

**PENGEMBANGAN SISTEM IDENTIFIKASI  
BIOMETRIK WAJAH MENGGUNAKAN METODE *NEURAL  
NETWORK* dan *PATTERN MATCHING***

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar ST**



**ELIZABETH**  
**06 06 04 2506**

**UNIVERSITAS INDONESIA  
FAKULTAS TEKNIK UNIVERSITAS INDONESIA  
PROGRAM STUDI TEKNIK ELEKTRO  
KEKHUSUSAN ELEKTRO  
DEPOK  
JULI 2008**

## HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya saya sendiri,  
Dan semua sumber baik yang dikutip maupun dirujuk  
telah saya nyatakan dengan benar.**

Nama : ELIZABETH

NPM : 0606042506

Tanda Tangan : .....

Tanggal : 17 juli 2008

## HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :  
Nama : Elizabeth  
NPM : 0606042506  
Program Studi : Teknik Elektro  
Judul Skripsi : **PENGEMBANGAN SISTEM IDENTIFIKASI  
BIOMETRIK WAJAH MENGGUNAKAN  
METODE *NEURAL NETWORK* dan *PATTERN  
MATCHING***

**Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Elektro Fakultas Teknik, Universitas Indonesia**

### DEWAN PENGUJI

Pembimbing : F. Astha Ekadiyanto. ST .Msc. (.....)  
Penguji : Dr.Ir. Dodi Sudiana .M.Eng (.....)  
Penguji : Dr. Ir. Arman Djohan Diponegoro (.....)

Ditetapkan di : Depok  
Tanggal : 17 juli 2008

## UCAPAN TERIMAKASIH

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Jurusan Elektro pada Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada: F. Astha Ekadiyanto. ST .Msc., selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini;

- orang tua dan keluarga saya yang telah memberikan dukungan doa dan semangat;
- Dr.Ir. Dodi Sudiana .M.Eng selaku dosen yang telah memberikan masukan dan kritik membangun untuk skripsi saya menjadi lebih baik lagi;
- Dr. Ir. Arman Djohan Diponegoro selaku dosen yang telah memberikan semangat, saran dan masukan untuk skripsi ini;
- teman sekosan yang telah banyak membantu saya dengan bersedia menjadi model dalam menyelesaikan skripsi ini terutama kepada Edit, mbak Iramaya dan temannya Ami, Irma, Nurina, Puji, Astri, Ira, Uli;
- Mbak Ratih teman kosan yang sudah membantu kesiapan sebelum sidang.
- Arthania Retno P. sebagai teman seangkatan atas kebersamaannya menyelesaikan skripsi, juga sebagai salah satu model dalam skripsi ini;
- Eka Dharma yang selalu memberikan perhatian, semangat dan mendukung pembuatan skripsi ini;
- Rina Safitri, Nurul Hikmah dan Pamela Alfa A sebagai sahabat yang selalu mengingatkan hal-hal berkaitan dengan perkuliahan termasuk skripsi ini;
- Maria Yulianty yang telah banyak memberikan dukungan dan semangat dalam pada saat proses pembuatan skripsi;
- Citra P sebagai salah satu model dalam skripsi ini;
- Fadly, Ingot, Dwi Kurnia, Gerald, Yudhi dan semua rekan ekstensi Teknik Elektro angkatan 2006 yang telah memberikan saran dan dukungannya;

- Semua pihak yang telah membantu pembuatan skripsi ini yang tidak dapat disebutkan satu-persatu.

Akhir kata, saya berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, 17 Juli 2006



Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI  
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

---

---

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Elizabeth  
NPM : 0606042506  
Program Studi : Teknik Elektro  
Departemen : Elektro  
Fakultas : Teknik  
Jenis karya : Skripsi

demikian demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul :

**”PENGEMBANGAN SISTEM IDENTIFIKASI BIOMETRIK WAJAH  
MENGUNAKAN METODE *NEURAL NETWORK* dan *PATTERN  
MATCHING*”**

berserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok  
Pada tanggal : 17 Juli 2007  
Yang menyatakan

( Elizabeth )

## ABSTRAK

Nama : Elizabeth  
Program Studi : Teknik Elektro  
judul : **PENGEMBANGAN SISTEM IDENTIFIKASI BIOMETRIK  
WAJAH MENGGUNAKAN METODE *NEURAL NETWORK*  
dan *PATTERN MATCHING***

