

# Pengenalan Wajah Menggunakan Metode Artificial Neural Network Dengan Fitur Ekstraksi Local Binary Pattern

*by* Mahtias Sutiyo T3115016

---

**Submission date:** 19-Nov-2019 08:45AM (UTC+0700)

**Submission ID:** 1216731398

**File name:** ral\_Network\_Dengan\_Fitur\_Ekstraksi\_Local\_Binary\_Pattern\_2019.pdf (1.68M)

**Word count:** 10304

**Character count:** 56054

**PENGENALAN WAJAH MENGGUNAKAN METODE  
5 ARTIFICIAL NEURAL NETWORK DENGAN  
FITUR EKSTRAKSI LOCAL BINARY  
PATTERN HISTOGRAM**

**Oleh**  
**MAHTIAS SUTIYO**  
**T3115016**

**SKRIPSI**

**51**  
**Untuk Memenuhi Salah Satu Syarat Ujian**  
**Guna Memperoleh Gelar Sarjana**



**PROGRAM SARJANA  
TEKNIK INFORMATIKA  
UNIVERSITAS IHSAN GORONTALO  
GORONTALO  
2019**

## **ABSTRACT**

*Online or computer-based examination systems do not use security systems. Implementation of online exam participant verification which is intended to avoid cheating from examinees / participants who have not been approved by online exams. Verification and confirmation of the examinees so that information security can be guaranteed well. Facial recognition is a biometric system that is widely used to understand individuals. There are two processes used in face recognition, namely the feature extraction process and the classification technique used. The purpose of this study is for face recognition using the Artificial Neural Network method with the extraction feature of the Local Binary Pattern Histogram for verification of examinees. Local Binary Pattern Histogram is used to extract facial features to get a weight value. The weight value of a test image will be matched with the weight value of the training image using the Artificial Neural Network method. Implementation of the online exam participant verification system uses the Python programming language. The results of the test were carried out with 30 training drawings consisting of 10 different people, and each person represented 3 images. With an experiment of 20 image tests conducted one by one on the images of 10 different people. The results of the study showed that facial recognition using the Artificial Neural Network method with the extraction feature of the Local Binary Pattern Histogram for the verification of the test participants was obtained at 80% using the Confusion Matrix test.*

**Keywords:** *Verification, Face Recognition, Local Binary Pattern Histogram, Artificial Neural Network.*

## ABSTRAK

Sistem ujian online atau berbasis komputer belum menggunakan sistem keamanan. Penerapan verifikasi peserta ujian online yang bertujuan untuk menghindari kecurangan dari peserta ujian / pihak yang tidak bersangkutan di ujian online. Verifikasi dan identifikasi terhadap peserta ujian agar keamanan informasi dapat terjamin dengan baik. Pengenalan wajah merupakan sistem biometrika yang banyak digunakan untuk mengenali individu. Ada dua proses yang digunakan di pengenalan wajah yaitu proses ekstraksi fitur dan teknik klasifikasi yang digunakan. Tujuan dari penelitian ini untuk pengenalan wajah menggunakan metode *Artificial Neural Network* dengan fitur ekstraksi *Local Binary Pattern Histogram* untuk verifikasi peserta ujian. *Local Binary Pattern Histogram* digunakan untuk mengextraksi fitur wajah untuk mendapatkan nilai bobot. Nilai bobot suatu citra testing akan dicocokan dengan nilai bobot citra training menggunakan metode *Artificial Neural Network*. Implementasi sistem verifikasi peserta ujian online ini menggunakan bahasa pemrograman Python. Hasil dari pengujian dilakukan dengan jumlah citra training sebanyak 30 citra yang terdiri dari 10 orang yang berbeda, dan setiap orang mewakili 3 citra. Dengan percobaan 20 citra testing yang dilakukan satu persatu terhadap citra dari 10 orang yang berbeda. Hasil dari penelitian menunjukkan bahwa pengenalan wajah menggunakan metode *Artificial Neural Network* dengan fitur ekstraksi *Local Binary Pattern Histogram* untuk verifikasi peserta ujian berhasil mendapatkan akurasi yaitu sebesar 80% dengan menggunakan pengujian Confusion Matrix.

49

Kata kunci : **Verifikasi, Pengenalan Wajah, Local Binary Pattern Histogram, Artificial Neural Network.**

**PENDAHULUAN****1.1 Latar Belakang**

Ujian Online atau berbasis komputer adalah suatu test atau evaluasi yang dapat dilakukan peserta menggunakan akses internet. Seiring dengan berkembangnya waktu teknologi saat ini menyebabkan aspek pendidikan berpartisipasi berkembangnya teknologi, dalam hal kenggulan, kecepatan dan kemudahan sehingga ujian manual atau konvensional pun beralih ke komputerisasi yaitu salah satunya dengan ujian online [1].

Seiring dengan perkembangan teknologi saat ini, masalah keamanan merupakan aspek yang sangat penting dalam kehidupan masyarakat, dalam hal ini adanya kecurangan dalam pelaksanaan ujian online, kecurangan yang dilakukan berupa peran pengganti sebagai peserta ujian online (*Joki*). Pada umumnya sistem ujian online atau berbasis komputer belum menggunakan sistem keamanan. Untuk itu perlu dilakukan verifikasi dan identifikasi terhadap peserta ujian agar keamanan informasi dapat terjamin dengan baik serta menghindari adanya kecurangan dari peserta ujian / pihak yang tidak bersangkutan di ujian online.

Sistem pengenalan wajah biasa disebut dengan sistem biometrik di *computer vision*. *Computer vision* (CV) adalah sebuah kemampuan sebuah komputer yang ke desain agar mampu melihat sebuah object sehingga mampu menampilkan objek digital dan bisa mengoleksi data secara visual komputer bisa melakukan beberapa pekerjaan yang tidak bisa dilakukan oleh manusia.

Proses identifikasi citra wajah adalah sebuah sistem rekognisi seseorang menggunakan ciri karakteristik yang ada pada wajah. Pengenalan wajah adalah bagian dari *computer vision* yang membahas dan mempelajari ilmu biometrik yang identitasnya ditentukan menggunakan karakter fisik[2]. Untuk mendukung sistem keamanan terhadap suatu akses baik secara langsung

maupun tidak, pelaksanaan ujian online yang bagaimana melakukan identifikasi terhadap keabsahan peserta ujian sehingga verifikasi peserta dapat terjamin dengan baik.

Sistem pengenalan terdiri dua tipe, yakni sistem verifikasi dan identifikasi. Sistem verifikasi berperan penting dalam menerima atau menolak suatu identitas seseorang, sedangkan sistem identifikasi bertujuan untuk mencari suatu identitas seseorang, dan biometrik itu sendiri adalah identifikasi dari karakteristik fisik atau perilaku seseorang [3].

Pada penelitian yang terkait, proses pengenalan wajah untuk validasi peserta ujian online menggunakan metode *haar cascade* dan *eigen face vector*. Berdasarkan hasil uji akurasi pengenalan wajah terhadap 10 orang sistem mampu menghasilkan rata-rata nilai akurasi sebesar 62 % untuk jarak 0.5 Meter dan error sebesar 38 % kemudian 30 % untuk jarak 1 Meter dengan error 70 % terakhir 0 % untuk jarak 2 Meter dan error 100 % [4]. Oleh karena itu pada penelitian ini perlu adanya penambahan metode klasifikasi untuk peningkatan akurasi pengenalan wajah

Salah satu metode fitur ekstrasi untuk pengenalan wajah yang akan digunakan di penelitian ini adalah *Local binary Pattern Histogram* (LBPH). Local Binary Pattern Histogram (LBPH) merupakan sistem terkini dari metode *local binary pattern* (LBP) untuk memperbaiki performa hasil pengenalan wajah. *Local binary pattern* merupakan descriptor tekstur yang dapat digunakan untuk mewakili wajah, karena gambar wajah dapat diamati seperti berupa arsitektur *micro-texture-pattern* yang merupakan suatu operator non parametrik yang mengilustrasikan pengolahan bagian lokal citra. Menurut penelitian terkait metode LBPH lebih baik dalam mengenali wajah secara real-time di banding metode *eigenfaciae*, rata-rata akurasi pada pengujian pertama LPBH : 95.42%, eigenface : 57.50%, pengujian kedua LBPH : 97.09%, eigenface : 72.50 % [5]. Kemudian metode yang digunakan untuk pengenalan wajah adalah metode *Artificial Neural Network* (ANN) yang merupakan representasi buatan dari otak manusia yang kerap mencoba menirukan proses

pembelajaran pada otak manusia. Istilah buatan yang digunakan oleh jaringan saraf ini untuk membuat dengan memanfaatkan suatu program komputer yang mampu menirukan beberapa cara perhitungan selama proses pembelajaran. Pada saat peninjauan metode ANN yang telah dilakukan menghasilkan konteks terbaik dalam jumlah unit pada lapisan tersembunyi 6, dengan jumlah unit input 225, jumlah unit output 10, maximum epoch 2500, target error 0.001 dan learning rate mencapai = 0.9 dengan rata-rata dikenali sebesar 95% [6]. Metode ini digunakan untuk mengenali wajah manusia didalam komputer vision, yang dianggap memiliki distribusi probabilitas yang tinggi untuk sebuah pengenalan wajah.

Berdasarkan uraian diatas, pada penelitian ini akan dibangun sebuah sistem verifikasi ujian online berbasis desktop dengan mengaplikasikan model pengenalan wajah menggunakan metode ANN dan LBPH. Berdasarkan uraian tersebut, maka dalam penelitian ini penulis mengambil judul **“Pengenalan Wajah Menggunakan Metode Artificial Neural Network Dengan Fitur Ekstrasi Local Binary Pattern Histogram Untuk Verifikasi Peserta Ujian”**.

44

## 1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah diuraikan maka identifikasi masalah adalah:

1. Dalam pelaksanaan ujian nasional adanya kecurangan yang dilakukan berupa peran pengganti sebagai peserta ujian online(*Joki*).
2. Penggunaan *Haar Cascade* dan *Eigen Face Vector* masih menghasilkan tingkat kesalahan yang cukup besar.

## 1.3 Rumusan Masalah

Berdasarkan identifikasi masalah, maka dapat dirumuskan masalah penelitian yaitu:

Bagaimana kinerja model pengenalan wajah menggunakan metode ANN dengan fitur ekstrasi LBPH untuk verifikasi peserta ujian online?

## 1.4 Tujuan Penelitian

Berdasarkan rumusan masalah, maka tujuan penelitian ini adalah:

Untuk mengetahui kinerja model pengenalan wajah untuk verifikasi peserta ujian online menggunakan metode ANN dan LBPH yang efektif".

## 11 1.5 Manfaat Penelitian

### 1.5.1 Manfaat Teoritis

Memberikan masukan bagi perkembangan ilmu pengetahuan dan teknologi, khususnya pada bidang computer vision berupa hasil uji coba metode ANN dan LBPH dalam sistem pengenalan wajah untuk verifikasi peserta ujian online.

### 1.5.2 Manfaat Praktis

Sumbangan pemikiran, karya, bahan pertimbangan, atau solusi bagi pembuat sistem guna untuk pengambilan keputusan dalam rangka untuk menghasilkan sistem yang berkualitas sehingga berdampak pula pada instansi terjamin dengan baik.

73  
**BAB II**

## **LANDASAN TEORI**

### **2. 1 Tinjauan Studi**

Sebagai bahan literatur untuk penelitian ini berikut disajikan beberapa penelitian terkait yang menjadi referensi.

**Tabel 2. 1 Penelitian Terkait**

NO	PENELITI	JUDUL	METODE	
1	Mika Tandililing (2016)	Aplikasi Pengenalan Wajah Untuk <sup>7</sup> Validasi Peserta Ujian Online Menggunakan Metode Haar Cascade Dan Eigen Face Vector.	<i>Haar Cascade Dan Eigen Face Vector</i>	
<b>HASIL</b>				
		Berdasarkan hasil implementasi sistem menunjukkan bahwa dari uji fungsional sistem dengan uji Blackbox didapatkan bahwa kinerja sistem secara keseluruhan mampu menghasilkan output sistem yang diinginkan oleh user, kemudian dengan uji akurasi pengenalan wajah <sup>31</sup> terhadap <sup>31</sup> 10 orang, sistem mampu menghasilkan rata-rata nilai akurasi sebesar 62 % untuk jarak 0.5 Meter dan error sebesar 38 %, kemudian 30 % untuk jarak 1 Meter dengan error 70 %, terakhir 0 % untuk jarak 2 Meter dan error 100 %, dari hasil uji fungsional maka dapat dikatakan bahwa sistem ini telah berjalan dengan baik untuk menghasilkan output yang diinginkan dan sesuai dengan input yang diberikan oleh pengguna. Sedangkan hasil dari uji manfaat implementasi sistem validasi ujian online, dengan menggunakan uji kuesioner didapatkan bahwa mayoritas 30 responden menjawab sistem validasi ujian online bermanfaat dan dapat digunakan sebagai sistem yang dapat mengurangi terjadinya kecurangan pada saat ujian online dengan rata-rata total presentase jawaban sangat setuju sebesar 56.3%.		

