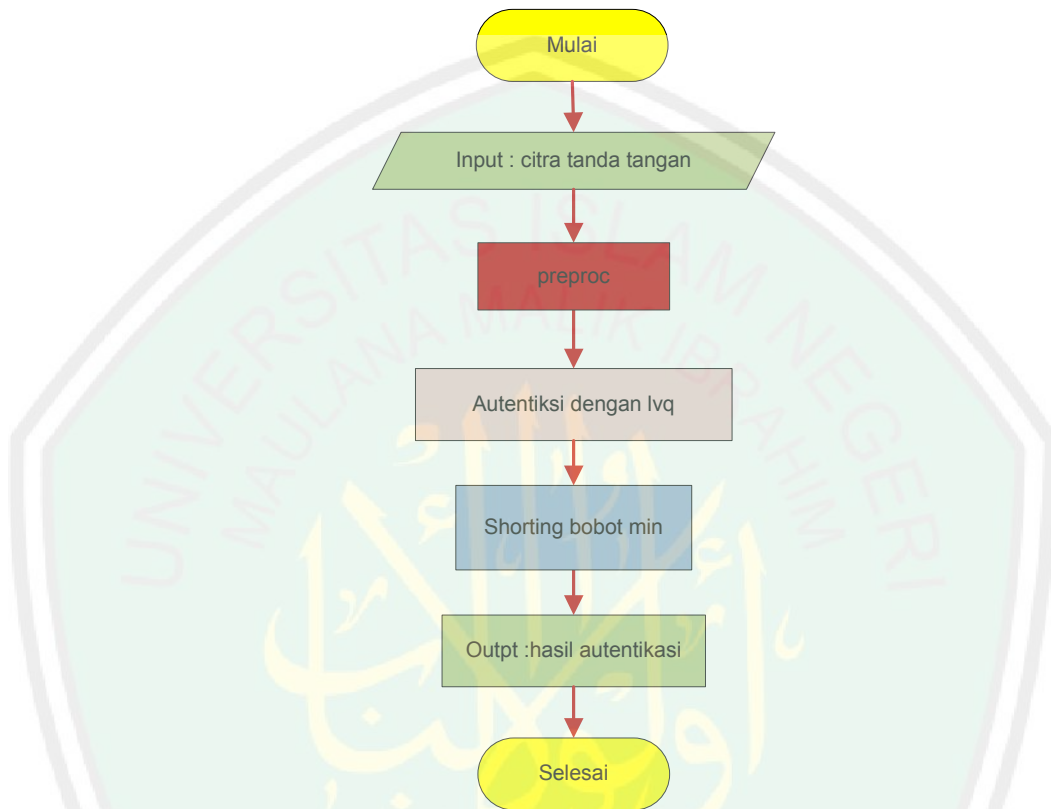
Table 3.1 Struktur Tabel .

Field	Tipe
Id	Int(5)
nama_u	Varchar(30)
jk	Varchar(2)
kelas	Varchar(5)
nama_g	Varchar(30)
bobot	longtext

3.1.5 Autentikasi Tanda Tangan Menggunakan LVQ

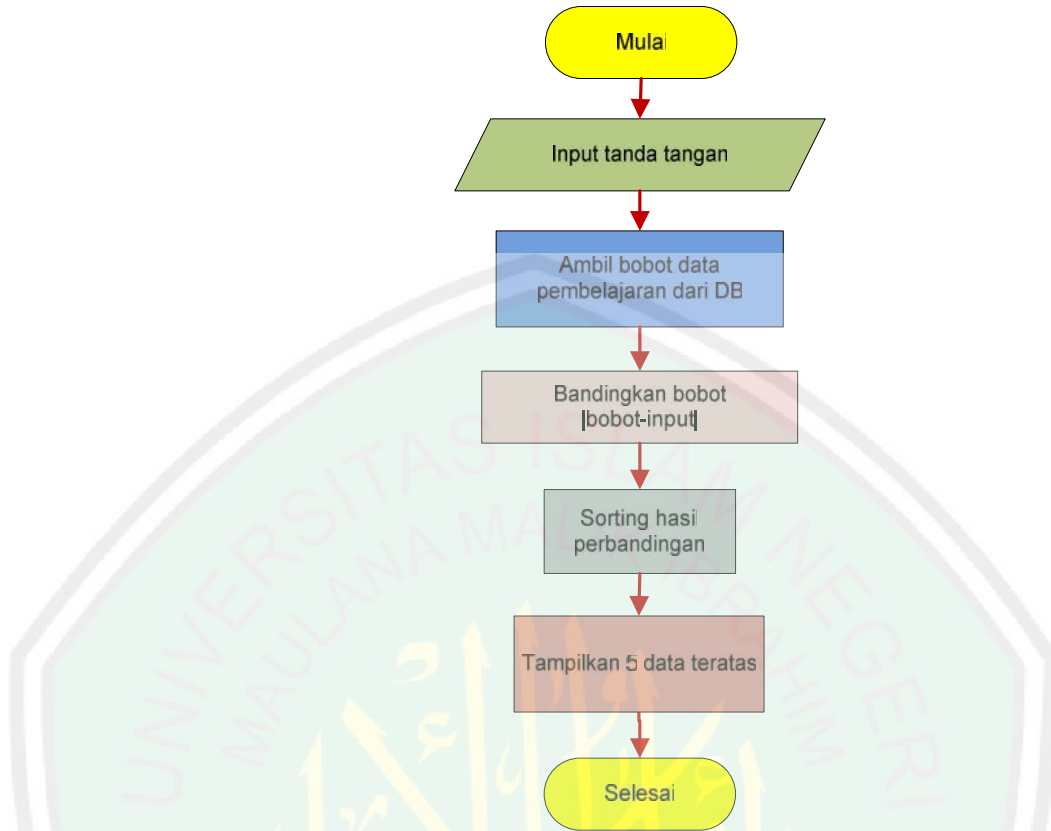
Proses autentikasi pada aplikasi ini adalah proses pencocokan citra tanda tangan yang diinputkan dengan citra yang sudah dilakukan pembelajaran yang hasilnya sudah disimpan di *database*.

Hasil pencocokan diranking menjadi lima teratas dari hasil perhitungan yang paling cocok. Gambar berikut adalah diagram alur proses autentikasi tanda tangan pada sistem :



Gambar 3.23 Diagram Alur Proses Autentikasi TTD

Proses autentikasi ini hampir sama dengan proses pembelajaran dengan LVQ yang sudah dibahas sebelumnya, yaitu pertama menginputkan citra tanda tangan yang akan dicocokkan dengan citra data yang sudah ada didatabase, yaitu citra yang sudah dilakukan pembelajaran sebelumnya, setelah itu dilakukan *preprocessing* agar citra berubah menjadi citra biner. Baru setelah itu dilakukan pencocokan dengan menggunakan algoritma LVQ, Berikut ini diagram alur dari autentikasi menggunakan lvq.



Gambar 3.24 Diagram Alur Proses Autentikasi Tanda Tangan Menggunakan LVQ.

Pertama-tama tentukan bobot, kelas dan citra input yang akan dilakukan pencocokan, Input citra gambar berektensi JPG dan berukuran 198x168 pixel yang sebelumnya sudah dirubah menjadi citra biner, setelah itu citra input dibandingkan dengan lima puluh bobot gambar yang sudah dilakukan pembelajaran yang berada didalam database.

Berikut ini adalah *source code* proses autentikasi menggunakan LVQ.

```

for k=1:x
  for ii=1:z
    J(ii)=0;
    for jj=1:y
      J(ii)=J(ii)+(hsl_bo(ii,jj)-inp(k,jj))^2;
    end
    J(ii)=sqrt(J(ii));
  end
end
  
```

Gambar 3.25 *Source Code* Autentikasi TTD Menggunakan LVQ.

Setelah mendapatkan hasil dari perbandingan bobot dan input dilakukan *shorting* dari bobot yang paling terkecil sampai yang terbesar, lalu dipilih lima nilai terkecil untuk ditampilkan gambarnya.

Berikut ini adalah *source code* proses *shorting*.

```

hasil=sort(J);
hasil1=hasil(1);
hasil2=hasil(2);
ke2=find(ismember(J,hasil2),1);
hasil3=hasil(3);
ke3=find(ismember(J,hasil3),1);
hasil4=hasil(4);
ke4=find(ismember(J,hasil4),1);
hasil5=hasil(5);
ke5=find(ismember(J,hasil5),1);

```

Gambar 3.26 *Source Code Shorting*

No	Signature Image	ID	Name	Gender
1	zila1.jpg	1	Nur Jaziloh Niom	PR
2	srleht1.jpg	6	Sholeh H	LK
3	riza1.jpg	3	Riza Yanuar	LK
4	agifi.jpg	4	Agil Saundra	LK
5	ido1.jpg	5	Idc	PR

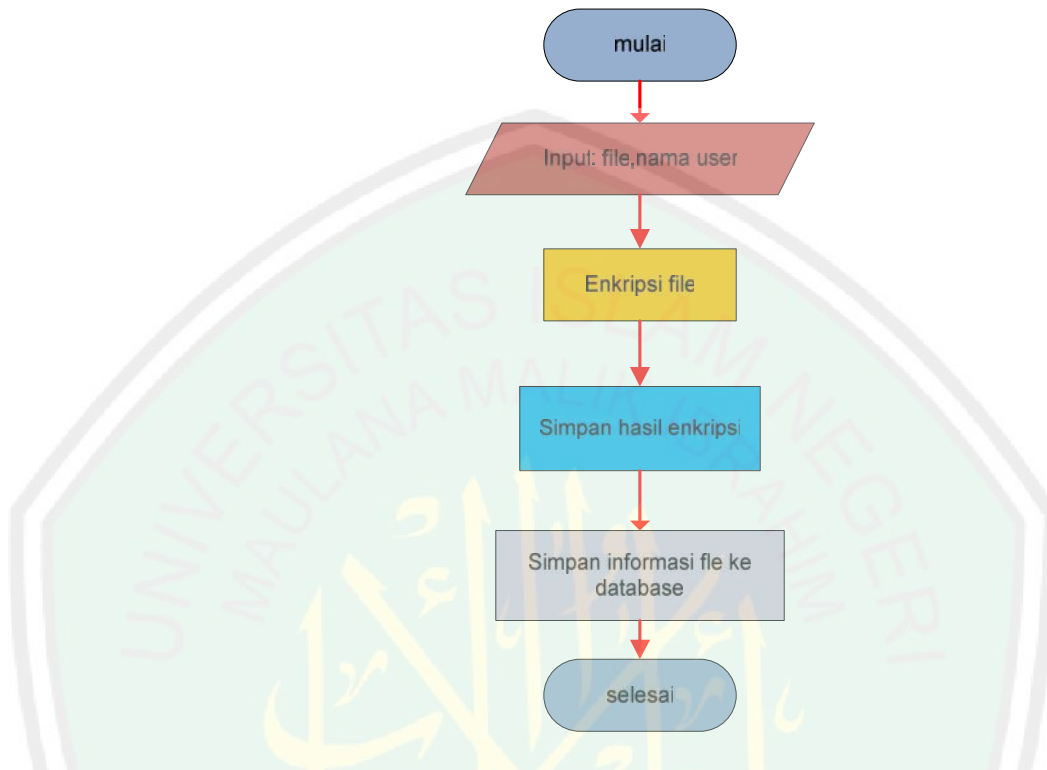
Gambar 3.27 Hasil Autentikasi Tanda Tangan

3.2 Keamanan File

3.2.1 Enkripsi File

Pengamanan file dalam penelitian ini menggunakan enkripsi *Caesar Chipper* dengan kunci pergeseran sebanyak 3 huruf. File berupa txt yang nanti setiap hurufnya akan diganti dengan huruf yang berada tiga blok setelahnya. Sebelum itu *user* diminta memasukkan nama *user* yang nantinya

akan digunakan untuk mendekripsikan *file* tersebut. Berikut ini merupakan diagram alur proses pengenkripsian *file*.