Biometrik adalah salah satu teknologi canggih yang banyak dipakai untuk menjadi bagian dari sistem keamanan di berbagai bidang. Teknologi biometrik yang ada di sekitar kita ada berbagai macam seperti sistem identifikasi retina, iris mata, telapak tangan, sidik jari dan wajah. Banyak komputer atau laptop yang dilengkapi oleh kamera digital atau webcam yang terintegrasi dengan sistem yang ada di komputer itu sendiri. Teknologi camera digital semakin hari juga semakin canggih dalam pengambilan gambar yang dapat disesuaikan dengan situasi apapun contohnya seperti didalam ruangan atau diluar ruangan. Hal ini memungkinkan untuk pengaplikasian pengenalan wajah sebagai sistem autentikasi pengganti password selain fingerprint. Penelitian ini difokuskan pada perancangan aplikasi sistem pengenalan wajah menggabungkan dua metode yakni jaringan saraf tiruan dan metode pencocokan pola. Input dari sistem pengenalan wajah ini diambil dari webcam yang sudah melalui proses *pre-processing* dan sudah difokuskan ke bagian wajah dengan sistem pendeteksi wajah dengan metode *pattern matching*. Selain itu hasil dari *pre-processing* juga digunakan sebagai data *training* atau pelatihan. Gambar wajah hasil dari *preprocessing* ini kemudian masuk ke proses pengenalan menggunakan algoritma jaringan saraf tiruan. Hasil dari proses pengenalan wajah adalah berupa nama dari wajah orang yang dikenali. Sistem ini telah diuji pada lebih dari 36 sampel wajah yang diambil dari 12 orang. Hasil akhir menunjukkan bahwa sistem ini berhasil mengidentifikasi sampel-sampel wajah tersebut dengan tingkat keberhasilan mencapai 86%.

**Kata Kunci : Biometrik, Wajah, Identifikasi, Deteksi, Jaringan saraf tiruan  
*Template Matching*, Pencocokan pola , Webcam, Citra digital.**

## ABSTRACT

Name : Elizabeth  
Study Program: Eletrical  
Title : **THE DEVELOPMENT OF SYSTEM IDENTIFICATION  
BIOMETRIC BY FACE RECOGNITION USING NEURAL  
NETWORK METHOD AND PATTERN MATCHING  
METHOD**

Biometric is one of the modern technology features that is used mostly as a part of security system in many types of application. There are so many biometric technology options this day such as retina identification, iris, eye, hand, finger print, and face. Many computers such as laptop are completed with digital camera or webcam which integrated with in the system computer it self. Camera technology is getting more sophisticated in nowadays in capturing image from many situations such as indoor or outdoor environmental. This technology allows the possibility to develop face recognition as an option to authentication system in computer, other than the most popular fingerprint. This final project focuses on the design of face identification application using combination of two methods, neural network method and pattern matching method. The input of the system is taken from face detection algorithm with pattern matching method on webcam images which focused on human face area and already pass preprocessing first. The digital images from preprocessing are also used as a training data. The preprocessed image is then passed into the recognition process using neural network algorithm. The result of the recognition process is the person's credential which in this case the name. This system has been tested over 36 samples taken from 12 people. Result show that the system has identified the samples with 86% success rate.

**Key Word : Biometric, face, Identification, Detection, *Neural Network*  
*Template Matching, Pattern Matching, Webcam, digital image.***

## DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERNYATAAN ORISINALITAS.....	ii
HALAMAN PENGESAHAN.....	iii
UCAPAN TERIMAKASIH.....	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI.....	vi
ABSTRAK.....	vii
ABSTRACT.....	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xii
<b>BAB 1 PENDAHULUAN.....</b>	<b>1</b>
1.1. LATAR BELAKANG .....	1
1.2. PERUMUSAN MASALAH.....	2
1.3. TUJUAN PENELITIAN.....	3
1.4. BATASAN MASALAH.....	4
1.5. METODOLOGI.....	4
1.6. SISTEMATIKA PENULISAN.....	5
<b>BAB 2 LANDASAN TEORI.....</b>	<b>6</b>
2.1. BIOMETRIC.....	6
2.2. CITRA DIGITAL.....	8
2.2.1. Operasi pemrograman sederhana pada citra.....	9
2.2.2. Tipe-tipe Citra.....	10
2.2.3. Operasi titik pada citra.....	14
2.2.4. Informasi Citra.....	23
2.3. PENGENALAN WAJAH.....	24
2.4. TEMPLATE MATCHING.....	25
2.5. JARINGAN SARAF TIRUAN.....	26
2.5.1. Supervised Learning.....	32
2.5.2. Multi-Layer Perceptron.....	33
2.5.3. Algoritma Backpropagation.....	34
2.6. SISTEM AUTENTIKASI.....	36
2.6.1. Faktor Autentikasi.....	37
<b>BAB 3 PERANCANGAN SISTEM.....</b>	<b>38</b>
3.1. PENDETEKSI WAJAH.....	39
3.2. PENGENALAN WAJAH.....	42
3.2.1. Pengolahan Citra ( <i>Pre-Processing</i> ).....	43
3.2.2. Pelatihan .....	50
3.2.3. Identifikasi.....	53
<b>BAB 4 PENGUJIAN DAN ANALISA.....</b>	<b>57</b>
4.1. PENDETEKSI WAJAH.....	57
4.1.1. Penentuan Pola pada Sistem Pendeteksi Wajah.....	57
4.1.2. Pendeteksi Wajah pada Posisi Wajah Yang Berbeda.....	60
4.2. PENGENALAN WAJAH.....	62