NO	PENELITI	JUDUL	METODE
2	Harris Simaremare, Agung Kurniawan (2016)	Perbandingan Akurasi Pengenalan 10 Wajah Menggunakan Metode LBPH dan Eigenface dalam Mengenali Tiga Wajah Sekaligus secara Real-Time	LBPH dan Eigenface

### HASIL

	<p>Berdasarkan hasil metode <i>LBPH</i> lebih baik dalam mengidentifikasi 3 wajah sekaligus secara real-time dibanding dengan metode <i>Eigenface</i>. Rata-rata akurasi pada pengujian pertama <i>LBPH</i> : 95.42%, <i>Eigenface</i> : 57.50%. Pengujian kedua <i>LBPH</i> : 97.09%, <i>Eigenface</i> : 72.50%. <i>False rejection rate</i> (FRR) pada metode <i>LBPH</i> lebih rendah dibanding dengan metode <i>Eigenface</i>, yang berarti semakin kecil FRR maka akan semakin besar tingkat akurasi pengenalan wajahnya. Rata-rata FRR pada pengujian pertama <i>LBPH</i> : 0%, <i>Eigenface</i> : 10,56%. Pengujian kedua <i>LBPH</i> : 0%, <i>Eigenface</i> : 3,05%. Sedangkan <i>False acceptance rate</i> (FAR) pada metode <i>LBPH</i> lebih rendah dibanding dengan metode <i>Eigenface</i>, yang berarti semakin kecil FAR maka akan semakin besar tingkat akurasi pengenalan wajahnya. Dengan rata rata FAR pada pengujian pertama <i>LBPH</i> : 1,53%, <i>Eigenface</i> : 7,09%. Pengujian kedua <i>LBPH</i> : 0,97%, <i>Eigenface</i> : 9,17%.</p>
--	--

	PENELITI	JUDUL	METODE
3	Sigit Kusmaryanto (2014)	Jaringan Saraf Tiruan Backpropagation 48 untuk Pengenalan Wajah Metode Ekstraksi Fitur Berbasis Histogram	Jaringan Saraf Tiruan <i>Backpropagation</i>

### HASIL

	<p>Pada saat peninjauan metode ANN yang telah dilakukan menghasilkan konteks terbaik dalam jumlah unit pada lapisan tersembunyi 6, dengan jumlah unit input 225, jumlah unit output 10, maximum epoch 2500, target error 0.001 dan learning rate mencapai = 0.9 dengan rata-rata dikenali sebesar 95%.</p>
--	--

NO	PENELITI	JUDUL	METODE
4	I Nyoman Piarsa, Riza Hisamuddin (2010)	Sistem Verifikasi Online Menggunakan biometrika Wajah	<i>Eigenface</i>
<b>HASIL</b>			
<p><b>6</b></p> <p>Sistem verifikasi online menggunakan biometrika wajah yang menggunakan metode <i>eigenface</i> dalam proses pengenalan dapat digunakan sebagai sistem keamanan tambahan dengan menggunakan nilai ambang 1.1 dan penggunaan 4 <i>eigenface</i>. Nilai ambang dan penggunaan <i>eigenface</i> bersifat berlawanan arah dimana semakin kecil nilai ambang dan semakin besar penggunaan <i>eigenface</i> maka keakuratan sistem semakin akurat, dan sebaliknya keakuratan sistem menjadi tidak akurat. Penggunaan <i>eigenface</i> sangat mempengaruhi akurasi sistem, pemilihan <i>eigenface</i> dapat disesuaikan dengan kepentingan keamanan yang diinginkan. Penggunaan metode <i>eigenface</i> yang rentan terhadap pencayaahan ditafsirkan kurang cocok untuk digunakan di sistem pengenalan dalam sistem verifikasi online, karena seorang pengguna sistem harus dapat terverifikasi pada situasi lingkungan yang memiliki tingkat pencahayaan yang berbeda dari lingkungan pada saat pendaftaran.</p>			

## 2. 2 Tinjauan Pustaka

### 2. 2. 1 Ujian Online

Pelaksanaan ujian merupakan suatu <sup>55</sup> evaluasi terhadap suatu pembelajaran yang dilakukan pada akhir materi. Tujuan ujian dapat berupa <sup>55</sup> sebagai standarisasi tes yang menentukan seseorang untuk memenuhi standar atau tidak berdasarkan kriteria yang telah ditetapkan sebelumnya. <sup>67</sup> Di era modern seperti sekarang ini dengan berkembangnya teknologi informasi sistem ujian mulai bertahap menjadi ujian yang terkomputerisasi. <sup>26</sup> Dan menghasilkan nilai akhir secara *real-time*. Keuntungan yang ditawarkan dari sebuah pelaksanaan ujian yang terkomputerisasi adalah kecepatan dan kemudahan dalam proses pemberian skor. Pengujii tidak lagi melakukan pemeriksaan satu persatu pada lembar jawaban peserta

kemudian menghitung skor melainkan nilai telah tersedia oleh perangkat lunak dalam basis data dengan penghitungan otomatis berdasarkan jawaban dari peserta. Keuntungan lain yang bisa diperoleh adalah pengurangan penggunaan kertas serta pengurangan bentuk kecurangan yang dilakukan peserta. Waktu dan tempat pelaksanaan dapat diatur sedemikian rupa sehingga menyulitkan bagi peserta untuk melaksanakan tindak kecurangan [7].

### 2.2.2 Computer Vision

Menurut Danny Agus Wahyudi, *Computer vision* merupakan salah satu turunan <sup>35</sup> ilmu komputer yang mempelajari bagaimana suatu komputer dapat melihat seperti manusia, faktor pencahayaan sangat penting dalam proses penglihatan. Pencahayaan sangat mempengaruhi kualitas hasil dari *computer vision*, semakin banyaknya cahaya membuat kualitas kontras gambar menjadi lebih baik. Dan sebaliknya jika cahaya <sup>35</sup> semakin berkurang akan membuat kualitas kontras gambar menjadi lebih buruk. Kemampuan di bidang vision dapat di kembangkan, contohnya salah satu <sup>30</sup> robot yang merupakan hasil dari *computer vision*, yang digunakan untuk mengenali lingkungan dan lengan robot yang dapat mengenali objek atau meletakkan sesuatu [8].

### 2.2.3 Pengenalan Wajah

<sup>19</sup> Pengenalan wajah merupakan teknologi baru yang makin kerap dikenal dengan sebutan *face recognition*. Pengenalan wajah memiliki fitur jauh lebih unggul yakni jauh lebih cepat dan akurat. Secara konvensional cara kerjanya dengan cara mengkonversi sebuah foto, sketsa, dan gambar video menjadi deretan angka, yang disebut dengan *faceprint* kemudian membandingkan dengan deretan angka lain yang menyubstitusikan wajah yang sudah dikenali [9]. Secara umum, sistem pengenalan wajah terdiri dari <sup>37</sup> 4 tahapan, yaitu akuisi citra wajah, pra-pengolahan, ekstraksi fitur dan klasifikasi citra wajah. Tahapan yang paling penting dalam sistem pengenalan wajah yaitu tahap ekstraksi fitur dan klasifikasi [10].

11

1. Akuisisi citra adalah bagian awal untuk mendapatkan citra digital.

Tujuan akuisisi citra untuk memastikan data yang diperlukan dan memilih metode perekaman citra digital. Bagian ini dimulai dari sebuah objek yang akan diambil gambarnya, mempersiapan alat-alat, dan sampai pada pencitraan. Pencitraan adalah kegiatan transformasi dari citra tampak (foto, gambar, lukisan, patung, pemandangan dan lain-lain) menjadi citra digital [11].

54

2. *Pre-processing* merupakan proses yang dilakukan untuk menghasilkan ciri atau fitur penting dari sebuah citra. Proses ini meliputi peningkatan kualitas citra (kontras, kecerahan, dan lain-lain), penghilangan noise, perbaikan citra (image restoration), ransformasi (image transformation) dan penentuan bagian citra yang akan diobservasi. *Pre-processing* berfungsi untuk menyeragamkan ukuran citra wajah dan memperbaiki kualitasnya sebelum melalui tahap ekstraksi fitur.

72

3. Ekstraksi fitur adalah suatu proses pengambilan ciri atau fitur dari citra yang nilainya akan digunakan untuk proses klasifikasi (pengenalan). Ekstraksi fitur dapat dibedakan menjadi 3 jenis (Kalel, Pisal, & Bagawade, 2016) :

- a. Ekstraksi Warna

Ruang warna mewakili warna dalam bentuk nilai intensitas. Kita bisa menentukan, memvisualisasikan dan menciptakan warna dengan menggunakan metode ruang warna. Metode yang sering digunakan untuk ekstraksi warna antara lain *Histogram Intersection*, *Zernike Chromaticity Distribution Moments* dan *Color Histogram*.

- b. Ekstraksi Bentuk

Bentuk adalah sumber informasi utama yang bisa digunakan untuk pengenalan objek. Tanpa bentuk, objek konten visual tidak bisa dikenali dengan baik. Dua objek tidak bisa memiliki bentuk yang sama persis tapi dengan menggunakan berbagai algoritma kita bisa mengenali bentuk yang serupa dengan mudah. Metode yang sering

54

digunakan untuk ekstraksi bentuk antara lain *Binary Image* dan segmentasi secara horizontal dan vertikal.

c. Ekstraksi Tekstur

Tekstur mengandung informasi penting tentang susunan dasar permukaan objek. Metode yang sering digunakan untuk ekstraksi tekstur antara lain *Gray Level Co-Occurrence Matrix* (GLCM), *Gray Level Run Length Matrix* (GLRLM), dan *Edge Detection*.

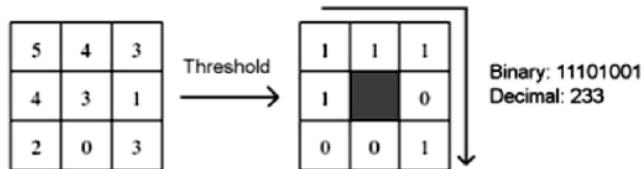
4. Klasifikasi citra wajah(Pengenalan) merupakan teknik yang digunakan untuk menentukan item dari suatu dataset kedalam kategori atau kelas tertentu. Tujuan klasifikasi adalah memprediksi kelas target secara akurat disetiap kasus yang terdapat dalam data. Pada bagian ini, pola data wajah yang dihasilkan pada tahap pengenalan yang akan dibandingkan dengan data pola yang tersimpan pada basis data wajah untuk mendapatkan hasil pola wajah yang dikenali.

#### 2. 2. 4 Metode Local Binary Pattern Histogram

Ada beberapa algoritma yang dapat digunakan untuk mengekstraksi fitur penting dari suatu gambar wajah. Salah satu algoritma yang digunakan untuk mengekstraksi fitur wajah adalah algoritma *Linier Binary Pattern Histogram*. Dengan algoritma ini memungkinkan untuk menggambarkan tekstur dalam bentuk citra digital. Hal ini dilakukan dengan cara membagi gambar menjadi beberapa bagian kecil dari fitur yang akan diekstraksi. Fitur-fitur ini terdiri dari pola biner yang menggambarkan lingkungan piksel di daerah tersebut. Fitur yang diperoleh kemudian dirubah menjadi histogram fitur tunggal yang membentuk representasi dari gambar [12].

*Local Binary Pattern Histogram* suatu metode yang mengklasifikasi dengan cara menggabungkan dengan histogram. *Local Binary Pattern Histogram* merupakan proses baru dari metode *local binary pattern* untuk mengubah performa hasil pengenalan wajah. Pada dasarnya metode *local binary pattern* hanya untuk pengenalan tekstur. *Local Binary Pattern Histogram* merupakan metode yang paling cocok untuk melakukan proses

<sup>16</sup> pengenalan citra wajah [13]. *Operator local binary pattern* diperkenalkan oleh Timo Ojala. Versi asli dari local binary pattern bekerja dalam sebuah blok  $3 \times 3$  piksel sebuah citra. Pada piksel tengah (pusat) threshold dengan nilai dari piksel tetangga akan menghasilkan sebuah angka binary [14].  
<sup>4</sup> Proses metode LBPH sama dengan original-LBP. Pembagian daerah suatu citra hasil dari cropping menjadi beberapa bagian, pada eksperimen <sup>4</sup> ini menggunakan komposisi citra menjadi  $8 \times 8$ , setelah melakukan proses ekstraksi fitur untuk mendapatkan nilai histogram dengan cara memperbarui setiap nilai pixel dari citra menjadi thresholding dengan ukuran  $3 \times 3$  tetangga, nilai tengah pixel akan digunakan sebagai pembeda.  
<sup>4</sup> Apabila nilai tetangga lebih besar atau sama besar dari nilai pembanding maka nilai pixel diubah menjadi bilangan biner 1, dan sebaliknya apabila nilai tetangga lebih kecil dari nilai pembanding maka nilai pixel diubah <sup>47</sup> menjadi bilangan biner 0. Setelah mendapatkan bilangan biner yang kemudian nilai akan diubah menjadi bilangan desimal yang menghasilkan nilai baru untuk nilai pixel tengah, selanjutnya bilangan biner menjadi nilai LBP. Seperti yang ditunjukkan pada gambar (2.1), nilai desimal 233 itu yang akan menjadi nilai tengah pixel selanjutnya [15].



Gambar 2.1 Operator LBP

4

Secara umum penghitungan LBP ini bisa dilakukan dengan rumus berikut:

$$L B P_{P,R} = \sum_{p=0}^{p=x-1} s(gp - gc)2^p \quad (2.1)$$

4

Dengan P adalah nilai sampling point yang digunakan dan R adalah besar radius.  $s(gp - gc)2^p$  gp adalah besar nilai pixel tetangga, gc besar nilai tengah pixel pada sampling point. Pada penghitungan ini dilakukan replacing pixel atau penggantian pixel dengan nilai nol dan satu seperti pada gambar 2.1. Untuk pengklasifikasian citra dengan menghitung perbedaan jarak terkecil antar histogram dari gambar yang dihasilkan [15].

1

## 2. 2. 5 Metode Artificial Neural Network

*Artificial neural network (ANN)* pertama kali di temukan oleh McCulloch dan Pitts pada tahun 1943. Bobot dalam jaringan yang diusulkan oleh McCulloch dan Pitts diatur untuk melakukan fungsi logika sederhana. Fungsi aktivasi yang dipakai adalah fungsi threshold [16].

*Artificial neural network (ANN)* yang merupakan contoh rekaan dari otak manusia yang kerap berusaha menirukan prosedur pembelajaran pada otak manusia. Istilah dari buatan yang diaplikasikan oleh jaringan saraf untuk membuat dengan cara memanfaatkan suatu program komputer yang sanggup menirukan beberapa cara perhitungan dalam prosedur pembelajaran. Jaringan Saraf Tiruan adalah salah satu contoh perhitungan yang biasa mengikuti proses kegiatan sistem otak manusia. Seperti halnya *Jaringan Saraf Biologis*, Jaringan Saraf Tiruan juga mempunyai keahlian dalam proses belajar dan menyesuaikan tentang masukan-masukan. Jaringan Saraf Tiruan yang cara kerjanya menyerupai otak manusia terdapat dua hal, yaitu:

- a. Ilmu yang didapat dari jaringan melalui proses belajar
- b. Kekuatan hubungan antara sel syaraf (neuron) yang dikenal sebagai bobot-bobot sinaptik digunakan untuk menyimpan pengetahuan.