Gambar 3.28 Diagram Alur Proses Enkripsi *File*.

Berikut ini *source code* pengenkripsian *file*.

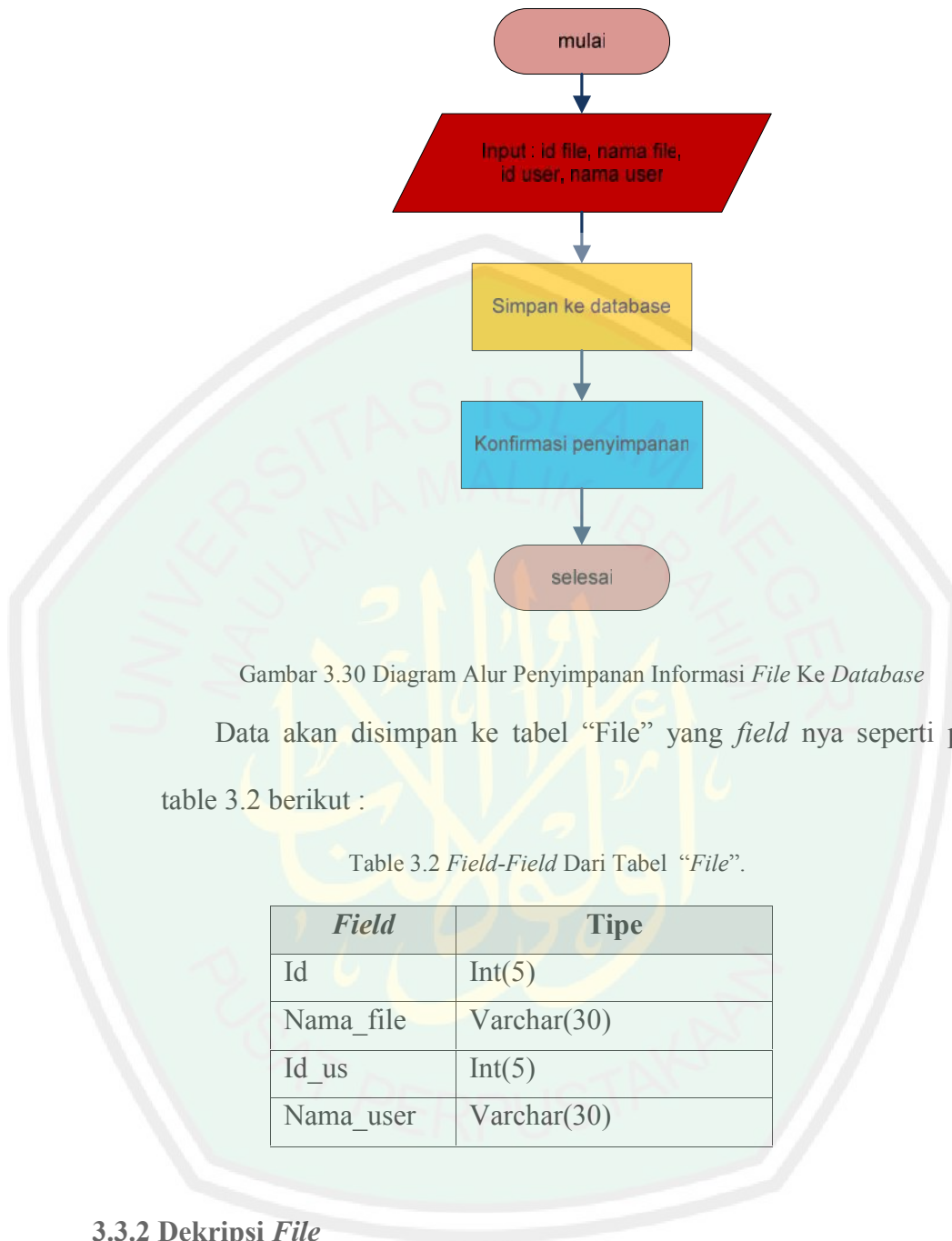
```

fil=tempat_file;
fid = fopen(fil);
tline = fgets(fid);
while ischar(tline)
disp(tline)
C=tline+3;
l=find(C>122);
C(l)=C(l)-26;
l=find(C>90);
l=find(C(l)<97);
C(l)=C(l)-26;
l=find(tline==32);
C(l)=32;
ciphertek=char(C);

tline = fgets(fid);
end
  
```

Gambar 3.29 *Source Code* Enkripsi *File*

Diagram alur penyimpanan informasi dari *file* ke *database*.



Gambar 3.30 Diagram Alur Penyimpanan Informasi *File* Ke *Database*

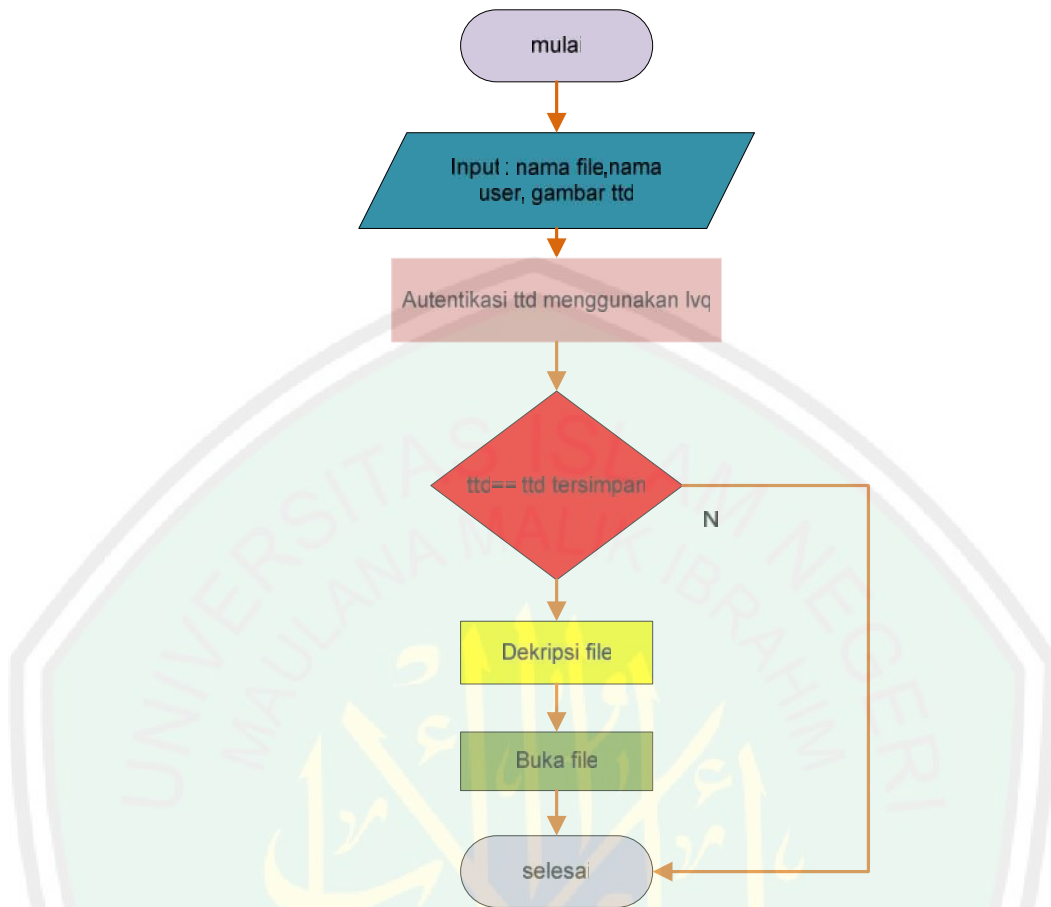
Data akan disimpan ke tabel “File” yang *field* nya seperti pada table 3.2 berikut :

Table 3.2 *Field-Field* Dari Tabel “File”.

<i>Field</i>	<i>Tipe</i>
Id	Int(5)
Nama_file	Varchar(30)
Id_us	Int(5)
Nama_user	Varchar(30)

3.3.2 Dekripsi *File*

Proses dekripsi merupakan proses perubahan kembali *file* seperti semula sebelum di enkripsi.



Gambar 3.31 Diagram Alur Dekripsi *File*

Untuk mendekripsikan *file*, pertama-tama kita harus memasukkan nama *file* dan nama *user*, setelah itu kita diminta untuk menggambar sebuah tanda tangan yang sesuai dengan tanda tangan dari tanda tangan dari nama *user* yang dimasukkan tadi. Setelah itu proses autentikasi dilakukan untuk mencocokkan apakah tanda tangan itu cocok dengan tanda tangan yang diminta. Bila cocok maka aplikasi akan mendekripsikan *file* dan *user* bisa membukanya.