4.2.1. Pengujian Parameter Sistem Pengenalan Wajah.....	62
4.2.2. Pengujian Tingkat Keberhasila Sistem Pengenalan Wajah.....	64
<b>BAB 5 KESIMPULAN .....</b>	<b>69</b>
DAFTAR REFERENSI .....	70
LAMPIRAN A.....	71
LAMPIRAN B.....	74



## DAFTAR GAMBAR

Gambar 1.1	Diagram Sistem Identifikasi Pengenalan Wajah .....	3
Gambar 2.1	Representasi Hexadesimal Citra RGB.....	11
Gambar 2.2	Kombinasi Citra RGB.....	11
Gambar 2.3	Struktur Citra Terindex.....	14
Gambar 2.4	Average Filtering .....	16
Gambar 2.5	Neuron Sensori .....	27
Gambar 2.6	Neuron Asosiasi.....	28
Gambar 2.7	Neuron Motor .....	28
Gambar 2.8	Model Generik Multilayer Perception.....	29
Gambar 2.9	Neuron pada saraf manusia .....	30
Gambar 2.10	Neuron pada JST.....	31
Gambar 2.11	<i>Step Function</i> .....	32
Gambar 2.12	<i>Sigmoid Function</i> .....	32
Gambar 2.13	Algoritma Propagasi Balik.....	35
Gambar 3.1	Diagram Sistem Identifikasi Pengenalan Wajah.....	39
Gambar 3.2	Diagram Blok Sistem Pendeteksi Wajah.....	40
Gambar 3.3	Blok korelasi 2 dimensi.....	40
Gambar 3.4	<i>Maximum block</i> .....	40
Gambar 3.5	<i>Draw Shape block</i> .....	41
Gambar 3.6	Diagram Blok Sistem Pengenalan Wajah.....	42
Gambar 3.7	Diagram Blok Proses Input Wajah Dari Webcam dengan Kotak pembatas.....	43
Gambar 3.8	Diagram alir <i>Pre-Processing</i> .....	44
Gambar 3.9	Citra Digital Sebelum di- <i>Crop</i> (a); Citra Digital Sesudah di- <i>Crop</i> (b).....	45
Gambar 3.10	Citra Digital Sebelum di- <i>Resize</i> (a); Citra Digital Setelah di- <i>Resize</i> (b).....	46
Gambar 3.11	Citra digital sebelum di- <i>greyscale</i> (a); Citra digital setelah di- <i>greyscale</i> (b).....	47
Gambar 3.12	Citra digital sebelum di- <i>histeq</i> (a); Citra digital setelah di- <i>histeq</i> (b).....	48
Gambar 3.13	Citra digital sebelum di- <i>Filter</i> (a); Citra digital setelah di- <i>Filter</i> (b).....	49
Gambar 3.14	Citra digital sebelum diubah ke <i>Binary Image</i> (a); Citra digital setelah diubah ke <i>Binary Image</i> (b).....	50
Gambar 3.15	Diagram alir proses <i>training</i> .....	51
Gambar 3.16	Diagram alir proses identifikasi.....	55
Gambar 3.17	Fungsi Transfer <i>Competitive</i> .....	56
Gambar 3.18	Tampilan GUI program pengenalan wajah.....	56
Gambar 4.1	Citra Pola Wajah .....	58
Gambar 4.2	Citra Template Wajah.....	58
Gambar 4.3.	Beberapa sample frame video input 160 x 120 hasil deteksi wajah.....	59
Gambar 4.4.	Kesalahan pada proses pengenalan karena mimik wajah.....	65
Gambar 4.5.	Contoh pengenalan wajah yang berhasil.....	66

## DAFTAR TABEL

<b>Tabel 3.1</b>	Sample data untuk proses pelatihan.....	52
<b>Tabel 4.1</b>	Hasil Percobaan Deteksi Wajah dengan Menentukan Ukuran Pola Wajah.....	58
<b>Tabel 4.2</b>	Hasil Percobaan Deteksi Wajah dengan Menentukan Ukuran Template Wajah.....	59
<b>Tabel 4.3</b>	Ujicoba Deteksi wajah pada Posisi Kordinat (x,y) Yang Berbeda.....	61
<b>Tabel 4.4.</b>	Hasil Uji Coba Deteksi Wajah pada Posisi Wajah yang Berbeda-beda.....	61
<b>Tabel 4.5</b>	Uji Hidden Layer 10 Neural Network pada pengenalan Wajah.....	62
<b>Tabel 4.6</b>	Uji Hidden Layer 40 Neural Network pada pengenalan Wajah.....	63
<b>Tabel 4.7</b>	Uji Hidden Layer 100 Neural Network pada pengenalan Wajah .....	63
<b>Tabel 4.8.</b>	Hasil identifikasi 12 data <i>training</i> .....	64
<b>Tabel 4.9.</b>	Hasil identifikasi 6 data <i>training</i> .....	65
<b>Tabel 4.10</b>	Hasil identifikasi 12 data <i>training</i> pada data wajah selain data <i>training</i> .....	66
<b>Tabel 4.11</b>	Hasil identifikasi 6 data <i>training</i> pada data wajah selain data <i>training</i> .....	67
<b>Tabel 4.12</b>	Hasil identifikasi seluruh data wajah.....	67

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Biometrik adalah salah satu teknologi canggih yang banyak dipakai untuk menjadi bagian dari system keamanan di berbagai bidang. Biometrik ini bahkan sudah digunakan sejak zaman mesir kuno. Salah satu teknologi biometrik adalah sistem pengenalan wajah. Selain itu teknologi biometrik yang ada di sekitar kita ada berbagai macam seperti sistem identifikasi retina, iris mata, telapak tangan dan sidik jari.