Arsitektur backpropagation menggunakan error output untuk mengubah nilai bobot-bobotnya dalam arah mundur (backward)

Kusumadewi (2004). Pada saat perambatan maju, neuron-neuron diaktifkan dengan menggunakan fungsi aktivasi yang didefinisikan, seperti:

Fungsi sigmoid:

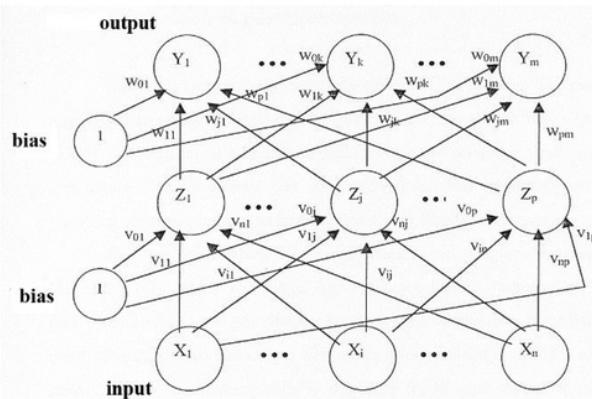
$$y = f(x) = \frac{1}{1+e^{-\theta x}} \quad (2.2)$$

Fungsi tansig:

$$y = f(x) = \frac{1-e^{-2x}}{1+e^{-2x}} \quad (2.3)$$

Gambar di bawah ini adalah arsitektur backpropagation dengan  $n$  buah input (ditambah sebuah bias), sebuah layer tersembunyi (hidden layer) yang terdiri dari  $p$  unit (ditambah sebuah bias), serta  $m$  buah unit output.

*v<sub>ji</sub>* bobot yang menghubungkan neuron input  $X_i$  ke neuron pada hidden layer  $Z_j$ .  $v_{j0}$  adalah bobot yang menghubungkan bias di neuron input ke neuron hidden layer  $Z_j$ .  $w_{kj}$  adalah bobot dari neuron hidden layer  $Z_j$  ke neuron keluaran  $y_k$ ,  $w_{k0}$  adalah bobot dari bias di hidden layer ke neuron keluaran  $y_k$ . Fungsi aktivasi yang digunakan, antara lapisan input dan hidden layer, dan antara hidden layer dengan lapisan output adalah fungsi aktivasi logsig [17].



30

Gambar 2. 2 Arsitektur Jaringan Backpropagation

Secara umum lapisan **atau** layer dalam suatu jaringan terdapat **3** bagian yaitu **input layer, hidden layer, dan output layer.**

Penjelasan dari tiga bagian yang ada dilapisan (layer), yaitu:[18]

- 1** a. **Input Layer** merupakan lapisan yang menerima masukan langsung dari lingkungan. Rata-rata jumlah input layer disesuaikan dengan jumlah masukan.
- 30** b. **Hidden Layer** merupakan lapisan-lapisan yang berada di antara **input layer** dan **output layer** (gambar 2.2). Terdapat dua lapisan didalam arsitektur *Backpropagation*, dan hidden layer tidak termasuk. Total neuron yang ada di hidden layer bermacam-macam. Total neuron dan total lapisan pada hidden layer digunakan untuk mendapatkan hasil penyelesaian yang mempunyai kesalahan minimum merupakan tahap penting dalam observasi ANN.
- c. **Output Layer** merupakan lapisan yang menciptakan output akhir dari suatu jaringan ANN. Total neuron pada lapisan akan disesuaikan dengan jumlah output yang ditentukan.

Pada *Backpropagation*, fungsi aktivasi yang digunakan wajib melengkapi sebagian **syarat**, yaitu : kontinyu, terdeferensial dengan mudah dan merupakan fungsi yang tidak turun. Fungsi yang dipakai adalah sigmoid biner, tansig dan purelin, dengan persamaan sebagai berikut:

Fungsi sigmoid:

$$y = f(x) = \frac{1}{1+e^{-\phi x}} \quad (2.4)$$

$$y^1 = f^1(x) = \phi f(x)[1 - f(x)] \quad (2.5)$$

Fungsi tansig:

$$y = f(x) = \frac{1}{1+e^{-2x}} \quad (2.6)$$

$$y^1 = f^1(x) = [1 + f(x)][1 - f(x)] \quad (2.7)$$

Algoritma Pelatihan

- a. Algoritma Pelatihan Algoritma Backpropagation [19].

1. Langkah awal, **Inisialisasi bobot dan bias, baik bobot maupun bias yang dapat diset dengan angka secara acak dan biasanya angka di sekitar 0 dan 1 atau -1 (bias positif atau negatif).**
2. Langkah pertama, apabila **stopping condition** <sup>70</sup> masih belum terlaksana, maka jalankan langkah ke-2 sampai ke-9
3. Langkah ke-2, untuk setiap data *training*, buat langkah ke-3 sampai ke-8
4. Langkah ke-3, setiap bagian *input* ( $X_i, i = 1, \dots, n$ ) memperoleh **sinyal *input*  $x_i$**  dan menyebarkan sinyal tersebut pada seluruh unit *hidden layer*. Perlu diketahui bahwa *input*  $x_i$  yang dipakai di sini adalah *input training data* yang sudah diskalakan.
5. Langkah ke-4, setiap *hidden unit* ( $Z_j, j = 1, \dots, p$ ) hendak menjumlahkan sinyal-sinyal *input* yang sudah berbobot, termasuk biasnya.

$$z_{in_j} = V_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (2.8)$$

dan menggunakan **fungsi aktivasi** yang sudah **ditentukan** untuk menghitung sinyal *output* dari *hidden unit* yang berkaitan,

$$z_j = f(z_{in_j}) \quad (2.9)$$

kemudian memberi **sinyal *output*** ini ke seluruh *unit* pada *unit output*

6. Langkah ke-5, setiap *unit output* ( $Y_k, k = 1, \dots, m$ ) hendak menjumlahkan sinyal-sinyal *input* yang sudah berbobot termasuk biasnya,

$$y_{in_k} = w_{0k} + \sum_{j=1}^p z_j w_{jk} \quad (2.10)$$

dan memanfaatkan peran **aktivasi** yang telah **ditentukan** untuk menghitung sinyal *output* dari *unit output* yang berkaitan

$$z_j = f(z_{in_j}) \quad (2.11)$$

7. Langkah ke-6, **propagasi balik *error* (backpropagation of error).** Setiap *unit output* ( $Y_k, k = 1, \dots, m$ ) memperoleh suatu target (*output yang diharapkan*) yang akan membandingkan dengan *output* yang diperoleh.

$$\delta_k = (t_k - y_k) f'(y_{in_k}) \quad (2.12)$$

Faktor  $\delta_k$  ini digunakan untuk menaksir koreksi *error* ( $\Delta w_{jk}$ ) yang nantinya hendak dipakai untuk memperbarui  $w_{jk}$ , di mana:

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (2.13)$$

5 Selain itu juga dihitung koreksi bias  $\Delta w_{0k}$  yang nantinya hendak dipakai untuk memperbarui  $w_{0k}$ , dimana:

$$\Delta w_{0k} = \alpha \delta_k \quad (2.14)$$

Faktor  $\delta_k$  ini kemudian dikirimkan ke layer di depannya

8. Langkah ke-7, Setiap *hidden unit* ( $z_j, j = 1, \dots, p$ ) menjumlah *input delta* (yang dikirim dari *layer* pada langkah ke-6) yang sudah berbobot.

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk} \quad (2.15)$$

Setelah itu hasil akan dikalikan dengan turunan dari fungsi aktivasi yang digunakan jaringan untuk menghasilkan faktor koreksi *error*  $\delta_j$ , di mana:

$$\delta_j = \delta_{in_j} f'(z_{in_j}) \quad (2.16)$$

5 Faktor  $\delta_j$  ini digunakan untuk menghitung koreksi *error* ( $\Delta V_{ij}$ ) yang nantinya akan dipakai untuk memperbarui  $v_{ij}$ , dimana:

$$\Delta V_{ij} = \alpha \delta_j x_i \quad (2.17)$$

Selain itu juga dihitung koreksi bias  $\Delta V_{0j}$  yang nantinya akan dipakai untuk memperbarui  $v_{0j}$ , di mana:

$$\Delta V_{0j} = \alpha \delta_j \quad (2.18)$$

9. Langkah ke-8 pembaharuan bobot dan bias:

Setiap *unit output* ( $Y_k, k = 1, \dots, m$ ) akan memperbarui bias dan bobotnya dengan setiap *hidden unit*.

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (2.19)$$

- 20 10. Demikian pula untuk setiap *hidden unit* akan memperbarui bias bobotnya dengan setiap *unit input*.

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (2.20)$$

11. Langkah ke-9 membuktikan <sup>3</sup> *stopping condition*, apabila *stop condition* telah terpenuhi, maka training jaringan dapat dihentikan. Untuk memastikan *stopping condition* diperoleh dua cara, yaitu: (1) memisahkan iterasi yang ingin dilakukan. Contoh jaringan akan dilatih sampai iterasi yang ke-500. Maksut dari literasi adalah <sup>1</sup> perulangan langkah ke-3 sampai langkah ke-8 untuk semua data <sup>3</sup> *training* yang ada. (2) memisahkan *error*.

Contoh untuk menentukan besar antara output yang dikehendaki dan output yang diperoleh dari jaringan. Jika diperoleh sebanyak *m* data *training*, maka untuk menghitung *Mean Square Error* digunakan persamaan berikut:

$$MSE = 0.5 \{(tk1 - yk1)^2 + (tk2 - yk2)^2 + \dots + (tkm - ykm)^2\} \quad (2.21)$$

Setelah penelitian selesai, *backpropagation network* (BPN) ditafsir sudah mampu, sehingga kalaupun jaringan diberi input tertentu, maka jaringan menghasilkan output yang diinginkan. Untuk mendapatkan output tersebut ialah dengan cara mengimplementasikan metode *backpropagation* yang menyerupai proses pelatihan, melainkan hanya pada tahap umpan majunya saja. Catatan yang digunakan untuk pengujian algoritma:

*X<sub>i</sub>* = *Unit Input* ke-*i*

*Z<sub>j</sub>* = *Hidden Unit* ke-*j*

*Y<sub>k</sub>* = *Unit output* ke-*k*

*V<sub>0j</sub>* = Bias untuk *hidden unit* ke-*j*

*V<sub>ij</sub>* = Bobot antara *unit input* ke-*I* dengan *hidden unit* ke-*j*

*W<sub>0k</sub>* = Bias *unit output* ke-*k*

*W<sub>jk</sub>* = Bobot antara *hidden unit* ke-*j* dengan *unit output* ke-*k*

- b. Langkah-langkah eksperimen pada model ANN, sebagai berikut:

1. Langkah awal, <sup>5</sup> Inisialisasi bobot sesuai dengan bobot yang telah diperoleh dari proses pelatihan di atas;
2. Langkah ke-1 Untuk setiap *input*, buat langkah ke2 sampai ke-4;

3. Langkah ke-<sup>5</sup> untuk setiap *input*  $i = 1, \dots, n$  skalakan bilangan dalam *range* fungsi aktivasi seperti yang dilakukan pada proses pelatihan di atas;
4. Langkah ke-<sup>6</sup> untuk  $j = 1, \dots, p$ :

$$z_{in_j} = v_{0j} + \sum_{i=1}^n x_i v_{ij}$$

$$z_j = f(z_{in_j})$$

5. Langkah ke-<sup>7</sup>: Untuk  $k=1, \dots, m$ :

$$y_{in_k} = w_{0k} + \sum_{j=1}^p z_j w_{jk}$$

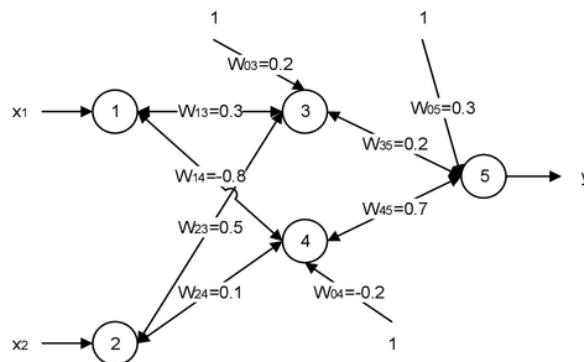
$$y_k = f(y_{in_k})$$

<sup>27</sup> Variabel  $y_k$  adalah *output* yang masih dalam skala menurut range fungsi aktivasi. Untuk mendapatkan nilai *output* yang sesungguhnya,  $y_k$  harus dikembalikan seperti semula.

Untuk memudahkan perhitungan, misalkan jaringan terdiri dari 2 unit input (X1 dan X7) dan 1 *output* yang diambil dari dataset untuk data ke 2, 16, 41, dan 66.cc

**Tabel 2. 2** Contoh Dataset

No	X1	X7	Y
2	2	3	0
16	2	6	1
41	1	4	1
62	1	2	0



Kondisi bobot saat inisialisasi seperti ditunjukkan pada gambar di atas dengan jumlah neuron pada layer tersembunyi = 2.

Laju pembelajaran ( $\eta$ ) = 0.1;

Fungsi aktivasi yang digunakan adalah Sigmoid Biner;

Target  $error = 0.0001$  dengan kriteria SSE;

Momentum yang digunakan = 0.95;

Maksimum jumlah iterasi pelatihan = 1,000 kali.