Berikut ini adalah *source code* dekripsi *file*.

```

while ischar(tline)
disp(tline)
plain=tline-3;
l=find(plain<65);
plain(l)=plain(l)+26;
l=find(plain<97);
l=find(plain(l)>90);
plain(l)=plain(l)+26;
l=find(tline==32);
plain(l)=32;
plain=char(plain);
tline = fgets(fid);
end

```

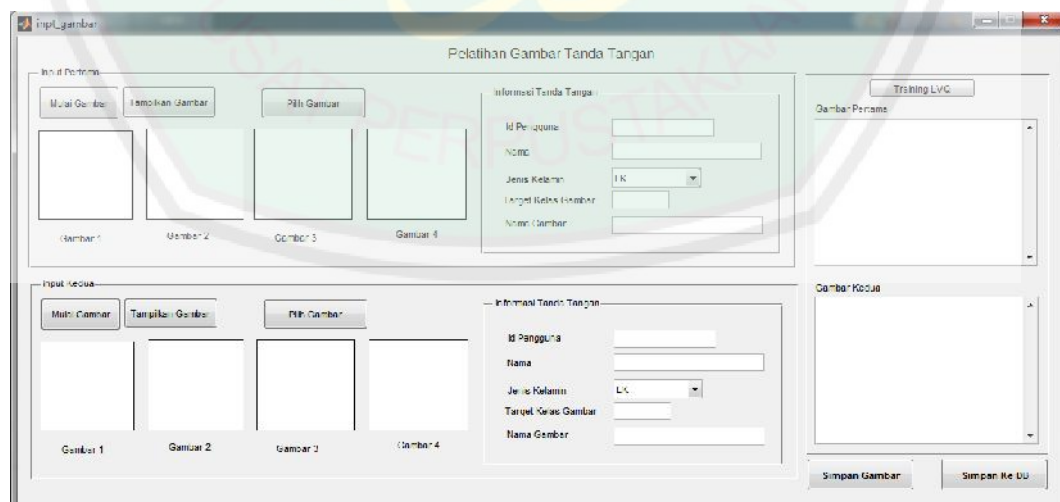
Gambar 3.32 *Source Code* Dekripsi *File*

Proses ini akan mengembalikan huruf menjadi tiga blok sebelumnya.

3.3 Desain GUI (*Graphical User Interface*)

Agar aplikasi ini lebih *user friendly* maka dibuatlah GUI (*Graphical User Interface*) atau grafis dari antar muka untuk pengguna. GUI dalam aplikasi ini dibuat tidak rumit agar pengguna tidak kesulitan dalam mengoperasikannya. Berikut ini adalah rancangan GUI pada aplikasi Autentikasi Tanda Tangan dan Pengamanan *File* yang akan dibuat.

a. Input Gambar dan Informasi dari Tanda Tangan



Gambar 3.33 Antarmuka Input Tanda Tangan

Ada beberapa tombol didalam *interface* ini, yaitu :

1. **Mulai Gambar**, tombol untuk memunculkan tempat untuk menggambar tanda tangan.



Gambar 3.34 Antarmuka Untuk Menggambar Tanda Tangan

Didalam *interface* ini terdapat beberapa tombol juga, yaitu :

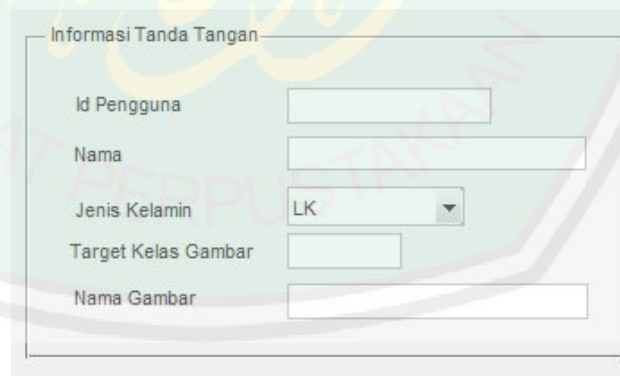
- **Gambar 1, Gambar 2, Gambar 3, Gambar 4** digunakan untuk menyimpan empat gambar yang dibutuhkan.
 - **Clear**, berfungsi untuk membersihkan kanvas bila ada kesalahan saat menggambar tanda tangan.
 - **Finish**, tombol untuk mengclose Layar bila penggambaran sudah selesai.
2. **Tampilkan Gambar**, tombol untuk menampilkan gambar yang sudah di gambar.
 3. **Pilih Gambar**, untuk memilih gambar tanda tangan yang sudah ada.
 4. **Training LVQ**, tombol untuk menghitung bobot akhir dari gambar yang sudah dibuat.
 5. **Simpan**, Tombol untuk menyimpan hasil ke dalam *database*.

Langkah yang harus dilalui *user* adalah sebagai berikut :

Ada dua cara untuk memulai, pertama user menekan tombol **Mulai Gambar** lalu menggambar tanda tangan sebanyak empat kali, tiap selesai menggambar user menekan tombol **Gambar 1** sampai **4**. bila terjadi kesalahan user tinggal menekan tombol **Clear** maka canvas akan kembali bersih dan user mulai menggambar. Bila sudah selesai user menekan tombol **Finish**. Setelah itu user menekan tombol **Tampilkan Gambar** untuk menampilkan hasil dari penggambaran tadi.

Cara yang kedua, user menekan tombol **Pilih Gambar**, lalu user memilih gambar yang dikehendaki. Maka otomatis gambar tanda tangan akan muncul pada axes.

setelah itu user mengisi informasi tentang dirinya di *textfield* yang sudah disediakan, User hanya perlu mengisi nama, jenis kelamin, dan nama gambar saja, dua *textfield* yang lainnya yaitu id user dan kelas otomatis diisi oleh aplikasi untuk menghindari kesamaan data.

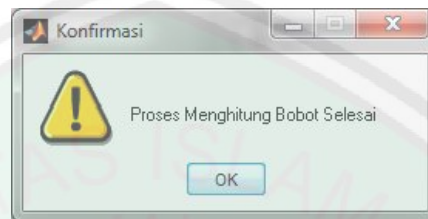


The image shows a screenshot of a web form titled "Informasi Tanda Tangan". The form contains five input fields arranged vertically. The first field is labeled "Id Pengguna" and is a text input. The second field is labeled "Nama" and is a text input. The third field is labeled "Jenis Kelamin" and is a dropdown menu with "LK" selected. The fourth field is labeled "Target Kelas Gambar" and is a text input. The fifth field is labeled "Nama Gambar" and is a text input. The form is set against a light green background with a faint watermark of the university's logo.

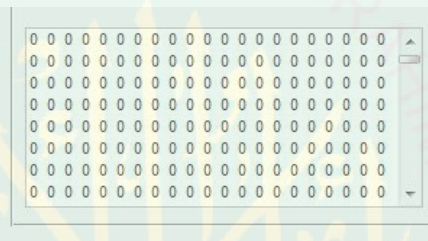
Gambar 3.35 *Textfield* Pengisian Informasi

Untuk gambar yang kedua, cara yang dilakukan sama dengan cara pada gambar pertama.

Baru setelah itu user menekan tombol **Training LVQ**, maka gambar tanda tangan akan di *preprocessing* yang akan menghasilkan citra biner, lalu dilakukan perhitungan LVQ untuk mendapatkan bobot akhir. bila perhitungan selesai akan muncul pesan konfirmasi bahwa perhitungan telah selesai.



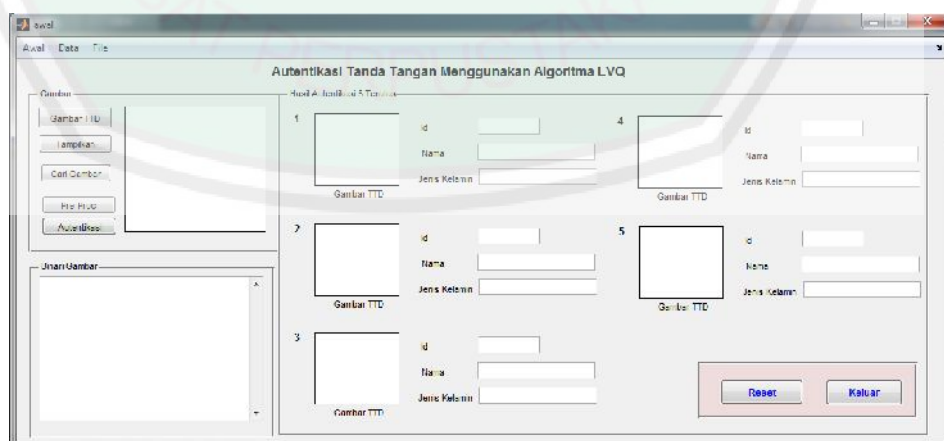
Gambar 3.36 Pesan Konfirmasi



Gambar 3.37 *Textfield* Hasil Perhitungan Bobot

Setelah muncul hasilnya, tekan tombol **Simpan** untuk menyimpan hasil kedalam *database*.

b. Autentikasi Tanda Tangan

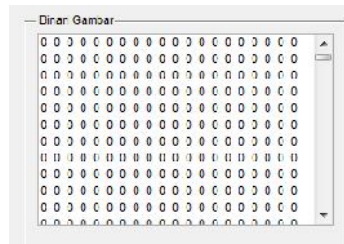


Gambar 3.38 Halaman Autentikasi Tanda Tangan

Ada beberapa tombol di halaman ini, yaitu :

1. **Gambar TTD**, tombol untuk menampilkan halaman yang digunakan untuk menggambar tanda tangan yang akan di autentikasi.
2. **Tampilkan**, tombol untuk menampilkan hasil gambar yang telah digambar.
3. **Cari Gambar**, tombol untuk mencari gambar yang sudah tersimpan.
4. **Pre-Proc**, tombol untuk melakukan proses *pre-processing* pada gambar, agar gambar menjadi citra biner.
5. **Autentikasi**, tombol untuk memulai proses autentikasi gambar tanda tangan.
6. **Reset**, tombol untuk membersihkan semua *axes* dan *textfield*
7. **Keluar**, tombol untuk keluar dan kembali ke halaman awal.

Dalam halaman ini ada beberapa proses yang dilakukan, yaitu input citra, *pre processing* dan autentikasi tanda tangan. Input citra bisa dilakukan dengan dua cara, yaitu dengan menggambar sendiri tanda tangan, atau menampilkan tanda tangan yang sudah disimpan sebelumnya. Kemudian citra dilakukan *preprocessing* dari konversi citra *RGB* ke *Grayscale* lalu *ditresholding* dan dijadikan citra biner. Setelah itu baru dilakukan proses autentikasi menggunakan algoritma LVQ.



Gambar 3.39 Hasil Dari *Pre-Processing*

Hasil akhir akan ditampilkan lima gambar teratas yang paling cocok dengan gambar yang di inputkan tadi. Yang paling cocok akan ditampilkan di axes2, yang ke dua di axes3, ke tiga di axes4, ke empat axes5 dan ke lima ada di axes6, dan masing-masing informasi dari gambar ada di *textfield* sebelah kanan masing-masing *axes*.