Pengenalan wajah belakangan ini semakin banyak dimanfaatkan, antara lain untuk sistem pengenalan biometrik seperti sistem pencarian dan pengindeksan database citra digital dan database video digital, sistem keamanan kontrol akses area terbatas, konferensi video, dan interaksi manusia dengan komputer.

Interaksi manusia dengan komputer menjadi latar belakang terciptanya ide untuk membangun sebuah sistem pengamanan dengan proses autentikasi biometrik. Peralatan tersebut cukup beresiko untuk mengundang kejahatan kriminal seperti pencurian, baik pencurian secara fisik maupun pencurian data. Dengan adanya sistem autentikasi pengenalan wajah yang terpasang pada komputer ataupun labtop diharapkan akan meminimalisir adanya tidak kriminal tersebut.

Disamping itu peralatan canggih seperti handphone, komputer terutama labtop sekarang ini sudah banyak dilengkapi oleh kamera digital atau webcam yang terintegrasi dengan sistem yang ada di komputer itu sendiri. Teknologi camera digital semakin hari juga semakin canggih dalam pengambilan gambar yang dapat disesuaikan dengan situasi apapun contohnya seperti didalam ruangan atau diluar ruangan. Hal ini juga memungkinkan untuk pengaplikasian pengenalan wajah sebagai sistem autentikasi pengganti *password* selain fingerprint. Didalam proses autentikasi ini terdapat proses identifikasi wajah untuk menunjukkan identitas pengguna.

Metode jaringan syaraf tiruan (JST) adalah sebuah sistem proses informasi dengan kemampuan untuk belajar, mengingat dan menyelesaikan masalah berdasarkan proses belajar yang diberikan. Sistem ini mempunyai struktur yang menyerupai jaringan saraf manusia. Dengan struktur demikian, maka JST dapat menyelesaikan masalah yang berhubungan dengan klasifikasi dan pencocokan pola, pendekatan fungsi, optimisasi, vektor kuantisasi, dan pengelompokan data. Tugas – tugas tersebut dapat diselesaikan secara efisien oleh komputer dimana komputer dapat menyelesaikan operasi aritmatika dan perhitungan logaritma lebih cepat [1].

Sistem biometrik wajah ini lebih mudah digunakan dari sisi user, karena tidak memerlukan kontak fisik yang membahayakan pengguna. Selain itu biometrik wajah juga lebih ekonomis karena hanya menggunakan kamera digital, yang pada penelitian ini menggunakan webcam, dibandingkan biometrik lain yang alat *scan*-nya jauh lebih mahal. Dengan metode *neural network* atau jaringan saraf tiruan yang di dukung dengan pengolahan citra yang sederhana diharapkan penelitian ini mendapatkan hasil yang lebih akurat, lebih murah dan lebih cepat pada system autentikasi pengenalan wajah.

## 1.2 Perumusan Masalah

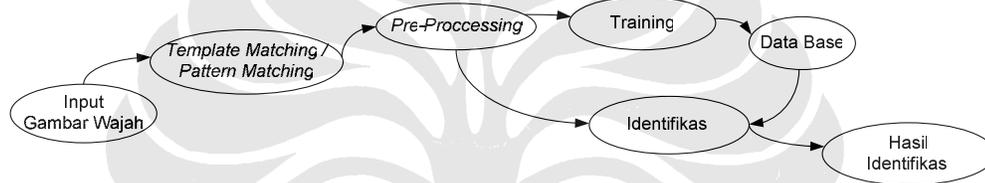
Untuk merealisasikan ide yang timbul dari latar belakang diatas, ada beberapa metode yang dapat digunakan diantaranya seperti metode jaringan saraf tiruan (*neural network*).

Input dari system autentikasi ini adalah gambar atau citra digital yang diambil dari hasil rekaman webcam setelah di deteksi wajah lalu baru dikenali dengan metode Jaringan Saraf Tiruan (JST). Sistem keseluruhan terdiri dari 4 tahap, tahap pertama deteksi wajah dengan *pattern matching*, tahap kedua yakni data-data yang sudah ada dilatih (*training*) sesuai dengan parameter-parameter JST yang sudah ditentukan melakuai uji coba, tahap ketiga proses identifikasi atau pengenalan, dan tahap terakhir evaluasi tingkat keberhasilan.