Maka:

1. Nilai pada neuron di hidden layer 3 dan 4:

$$v_j(p) = \sum_{i=1}^n x_i(p)w_{ij}(p)$$

$$y_j(p) = \frac{1}{1 + e^{-v_j(p)}}$$

(P) adalah iterasi ke-i.

$$v_3(1) = x1w_{13} + x2w_{23} + 1w_{03} = 2 * 0.3 + 3 * 0.5 + 1 * 0.2 = 2.3$$

$$y_3(1) = \frac{1}{1+e^{-2.3}} = \frac{1}{1+2.71828^{-2.3}} = 0.908877$$

$$v_4(1) = x1w_{14} + x2w_{24} + 1w_{04} = 2 * (-0.8) + 3 * 0.1 + 1 * (-0.2) = 1.7$$

$$y_4(1) = \frac{1}{1+e^{-1.7}} = 0.845535$$

2. Nilai pada neuron di output layer:

$$v_k(p) = \sum_{j=1}^m x_j(p)w_{jk}(p)$$

$$y_k(p) = \frac{1}{1 + e^{-v_k(p)}}$$

$$v_5(1) = Y3(1)w_{35} + Y4(1)w_{45} + 1w_{05} = 0.908877 * 0.2 + 0.845535 * 0.7 + 1 * 0.3 = 2.054411496$$

$$y_5(1) = \frac{1}{1+e^{-2.054411496}} = 0.886392478$$

3. Hitung gradient error untuk neuron pada output layer:

$$e_k(p) = y_{dk}(p) - y_k(p)$$

$$\delta_k(p) = y_k(p) * [1 - y_k(p)] * e_k(p)$$

Untuk data pertama, target/class nilai yang diharapkan adalah  $y_d = 0$ , sedangkan keluaran yang didapatkan  $y_5(1) = 0.886392478$ . Maka hitung *error* pada iterasi pertama untuk data pertama pada neuron 5 di output layer:

$$e_5^1(1) = y_d - y_5(1) = 0 - 0.886392478 = \\ \mathbf{-0.886392478}$$

$$\delta_5(1) = y_5(1) * (1 - y_5(1)) * e_5^1(1) = 0.886392478 * (1 - 0.886392478) * (-0.886392478) = -0.089260478$$

4. Hitung koreksi bobot untuk output layer,  $\Delta w_{35}, \Delta w_{45}$ , dan  $\Delta w_{05}$ :

$$\Delta w_{jk}(\mathbf{p}) = \eta \cdot y_j(\mathbf{p}) * \delta_k(\mathbf{p})$$

$$\Delta w_{35} = \eta * y_3(1) * \delta_5(1) = 0.1 * 0.91 * (-0.089) = \\ -0.008112679$$

$$\Delta w_{45} = \eta * y_4(1) * \delta_5(1) = 0.1 * 0.85 * (-0.089) = \\ -0.007547282$$

$$\Delta w_{05} = \eta * 1 * \delta_5(1) = 0.1 * 1 * (-0.089) = \\ -0.0089266048$$

5. Hitung *gradient error* pada hidden layer  $\delta_3(1)$  dan  $\delta_4(1)$ :

$$\delta_j(\mathbf{p}) = y_j(\mathbf{p}) * [(1 - y_j(\mathbf{p})) + \sum_{k=1}^1 \delta_k(\mathbf{p}) * w_{jk}(\mathbf{p})]$$

$$\delta_3(1) = y_3(1) * (1 - y_3(1)) * \sum_{k=1}^1 \delta_k(1) * w_{3k}(1) = y_3(1) * (1 - y_3(1)) * \delta_5(1) * w_{35}(1) \\ = 0.91 * (1 - 0.91) * (-0.089) * 0.2 = -0.001478505$$

$$\delta_4(1) = y_4(1) * (1 - y_4(1)) * \sum_{k=1}^1 \delta_k(1) * w_{4k}(1) = y_4(1) * (1 - y_4(1)) * \delta_5(1) * w_{45}(1) \\ = 0.85 * (1 - 0.85) * (-0.089) * 0.7 = -0.008160558$$

6. Hitung koreksi bobot untuk hidden layer  $\Delta w_{13}, \Delta w_{23}, \Delta w_{03}, \Delta w_{14}, \Delta w_{24}$ , dan  $\Delta w_{04}$

$$\Delta w_{ij}(\mathbf{p}) = \eta * x_i(\mathbf{p}) * \delta_j(\mathbf{p})$$

$$\Delta w_{13} = \eta * x_1 * \delta_3(1) = 0.1 * 2 * (-0.0015) = -0.0003$$

$$\Delta w_{23} = \eta * x2 * \delta_3(1) = 0.1 * 3 * (-0.0015) = -0.00044$$

$$\Delta w_{03} = \eta * 1 * \delta_3(1) = 0.1 * 1 * (-0.0015) = -0.00015$$

$$\Delta w_{14} = \eta * x1 * \delta_4(1) = 0.1 * 2 * (-0.0082) = -0.00163$$

$$\Delta w_{24} = \eta * x2 * \delta_4(1) = 0.1 * 3 * (-0.0082) = -0.00245$$

$$\Delta w_{04} = \eta * 1 * \delta_4(1) = 0.1 * 1 * (-0.0082) = -0.00082$$

7. Perbaharui bobot untuk neuron pada hidden layer  $w_{35}, w_{45}, w_{05}$ :

$$w_{ij}(p + 1) = w_{ij}(p) + \Delta w_{ij}(p)$$

$$w_{35}(2) = w_{35}(1) + \Delta w_{35} = 0.2 + (-0.008112679) =$$

$$0.191887321$$

$$w_{45}(2) = w_{45}(1) + \Delta w_{45} = 0.7 + (-0.007547282) =$$

$$0.692452718$$

$$w_{05}(2) = w_{05}(1) + \Delta w_{05} = 0.3 + (-0.0089266048) =$$

$$0.291073952$$

8. Perbaharui bobot pada layer tersembunyi,  
 $w_{13}, w_{23}, w_{03}, w_{14}, w_{24}, w_{04}$ :

$$w_{13}(2) = w_{13}(1) + \Delta w_{13} = 0.3 + (-0.0003) = 0.299704$$

$$w_{23}(2) = w_{23}(1) + \Delta w_{23} = 0.5 + (-0.00044) = 0.499556$$

$$w_{03}(2) = w_{03}(1) + \Delta w_{03} = 0.2 + (-0.00015) = 0.199852$$

$$w_{14}(2) = w_{14}(1) + \Delta w_{14} = 0.8 + (-0.00163) = \\ -0.798368$$

$$w_{24}(2) = w_{24}(1) + \Delta w_{24} = 0.1 + (-0.00245) = 0.097552$$

$$w_{04}(2) = w_{04}(1) + \Delta w_{04} = (-0.2) + (-0.00082) = \\ -0.20082$$

9. Naikkan 1 langkah iterasi (p), kembali ke langkah 2 dan ulangi proses tersebut sampai *kriteria error* tercapai.

10. Dengan demikian, didapatkan bobot akhir pada iterasi pertama untuk data pertama[2, 3]:

$$w_{13} = 0.299704;$$

$$w_{23} = 0.499556 ;$$

$$w_{03} = 0.199852;$$

$$w_{14} = -0.798368;$$

$$w_{24} = 0.097552;$$

$$w_{04} = -0.20082;$$

$$w_{35} = 0.191887321;$$

$$w_{45} = 0.692452718;$$

$$w_{05} = 0.291073952$$

Error yang didapatkan dadalah = = **-0.886392478**.

Selanjutnya proses di atas dulangi untuk data ke-dua [2, 6], data ke-tiga [1, 4], dan data ke-empat [1, 2] sehingga iterasi pertama selesai dilakukan. Pada setiap data yang diproses dalam satu iterasi, error keluaran disimpan untuk dihitung sebagai kriteria error, seperti SSE, MSE, dsb. Apabila hasil MSE < error yang didapatkan maka iterasi berhenti, sebaliknya dilakukan perambatan terus hingga batas perulangan/epoch.

## 2. 2. 6 Evaluasi Model

Evaluasi model dalam penelitian ini dilakukan untuk menguji performansi dari sistem yang dibangun. Beberapa cara untuk melakukan evaluasi adalah dengan menghitung akurasi (*accuracy*), presisi (*precision*), dan *recall* dari hasil analisis sistem. Metode yang digunakan pada penelitian ini adalah *Confusion Matrix*.

*Confusion matrix* merupakan salah satu metode yang dapat digunakan untuk mengukur kinerja suatu proses klasifikasi. Dengan

*confusion matrix* dapat menganalisa seberapa baik *classifier* dapat mengidentifikasi *record* dari kelas-kelas yang berbeda.

**Tabel 2, 3 Model Confusion Matrix**

Klasifikasi Sebenarnya	Hasil Klasifikasi	
	+	-
+	TP 17	FP
-	FN	TN

**Keterangan :**

1. TP (*True Positive*) merupakan banyaknya data yang nilai aktualnya adalah nilai positif dengan hasil prediksinya merupakan nilai positif,
2. 2. FP (*False Positive*) merupakan banyaknya data yang nilai aktualnya adalah nilai negatif dengan hasil prediksinya merupakan nilai positif,
3. FN (*False Negative*) merupakan banyaknya data yang nilai aktualnya adalah nilai positif dengan hasil prediksinya merupakan nilai negatif,
4. TN (*True Negative*) merupakan banyaknya data yang nilai aktualnya adalah nilai negatif dengan hasil prediksinya merupakan nilai negatif.

65

Perhitungan akurasi dengan tabel *confusion matrix* adalah sebagai berikut:

1. *Accuracy* merupakan sistem pengujian atas fase kedekatan antara nilai prediksi dengan nilai aktual. Dengan mengetahui jumlah data yang diklasifikasi secara benar maka dapat diketahui hasil akurasi prediksi. Persamaan akurasi sebagai berikut.

$$Accuracy = \frac{N_{Benar}}{N} \times 100\%$$

2. *Precision* merupakan sistem pengujian dengan melakukan perbandingan jumlah informasi relevan yang didapatkan sistem dengan

jumlah seluruh informasi yang terambil oleh sistem baik yang relevan maupun tidak. Persamaan presisi sebagai berikut.

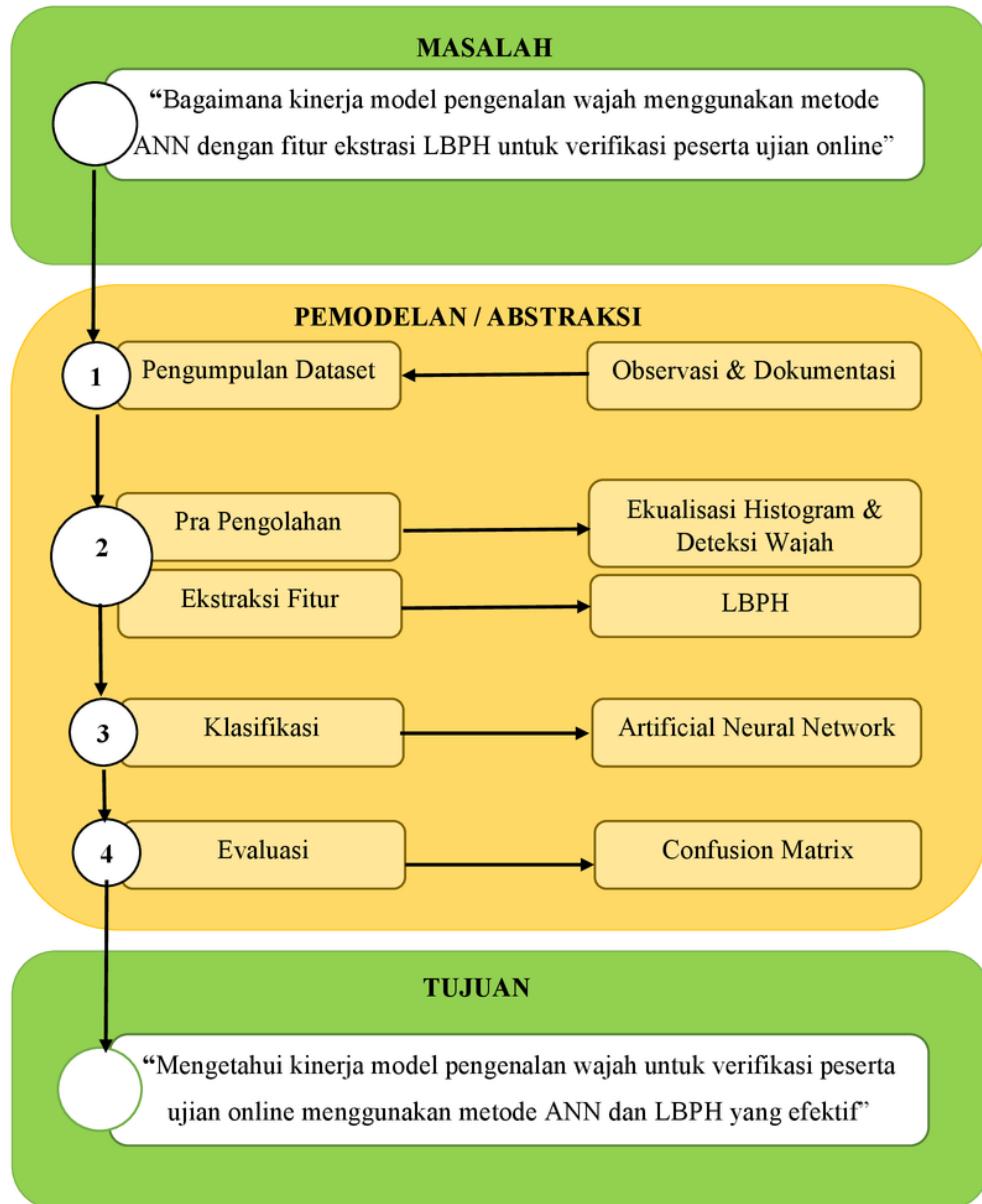
$$Precision = \frac{TP}{TP + FP}$$

2

3. *Recall* merupakan sistem pengujian yang membandingkan jumlah informasi relevan yang didapatkan sistem dengan jumlah seluruh informasi relevan yang ada dalam koleksi informasi. Persamaan *recall* sebagai berikut.

$$Recall = \frac{TP}{TP + FN}$$

### 2. 3 Kerangka Pikir



**Gambar 2. 3** Bagan Kerangka Pikir

## BAB III

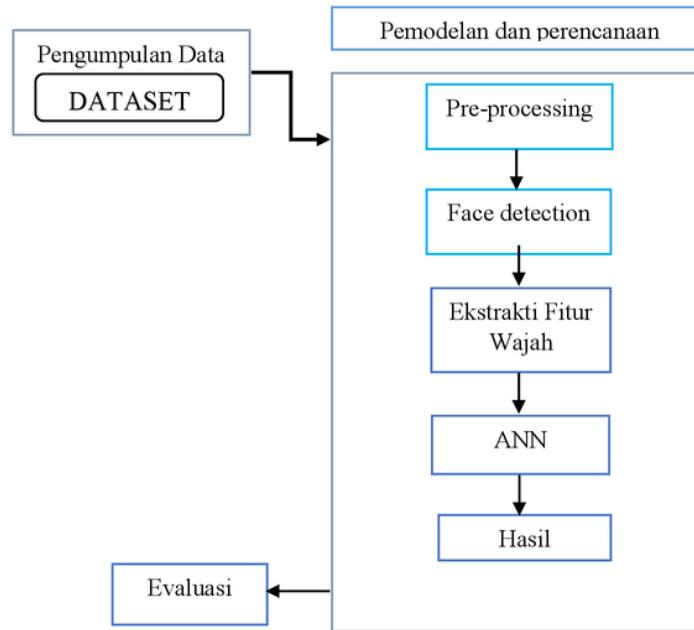
### METODE PENELITIAN

#### 3.1 Jenis, Metode, Subjek, Objek, Waktu, Dan Lokasi Penelitian

Dipandang dari tingkat penerapannya, maka penelitian ini merupakan penelitian terapan karena penelitian ini berfokus penerapannya untuk memberikan solusi atas permasalahan secara praktis. Dipandang dari jenis informasi yang diolah, maka penelitian ini merupakan penelitian kuantitatif. Dipandang dari perlakuan terhadap data, maka penelitian ini merupakan penelitian konfirmatori.

Penelitian ini menggunakan metode penelitian eksperimen. Dengan demikian jenis penelitian ini adalah penelitian eksperimental. Subjek penelitian ini adalah verifikasi pengenalan wajah pada objek peserta ujian Online.