Gambar 3.40 Hasil Dari Autentikasi

Langkah-langkah untuk proses autentikasi, yang pertama, untuk memasukkan gambar, user bisa memilih mau menggambar sendiri atau mau mencari gambar dari *file* yang sudah ada. Bila user mau menggambar sendiri, user bisa menekan tombol **Mulai Gambar** maka akan muncul halaman untuk menggambar.



Gambar 3.41 Tampilan Tempat Untuk Menggambar Tanda Tangan

Bila sudah selesai menggambar tekan tombol selesai, maka halaman akan hilang dan kembali ke halaman awal. Setelah itu tekan tombol **Tampilkan** untuk menampilkan gambar yang sudah digambar tadi. Bila user memilih untuk mencari saja gambar yang sudah ada, maka user tinggal menekan tombol **Cari Gambar**, maka akan muncul kotak dialog untuk mencari file. Setelah memilih *file* tekan tombol Open untuk menampilkan gambar ke *axes1*.

Setelah itu dilakukan *preprocessing* untuk merubah citra gambar menjadi citra biner. Dengan cara menekan tombol **Pre-Proc**. Setelah proses ini selesai maka di *edit1* akan muncul hasil citra biner dari gambar. Barulah proses autentikasi bisa dilakukan. User menekan tombol **Autentikasi** untuk memulai proses Autentikasi tanda tangan input yang dibandingkan dengan tanda tangan yang sudah ada di *database*, setelah didapat hasilnya, hasil itu akan di *shorting* oleh sistem agar bisa mendapatkan lima gambar teratas yang paling cocok dengan input gambar tanda tangan. Bila proses autentikasi selesai maka akan muncul pesan konfirmasi bahwa proses telah selesai.

Tombol **Reset** digunakan bila *user* ingin melakukan autentikasi dengan gambar yang lain. Dan menekan tombol **Keluar** bila ingin keluar dari halaman ini.

c. Input Pengamanan *File*

Pengamanan file pada penelitian ini adalah enkripsi file yang menggunakan metode Caesar chipper dengan key 3. User tidak perlu menggambar tanda tangan, karena hanya perlu memilih nama *user*nya saja yang sudah tersedia dalam database, ini dilakukan untuk mempermudah *user* dalam menggunakan aplikasi ini. Berikut ini adalah tampilan muka dari pengamanan *file*.

Gambar 3.42 Tampilan Halaman Pengaman *File*

Ada beberapa tombol dalam halaman ini, yaitu :

1. **Cari**, tombol untuk mencari *user* yang akan digunakan untuk mendekripsikan *file* nantinya.
2. **Pilih File**, tombol untuk memilih *file* yang akan di Enkripsi.
3. **Enkripsi File**, tombol untuk mengenkripsi *file* yang sudah dipilih

4. **Simpan**, tombol untuk menyimpan *file* yang sudah di enkripsi lalu menyimpan informasinya ke *database*.
5. **Keluar**, tombol untuk keluar dari halaman.

Untuk melakukan proses pengamanan *file* dengan enkripsi di aplikasi ini, langkah yang harus dilakukan adalah : pertama masukkan nama *user* yang anda pilih ke *edit1* lalu tekan tombol **Cari** untuk mencari data dari *user* yang anda pilih di dalam *database*. Setelah ditemukan maka informasi akan tampil untuk memudahkan user memeriksa apa benar *user* itu yang dia pilih.

Data yang akan tampil adalah : gambar tanda tangan akan tampil di *axes1* beserta nama gambarnya di *text3*. Id user tampil di *edit2*, nama *user* di *edit3*, jenis kelamin di *edit4* dan yang terakhir kelas gambar di *edit5*.

Proses ke dua adalah memilih *file* yang akan dienkripsi dengan menekan tombol **Pilih File**. Maka akan muncul kotak dialog untuk memilih *file*. *File* yang bisa dipilih khusus berektensi txt saja. Setelah memilih tekan tombol **Open**. Maka dalam *edit6* akan muncul *path* beserta nama *file* yang dipilih.

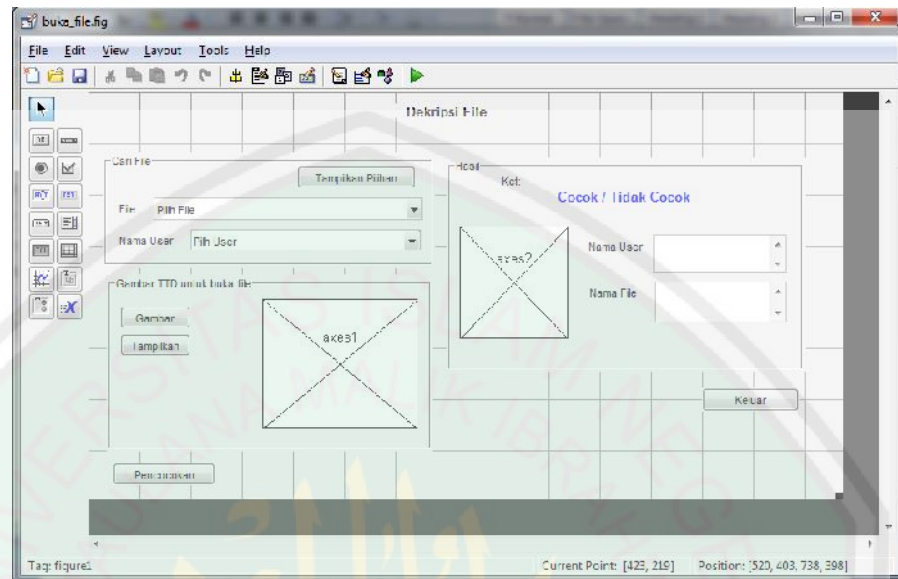
Tekan tombol **Enkripsi File** untuk mengenkripsi *file* yang sudah dipilih, enkripsi *file* disini yang digunakan adalah enkripsi *Caesar Chiper* dengan *key* Setelah proses enkripsi selesai, Tekan tombol **Simpan** untuk menyimpan *file* hasil enkripsi dan data *file* ke *database*.

Bila sudah selesai tekan tombol **Keluar** untuk keluar dari halaman ini.

d. Dekripsi File

Dekripsi dalam aplikasi ini adalah mengembalikan *file* seperti semula agar bisa dimengerti oleh user. Dekripsi *file* harus menggunakan tanda tangan

yang sudah dipilih sendiri oleh user sebelumnya, berikut ini adalah tampilan muka dekripsi *file* pada aplikasi ini.



Gambar 3.43 Tampilan Halaman Dekripsi *File*

Dalam tampilan muka dekripsi *file* terdapat beberapa tombol yang bisa digunakan oleh user, yaitu :

1. **Tampilkan Pilihan**, tombol untuk menampilkan pilihan nama user dan nama *file* yang ada dalam *database*.
2. **Gambar**, tombol untuk menampilkan halaman muka untuk menggambar tanda tangan.
3. **Tampilkan**, tombol untuk menampilkan gambar tanda tangan yang telah digambar pada *axes*.
4. **Pencocokan**, tombol untuk mencocokkan gambar tanda tangan dengan gambar tanda tangan yang diminta oleh *database*.
5. **Buka**, tombol yang keluar pada saat hasil pencocokan adalah cocok, tombol ini berfungsi untuk mendeskripsikan dan membuka *file*.

6. **Keluar**, tombol untuk keluar dari halaman muka.

Langkah-langkah yang harus dilakukan *user* adalah sebagai berikut :

Pertama user menekan tombol **Tampilkan Pilihan** untuk menampilkan pilihan nama *file* yang akan dibuka nanti dan nama user yang digunakan untuk membuka *filenya*. Setelah pilihan tampil pada *popupmenu1* untuk nama *file* dan *popupmenu2* untuk nama *user*, *user* harus memilih dengan cara mengklik pilihan tersebut.

Setelah itu *user* diminta menggambar tanda tangan dengan menekan tombol **Gambar**, maka akan muncul tampilan muka untuk menggambar. Setelah *user* menggambar ditempat yang sudah disediakan *user* menekan tombol **Selesai** pada tampilan muka yang berada dikanan bawah.



. Gambar 3.44 Halaman Muka Untuk Menggambar Tanda Tangan

Setelah itu *user* menekan tombol **Tampilkan** untuk menampilkan gambar yang sudah digambar pada *axes1*. Tekan tombol **Pencocokan** untuk mencocokkan apakah gambar input sama dengan gambar yang diminta dalam *database*. Proses pencocokan ini menggunakan algoritma LVQ. Bila hasilnya adalah cocok maka tombol **Buka** akan muncul, tombol ini

digunakan untuk mendeskripsikan *file* dengan metode *Caesar Chiper key 3* agar *file* bisa dimengerti lagi. Dan membuka *file* yang dimaksud.

Tombol **Keluar** ditekan bila *user* sudah selesai dan ingin keluar dari halaman ini.

3.5 Implementasi Aplikasi

Ada dua kebutuhan dalam mengimplementasikan aplikasi dalam penelitian ini, yaitu *hardware* dan *software*.

a. *Hardware* (Perangkat Keras)

Untuk membuat aplikasi ini penulis menggunakan perangkat keras dengan spesifikasi sebagai berikut :

- Processor inter core™ 2 duo CPU T6500 @1.10GHz
- RAM 2.00 GB
- Tipe Sistem 32Bit

b. *Software* (Perangkat Lunak)

Digunakan beberapa *software* dalam membuat aplikasi ini, yaitu:

1. *Windows 7 Profesional*

Sebagai operasi Sistem dasar yang menghubungkan *Hardware* dan *software*.

2. Matlab R2012b

Matlab digunakan untuk membuat berbagai tampilan muka, dan melakukan berbagai pemograman dalam pembuatan aplikasi ini.

3. Microsoft Office 2007

Aplikasi ini adalah aplikasi dibawah naungan *Microsoft* yang berguna untuk berbagai pengolahan teks, Untuk pembuatan aplikasi ini berguna untuk merancang dan pembuatan laporan ini.

4. Microsoft Visio 2007

Adalah aplikasi yang digunakan untuk berbagai macam kegiatan. Dalam penelitian ini aplikasi ini dipakai untuk pembuatan diagram alur dan arsitektur.

5. Appserv dan phpmyadmin

Appserv merupakan *server* lokal yang didalamnya ada berbagai aplikasi, diantaranya adalah phpmyadmin (mysql), apache dan php. Phpmyadmin digunakan untuk pembuatan dan menyimpan database aplikasi ini.

BAB IV

EKSPERIMEN DA PEMBAHASAN

Pada bab ini akan menjelaskan tentang rangkaian uji coba dan evaluasi yang telah dilakukan pada penelitian ini. Uji coba bertujuan untuk melihat sejauh mana tingkat keberhasilan dari implementasi dari aplikasi ini. Dan evaluasi dilakukan untuk analisa hasil uji coba agar mendapatkan kesimpulan dan saran untuk perkembangan aplikasi ini selanjutnya.