Berbagai macam cara sudah di uji coba dan diterapkan untuk mendeteksi wajah manusia. Seperti metode pencocokan *template* (*Template Matching*), Model Markov Tersembunyi (*Hidden Markov Model/HMM*), Wajaheigen (*Eigenface*),

Jaringan Syaraf Tiruan (JST), *Fuzzy logic*, . Metode yang dipilih sebagai metode pendukung system identifikasi ini adalah gabungan dari metode Jaringan Syaraf Tiruan (JST) dan metode pencocokan pola (*Pattern Matching*).

Penggabungan kedua metode tersebut merupakan kolaborasi metode pada saat deteksi wajah dan pengenalan wajah. Deteksi wajah pada saat pengambilan gambar pada waktu pertama kali, baik untuk data *training* maupun untuk data gambar yang akan dikenali, menggunakan metode *template matching* atau yang biasa disebut juga *pattern matching*. Sementara itu pengenalan wajah dilakukan pada saat proses identifikasi menggunakan metode *neural network* dengan algoritma propagasi balik atau *back propagation*.



Gambar 1.1 Diagram Sistem Identifikasi Pengenalan Wajah

Dari Gambar 1.1 diatas dapat dilihat secara garis besar sistem pengenalan wajah yang akan dibuat sebagai program untuk menunjukkan proses identifikasi sebagai bagian dari proses autentikasi biometrik wajah.

### 1.3 Tujuan Penulisan

Tujuan dari penulisan ini adalah untuk melengkapi sebuah penelitian sistem pengenalan wajah menggunakan metode *neural network* dan *template matching*. Tulisan ini juga sebagai suatu sarana menuangkan ide skripsi tentang teknologi biometrik pengenalan wajah, yang akan diaplikasikan ke suatu sistem autentikasi yang pada umumnya bertujuan untuk pengamanan. Teknologi biometrik wajah disini akan lebih ditonjolkan kepada sistem pendeteksian wajah dengan metode jaringan syaraf tiruan (*Neural Network*) yang akan digabungkan dengan metode

pencocokan pola wajah (*Pattern Matching*) dengan tujuan agar system menjadi lebih presisi.

#### 1.4 Batasan Masalah

Permasalahan pada sistem pengenalan wajah yang umumnya terjadi pada data gambar wajah yang ada dengan berbagai ukuran, posisi, dan latar belakang pencahayaan yang kurang baik.

- Dengan permasalahan yang pada sistem pengenalan wajah, maka wajah yang akan di proses untuk sistem autentikasi ini adalah wajah yang menghadap ke depan (frontal), dalam posisi tegak, pada jarak tertentu dan tidak terhalangi sebagian oleh objek lain, dengan pencahayaan yang cukup.
- Metode deteksi dan verifikasi wajah yang dipakai adalah jaringan syaraf tiruan (*Neural Network*) yang akan digabungkan dengan metode pencocokan pola wajah (*Pattern Matching*).
- Sistem autentikasi yang akan dibuat adalah sebuah simulasi program yang akan mengaplikasikan teknologi biometrik pengenalan wajah, pada komputer dengan menggunakan webcam.

#### 1.5 Metodologi

##### 1. Studi literatur

Studi literatur baik terhadap teori algoritma jaringan saraf tiruan dan metode pencocokan pola wajah. Juga pembelajaran system autentikasi sebagai referensi dari program yang akan dibangun.

##### 2. Analisis

Analisis terhadap hasil studi literatur sesuai dengan masalah yang dirumuskan, yaitu bagaimana gabungan metode jaringan saraf tiruan untuk identifikasi pengguna pada system autentikasi.

### 3. Perancangan

Perancangan perangkat lunak untuk pengenalan wajah dengan program guna mengaplikasikan penggabungan metode *pattern matching* dan metode jaringan saraf tiruan yakni dengan menggabungkan sistem pendeteksi wajah dengan metode *pattern matching* lalu output-nya berupa gambar wajah yang sudah di crop otomatis menjadi input di sistem pengenalan wajah menggunakan metode *neural network*.

#### 1.6 Sistematika Penulisan

Sistematika penulisan laporan Tugas Akhir ini adalah sebagai berikut:

**Bab 1 Pendahuluan**, berisi penjelasan mengenai latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi, serta sistematika penulisan yang digunakan untuk menyusun laporan Tugas Akhir.

**Bab 2 Dasar Teori**, berisi dasar teori yang digunakan dalam analisis dan perancangan pada skripsi yang dipresentasikan. Dasar teori yang diangkat antara lain teknologi biometri, citra digital, metode pencocokan wajah template, metode deteksi dengan jaringan syaraf tiruan, dan Sistem Autentikasi.

**Bab 3 Perancangan**, berisi rancang bangun system pengenalan wajah menggunakan metode jaringan saraf tiruan dan pencocokan pola wajah, mencakup analisis metode serta proses pembelajaran jaringan saraf tiruan, analisis metode pencocokan wajah, representasi wajah penggabungan kedua metode, untuk kemudian di aplikasikan kedalam simulasi system autentikasi.