Adapun metode penelitian ini diusulkan seperti gambar dibawah ini:



**Gambar 3. 1 Metode Penelitian**

### **1 3.2 Pengumpulan Data**

Dataset yang digunakan dalam penelitian ini adalah dataset public dengan  
22  
nama The MIT-CBCL face recognition database, dataset yang di unduh dari  
36  
<http://cbcl.mit.edu/software-datasets/heisele/facerecognition-database.html>.

Dataset ini merupakan data yang pada umumnya digunakan dalam eksperiment  
untuk verifikasi deteksi wajah.

Berikut penjelasan IMAGES DATA SET :

22  
Nama Dataset : The MIT-CBCL face recognition database

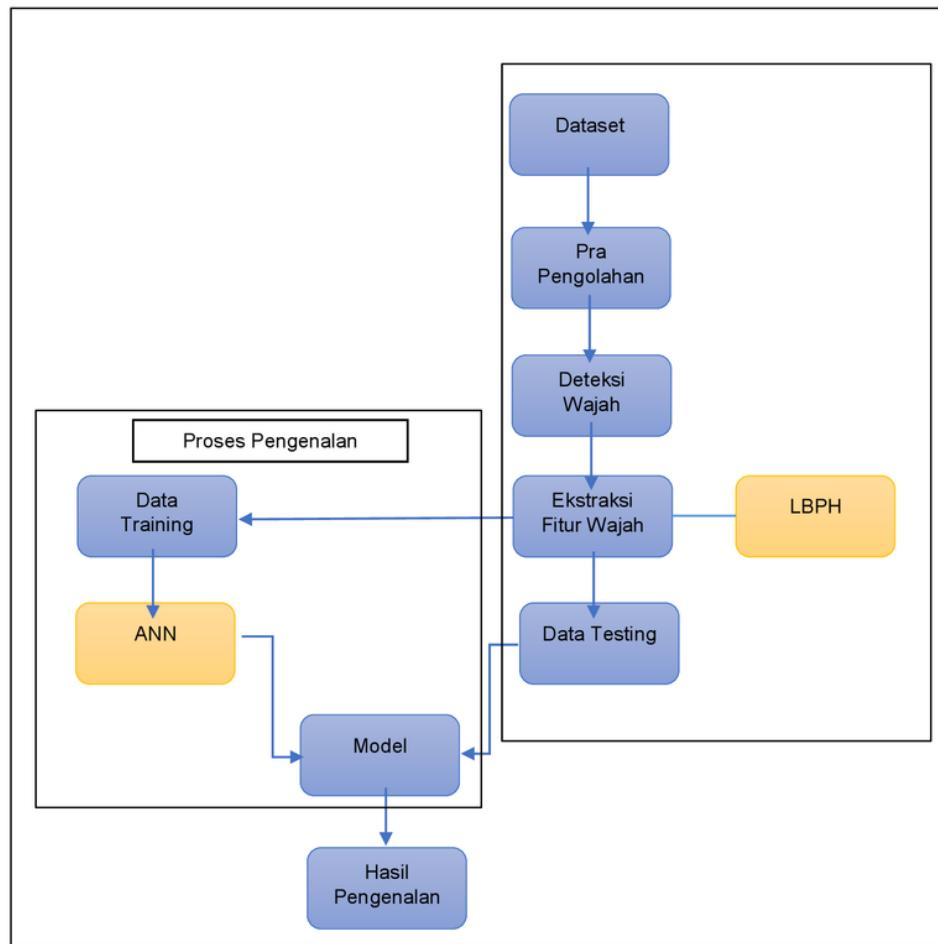
Jumlah Individu : 10 Subjek, Wanita (3), Pria (7)

Directory : training-originals 59 image

Subdirektori : training\_used\_in\_fpiv04\_paper 30 image  
22  
URL

<http://cbcl.mit.edu/software-datasets/heisele/facerecognition-database.html>

### 3.3 Pemodelan



**Gambar 3. 2 Pemodelan**

#### 3.3.1 Pra-pengolahan

*Pre-processing* dilakukan untuk menghasilkan ciri atau fitur penting dari sebuah citra. Proses ini meliputi peningkatan kualitas citra dan penetapan elemen citra yang akan diobservasi. Pada penelitian ini <sup>43</sup> *Pre-processing* menggunakan *Histogram Equalization* dan *Face Detection* yang berfungsi untuk menyeragamkan ukuran citra wajah dan memperbaiki kualitasnya sebelum melalui tahap ekstraksi fitur.

### 3.3.2 1 Ekstraksi Fitur Wajah

Ekstraksi fitur berfungsi sebagai pendekripsi ciri dari suatu citra. Ciri yang dapat digunakan untuk membedakan objek satu dengan objek yang lainnya, diantaranya ciri bentuk, ciri ukuran, ciri geometri, ciri tekstur dan ciri warna. Pada penelitian ini 1 digunakan ekstraksi ciri menggunakan Local 32 Binary Pattern Histogram. Masing-masing objek diekstrak cirinya berdasarkan 32 parameter-parameter tertentu dan dikelompokan pada kelas tertentu. Nilai dan parameter-parameter tersebut kemudian dijadikan sebagai data masukan dalam proses identifikasi/klasifikasi.

### 3.3.3 Data Training

Data training berupa data yang telah terekstraksi cirinya yang selanjutnya akan dilatih menggunakan algoritma *Artificial Neural Network*, algoritma ini akan menentukan / mencari bobot yang terbaik. Data training ini berupa hasil dari ekstraksi *Gray Level Co-occurrence Matrix* terhadap pengenalan wajah manusia

### 3.3.4 Training Menggunakan ANN

*Training* menggunakan *Artificial Neural Network* yaitu menjadikan data training menjadi data inputan. untuk menemukan model yang tepat. Arsitektur dari algoritma *Artificial Neural Network* akan di lakukan secara eksperimen untuk menentukan komposisi jumlah layer dan neuron yang tepat untuk mendapatkan hasil kinerja terbaik.

### 3.3.5 Model

Model merupakan hasil dari proses *training* dari algoritma *Artificial Neural Network* menggunakan 1 data training

### 3.3.6 Data Testing

Data testing merupakan data yang telah terekstrasi cirinya, untuk menguji data yang telah dilatih. Prosedur atau 15 langkah-langkah pokok dalam pengenalan wajah menggunakan *Artificial Neural Network* dengan Fitur Ekstraksi *local binary pattern Histogram* dan pada proses verifikasi untuk pengenalan dengan menggunakan alat bantu *Python*.

### 3.3.7 Evaluasi<sup>77</sup>

Model yang telah dihasilkan kemudian dievaluasi dengan menggunakan *Confusion Matrix* untuk mengetahui *Akurasi*. Evaluasi model dalam penelitian ini dilakukan untuk menguji performansi dari sistem yang dibangun.

5  
**BAB IV**

## **HASIL PENELITIAN**

### **4.1 Hasil Pengumpulan Data**

Dataset yang digunakan dalam penelitian ini adalah dataset *public* dengan <sup>22</sup> nama *The MIT-CBCL face recognition database*, dataset yang di unduh dari <sup>36</sup> <http://cbcl.mit.edu/software-datasets/heisele/facerecognition-database.html>.

Dataset ini merupakan data yang pada umumnya digunakan dalam eksperiment untuk deteksi wajah. Dengan jumlah data direktori *training original* 59 image, dan subdirektori *training used in fpiv04 paper* 30 image.



**Gambar 4. 1** Gambar Beberapa Contoh Dataset

## 4.2 Hasil Pemodelan

58

### 4.2.1 Pengambilan Citra Wajah

Data yang digunakan pada penelitian ini adalah berupa data citra digital untuk pengenalan wajah. Pada verifikasi peserta Ujian Online data gambar menggunakan Data Publik. Citra wajah akan dijadikan sebagai data *training* yakni dua citra wajah untuk setiap pengenalan dengan percobaan 10 pengenalan dan dua citra wajah untuk data test. Untuk data *training* diambil satu citra dengan pose tegak kedepan dan dua citra dengan pose menghadap ke kiri dan ke kanan sekitar 15 derajat. Sehingga terdapat 30 data *training* dan 20 data *testing*.

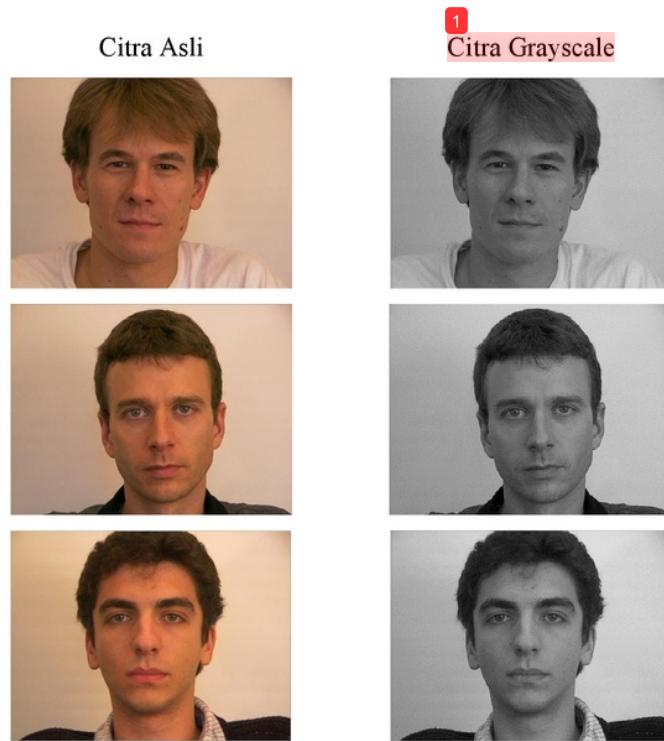
### 4.2.2 Pra-Pengolahan

23

Pra-pengolahan citra (image pre-processing), yaitu proses paling awal dalam pengolahan citra sebelum proses utama dilakukan. Pada tahap ini citra wajah yang sudah ada dikonversi agar diperoleh data citra wajah yang sesuai kebutuhan. Tahap ini berfungsi untuk menormalisasi citra wajah dari permasalahan luminasi yang terlalu gelap atau terlalu terang sehingga dapat meningkatkan performansi dari sistem pengenalan wajah. Pra-pengolahan dilakukan dengan 2 tahap, yaitu:

1. Pengubahan citra warna ke *grayscale*

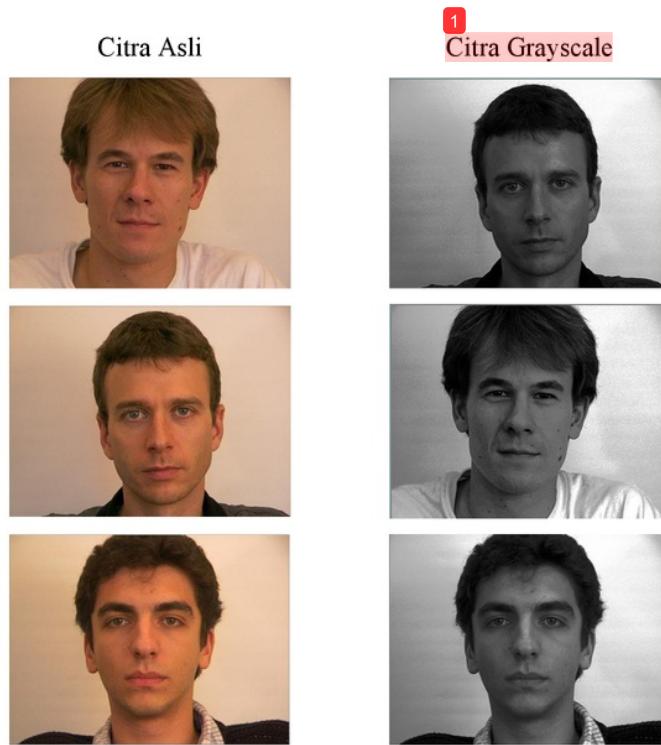
Pra-pengolahan proses awal yang merubah citra *training* atau citra *testing* yang awalnya citra dari RGB menjadi citra *grayscale*, perubahan ini dilakukan karena citra *grayscale* memiliki persamaan yang sederhana dan mampu mengurangi kebutuhan memory dimana nilai warna putih diwakili dengan angka 255 dan nilai warna hitam diwakili dengan angka 0.



**Gambar 4. 2** Hasil Konversi Citra Warna ke *Grayscale*

## 2. Normalisasi *citra*

Setelah citra asli dikonversi ke citra *grayscale* maka prapengolahan selanjutnya adalah normalisasi citra dengan histogram ekualisasi. Histogram ekualisasi adalah proses yang mengonversikan bagian nilai derajat keabuan pada sebuah citra sehingga menjadi sejenis. Maksut dari proses *histogram equalization* adalah untuk mencapai diseminasi histogram yang menyeluruh sehingga setiap derajat keabuan mendapat jumlah pixel yang relatif sama. Berikut gambar normalisasi citra abu-abu dengan histogram ekualisasi.



**Gambar 4.3** Hasil Konversi Citra Warna ke Histogram

#### 4.2.3 Ekstraksi Fitur

Setelah diperoleh hasil dari pra-pengolahan citra wajah, selanjutnya tahapan mengekstraksi ciri dari objek di dalam citra wajah baik citra latih ataupun yang ingin dikenali.