1.1 Langkah-Langkah Eksperimen

Langkah-langkah dari uji coba aplikasi yang dilakukan pada penelitian ini adalah :

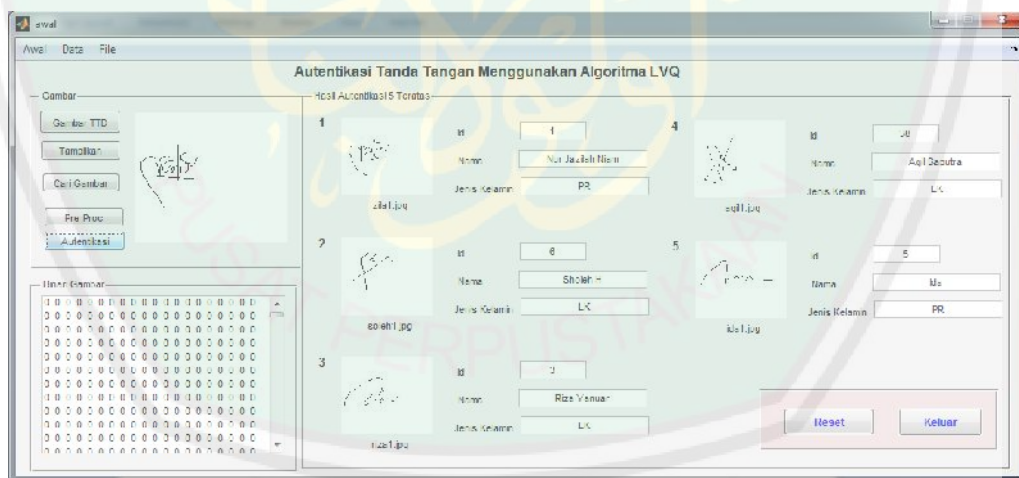
- a. Pembuatan citra tanda tangan, citra ini dibuat dengan cara menggambar tanda tangan di tempat yang sudah disediakan atau memilih gambar yang sudah ada.
- b. Preprocessing meliputi konversi *RGB* ke *Grayscale*, proses *thresholding* dan konversi citra menjadi citra biner.
- c. Menentukan parameter maksimal epoch, error minimal dan learning rate.
- d. Pelatihan citra yang sudah dibuat.
- e. Menyimpan hasil perhitungan bobot dari pelatihan dan informasi user ke dalam database.
- f. Uji coba, setelah di simpan di database, proses selanjutnya yaitu uji coba implementasi dari aplikasi sesuai dengan parameter yang sudah ditentukan.

- g. Hasil ditampilkan dalam lima besar teratas yang paling cocok dengan epoch, min error dan learning rate yang berbeda.

1.2 Eksperimen atau Uji Coba

Eksperimen dilakukan setelah proses penyimpanan hasil bobot akhir citra kedalam database yang didapatkan dari pembelajaran citra tanda tangan. Hasil inilah yang nanti dijadikan referensi bobot pada saat proses *testing* dilakukan.

Proses uji coba dilakukan dengan cara autentikasi tanda tangan menggunakan aplikasi yang sudah dibuat. Pengujian dilakukan satu persatu dengan memasukkan inputan citra. dari lima puluh data tanda tangan dua puluh data dijadikan sample testing. Hasil autentikasi menunjukkan perbandingan Minimal citra input dan bobot yang sudah tersimpan dalam database.



Gambar 4.1 Hasil Autentikasi Tanda Tangan

Dua puluh data yang akan digunakan untuk proses *testing*, yaitu :

Tabel 4.1 Data Yang Digunakan Untuk Proses *Testing*

No	Nama	JK	Target Kelas
1	adis	LK	1
2	agil	PR	2
3	ahsan	LK	3
4	ahsin	LK	4
5	aji	PR	5
6	andis	LK	6
7	anis	LK	7
8	ari	PR	8
9	awalia	PR	9
10	bila	PR	10
11	bintang	LK	11
12	coki	PR	12
13	dandi	PR	13
14	desi	LK	14
15	dicki	LK	15
16	eko	LK	16
17	fikar	LK	17
18	haqi	LK	18
19	hicha	PR	19
20	ida	PR	20

Uji coba pada penelitian ini dilakukan dengan beberapa epoch, min error dan learning rate yang berbeda. Parameter yang digunakan sama antara proses *learning* dan *testing*.

Minimal errornya sama dengan *learning rate*.

- a. Hasil uji coba dengan maksimal Epoch 3 min error 0.1 dan learning rate 0.1.

Tabel 4.2 Hasil Uji Coba Dengan Maksimal Epoch 10 Min Error 0.1 Dan Learning Ratenya 0.1

No	Nama	Ada Pada No					Salah	Waktu (sec)
		1	2	3	4	5		
1	adis	√						10.58
2	agil	√						8.29
3	ahsan	√						8.97

4	ahsin	√						8.97
5	aji						√	9.04
6	andis	√						9.08
7	anis	√						8.91
8	ari	√						9.46
9	awalia	√						8.95
10	bila	√						8.95
11	bintang	√						9.75
12	coki						√	8.95
13	dandi	√						9.53
14	desi						√	8.95
15	dicki	√						8.99
16	eko	√						8.97
17	fikar	√						8.99
18	haqi						√	8.92
19	hicha	√						9.24
20	ida	√						9.09

b. Hasil uji coba dengan maks Epoch 10 min error 0.1 dan learning rate 0.1.

Tabel 4.3 Hasil Uji Coba Dengan Maksimal Epoch 10 Min Error 0.1 Dan Learning Ratenya 0.1

No	Nama	Ada Pada No					Salah	Waktu (sec)
		1	2	3	4	5		
1	adis	√						10.50
2	agil		√					9.42
3	ahsan	√						9.58
4	ahsin	√						9.17
5	aji						√	9.20
6	andis	√						9.13
7	anis	√						9.50
8	ari	√						8.60
9	awalia	√						9.13
10	bila	√						9.24
11	bintang	√						9.22
12	coki						√	9.56
13	dandi	√						9.46
14	desi						√	9.11
15	dicki	√						9.74

16	eko	√						9.25
17	fikar	√						9.26
18	haqi						√	9.01
19	hicha	√						8.90
20	ida	√						9.07

c. Hasil uji coba dengan maks Epoch 100 min error 0.1 dan *learning rate* 0.1.

Tabel 4.4 Hasil Uji Coba Dengan Maksimal Epoch 100 Min Error 0.1 Dan Learning Ratenya 0.1

No	Nama	Ada Pada No					Salah	Waktu (sec)
		1	2	3	4	5		
1	adis	√						12.01
2	agil		√					11.86
3	ahsan	√						12.29
4	ahsin	√						11.71
5	aji						√	12.00
6	andis	√						11.62
7	anis	√						11.84
8	ari	√						11.90
9	awalia	√						11.97
10	bila	√						11.32
11	bintang	√						11.78
12	coki						√	11.21
13	dandi	√						11.29
14	desi						√	11.22
15	dicki	√						11.27
16	eko	√						11.30
17	fikar	√						11.39
18	haqi						√	11.26
19	hicha	√						11.22
20	ida	√						11.24

Minimal errornya lebih besar dari *learning rate*.

a. Hasil uji coba dengan maks Epoch 3 min error 0.1 dan *learning rate* 0.01.

Tabel 4.5 Hasil Uji Coba Dengan Maksimal Epoch 3 Min Error 0.1 Dan *Learning Ratenya* 0.01

No	Nama	Ada Pada No					Salah	Waktu (Sec)
		1	2	3	4	5		
1	adis	√						8.87
2	agil		√					8.63
3	ahsan	√						8.39
4	ahsin	√						8.65
5	aji						√	9.41
6	andis	√						8.70
7	anis	√						8.72
8	ari	√						8.66
9	awalia	√						8.77
10	bila	√						8.73
11	bintang	√						9.33
12	coki						√	8.70
13	dandi	√						8.70
14	desi						√	9.05
15	dicki	√						8.80
16	eko	√						8.85
17	fikar	√					√	9.39
18	haqi							8.76
19	hicha	√						8.74
20	ida	√						8.73

b. Hasil uji coba dengan maks Epoch 10 min error 0.1 dan learning rate 0.01.

Tabel 4.6 Hasil Uji Coba Dengan Maksimal Epoch 10 Min Error 0.1 Dan *Learning Ratenya* 0.01

No	Nama	Ada Pada No					Salah	Waktu (Sec)
		1	2	3	4	5		
1	adis	√						10.75
2	agil		√					1032
3	ahsan	√						9.07
4	ahsin	√						9.07
5	aji						√	9.40
6	andis	√						9.52
7	anis	√						9.05
8	ari	√						8.99
9	awalia	√						6.75

10	bila	√						8.95
11	bintang	√						9.04
12	coki						√	8.98
13	dandi	√						8.99
14	desi						√	9.05
15	dicki	√						8.99
16	eko	√						9.79
17	fikar	√					√	9.25
18	haqi							9.17
19	hicha	√						8.92
20	ida	√						8.98

c. Hasil uji coba dengan maks Epoch 100 min error 0.1 dan learning rate 0.01.

Tabel 4.7 Hasil Uji Coba Dengan Maksimal Epoch 100 Min Error 0.1 Dan Learning Ratenya 0.01

No	Nama	Ada Pada No					Salah	Waktu (Sec)
		1	2	3	4	5		
1	adis	√						11.16
2	agil		√					11.28
3	ahsan	√						11.04
4	ahsin	√						11.11
5	aji						√	11.00
6	andis	√						11.36
7	anis	√						11.96
8	ari	√						11.75
9	awalia	√						11.41
10	bila	√						11.47
11	bintang	√						11.92
12	coki						√	11.00
13	dandi	√						11.54
14	desi						√	11.45
15	dicki	√						11.49
16	eko	√						11.57
17	fikar	√					√	11.65
18	haqi							11.89
19	hicha	√						11.58
20	ida	√						11.49

Minimal errornya lebih kecil dari *learning rate*.

a. Hasil uji coba dengan maks Epoch 3 min error 0.01 dan learning rate 0.1.

Tabel 4.8 Hasil Uji Coba Dengan Maksimal Epoch 3 Min Error 0.01 Dan Learning Ratenya 0.1

No	Nama	Ada Pada No					Salah	Waktu (Sec)
		1	2	3	4	5		
1	adis	√						9.42
2	agil		√					9.27
3	ahsan	√						9.24
4	ahsin	√						9.34
5	aji						√	9.20
6	andis	√						10.19
7	anis	√						9.61
8	ari	√						9.52
9	awalia	√						10.00
10	bila	√						9.45
11	bintang	√						9.50
12	coki						√	9.39
13	dandi	√						9.33
14	desi						√	10.52
15	dicki	√						9.28
16	eko	√						9.68
17	fikar	√						9.33
18	haqi						√	9.10
19	hicha	√						9.38
20	ida	√						9.74

b. Hasil uji coba dengan maks Epoch 10 min error 0.01 dan learning rate 0.1.