**Bab 4 Pengujian dan Analisa**, Bab ini membahas pengujian dari sistem yang sudah di bangun. Disini juga terdapat cara penggunaan sistem. Dari Bab ini maka akan terlihat seberapa persentase keberhasilan pengenalan wajah menggunakan *neural network*.

**Bab 5 Penutup**, berisi kesimpulan dan saran yang didapatkan selama pelaksanaan Skripsi.

## BAB 2

### DASAR TEORI

Pada Bab 2 ini dasar – dasar teori yang akan dibahas adalah dasar teori pendukung sistem pengenalan wajah menggunakan metode neural network dan *pattern matching*. Dasar – dasar teori pendukung tersebut antara lain tentang; Biometrik, Citra Digital, Deteksi Wajah, *Pattern Matching*, Jaringan Saraf Tiruan (*Neural Network*) dan Sistem autentikasi.

#### 2.1 Biometrik

Teknologi biometrik adalah teknologi keamanan yang menggunakan bagian tubuh atau kebiasaan yang berasal dari tubuh kita sebagai identitas. Dunia medis mengatakan bahwa ada berapa bagian tubuh kita yang sangat unik juga kebiasaan yang diciptakannya, seperti tanda tangan. Artinya, tidak dimiliki oleh lebih dari satu individu. Contoh bagian tubuh seperti sidik jari atau retina mata. Meskipun bentuk atau warna mata bisa saja sama, namun retina mata belum tentu sama. Begitu juga dengan suara dan struktur wajah. Bagian-bagian unik inilah yang kemudian dikembangkan sebagai atribut keamanan.

Sebagai bagian dari teknologi keamanan, biometrik memiliki dua fungsi sekaligus yang dapat dijalankan terpisah maupun secara bersamaan. Yang pertama sebagai pencatat ID atau sebagai alat verifikasi (*password*).

Teknologi biometrik hampir dapat diterapkan di mana saja. Mulai untuk melindungi sebuah barang tertentu dari akses yang tidak diinginkan, seperti komputer. Sampai untuk melindungi sebuah ruangan yang ramai dari orang-orang tertentu. Misalnya, untuk mengetahui keberadaan teroris atau penjahat lain di bandara.

Cara kerja teknologi keamanan yang satu ini hampir sama dengan teknologi keamanan lain yang sangat bergantung kepada sensor. Sensor yang

digunakan pada teknologi biometrik cenderung mahal dan semakin akurat ketajamannya maka akan semakin mahal.

Selain sensor, bagian yang tidak kalah penting dari biometrik adalah data. Bagaimana Anda menyimpan data pada sebuah sistem sangat penting. Sebab biometrik adalah teknologi yang bergantung kepada data. Bila data yang disimpan tidak aman atau lengkap, kemungkinan adanya penyusup ke system ini akan lebih besar.

Menurut sistemnya biometrik sendiri terbagi atas tiga macam, sebagai berikut:

- a. Sistem yang menyimpan data langsung pada alat.

Dengan sistem ini, data akan disimpan pada media penyimpanan yang berada dalam alat detektor. Jika sewaktu-waktu mesin harus di-reset atau dikembalikan ke posisi awal, maka data yang ada dapat saja ikut terhapus. Sehingga petugas harus meng-input ulang. Jika data yang dimasukkan sangat banyak tentu akan sangat merepotkan, lain halnya bila data tidak terlalu banyak. Biometrik dengan sistem ini sangat cocok untuk diterapkan pada sebuah alat tertentu yang tidak digunakan oleh banyak orang atau untuk melindungi sebuah ruang khusus, yang juga tidak diakses oleh banyak pengunjungnya.

- b. Sistem yang menyimpan data pada jaringan.

Sistem yang kedua memanfaatkan jaringan untuk menyimpan datanya. Sistem yang kedua sangat efektif bagi aplikasi yang memang dipergunakan untuk banyak user. Misalnya saja untuk data absen karyawan atau siswa. Bentuk fisik yang ditampilkan oleh alat juga tidak perlu terlalu besar. Karena data tidak akan diproses langsung pada alat. Melainkan dikirim dahulu ke sebuah jaringan baru kemudian diproses dan disimpan. Sistem ini memang membutuhkan waktu lama. Tetapi cukup efektif untuk data yang besar. Karena tidak akan terkena risiko data hilang pada saat proses reset pada alat harus dilakukan.

- c. Sistem yang menyimpan data pada sebuah chip.

Sistem yang terakhir ini menggunakan media tambahan berupa chip untuk menyimpan data si pemilik ID. Sehingga untuk menggunakannya

seseorang harus membawanya. Untuk sistem yang terakhir ini, akan sangat efektif diterapkan untuk yang memiliki pengguna sangat banyak atau bila alatnya hendak diletakkan di tempat umum. Misalnya saja untuk keamanan di mesin ATM atau hanya sekadar sebagai ID masuk dalam sebuah gedung.