##### 1. Ekstraksi ciri menggunakan LBPH

8

Metode yang digunakan untuk mengekstraksi ciri yakni LBPH (*Local Binary Pattern Histogram*). Semua citra wajah direpresentasikan dalam bentuk matrix (*pixel*) untuk diproses. Perhitungan manual metode LBPH untuk menyederhanakan data asli, berikut ini adalah contoh proses konvolusi terhadap filter  $LBPH$  dengan size =  $49 \times 49$ ,  $f=1$ ,  $\theta=0$ . Konvolusi nilai-nilai filternya di balik  $180^0$ . Contoh, bila sebuah citra  $f(x,y)$  akan dikonvolusi dengan filter  $s(gp - gc)$ :

$f(x,y) =$ 

x,y	0	1	3	4	....	46	47	48
0	215	215	222	223	....	193	184	173
1	212	217	221	219	....	176	165	150
2	219	219	219	215	....	155	140	136
....	....	....	....	....	....	....	....	....
46	212	212	212	212	....	171	183	193
47	215	215	215	215	....	179	189	178
48	199	206	217	217	....	185	192	212

(Citra)

## 2. Korelasi sebuah citra

## a. Contoh korelasi 1

 $f(x,y) =$ 

x,y	0	1	3	4	....	46	47	48
0	215	215	222	223	....	193	184	173
1	212	217	221	219	....	176	165	150
2	219	219	219	215	....	155	140	136
....	....	....	....	....	....	....	....	....
46	212	212	212	212	....	171	183	193
47	215	215	215	215	....	179	189	178
48	199	206	217	217	....	185	192	212

Nilai pixel Grayscale

201	215	222
212	217	221
219	219	219

Threshold

0	0	1
0	1	1
1	1	1

Untuk perhitungan konvers binary ke decimal :

Binary : 00111110<sub>2</sub>

42

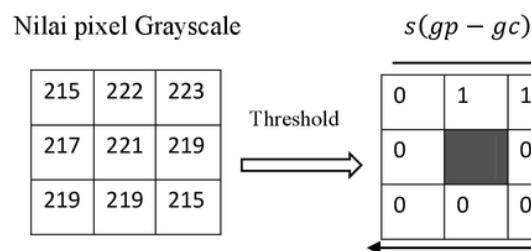
$$(0 \times 2) + (0 \times 2) + (1 \times 2) + (0 \times 2) = 62$$

217 diganti oleh 62, tempatkan pada matrik yang baru, hasilnya adalah :

x,y	0	1	3	4	....	46	47	48
0	201	215	222	223	....	193	184	173
1	212	62	221	219	....	176	165	150
2	219	219	219	215	....	155	140	136
....	....	....	....	....	....	....	....	....
46	212	212	212	212	....	171	183	193
47	215	215	215	215	....	179	189	178
48	199	206	217	217	....	185	192	212

- b. Contoh korelasi 2, <sup>18</sup> Geser citra  $f(x,y)$  ukuran 3x3 satu piksel ke kanan, kemudian hitung korelasinya dengan filter  $s(gp - gc)$ .

$f(x,y) =$	x,y	0	1	3	4	....	46	47	48
	0	215	215	222	223	....	193	184	173
	1	212	217	221	219	....	176	165	150
	2	219	219	219	215	....	155	140	136
	....	....	....	....	....	....	....	....	....
	46	212	212	212	212	....	171	183	193
	47	215	215	215	215	....	179	189	178
	48	199	206	217	217	....	185	192	212



Untuk perhitungan konvers binary ke decimal :

Binary : 011000000<sub>2</sub>

$$42 \\ (0x2) + (1x2) + (1x2) + (0x2) + (0x2) + (0x2) + (0x2) + (0x2) = 96$$

221 diganti oleh 96, tempatkann pada matrik yang baru, hasilnya adalah :

x,y	0	1	3	4	....	46	47	48
0	201	215	222	223	....	193	184	173
1	212	62	96	219	....	176	165	150
2	219	219	219	215	....	155	140	136
....	....	....	....	....	....	....	....	....
46	212	212	212	212	....	6	3	2
47	215	215	215	215	....	7	6	2
48	199	206	217	217	....	0	7	2

Setelah semua telah dikonvolusi selanjutnya mengkualisasi histogram.

- 7  
3. Histogram merupakan sebuah diagram yang membuktikan jumlah titik yang terdapat pada sebuah citra untuk setiap tingkat keabuan (Hestiningsih, 2009). Sumbu x (absis) pada histogram menunjukkan tingkat warna, sedangkan sumbu y (kordinat) menunjukkan frekuensi kemunculan titik.

Distribusi Kumulatif dapat dibuat menggunakan rumus:

$$k_o = \text{round} \left( \frac{c_i(2^k - 1)}{w.h} \right)$$

Dimana:

- <sup>21</sup>  
 $c_i$  = distribusi kumulatif dari nilai skala keabuan ke  $-I$  dari citra asli  
 Round = fungsi pembulatan ke bilangan yang terdekat  
 $k_o$  = nilai keabuan hasil histogram equalization  
 $w$  = lebar citra  
 $h$  = tinggi citra

Contoh perhitungan manual ekualisasi **histogram** :

1. Terdapat sebuah **citra** keabuan dengan nilai pixel 8x8 sebagai berikut:

**Tabel 4. 1** Citra Keabuan

x,y	0	1	2	3	4	5	6	7
0	201	215	222	223	219	212	206	206
1	212	62	96	227	195	223	159	6
2	219	55	99	231	255	255	60	8
3	221	239	207	207	159	63	118	88
4	219	215	155	29	31	7	103	243
5	219	231	243	16	0	5	207	231
6	223	7	243	241	248	200	143	143
7	225	129	225	255	252	240	144	1

2. Penyederhanaan Nilai pixel pada matrix citra, <sup>8</sup> dihitung frekuensi dan distribusi kumulatif dari nilai skala keabuan.

**Tabel 4. 2** Penyederhanaan Nilai Pixel

Skala Keabuan	Frek	Distribusi Komuatif	Skala Keabuan	Frek	Distribusi Komuatif
0	1	1	159	2	25 + 2 = 27
1	1	1 + 1 = 2	195	1	27 + 1 = 28
5	1	2 + 1 = 3	200	1	28 + 1 = 29
6	1	3 + 1 = 4	201	1	29 + 1 = 30
7	2	4 + 2 = 6	206	2	30 + 1 = 31
8	1	6 + 1 = 7	207	3	31 + 1 = 32
16	1	7 + 2 = 9	212	2	32 + 2 = 34
29	1	9 + 1 = 10	215	2	34 + 1 = 35

31	1	$10 + 1 = 11$	219	4	$35 + 1 = 36$
55	1	$11 + 1 = 12$	221	1	$36 + 3 = 39$
60	1	$12 + 1 = 13$	222	1	$39 + 1 = 40$
62	1	$13 + 1 = 14$	223	3	$40 + 1 = 41$
63	1	$14 + 1 = 15$	225	2	$41 + 1 = 42$
88	1	$15 + 1 = 16$	227	1	$42 + 1 = 43$
96	1	$16 + 1 = 17$	231	3	$43 + 1 = 44$
99	1	$17 + 1 = 18$	239	1	$44 + 1 = 45$
103	1	$18 + 1 = 19$	240	1	$45 + 1 = 46$
118	1	$19 + 1 = 20$	241	1	$45 + 2 = 47$
129	1	$20 + 1 = 21$	243	3	$47 + 1 = 48$
143	2	$21 + 2 = 23$	248	1	$48 + 1 = 49$
144	1	$23 + 1 = 24$	252	1	$49 + 1 = 50$
155	1	$24 + 1 = 25$	255	3	$50 + 1 = 51$

7. 3. Menghitung nilai keabuan dari hasil perhitungan distribusi kumulatif

Tabel 4. 3 Hasil Distribusi Kumulatif

Keabuan Awal	Frk	Keabuan Hasil
0	1	4
1	1	8
5	1	12
6	1	17
7	2	24
8	1	28
16	1	36
29	1	40
31	1	44
55	1	48

Keabuan Awal	Frek	Keabuan Hasil
159	2	107
195	1	111
200	1	155
201	1	119
206	2	123
207	3	127
212	2	135
215	2	139
219	4	143
221	1	155

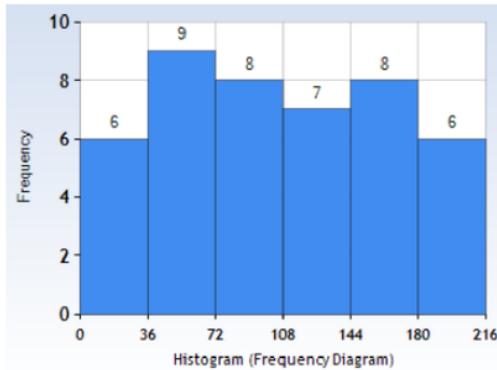
60	1	52		222	1	159	
62	1	56		223	3	163	
63	1	60		225	2	167	
88	1	64		227	1	171	
96	1	68		231	3	175	
99	1	72		239	1	179	
103	1	76		240	1	183	
118	1	80		241	1	187	
129	1	84		243	3	191	
143	2	92		248	1	195	
144	1	96		252	1	199	
155	1	100		255	3	203	

4. Hasil matrix citra yang telah diproses.

**Tabel 4.4 Hasil Matrix Citra**

x,y	0	1	2	3	4	5	6	7
0	119	139	159	163	143	135	123	123
1	135	56	86	171	111	163	107	17
2	143	48	72	175	203	203	52	28
3	155	179	127	127	107	60	80	64
4	143	139	100	40	44	24	76	191
5	143	175	191	36	4	12	127	175
6	163	24	191	187	195	155	92	92
7	167	64	167	203	199	183	96	8

## 5. Histogram



Gambar 4. 4 Diagram histogram

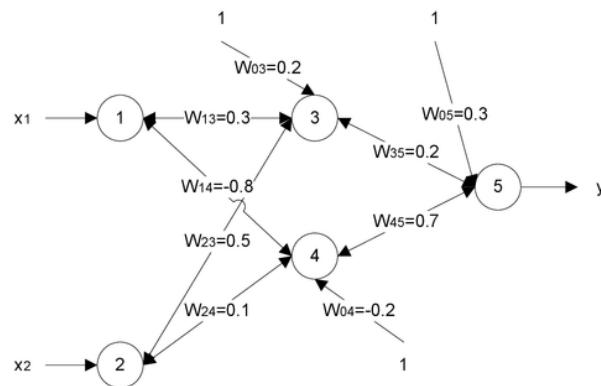
Frequency Table	
class	count
0 -35	6
36 -71	9
72 -107	8
108 -143	7
244 -179	8
280 -215	6

Histogram	
Mean	105.61364
Standard (S)	60.83382
Skewness	-0.01196
Kurtosis	-1.28267
29	
Lowest Score	4
Highest Score	203
Distribusi Range	199
Total Number Of Scores	44
Number Of Distinct Scores	43
Lowest Class Value	0
Highest Class Value	215
Number Of Classes	6
Class Range	36

Nilai Mean yang dihasilkan dari hasil ekualisasi histogram adalah =  
105.61364

#### 4.2.4 Klasifikasi

Penelitian ini menggunakan algoritma *Artificial Neural Network* untuk klasifikasi. Pada proses *training* ini berguna untuk mencari bobot terbaik untuk setiap neuron. Arsitektur *Artificial Neural Network* sebagai berikut:



Gambar 4.5 Arsitektur Artificial Neural Network

Tabel 4.5 Dataset

No.	X1	X7	y
1.	143.1591	105.6136	0

Kondisi bobot saat inisialisasi seperti ditunjukkan pada gambar di atas dengan jumlah neuron pada layer tersembunyi = 2.

Laju pembelajaran ( $\eta$ ) = 0.1;

Fungsi aktivasi yang digunakan adalah *Sigmoid Biner*;

Target *error* = 0.0001 dengan kriteria SSE;

Momentum yang digunakan = 0.95;

Maksimum jumlah iterasi pelatihan = 1,000 kali.

Maka:

- Nilai pada neuron di hidden layer 3 dan 4:

$$v_j(p) = \sum_{i=1}^n x_i(p) w_{ij}(p)$$

$$y_j(p) = \frac{1}{1 + e^{-v_j(p)}}$$

(P) adalah iterasi ke-i.

$$v_3(1) = x1w_{13} + x2w_{23} + 1w_{03} = 143.1591 * 0.3 + 105.6136 *$$

$$0.5 + 1 * 0.2 = 95.95453$$

$$y_3(1) = \frac{1}{1+e^{-95.95453}} = \frac{1}{1+2.71828^{-95.95453}} = 2.125578$$

$$v_4(1) = x1w_{14} + x2w_{24} + 1w_{04} = 143.1591 * (-0.8) +$$

$$105.6136 * 0.1 + 1 * (-0.2) = -104.16592$$

$$y_4(1) = \frac{1}{1+e^{-104.16592}} = 5.771859$$

- Nilai pada neuron di output layer:

$$v_k(p) = \sum_{j=1}^m x_j(p) w_{jk}(p)$$

$$y_k(p) = \frac{1}{1 + e^{-v_k(p)}}$$

$$v_5(1) = Y3(1)w_{35} + Y4(1)w_{45} + 1w_{05} = 2.125578 * 0.2 +$$

$$5.771859 * 0.7 + 1 * 0.3 = 4.7654163$$

$$y_5(1) = \frac{1}{1+e^{-4.7654163}} = 0.0085193409$$

- Hitung gradient error untuk neuron pada output layer:

$$e_k(p) = y_{dk}(p) - y_k(p)$$

$$\delta_k(p) = y_k(p) * [1 - y_k(p)] * e_k(P)$$

Untuk data pertama, target/class nilai yang diharapkan adalah  $y_d = \mathbf{0}$ , sedangkan keluaran yang didapatkan  $y_5(1) = \mathbf{0.0085193409}$ . Maka hitung *error* pada iterasi pertama untuk data pertama pada neuron 5 di output layer:

$$e_5^1(1) = y_d - y_5(1) = 0 - 0.0085193409 = -0.0085193409$$

$$\delta_5(1) = y_5(1) * (1 - y_5(1)) * e_5^1(1) = 0.0085193409 * (1 - 0.0085193409) * (-0.0085193409) = -0.0000719608$$

4. Hitung koreksi bobot untuk output layer,  $\Delta w_{35}$ ,  $\Delta w_{45}$ , dan  $\Delta w_{05}$ :

$$\Delta w_{jk}(p) = \eta \cdot y_j(p) * \delta_k(p)$$

$$\Delta w_{35} = \eta * y_3(1) * \delta_5(1) = 0.1 * 2.13 * (-0.0000719608) = -0.0000153277$$

$$\Delta w_{45} = \eta * y_4(1) * \delta_5(1) = 0.1 * 5.77 * (-0.0000719608) = -0.0000415214$$

$$\Delta w_{05} = \eta * 1 * \delta_5(1) = 0.1 * 1 * (-0.0000719608) = -0.0000071961$$