Tabel 4.9 Hasil Uji Coba Dengan Maksimal Epoch 10 Min Error 0.01 Dan Learning Ratenya 0.1

No	Nama	Ada Pada No					Salah	Waktu (Sec)
		1	2	3	4	5		
1	adis	√						9.79
2	agil		√					9.52
3	ahsan	√						9.73
4	ahsin	√						9.24
5	aji						√	9.31

6	andis	√						9.30
7	anis	√						9.73
8	ari	√						9.60
9	awalia	√						9.34
10	bila	√						9.47
11	bintang	√						9.30
12	coki						√	9.43
13	dandi	√						10.00
14	desi						√	9.43
15	dicki	√						9.49
16	eko	√						9.43
17	fikar	√						9.40
18	haqi						√	9.35
19	hicha	√						9.84
20	ida	√						9.97

c. Hasil uji coba dengan maks Epoch 100 min error 0.01 dan learning rate 0.1.

Tabel 4.10 Hasil Uji Coba Dengan Maks Epoch 100 Min Error 0.01 Dan Learning Ratenya 0.1

No	Nama	Ada Pada No					Salah	Waktu (Sec)
		1	2	3	4	5		
1	adis	√						11.14
2	agil		√					11.14
3	ahsan	√						11.40
4	ahsin	√						11.20
5	aji						√	11.74
6	andis	√						11.11
7	anis	√						11.11
8	ari	√						11.19
9	awalia	√						11.07
10	bila	√						11.68
11	bintang	√						11.25
12	coki						√	11.12
13	dandi	√						11.15
14	desi						√	11.26
15	dicki	√						11.22
16	eko	√						11.65
17	fikar	√						11.23

18	haqi						√	11.13
19	hicha	√						11.07
20	ida	√						11.13

1.3 Pembahasan

Ada dua puluh data yang digunakan untuk proses pengujian, masing-masing data dilakukan sebanyak sembilan kali pengujian dengan parameter maksimal epoch, error minimal dan *learning rate* yang berbeda-beda.

Dari hasil pengujian didapatkan perhitungan akurasi sebagai berikut:

Percobaan pertama, error minimal dan learning rate sama, hasilnya adalah :

a. Gambar yang muncul pada nomor satu

Tabel 4.11 Hasil Dari Banyaknya Gambar Yang Muncul Pada Nomor Satu

No	Epoch	Learning Rate	Min Error	Jumlah Gambar Yang Muncul	Persentase
1	3	0.1	0.1	15	75%
2	10	0.1	0.1	15	75%
3	100	0.1	0.1	15	75%
4	3	0.01	0.1	15	75%
5	10	0.01	0.1	15	75%
6	100	0.01	0.1	15	75%
7	3	0.1	0.01	15	75%
8	10	0.01	0.1	15	75%
9	100	0.1	0.01	15	75%

b. Gambar yang muncul pada nomor dua

Tabel 4.12 Hasil Dari Banyaknya Gambar Yang Muncul Pada Nomor Dua

No	Epoch	Learning Rate	Min Error	Jumlah Gambar Yang Muncul	Persentase
1	3	0.1	0.1	1	5%
2	10	0.1	0.1	1	5%
3	100	0.1	0.1	1	5%
4	3	0.01	0.1	1	5%
5	10	0.01	0.1	1	5%
6	100	0.01	0.1	1	5%
7	3	0.1	0.01	1	5%
8	10	0.01	0.1	1	5%
9	100	0.1	0.01	1	5%

c. Gambar yang muncul pada nomor tiga

Tabel 4.13 Hasil Dari Banyaknya Gambar Yang Muncul Pada Nomor Tiga

No	Epoch	Learning Rate	Min Error	Jumlah Gambar Yang Muncul	Persentase
1	3	0.1	0.1	0	0%
2	10	0.1	0.1	0	0%
3	100	0.1	0.1	0	0%
4	3	0.01	0.1	0	0%
5	10	0.01	0.1	0	0%
6	100	0.01	0.1	0	0%

7	3	0.1	0.01	0	0%
8	10	0.01	0.1	0	0%
9	100	0.1	0.01	0	0%

d. Gambar yang muncul pada nomor empat

Tabel 4.14 Hasil Dari Banyaknya Gambar Yang Muncul Pada Nomor Empat

No	Epoch	Learning Rate	Min Error	Jumlah Gambar Yang Muncul	Persentase
1	3	0.1	0.1	0	0%
2	10	0.1	0.1	0	0%
3	100	0.1	0.1	0	0%
4	3	0.01	0.1	0	0%
5	10	0.01	0.1	0	0%
6	100	0.01	0.1	0	0%
7	3	0.1	0.01	0	0%
8	10	0.01	0.1	0	0%
9	100	0.1	0.01	0	0%

e. Gambar yang muncul pada nomor lima

Tabel 4.15 Hasil Dari Banyaknya Gambar Yang Muncul Pada Nomor Lima

No	Epoch	Learning Rate	Min Error	Jumlah Gambar Yang Muncul	Persentase
1	3	0.1	0.1	0	0%
2	10	0.1	0.1	0	0%

3	100	0.1	0.1	0	0%
4	3	0.01	0.1	0	0%
5	10	0.01	0.1	0	0%
6	100	0.01	0.1	0	0%
7	3	0.1	0.01	0	0%
8	10	0.01	0.1	0	0%
9	100	0.1	0.01	0	0%

Jadi, bisa disimpulkan dari hasil autentikasi yang sudah dilakukan, keakuratan gambar yang bisa dikenali oleh aplikasi yang menggunakan metode LVQ ini dari yang nomor satu sampai nomor lima ada enam belas gambar, berarti sama dengan 80%.

Sedangkan untuk waktu yang dibutuhkan dalam proses autentikasi ditunjukkan pada tabel 4.15 Berikut :

Tabel 4.16 Rata-Rata Waktu Autentikasi

Parameter	Maksimal epoch	Rata-rata waktu yang dibutuhkan (secon)
Eps == Alpha	3	8.915
	10	9.272
	100	11.585
Eps >= Alpha	3	8.829
	10	9.1515
	100	11.456
Eps <= Alpha	3	9.5245
	10	9.5335
	100	11.2495

Dari hasil pengujian bisa dilihat pada parameter yang manapun bila epochnya sama maka waktu yang dibutuhkan untuk proses autentikasinya pun hampir sama.

1.4 Integrasi Autentikasi Tanda Tangan Menggunakan Metode LVQ (Learning Vector Quantization) untuk Keamanan Dokumen dengan AI Qur'an

Di dalam Al Qur'an Surat An nisa' ayat 94, Allah SWT berfirman :

يٰۤاَيُّهَا الَّذِيْنَ ءَامَنُوْا اِذَا ضَرَبْتُمْ فِى سَبِيْلِ اللّٰهِ فَتَبَيَّنُوْا وَّلَا تَقُوْلُوْا لِمَنْ اَلْقَىٰ
 اِلَيْكُمْ السَّلَامَ لَسْتُمْ مُّؤْمِنًا تَبْتَغُوْنَ عَرَضَ الْحَيٰوةِ الدُّنْيَا فَعِنْدَ اللّٰهِ
 مَغٰنِمٌ كَثِيْرَةٌ ...

Artinya :”Hai orang-orang yang beriman, apabila kamu pergi (berperang) di jalan Allah, Maka telitilah dan janganlah kamu mengatakan kepada orang yang mengucapkan "salam" kepadamu: "Kamu bukan seorang mukmin" (lalu kamu membunuhnya), dengan maksud mencari harta benda kehidupan di dunia...”.

Ayat diatas menjelaskan manusia dalam memeriksa sesuatu harus dengan teliti, agar nantinya tidak ada seorangpun yang merasa dirugikan atas kesalahan dan kelalaian kita.

Tanda tangan dalam bahasa Inggris: *signature* berasal dari *Latin: signare* yang berarti "tanda". Jadi tanda tangan adalah tanda sebagai lambang nama yang dituliskan dengan tangan oleh orang itu sendiri sebagai penanda pribadi. Dalam Surat Al a'la ayat dua dan tiga, Allah SWT berfirman :

الَّذِي خَلَقَ فَسَوَّىٰ ﴿٢﴾ وَالَّذِي قَدَّرَ فَهَدَىٰ ﴿٣﴾

Artinya : “Yang Menciptakan, dan menyempurnakan (penciptaan-Nya), Dan yang menentukan kadar (masing-masing) dan memberi petunjuk”.

Ayat tersebut menjelaskan bahwa setiap ciptaan tuhan pasti mempunyai kadar (ukuran) masing-masing, sesuai dengan citra tanda tangan yang mempunyai ukuran (jumlah) pixel yang berbeda-beda.

Sedangkan tentang algoritma LVQ dalam Alquran dijelaskan di dalam surat Al mulk ayat dua sampai tiga :

الَّذِي خَلَقَ سَبْعَ سَمَاوَاتٍ طِبَاقًا ۗ مَا تَرَىٰ فِي خَلْقِ الرَّحْمَنِ مِن تَفَوتٍ فَأَرْجِعِ

الْبَصَرَ هَل تَرَىٰ مِن فُطُورٍ ﴿٢﴾ ثُمَّ أَرْجِعِ الْبَصَرَ كَرَّتَيْنِ يَنقَلِبْ إِلَيْكَ الْبَصَرُ خَاسِئًا

وَهُوَ حَسِيرٌ ﴿٤﴾

Artinya :”Yang Telah menciptakan tujuh langit berlapis-lapis. kamu sekali-kali tidak melihat pada ciptaan Tuhan yang Maha Pemurah sesuatu yang tidak seimbang. Maka Lihatlah berulang-ulang, Adakah kamu lihat sesuatu yang tidak seimbang?. Kemudian pandanglah sekali lagi niscaya penglihatanmu akan

kembali kepadamu dengan tidak menemukan sesuatu cacat dan penglihatanmu itupun dalam keadaan payah”.

Jadi bila ada citra tanda tangan yang tidak seimbang maka dilakukan pembelajaran berulang-ulang sampai batas yang ditentukan, agar nantinya saat dilihat kembali (autentikasi) citra yang ditampilkan benar.