## 2.2 Citra Digital

Citra digital merupakan suatu matriks yang terdiri dari baris dan kolom, dimana setiap pasangan indeks baris dan kolom menyatakan suatu titik pada citra. Nilai matriksnya menyatakan nilai kecerahan titik tersebut. Titik-titik tersebut dinamakan sebagai elemen citra, atau pixel (*picture elemen*). [2].

Pengolahan citra atau *image processing* merupakan proses untuk memperbaiki kualitas citra, khususnya dengan menggunakan komputer, menjadi citra yang kualitasnya lebih baik agar lebih mudah untuk diinterpretasi oleh manusia ataupun mesin. Umumnya operasi – operasi pengolahan citra dilakukan bila:

- a. Perbaikan atau modifikasi citra perlu dilakukan untuk meningkatkan kualitas penampakan atau untuk menonjolkan beberapa aspek informasi yang terkandung dalam citra.
- b. Elemen dalam citra perlu di cocokkan, dikelompokkan atau diukur.
- c. Sebagian citra perlu digabung dengan bagian citra yang lain.

Terminologi yang berkaitan dengan *image processing* adalah *computer vision*. Pada hakikatnya *computer vision* mencoba meniru cara kerja sistem visual manusia (*human vision*). *Human vision* sesungguhnya sangat kompleks. manusia melihat objek dengan indra penglihatan (mata), lalu citra objek diteruskan ke otak untuk di interpretasi sehingga manusia mengerti objek apa yang tampak dalam pandangan matanya. Hasil interpretasi mungkin digunakan untuk mengambil keputusan (misal menghindar kalau melihat mobil yang melaju di depan).

*Computer vision* merupakan proses otomatis yang mengintegrasikan sejumlah besar proses untuk persepsi visual, seperti akuisisi citra, pengolahan citra, klasifikasi, pengenalan (*recognition*), dan membuat keputusan.

*Computer vision* terdiri atas teknik – teknik untuk mengestimasi ciri – ciri objek di dalam citra, pengukuran citra yang berkaitan dengan geometry objek, dan menginterpretasi informasi geometry tersebut. Proses – proses dalam *Computer vision* dapat dibagi menjadi tiga proses:

- a. Memperoleh atau mengakuisisi citra digital
- b. Melakukan teknik komputasi untuk memproses atau memodifikasi data citra (operasi – operasi pengolahan citra)
- c. Menganalisis dan menginterpretasi citra dan menggunakan hasil pemrosesan untuk tujuan tertentu, misalnya memandu robot, mengontrol peralatan, memantau proses manufaktur dsb.

### 2.2.1 Operasi pemrograman sederhana pada citra

Operasi pemrograman citra yang masih sederhana dibagi menjadi 3 tahap membaca citra, menampilkan citra dan menulis citra.

- a. Membaca citra

Citra digital dapat dibaca dengan menggunakan fungsi `imread` dengan sintaks :

```
Imread ('nama file');
```

Misalkan nama filenya adalah `muka.jpg` maka sintaksnya menjadi

```
Imread ('muka.jpg');
```

Artinya program akan membaca file dengan nama `muka` dengan ekstensi `.jpg` pada *current directory*.

Ada beberapa ekstensi citra yang dapat dibaca oleh fungsi `imread`, diantaranya adalah `tif`, `jpg`, `gif`, `png`, `bmp`, `xwd`, `ras`, `pnm`, dan lain sebagainya.

- b. Menampilkan citra

Citra akan ditampilkan dalam program dengan menggunakan fungsi `imshow` tetapi hanya jika telah melewati proses pembacaan menggunakan fungsi `imread`, dengan sintaks :

```

imshow ('nama file')
imshow ('f,G')

```

Dengan *f* adalah array citranya sedangkan *G* adalah tingkat intensitas warnanya. Jika *G* tidak ditentukan maka secara *default* dikenali sebagai tingkat warna 256 (0-255).

Misalkan dengan nama file *muka.jpg* maka proses menampilkannya adalah sebagai berikut :

```

>> imread ('muka.jpg') ; dan dilanjutkan dengan
>> imshow ('muka.jpg');

```

Atau dapat juga dengan memisalkan *muka.jpg* dengan array *f*.

```

>> f = 'muka.jpg' ; dilanjutkan dengan
>> imread (f); kemudian ditampilkan dengan
>> imshow (f);

```

### c. Menulis Citra

Citra dapat ditulis dan disimpan dalam disc dengan menggunakan fungsi *imwrite*, yang memiliki sintaks dasar sebagai berikut :

```

imwrite (f, 'nama file')

```

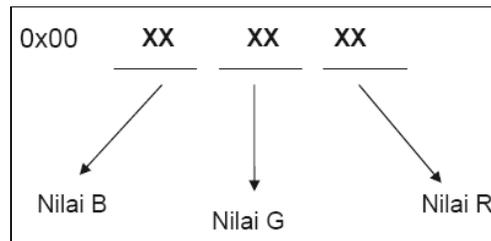
perlu diketahui bahwa dengan menggunakan fungsi *imwrite* ini maka citra yang akan disimpan harus memiliki ekstensi file yang diketahui oleh program.