5. Hitung *gradient error* pada hidden layer  $\delta_3(1)$  dan  $\delta_4(1)$ :

$$\delta_j(p) = y_j(p) * [(1 - y_j(p))] + \sum_{k=1}^1 \delta_k(p) * w_{jk}(p)$$

$$\begin{aligned} \delta_3(1) &= y_3(1) * (1 - y_3(1)) * \sum_{k=1}^1 \delta_k(1) \cdot w_{3k}(1) = y_3(1) * (1 - y_3(1)) * \delta_5(1) \cdot w_{35}(1) \\ &= 2.13 * (1 - 2.13) * (-0.0000719608) * 0.2 = 0.0000346405 \end{aligned}$$

$$\delta_4(1) = y_4(1) * (1 - y_4(1)) * \sum_{k=1}^1 \delta_k(1) \cdot w_{4k}(1) = y_4(1) * (1 - y_4(1)) * \delta_5(1) \cdot w_{45}(1)$$

$$= 5.77 * (1 - 5.77) * (-0.0000719608) * 0.7 = 0.0013863989$$

6. Hitung koreksi bobot untuk hidden layer  $\Delta w_{13}, \Delta w_{23}, \Delta w_{03}, \Delta w_{14}, \Delta w_{24},$  dan  $\Delta w_{04}$

$$\Delta w_{ij}(p) = \eta * x_i(p) * \delta_j(p)$$

$$\Delta w_{13} = \eta * x_1 * \delta_3(1) = 0.1 * 2 * (-0.0000356) = -0.00000712$$

$$\Delta w_{23} = \eta * x_2 * \delta_3(1) = 0.1 * 3 * (-0.0000356) = -0.00001068$$

$$\Delta w_{03} = \eta * x_1 * \delta_3(1) = 0.1 * 1 * (-0.0000356) = -0.00000356$$

$$\Delta w_{14} = \eta * x_1 * \delta_4(1) = 0.1 * 2 * (-0.0013864) = -0.00027728$$

$$\Delta w_{24} = \eta * x_2 * \delta_4(1) = 0.1 * 3 * (-0.0013864) = -0.00041592$$

$$\Delta w_{04} = \eta * x_1 * \delta_4(1) = 0.1 * 1 * (-0.0013864) = -0.00013864$$

7. Perbaharui bobot untuk neuron pada hidden layer  $w_{35}, w_{45}, w_{05}$ :

$$w_{ij}(p + 1) = w_{ij}(p) + \Delta w_{ij}(p)$$

$$w_{35}(2) = w_{35}(1) + \Delta w_{35} = 0.2 + (-0.0000153277) =$$

$$0.1999846723$$

$$w_{45}(2) = w_{45}(1) + \Delta w_{45} = 0.7 + (-0.0000415214) =$$

$$0.6999584786$$

$$w_{05}(2) = w_{05}(1) + \Delta w_{05} = 0.3 + (-0.0000071961) =$$

$$0.2999928039$$

8. Perbaharui bobot pada layer tersembunyi,  $w_{13}, w_{23}, w_{03}, w_{14}, w_{24}, w_{04}$ :

$$w_{13}(2) = w_{13}(1) + \Delta w_{13} = 0.3 + (-0.00000712) = 0.29999288$$

$$w_{23}(2) = w_{23}(1) + \Delta w_{23} = 0.5 + (-0.00001068) = 0.49998932$$

$$w_{03}(2) = w_{03}(1) + \Delta w_{03} = 0.2 + (-0.00000356) = 0.19999644$$

$$w_{14}(2) = w_{14}(1) + \Delta w_{14} = 0.8 + (-0.00027728) = 0.79972272$$

$$w_{24}(2) = w_{24}(1) + \Delta w_{24} = 0.1 + (-0.00041592) = 0.09958408$$

$$w_{04}(2) = w_{04}(1) + \Delta w_{04} = (-0.2) + (-0.00013864) = -0.20013864$$

9. Naikkan 1 langkah iterasi ( $p$ ), kembali ke langkah 2 dan ulangi proses tersebut sampai *kriteria error* tercapai.
10. Dengan demikian, didapatkan bobot akhir pada iterasi pertama untuk data pertama[2, 3]:

$$w_{13} = 0.29999288 ;$$

$$w_{23} = 0.49998932 ;$$

$$w_{03} = 0.19999644 ;$$

$$w_{14} = 0.79972272 ;$$

$$w_{24} = 0.09958408 ;$$

$$w_{04} = -0.20013864 ;$$

$$w_{35} = 0.1999846723 ;$$

$$w_{45} = 0.6999584786 ;$$

$$w_{05} = 0.2999928039$$

*Error* yang didapatkan adalah  $= -0.0085193409$ . Selanjutnya proses di atas diulangi untuk data selanjutnya sehingga iterasi pertama selesai dilakukan.

Berikut merupakan bobot terbaik yang dihasilkan dari proses *training artificial neural network* :

**Tabel 4. 6** Bobot hasil Train ANN

NO	Hidden Layer	Bobot	Total
1	Input – HL 1	-0.825624, 0.137009	-0.688615
2	HL 1 - HL 2	-0.621031, 15.2097	14.588669
3	52 HL 2 – HL 3	0.36926, -0.735098	-0.365838
4	HL 3 – HL 4	-0.15887, 0.359355	0.200485
5	HL 4 – Output	-0.158642, -16.0635	-16.222142

Testing menggunakan algoritma ANN:

1. inputan menerima sinyal *input* dan menyebarkan sinyal tersebut ke *hidden layer* :

$$\begin{aligned}
 v_1 &= x_1 \cdot w_{11} + x_2 \cdot w_{21} + 1 \cdot w_{01} \\
 &= 143.1591 * (-0.825624) + 105.6136 * 0.137009 + 1 * 0.1 \\
 &= -103.6255751
 \end{aligned}$$

$$y_1(1) = \frac{1}{1 + e^{-103.6255751}} = \frac{1}{1 + 2.71828^{-103.6255751}} = 9.907966967$$

- 1  
2. Setiap *hidden unit* akan menjumlahkan sinyal-sinyal *input* yang sudah berbobot, termasuk biasnya

$$v_{33}(1) = y_1(1) \cdot w_{1.33} + 1 \cdot w_{0.33} \\ = (5.771859 * (-0.621031)) + (1 * 0.1) = -3.4845033666$$

$$y_{33}(1) = \frac{1}{1+e^{-3.4845033666}} = 0,0306689859$$

- 1  
3. Setiap *hidden unit* akan menjumlahkan sinyal-sinyal yang sudah berbobot ke *output*, termasuk biasnya

$$v = y_1 \cdot w_{34.1} + 1 \cdot w_{0.34} \\ = (5.771859 * (-0.158642)) + (1 * 0.1) = -0,8156592555$$

$$y = \frac{1}{1+e^{-0,8156592555}} = 0,44$$

#### 4.2.5 Evaluasi Model

Tabel 4.7 Hasil Pengenalan

No	Id Sebenarnya	Id yang Dikenali	Ket	No	Id Sebenarnya	Id yang Dikenali	Ket
1	ID19011001	ID19011009	S	11	ID19011006	ID19011009	S
2	ID19011001	ID19011009	S	12	ID19011006	ID19011002	S
3	ID19011002	ID19011002	B	13	ID19011007	ID19011007	B
4	ID19011002	ID19011002	B	14	ID19011007	ID19011009	S
5	ID19011003	ID19011003	B	15	ID19011008	ID19011008	B
6	ID19011003	ID19011005	S	16	ID19011008	ID19011008	B
7	ID19011004	ID19011002	S	17	ID19011009	ID19011001	S
8	ID19011004	ID19011002	S	18	ID19011009	ID19011001	S
9	ID19011005	ID19011003	S	19	ID19011010	ID19011001	B
10	ID19011005	ID19011003	S	20	ID19011010	ID19011001	B

Keterangan:

S = Salah

B = Benar

Dari hasil pengujian data testing dihasilkan klasifikasi pengenalan, dengan percobaan 20 citra testing dari 10 orang berbeda. Untuk mencari akurasi dari pengenalan menggunakan *confusion matrix*. Berikut ini merupakan table *confusion matrix*:

		Prediksi	
		Id Sebenarnya	Id Dikenali
Aktual	Id Sebenarnya	TP	FP
	Id Dikenali	FN	TN

64  
Keterangan:

TP = True Positif

FN = False Negatif

FP = False Negative

TN = True Negatif

11

Secara umum precision, recall dan accuracy dapat dirumuskan sebagai berikut:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Accuracy = \frac{TP + TN}{TP + FN + FP + FN}$$

Untuk mencari akurasi dari pengenalan:

		Prediksi	
		Id Sebenarnya	Id Dikenali
Aktual	Id Sebenarnya	6	2
	Id Dikenali	2	10

- 17  
1. Menghitung rasio prediksi benar positif dibandingkan dengan keseluruhan hasil prediksi positif

$$precision = \frac{6}{6 + 2} = 0.75 = 75\%$$

- 17  
2. Menghitung rasio prediksi benar positif dibandingkan dengan keseluruhan data hasil benar positif

$$recall = \frac{6}{6 + 2} = 0.75 = 75\%$$

- 17  
3. Menghitung rasio prediksi benar (positif dan negative) dengan keseluruhan hasil prediksi positif

$$akurasi = \frac{6 + 10}{6 + 2 + 2 + 10} = \frac{16}{20} = 0.8 = 80\%$$

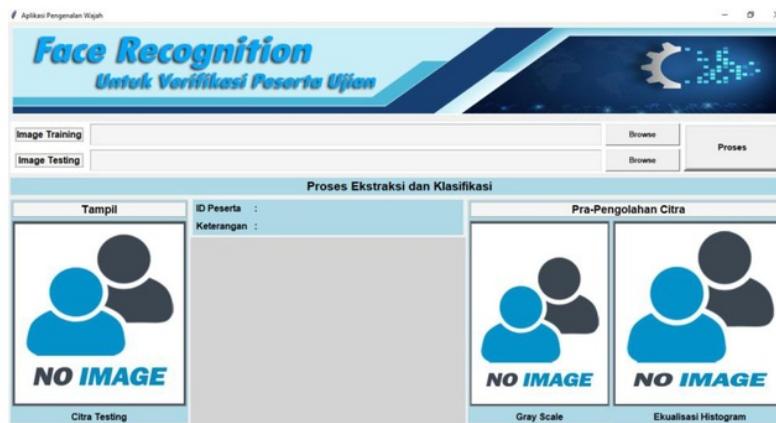
## BAB V

### PEMBAHASAN

#### 5.1 Pembahasan Sistem

##### 1. Tampilan menu utama

Form menu utama merupakan form utama yang muncul saat program <sup>11</sup> dijalankan. Pada form menu utama menampilkan menu – menu yang ada pada sistem yaitu menu load training, menu load testing, dan proses.



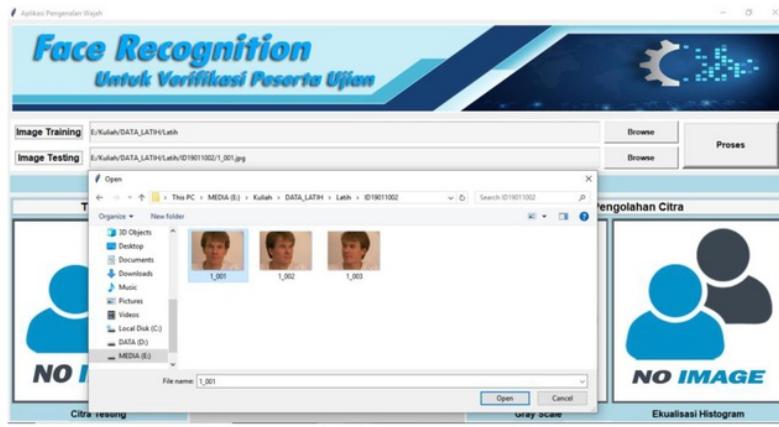
Gambar 5. 1 Tampilan Awal Aplikasi

##### 2. Tampilan Proses Load Data Training



Gambar 5. 2 Tampilan Proses Load Data Training

### 3. Tampilan Proses Load Data Testing



45  
Gambar 5. 3 Tampilan Proses Load Data Testing

### 4. Tampilan Hasil Pengenalan



45  
Gambar 5. 4 Tampilan Hasil Pengenalan

## 5.2 Pembahasan Model

Penelitian menggunakan *Local Binary Pattern Histogram* sebagai fitur ekstraksi untuk citra wajah dan *Artificial Neural Network* sebagai algoritma klasifikasi. Tujuan dari penelitian ini untuk mengetahui kinerja model pengenalan wajah untuk verifikasi peserta ujian online menggunakan metode *Artificial Neural Network* dan *Local Binary Pattern Histogram* yang efektif.

Data yang digunakan pada penelitian ini adalah berupa data citra digital untuk pengenalan wajah. Pada verifikasi peserta Ujian Online data gambar menggunakan Data Publik. Citra wajah akan dijadikan sebagai data *training* yakni dua citra wajah untuk setiap pengenalan dengan percobaan 10 pengenalan dan dua citra wajah untuk data test. Untuk data *training* diambil satu citra dengan pose tegak kedepan dan dua citra dengan pose mengahadap ke kiri dan ke kanan sekitar 15 derajat. Sehingga terdapat 30 data *training* dan 20 data *testing*.

Pra-pengolahan citra yaitu proses paling awal dalam pengolahan citra sebelum proses utama dilakukan. Pada tahap ini citra wajah yang sudah ada dikonversi agar diperoleh data citra wajah yang sesuai kebutuhan. Pra-pengolahan dilakukan dengan 2 tahap, yaitu proses pengubahan citra RGB menjadi citra bentuk *grayscale*, hasil dari konversi citra menjadi abu-abu terang. Setelah citra asli dikonversi ke citra *grayscale* maka pra-pengolahan selanjutnya adalah normalisasi citra dengan histogram ekualisasi. Hasil dari konversi histogram ekualisasi menjadi abu-abu gelap.

Deteksi wajah (*face detection*) merupakan salah satu tahap (*preprocessing*) yang sangat penting sebelum dilakukan proses pengenalan wajah (*face recognition*). Pengenalan wajah merupakan proses untuk membandingkan citra wajah *testing* dengan data *training*.

Ekstraksi fitur berfungsi sebagai pendekripsi ciri dari suatu citra. Ciri yang dapat digunakan untuk membedakan objek satu dengan objek yang lainnya, diantaranya ciri bentuk, ciri ukuran, ciri geometri, ciri tekstur dan ciri warna. Ekstraksi fitur menggunakan metode *Local binary pattern* membutuhkan waktu sedikit lama untuk memproses sebuah citra, karena

semua citra wajah direprestasikan dalam bentuk matrix (*pixel*) untuk diproses.  
Untuk menentukan hasil dari konvolusi, citra asli akan dikorelasi menjadi nilai biner maka akan menghasilkan deret biner yang kemudian jika diubah menjadi nilai decimal akan menghasilkan nilai baru untuk nilai tengah pixel.  
Selanjutnya proses ekualisasi histogram untuk mendapatkan nilai mean , dalam penyerahan nilai *pixel* pada matrix citra dengan cara menghitung frekuensi dan distribusi kumulatif dari nilai skala keabuan. nilai mean yang dihasilkan akan digunakan untuk mencari bobot dalam klasifikasi.