Sedangkan dalam masalah pentingnya keamanan, Al qur'an beberapa kali menyebutkan masalah itu, salah satunya adalah:

هُوَ اللَّهُ الَّذِي لَا إِلَهَ إِلَّا هُوَ الْمَلِكُ الْقُدُّوسُ السَّلَامُ الْمُؤْمِنُ الْمُهَيْمِنُ الْعَزِيزُ
الْجَبَّارُ الْمُتَكَبِّرُ سُبْحَانَ اللَّهِ عَمَّا يُشْرِكُونَ

Artinya : “Dialah Allah yang tiada Tuhan selain Dia, raja, yang Maha suci, yang Maha Sejahtera, yang Mengaruniakan Keamanan, yang Maha Memelihara, yang Maha Perkasa, yang Maha Kuasa, yang memiliki segala Keagungan, Maha Suci Allah dari apa yang mereka persekutukan”.

(QS Al hasyr : 23)

Sungguh penting keamanan dan rasa aman itu, hingga nabi Ibrahim mendahulukan berdoa agar negrinya aman sebelum dia berdoa untuk meminta rizki. Karena jika manusia sudah merasa aman maka dia akan merasakan sebuah tenang dalam menjalani hidup.

BAB V

KESIMPULAN DAN SARAN

1.1 Kesimpulan

Dari hasil implementasi dan serangkaian uji coba yang telah dilakukan pada aplikasi Autentikasi Tanda Tangan menggunakan metode LVQ untuk Keamanan Dokumen ini dapat disimpulkan bahwa :

- a. Metode LVQ bisa diimplementasikan pada aplikasi Autentikasi Tanda Tangan sebagai sebuah metode untuk melakukan pembelajaran dan autentikasi citra tanda tangan. Data citra tanda tangan yang sudah di simpan di dalam *database* bisa digunakan untuk proses autentikasi tanda tangan. Hasil dari autentikasi diambil lima teratas.
- b. Hasil dari proses autentikasi tanda tangan menggunakan metode LVQ dibagi menjadi tiga parameter yang berbeda, tiap parameter dibagi lagi menjadi tiga epoch yang berbeda, dari Sembilan uji coba yang telah dilakukan, gambar yang berhasil dikenali sebanyak 80%.

1.2 Saran

Ada banyak kekurangan dan hal-hal yang harus diperbaiki dalam penelitian aplikasi Autentikasi Tanda Tangan Menggunakan Metode LVQ untuk Keamanan Dokumen ini. Karena itu penulis menyarankan beberapa hal untuk pengembangan aplikasi ini selanjutnya, yaitu :

- a. Dalam proses input tanda tangan tidak lagi dilakukan dengan cara menggambar pada halaman yang telah disediakan dengan menggunakan *mouse*, karena kebanyakan *user* kesulitan dalam menggambar.
- b. Metode autentikasi tanda tangan bisa menggunakan metode yang lain agar tingkat keakuratan pada saat proses autentikasi lebih baik.
- c. Autentikasi tanda tangan bisa diterapkan pada hal lain, untuk pengamanan dokumennya bisa menggunakan cara yang lain juga.



DAFTAR PUSTAKA

- Chandra U, Rifky. 2013. *Implementasi Algoritma Enkripsi Abacusan Dan Caesar Cipher Pada Sms Dan Koordinat Lokasi Berbasis Android Menggunakan App Inventor* . Yogyakarta: Sekolah Tinggi Manajemen Informatika Dan Komputer Amikom Yogyakarta .
- Frebiansyah dkk. 2007. *Autentikasi Tanda Tangan Menggunakan Metode C-Means*. Jakarta :Jurusan Teknik Informatika Universitas Bina Nusantara.
- Hermawan, Arif. 2006. *Jaringan Saraf Tiruan, Teori dan Aplikasi*. Yogyakarta: CV Andi Offsed
- Hidayatno, Achmad dkk. 2008. *Identifikasi Tanda-Tangan Menggunakan Jaringan Saraf Tiruan Perambatan-Balik (BACKPROPAGATION)*. Teknik Elektro Fakultas Teknik Universitas Diponegoro
- Kusuma, Dewi, S. 2003. *Membangun jaringan saraf tiruan menggunakan matlab dan excel link*. Yogyakarta: Graha Ilmu.
- Nugroho, Andy. 2012. *Implementasi Algoritma Caesar Cipher Rot13 Dan Base64 Untuk Enkripsi Dan Dekripsi Pesan Sms Pada Handphone Berbasis Android*. Yogyakarta: Sekolah Tinggi Manajemen Informatika Dan Komputer Amikom Yogyakarta .
- Putra, Darma. 2010. *Pengolahan Citra Digital*. Yogyakarta: CV.Andi Offset
- Rakhmanullah, Afif. 2010. *Autentifikasi Pengenalan Pola Tanda Tangan Manual Menggunakan Jaringan Saraf Tiruan LVQ (Learning Vector Quantization) dan Tanda Tangan Digital Menggunakan Algoritma RSA (Riset Shamir Adleman*. UIN malang.
- Ricardo, Ignatius. *Pengenalan Tanda Tangan Melalui Pengolahan Citra Digital dan Jaringan Saraf Tiruan RBF*. Sistem Informasi Universitas Pelita Harapan Surabaya
- Seftianto, Donny dkk. 2012. *Peran Algoritma Caesar Cipher Dalam Membangun Karakter Akan Kesadaran Keamanan Informasi*. Sekolah Tinggi Sandi Negara

Sutikno. *Aplikasi Pengenalan Tanda Tangan dengan Metode Learning Vektor Quantization dan Backpropagation*. Universitas Mercu Buana

Zainal, Andam dkk. 2002. *Aplikasi Neural Network pada Pengenalan Pola Tanda Tangan*. Politeknik Elektronika Negeri Surabaya Institut Teknologi Sepuluh November (ITS) Surabaya



LAMPIRAN 1 Database Tanda Tangan

id	nama_user	jk	kelas	Nama_gambar	Bobot
1	Nur Jazilah Niam	PR	1	zila1.jpg	
2	Taufan	LK	2	taufan1.jpg	
3	Riza Yanuar	LK	3	riza1.jpg	
4	Hicha M	PR	4	hicha1.jpg	
5	Ida	PR	5	ida1.jpg	
6	Sholeh H	LK	6	soleh1.jpg	
7	Choki S	LK	7	coki1.jpg	
8	Mustamilatun	PR	8	mus1.jpg	
9	Qlswa S	PR	9	qiswa1.jpg	
10	Prayitno	LK	10	prayitno1.jpg	
11	Reza S	LK	11	reza1.jpg	
12	Ahmad Mustofa	LK	12	tofa1.jpg	
13	Ari Muwafa	LK	13	ari1.jpg	
14	Andis	PR	14	andis1.jpg	
15	Awalia	PR	15	awalia1.jpg	
16	Nur Lembang	PR	16	nur1.jpg	
17	Sita N	PR	17	sita1.jpg	
18	Wawan	LK	18	wawan1.jpg	
19	Jimmi	LK	19	jimmi1.jpg	
20	Murdiayah	LK	20	murdiyah1.jpg	
21	Ahsin Nuh	LK	21	ahsin1.jpg	
22	Zulianto	LK	22	zul1.jpg	

id	nama_user	jk	kelas	Nama_gambar	Bobot
23	Sinta	PR	23	sinta1.jpg	
24	Ilfadon Uta Arsada	LK	24	uta1.jpg	
25	Yudhara	LK	25	yuda1.jpg	
26	Reni F	PR	26	reni1.jpg	
27	Uzi	LK	27	uzi1.jpg	
28	Bintang	LK	28	bintang1.jpg	
29	Baihaqi	LK	29	haqi1.jpg	
30	Maiya	PR	30	maiya1.jpg	
31	Anis	PR	31	anis1.jpg	
32	Tata	PR	32	tata1.jpg	
33	Munip	LK	33	munip1.jpg	
34	Mulia	PR	34	mulia1.jpg	
35	Sasi	PR	35	sasi1.jpg	
36	Bila	PR	36	bila1.jpg	
37	Adis	LK	37	adis1.jpg	
38	Agil Saputra	LK	38	agil1.jpg	
39	Desi	PR	39	desi1.jpg	
40	Syafa B	PR	40	syafa1.jpg	
41	Sulfikar	LK	41	fikar1.jpg	
42	Sulis	PR	42	sulis1.jpg	
43	Ahsan	LK	43	ahsan1.jpg	
44	Eko Prasetio	LK	44	eko1.jpg	

id	nama_user	jk	kelas	Nama_gambar	Bobot
45	Dicki	LK	45	dicki1.jpg	
46	Mubarak	LK	46	mubarak1.jpg	
47	Dandi	LK	47	dandi1.jpg	
48	Rangga	LK	48	rangga1.jpg	
49	Qayla	PR	49	qayla1.jpg	
50	Hendy Satrio Aji	LK	50	aji1.jpg	

NB : karena data bobot terlalu panjang, maka data tidak ditampilkan disini.



LAMPIRAN 2 Source Code Proses Pembelajaran Tanda Tangan

Menggunakan LVQ

```
image_input1 = handles.data1;
image_input2 = handles.data2;
image_input3 = handles.data3;
image_input4 = handles.data4;

%input 1
% 0=hitam dan 255=putih
data_proses1=uint8(zeros(size(image_input1,1),size(image_input1,2)));
for i=1:size(image_input1,1)
    for j=1:size(image_input1,2)
        data_proses1(i,j)=0.2989*image_input1(i,j,1)+0.5870*image_input1(i,j,2)+0.1140*image_input1(i,j,3);
    end
end

threshold = thres(data_proses1);

baris = size(data_proses1,1);
kolom = size(data_proses1,2);
for i=1:baris
    for j=1:kolom
        if data_proses1(i,j) > threshold
            data_proses1(i,j)=255;
        else
            data_proses1(i,j)=0;
        end
    end
end

sz1=size(data_proses1);
bina1=zeros(size(data_proses1));
for i2=1:sz1(1)
    for j2=1:sz1(2)
        if(data_proses1(i2,j2)<threshold)
            bina1(i2,j2)=1;
        end
    end
end

im_in1 = logical(bina1);
vec_bi_in1 = reshape(im_in1,1,[]);

%input 2
data_proses2=uint8(zeros(size(image_input2,1),size(image_input2,2)));
for i=1:size(image_input2,1)
    for j=1:size(image_input2,2)
        data_proses2(i,j)=0.2989*image_input2(i,j,1)+0.5870*image_input2(i,j,2)+0.1140*image_input2(i,j,3);
    end
end

threshold = thres2(data_proses2);
```

```

threshold = thres2(data_proses2);

baris = size(data_proses2,1);
kolom = size(data_proses2,2);
for i=1:baris
    for j=1:kolom
        if data_proses2(i,j) > threshold
            data_proses2(i,j)=255;
        else
            data_proses2(i,j)=0;
        end
    end
end

sz2=size(data_proses2);

bina2=zeros(size(data_proses2));

for i2=1:sz2(1)
    for j2=1:sz2(2)
        if(data_proses2(i2,j2)<threshold)
            bina2(i2,j2)=1;
        end
    end
end

im_in2 = logical(bina2);
vec_bi_in2 = reshape(im_in2,1,[]);

%input 3
data_proses3=uint8(zeros(size(image_input3,1),size(image_input3,2)));
for i=1:size(image_input3,1)
    for j=1:size(image_input3,2)
        data_proses3(i,j)=0.2989*image_input3(i,j,1)+0.5870*image_input3(i,j,2)+0.1140*image_input3(i,j,3);
    end
end

threshold = thres3(data_proses3);

baris = size(data_proses3,1);
kolom = size(data_proses3,2);
for i=1:baris
    for j=1:kolom
        if data_proses3(i,j) > threshold
            data_proses3(i,j)=255;
        else
            data_proses3(i,j)=0;
        end
    end
end

sz3=size(data_proses3);

bina3=zeros(size(data_proses3));

for i2=1:sz3(1)
    for j2=1:sz3(2)
        if(data_proses3(i2,j2)<threshold)