Evaluasi model dalam penelitian ini dilakukan untuk menguji performa menggunakan *confusion matrix*. ada tiga cara untuk melakukan evaluasi, yaitu: *Precision* melakukan perbandingan jumlah dari data relevan yang diperoleh dari sistem dengan jumlah dari seluruh data yang diambil oleh sistem baik yang relevan maupun tidak. *Recall* akan membandingkan jumlah data yang didapatkan dari sistem dengan jumlah seluruh data yang ada dalam koleksi data. *Accuracy* merupakan pengujian berdasarkan tingkat kedekatan antara nilai prediksi dengan nilai aktual. Dengan mengetahui jumlah data yang diklasifikasi secara benar maka dapat diketahui hasil akurasi prediksi. Pada percobaan pengenalan wajah menggunakan *Local Binary Pattern Histogram* sebagai fitur ekstraksi untuk citra wajah dan *Artificial Neural Network* untuk klasifikasi, diperoleh akurasi sebesar 80%.

11  
**BAB VI**

**PENUTUP**

**6.1 Kesimpulan**

Berdasarkan hasil penelitian yang telah dilakukan **dan pembahasan** yang telah diuraikan sebelumnya maka dapat ditarik kesimpulan bahwa :

1. Dari hasil yang diperoleh dalam pengenalan wajah menggunakan metode <sup>15</sup> *Artificial Neural Network* dengan fitur ekstraksi *Local Binary Pattern Histogram* untuk mengenali wajah mendapatkan hasil akurasi 80%. Untuk pengenalan dari 30 data training dan 20 data testing dengan satu citra dengan pose tegak kedepan dan dua citra dengan pose mengahadap ke kiri dan ke kanan sekitar 15 derajat. Hal ini menunjukan bahwa untuk mengenali wajah berkinerja dengan baik, dan sudah mampu mengklasifikasikan peserta ujian.

**6.2 Saran**

Setelah melakukan penelitian dan pembuatan aplikasi pengenalan wajah untuk verifikasi peserta ujian, ada beberapa saran yang perlu di perhatikan untuk mencapai tujuan yang di harapkan, yaitu :

1. Sistem pengenalan wajah ini dapat dikembangkan dengan model pengenalan secara *realtime*.
2. Perlu diadakan penelitian lebih lanjut, dengan penggunaan metode lainnya terutama pada metode ekstraksi ciri citra wajah sebagai perbandingan untuk mendapatkan hasil terbaik.

## DAFTAR PUSTAKA

- [1] D. Kurniawan, R. Andrian, and L. Ertina, “Pengembangan Sistem Ujian Online Sekolah,” *J. Komputasi*, vol. 2, no. 2, pp. 36–40, 2014.  
8
- [2] A. Budi, S. Suma’inna, and H. Maulana, “Pengenalan Citra Wajah Sebagai Identifier Menggunakan Metode Principal Component Analysis (PCA),” *J. Tek. Inform.*, vol. 9, no. 2, pp. 166–175, 2016.
- [3] I. Piarsa and R. Hisamuddin, “Sistem Verifikasi Online Menggunakan Biometrika Wajah,” *Teknol. Elektro*, vol. 9, no. 1, pp. 88–94, 2010.
- [4] M. Tandililing, “Aplikasi Pengenalan Wajah Untuk Validasi Peserta Ujian Online Menggunakan Metode Haar Cascade dan Eigen Face Vektor,” *SIBerPro*, vol. 1, no. 1, pp. 51–57, 2016.  
40
- [5] H. Simaremare and A. Kurniawan, “Perbandingan Akurasi Pengenalan Wajah Menggunakan Metode LBPH dan Eigenface dalam Mengenali Tiga Wajah Sekaligus secara Real-Time,” *J. Sains, Teknol. dan Ind.*, vol. 14, no. 1, pp. 66–71, 2016.  
50
- [6] S. Kusmaryanto, “Jaringan Saraf Tiruan Backpropagation untuk Pengenalan Wajah Metode Ekstraksi Fitur Berbasis Histogram,” *J. EECCIS*, vol. 8, no. 2, pp. 193–198, 2014.  
33
- [7] N. W. S. Saraswati and D. M. D. U. Putra, “Sistem Ujian Online Berbasis Website,” *S@Cies*, vol. 6, no. 1, pp. 21–29, 2015.
- [8] D. A. Wahyudi and I. H. Kartowisastro, “Menghitung Kecepatan Menggunakan Computer Vision,” *J. Teknik Komput.*, vol. 19, no. 2, pp. 89–101, 2011.
- [9] R. Wiryadinata, R. Sagita, S. Wardoyo, and P. Priswanto, “Pengenalan Wajah Pada Sistem Presensi Menggunakan Metode Dynamic Times Wrapping, Principal Component Analysis Dan Gabor Wavelet,” *Din. REKAYASA*, vol. 12, no. 1, pp. 1–8, 2016.  
37
- [10] F. Fandiansyah, J. Y. Sari, and I. P. Ningrum, “Pengenalan Wajah Menggunakan Metode Linear Discriminant Analysis dan k Nearest Neighbor,” *J. Inform.*, vol. 11, no. 2, pp. 48–59, 2017.
- [11] A. B. W.P and D. L. Pandiangan, “Pengenalan Pola Wajah untuk Simulasi Presensi Mahasiswa,” *SMARTICS J.*, vol. 2, no. 1, pp. 15–20, 2016.  
63
- [12] I. Saputra, “Manusia Melalui Webcam Dengan Menerapkan,” *J. Ilm. INFOTEK*, vol. 1, no. 1, pp. 55–64, 2016.
- [13] A. Fauzan, L. Novamizanti, and Y. N. Fuadah, “Perancangan Sistem Deteksi Wajah Untuk Presensi Kehadiran Menggunakan Metode Lbph ( Local

Binary Pattern Histogram ) Berbasis Android,” *e-Proceeding Eng.*, vol. 5, no. 3, pp. 5403–5413, 2018.

- [14] M. Muthia, K. Munadi, and F. Arnia, “Studi Kuantisasi Bin Terhadap Metode Local Binary Pattern dan Local Binary Pattern Variance pada Deteksi Citra Asap,” *Karya Ilm. Mhs. Tek. Elektro*, vol. 2, no. 3, pp. 31–39, 2017.
- [15] S. A. Wijaya, “Perbandingan Metode Pengenalan Wajah Secara Real-Time Pada Perangkat Bergerak Berbasis Android,” pp. 1–6.
- [16] A. Salim and W. S. Pambudi, “Implementasi Metode Hybrid Artificial Neural Network ( Ann ) – Pid Untuk P<sub>68</sub>aikan Proses Berjalan Pada Prototype Robot Material Handling,” *J. Ilm. Mikrotek*, vol. 1, no. 3, pp. 155–164, 2015.
- [17] A. S. Ritonga and S. Atmojo, “Pengembangan Model Jaringan Syaraf Tiruan untuk Memprediksi Jumlah Mahasiswa Baru di PTS Surabaya (Studi Kasus Universitas Wijaya Putra),” *J. Ilm. Teknol. Inf. Asia*, vol. 12, no. 1, pp. 15–24, 2018.
- [18] U. Hamida, “Penggunaan Artificial Neural Network (Ann) Untuk Memodelkan Kebutuhan Energi Untuk Transportasi,” *J. Teknol. dan Manaj.*, vol. 12, no. 2, pp. 57–65, 2014.
- [19] S. T. E. Mulyanto, and V. Suhartono, *Kecerdasan Buatan*. Yogyakarta: Andi Offset, 2011.

# Pengenalan Wajah Menggunakan Metode Artificial Neural Network Dengan Fitur Ekstraksi Local Binary Pattern

ORIGINALITY REPORT



PRIMARY SOURCES

- |   |   |    |
|---|---|----|
| 1 | Submitted to Universitas Brawijaya<br>Student Paper       | 2% |
| 2 | eprints.umm.ac.id<br>Internet Source                      | 2% |
| 3 | media.neliti.com<br>Internet Source                       | 2% |
| 4 | de.scribd.com<br>Internet Source                          | 2% |
| 5 | Submitted to Universitas Dian Nuswantoro<br>Student Paper | 2% |
| 6 | ejournal.unud.ac.id<br>Internet Source                    | 1% |
| 7 | docplayer.info<br>Internet Source                         | 1% |
| 8 | Submitted to Sriwijaya University<br>Student Paper        | 1% |
| 9 | jurnal.stmikasia.ac.id                                    |    |

10 ejournal.uin-suska.ac.id  
Internet Source

1 %

11 www.scribd.com  
Internet Source

1 %

12 heregeralds.blogspot.com  
Internet Source

&lt;1 %

13 text-id.123dok.com  
Internet Source

&lt;1 %

14 pasca.unhas.ac.id  
Internet Source

&lt;1 %

15 kinetik.umm.ac.id  
Internet Source

&lt;1 %

16 www.jurnal.unsyiah.ac.id  
Internet Source

&lt;1 %

17 Submitted to Forum Komunikasi Perpustakaan  
Perguruan Tinggi Kristen Indonesia (FKPPTKI)  
Student Paper

&lt;1 %

18 id.scribd.com  
Internet Source

&lt;1 %

19 Submitted to Politeknik Negeri Bandung  
Student Paper

&lt;1 %

goindoti.blogspot.com

20	Internet Source	<1 %
21	<a href="http://maghfirahmaulani82.blogspot.com">maghfirahmaulani82.blogspot.com</a> Internet Source	<1 %
22	Dharavath, Krishna, Rabul H Laskar, and Fazal A Talukdar. "Qualitative study on 3D face databases: A review", 2013 Annual IEEE India Conference (INDICON), 2013. Publication	<1 %
23	<a href="http://diansulistyo.wordpress.com">diansulistyo.wordpress.com</a> Internet Source	<1 %
24	Submitted to University of New England Student Paper	<1 %
25	<a href="http://www.sigtkus.lecture.ub.ac.id">www.sigtkus.lecture.ub.ac.id</a> Internet Source	<1 %
26	<a href="http://ejurnal.stiki-indonesia.ac.id">ejurnal.stiki-indonesia.ac.id</a> Internet Source	<1 %
27	<a href="http://informatikaunindra.org">informatikaunindra.org</a> Internet Source	<1 %
28	<a href="http://pt.scribd.com">pt.scribd.com</a> Internet Source	<1 %
29	Submitted to Aston University Student Paper	<1 %
30	Submitted to Universitas Islam Indonesia	

---

31	Submitted to Udayana University Student Paper	<1 %
32	pemrogramanmatlab.com Internet Source	<1 %
33	Submitted to Universitas Atma Jaya Yogyakarta Student Paper	<1 %
34	es.scribd.com Internet Source	<1 %
35	panggihpanges2.blogspot.com Internet Source	<1 %
36	journals.sagepub.com Internet Source	<1 %
37	jurnal.polgan.ac.id Internet Source	<1 %
38	id.123dok.com Internet Source	<1 %
39	Submitted to Politeknik Negeri Jember Student Paper	<1 %
40	www.matec-conferences.org Internet Source	<1 %
41	Submitted to Universitas Muhammadiyah	<1 %

---

42	Submitted to Roehampton University Student Paper	<1 %
43	ejournals.umn.ac.id Internet Source	<1 %
44	Submitted to Universitas Putera Batam Student Paper	<1 %
45	Submitted to Universitas International Batam Student Paper	<1 %
46	Submitted to UIN Sultan Syarif Kasim Riau Student Paper	<1 %
47	ejurnal.unmerpas.ac.id Internet Source	<1 %
48	www.neliti.com Internet Source	<1 %
49	Submitted to UIN Sunan Gunung Djati Bandung Student Paper	<1 %
50	ojs.uho.ac.id Internet Source	<1 %
51	www.fikom-unisan.ac.id Internet Source	<1 %
52	H TRUSSELL. "Color and Multispectral Image Representation and Display", Handbook of	<1 %

# Image and Video Processing, 2005

Publication

---

53	<a href="http://www.tandfonline.com">www.tandfonline.com</a>	<1 %
54	Submitted to Universitas Gunadarma	<1 %
55	Ni Wayan Sumartini Saraswati, Desak Made Dwi Utami Putra. "Sistem Ujian Online Berbasis Website", S@CIES, 2015	<1 %
56	Submitted to Universitas Muria Kudus	<1 %
57	<a href="http://core.ac.uk">core.ac.uk</a>	<1 %
58	<a href="http://lib.unnes.ac.id">lib.unnes.ac.id</a>	<1 %
59	Submitted to Syiah Kuala University	<1 %
60	<a href="http://umar.student.unidar.ac.id">umar.student.unidar.ac.id</a>	<1 %
61	<a href="http://repository.maranatha.edu">repository.maranatha.edu</a>	<1 %
62	<a href="http://koleksidatajurnal.blogspot.com">koleksidatajurnal.blogspot.com</a>	<1 %

---

63	<a href="http://ejurnal.stmik-budidarma.ac.id">ejurnal.stmik-budidarma.ac.id</a> Internet Source	<1 %
64	<a href="http://eprints.mdp.ac.id">eprints.mdp.ac.id</a> Internet Source	<1 %
65	<a href="http://tenggarakahskom.blogspot.com">tenggarakahskom.blogspot.com</a> Internet Source	<1 %
66	<a href="http://eprints.akakom.ac.id">eprints.akakom.ac.id</a> Internet Source	<1 %
67	<a href="http://lehamaratus.student.umm.ac.id">lehamaratus.student.umm.ac.id</a> Internet Source	<1 %
68	Riza Agung Firmansyah, Wahyu S. Pambudi, Titiek Suheta, Efrita Arfah Zuliari, Syahri Muharom, M. Bayu Syarif Hidayatullah. "Implementation of Artificial Neural Networks for Localization System on Rescue Robot", 2018 Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS), 2018 Publication	<1 %
69	<a href="http://jurnal.ensiklopediaku.org">jurnal.ensiklopediaku.org</a> Internet Source	<1 %
70	<a href="http://e-journal.uajy.ac.id">e-journal.uajy.ac.id</a> Internet Source	<1 %
71	Submitted to University of Leeds Student Paper	<1 %

72	Submitted to President University Student Paper	<1 %
73	fr.slideshare.net Internet Source	<1 %
74	Submitted to STMIK STIKOM Bali Student Paper	<1 %
75	Submitted to Melbourne Institute of Technology Student Paper	<1 %
76	Submitted to iGroup Student Paper	<1 %
77	Submitted to Lambung Mangkurat University Student Paper	<1 %
78	Winda Usman, Irfan Sudahri Damanik, Jaya Tata Hardinata. "Jaringan Syaraf Tiruan dengan Metode Learning Vector Quantization (LVQ) dalam Menentukan Klasifikasi Jenis Tilang Berdasarkan Kendaraan", Prosiding Seminar Nasional Riset Information Science (SENARIS), 2020 Publication	<1 %

Exclude quotes

On

Exclude matches

< 5 words

Exclude bibliography

On