```

```

        bina3(i2,j2)=1;
    end

end

end

im_in3 = logical(bina3);
vec_bi_in3 = reshape(im_in3,1,[]);

%input 4
data_proses4=uint8(zeros(size(image_input4,1),size(image_input4,2)));
for i=1:size(image_input4,1)
    for j=1:size(image_input4,2)
        data_proses4(i,j)=0.2989*image_input4(i,j,1)+0.5870*image_input4(i,j,2)+0.1140*image_input4(i,j,3);
    end
end

threshold = thres3(data_proses4);

baris = size(data_proses4,1);
kolom = size(data_proses4,2);
for i=1:baris
    for j=1:kolom
        if data_proses4(i,j) > threshold
            data_proses4(i,j)=255;
        else
            data_proses4(i,j)=0;
        end
    end
end

sz4=size(data_proses4);
bina4=zeros(size(data_proses4));
for i2=1:sz4(1)
    for j2=1:sz4(2)
        if(data_proses4(i2,j2)<threshold)
            bina4(i2,j2)=1;
        end
    end
end

im_in4 = logical(bina4);
vec_bi_in4 = reshape(im_in4,1,[]);

%pembelajaran LVQ
target=get(handles.edit3, 'String');
P=[vec_bi_in1;vec_bi_in2;vec_bi_in3;vec_bi_in4];
T=[target target target target];
w=pembelajaran_lvq(P,T);
w1= int2str(w);
set(handles.edit5, 'String', w1);

msgbox('Proses Menghitung Bobot Selesai','Konfirmasi','warn');

```

LAMPIRAN 3 Source Code Autentikasi Tanda Tangan Lima Teratas

```
image_in = handles.data1;

%input 1
% 0=hitam dan 255=putih
data_proses1=uint8(zeros(size(image_in,1),size(image_in,2)));
for i=1:size(image_in,1)
    for j=1:size(image_in,2)
        data_proses1(i,j)=0.2989*image_in(i,j,1)+0.5870*image_in(i,j,2)+0.1140*image_in(i,j,3);
    end
end

threshold = thres(data_proses1);

baris = size(data_proses1,1);
kolom = size(data_proses1,2);
for i=1:baris
    for j=1:kolom
        if data_proses1(i,j) > threshold
            data_proses1(i,j)=255;
        else
            data_proses1(i,j)=0;
        end
    end
end

sz1=size(data_proses1);
bina1=zeros(size(data_proses1));

for i2=1:sz1(1)
    for j2=1:sz1(2)
        if(data_proses1(i2,j2)<threshold)
            bina1(i2,j2)=1;
        end
    end
end

im_inp = logical(bina1);
vec_bi_inp = reshape(im_inp,1,[]);
save vec_bi_inp

vec_bi_inp1= int2str(vec_bi_inp);
set(handles.edit1, 'String', vec_bi_inp1);

%# add path to the JAR file you just installed to Java dynamic classpath
javaaddpath('F:\skripsiZila\programbaru\mysql-connector-java-5.1.6-bin.jar')
%# connection parameteres
host = 'localhost';
user = 'root';
password = 'root';
dbName = 'skripsi';
%# JDBC parameteres
jdbcString = sprintf('jdbc:mysql://%s/%s', host, dbName);
jdbcDriver = 'com.mysql.jdbc.Driver';
```

```
## Create the database connection object
conn = database(dbName, user , password, jdbcDriver, jdbcString);
```

```
%image yang di aut
load vec_bi_inp
inp=vec_bi_inp;
[x y]=size(inp);
Diag=zeros(x,1);
% Diag2=zeros(x,2);
```

```
%menampilkan data
sql_bo='select * from ttd';
hsl_b=fetch(conn,sql_bo);
hsl_b1=hsl_b(:,6);
hsl=char(hsl_b1);
hsl_b2=str2num(hsl);
hsl_bo=logical(hsl_b2);
```

```
z=size(hsl_bo);% x baris
```

```
J=zeros(z,1);
```

```
for k=1:x
    for ii=1:z
        J(ii)=0;
        for jj=1:y
            J(ii)=J(ii)+(hsl_bo(ii,jj)-inp(k,jj))^2;
        end
        J(ii)=sqrt(J(ii));
    end
end
```

```
[Jmin idx ]= min(J);
Diag(k,:)=idx;
```

```
J
hasil=sort(J);
hasil
hasil1=hasil(1);
hasil2=hasil(2);
ke2=find(ismember(J,hasil2),1);
hasil3=hasil(3);
ke3=find(ismember(J,hasil3),1);
hasil4=hasil(4);
ke4=find(ismember(J,hasil4),1);
hasil5=hasil(5);
ke5=find(ismember(J,hasil5),1);
```

```
end
```

```
%menampilkan hasil uji
% var=num2str(Diag);
% sql_uji='select * from ttd where kelas=';
% variabel=var;
% hasil_uji= strcat(sql_uji,variabel);
% Data=fetch(conn, hasil_uji);
```

```
sql_uji='select * from ttd';
Data=fetch(conn,sql_uji);
```

```
namefile=char(Data(Diag,5));
name_path='F:\skripsiZila\programbaru\gambarTTD';
handles.data1 = imread(fullfile(name_path,namefile));
guidata(hObject,handles);
axes(handles.axes2);
imshow(handles.data1);
```

```

set(handles.edit2, 'String', Data(Diag,1));
set(handles.edit3, 'String', Data(Diag,2));
set(handles.edit4, 'String', Data(Diag,3));
set(handles.text2, 'String', Data(Diag,5));

%hasil ke-2
namefile=char(Data(ke2,5));
name_path='F:\skripsiZila\programbaru\gambarTTD';
handles.data1 = imread(fullfile(name_path,namefile));
guidata(hObject,handles);
axes(handles.axes3);
imshow(handles.data1);

set(handles.edit5, 'String', Data(ke2,1));
set(handles.edit6, 'String', Data(ke2,2));
set(handles.edit7, 'String', Data(ke2,3));
set(handles.text7, 'String', Data(ke2,5));

%hasil ke-3
namefile=char(Data(ke3,5));
name_path='F:\skripsiZila\programbaru\gambarTTD';
handles.data1 = imread(fullfile(name_path,namefile));
guidata(hObject,handles);
axes(handles.axes4);
imshow(handles.data1);

set(handles.edit8, 'String', Data(ke3,1));
set(handles.edit9, 'String', Data(ke3,2));
set(handles.edit10, 'String', Data(ke3,3));
set(handles.text12, 'String', Data(ke3,5));

%hasil ke-4
namefile=char(Data(ke4,5));
name_path='F:\skripsiZila\programbaru\gambarTTD';
handles.data1 = imread(fullfile(name_path,namefile));
guidata(hObject,handles);
axes(handles.axes5);
imshow(handles.data1);

set(handles.edit11, 'String', Data(ke4,1));
set(handles.edit12, 'String', Data(ke4,2));
set(handles.edit13, 'String', Data(ke4,3));
set(handles.text17, 'String', Data(ke4,5));

%hasil ke-5
namefile=char(Data(ke5,5));
name_path='F:\skripsiZila\programbaru\gambarTTD';
handles.data1 = imread(fullfile(name_path,namefile));
guidata(hObject,handles);
axes(handles.axes6);
imshow(handles.data1);

set(handles.edit14, 'String', Data(ke5,1));
set(handles.edit15, 'String', Data(ke5,2));
set(handles.edit16, 'String', Data(ke5,3));
set(handles.text22, 'String', Data(ke5,5));
msgbox('Proses Autentikasi Selesai','Konfirmasi','warn');

```

LAMPIRAN 4 Source Code Enkripsi File

```
clc
tempat_file=get(handles.edit6, 'String');

fil=tempat_file;
fid = fopen(fid);
tline = fgets(fid);
while ischar(tline)
disp(tline)
C=tline+3;
l=find(C>122);
C(l)=C(l)-26;
l=find(C>90);
l=find(C(l)<97);
C(l)=C(l)-26;
l=find(tline==32);
C(l)=32;
ciphertek=char(C);

tline = fgets(fid);
end
disp('cipherteks adalah');
disp(ciphertek)
save ciphertek
set(handles.text11, 'String','Proses Selesai');
```


LAMPIRAN 5 Source Code Dekripsi File

```
javaaddpath('F:\skripsiZila\programbaru\mysql-connector-java-5.1.6-bin.jar')
%# connection parameteres
host = 'localhost';
user = 'root';
password = 'root';
dbName = 'skripsi';
%# JDBC parameters
jdbcString = sprintf('jdbc:mysql://%s/%s', host, dbName);
jdbcDriver = 'com.mysql.jdbc.Driver';

%# Create the database connection object
conn = database(dbName, user , password, jdbcDriver, jdbcString);

sql1='select * from file';
sql2=fetch(conn,sql1);
sql3=char(sql2(2,2));
whos sql3
sql3
fil=sql3;
fid = fopen(fil);

tline = fgets(fid);
while ischar(tline)
disp(tline)
plain=tline-3;
l=find(plain<65);
plain(l)=plain(l)+26;
l=find(plain<97);
l=find(plain(l)>90);
plain(l)=plain(l)+26;
l=find(tline==32);
plain(l)=32;
plain=char(plain);
tline = fgets(fid);
end
fid1 = fopen(fil , 'w');
fwrite(fid1,plain);
open(fil);
fclose(fid1);
```