## 2.2.2 Tipe-tipe Citra

Tipe-tipe citra antara lain adalah citra RGB, citra *grayscale*, citra negatif, citra biner dan citra terindex. Masing – masing citra akan dibahas lebih lanjut sebagai berikut.

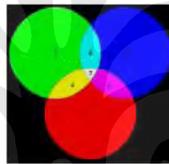
### a. Citra RGB

Dasar dari pengolahan citra adalah pengolahan warna RGB pada posisi tertentu. Dalam pengolahan citra warna direpresentasikan dengan nilai hexadesimal dari 0x00000000 sampai dengan 0x00ffffff. Warna hitam dengan nilai 0x00000000, dan warna putih bernilai 0x00ffffff. Presentasi pla citra RGB digambarkan pada Gambar 2.1 berikut.



**Gambar 2.1** Representasi Hexadesimal Citra RGB

Nilai warna diatas merupakan gabungan antara 3 warna primer yaitu merah (Red), hijau (Green), biru (Blue), seperti yang terlihat pada Gambar 2.2. Sehingga untuk memperoleh warna lain selain warna primer, perlu menggabungkan skala kecerahan dari tiap warna.



**Gambar 2.2** Kombinasi Citra RGB

Untuk mengetahui kombinasi warna, perlu dibuat suatu program yang dapat menampilkan warna sesuai dengan nilai masukan. Sehingga dapat dicoba berbagai macam kombinasi warna RGB.

b. Citra *Grayscale*

Proses mengubah citra warna menjadi citra *grayscale* digunakan dalam *image processing* untuk menyederhanakan model citra. Citra berwarna terdiri dari 3 layer matriks yaitu R-layer, G-layer, B-layer. Untuk melakukan proses – proses selanjutnya perlu diperhatikan 3 layer diatas, Jika setiap proses perhitungan dilakukan menggunakan 3 layer maka dilakukan 3 perhitungan yang sama.

Mengubah 3 layer diatas menjadi 1 layer matrik *grayscale*. Dalam citra ini tidak ada lagi warna, yang ada adalah derajat keabuan. Untuk mengubah citra berwarna yang mempunyai nilai matrik masing – masing r, g, b menjadi citra

*grayscale* dengan nilai  $s$ . Konversi dapat dilakukan dengan mengambil rata – rata dari nilai  $r$ ,  $g$ , dan  $b$  sehingga dapat dituliskan menjadi persamaan 2.1 berikut:

$$s = \frac{r + g + b}{3} \dots\dots\dots 2.1$$

Dengan:

S : Nilai derajat keabuan

r : Nilai warna merah

g : Nilai warna hijau

b : Nilai warna biru

Dalam program untuk melakukan konversi citra dari RGB menjadi *grayscale* dapat dilakukan dengan mempergunakan fungsi `rgb2gray`.

#### c. Citra Negatif

Citra negatif (*negative image*) merupakan pemrosesan yang hanya melibatkan satu pixel saja (tidak menggunakan jendela ketetanggaan). Hasil dari gambar negatif ini seperti klise foto.

Perubahan grey level pada gambar negatif yaitu dengan sistematis dipresentasikan oleh persamaan 2.2 berikut:

$$G_{baru} = 255 - G_{lama} \dots\dots\dots 2.2$$

#### d. Citra Biner

Dalam banyak aplikasi yang melibatkan citra, sangat berguna untuk memisahkan wilayah objek yang kita perlukan dari wilayah-wilayah lain yang tidak kita perlukan yang dikenal sebagai *background*.

Proses *Thresholding* sering menyediakan cara yang mudah dan nyaman untuk melakukan segmentasi citra ini dengan cara membedakan intensitas atau warna antara objek (*foreground*) dengan latar belakang (*background*). Kadang *thresholding* juga sangat penting untuk dapat melihat daerah dari sebuah citra yang nilai pixelnya terletak dalam range atau band intensitas (atau warna). Proses *thresholding* ini akan menghasilkan citra biner.

Citra biner (*binary image*) adalah citra yang hanya mempunyai 2 nilai derajat keabuan yaitu hitam dan putih, biasanya 0 menyatakan hitam (*background*) dan 1 atau 255 untuk putih (*foreground*). Tetapi ketentuan tersebut juga tergantung dari citra yang *dithresholdkan*. Meskipun saat ini citra yang berwarna lebih disukai karena memberi kesan yang lebih kaya daripada citra biner, namun tidak membuat citra biner mati. Pada beberapa aplikasi citra biner masih tetap dibutuhkan, misalnya citra logo instansi (yang hanya terdiri atas wana hitam dan putih), citra kode batang (*bar code*) yang tertera pada label barang, citra hasil *scanning* dokumen teks, dan sebagainya.

Seperti yang telah diterangkan citra biner hanya memiliki 2 derajat keabuan, pixel-pixel objek bernilai 1 dan pixel-pixel latar belakang bernilai 0. Pada waktu menampilkan gambar, 0 adalah putih dan 1 adalah hitam. Jadi pada citra biner, latar belakang berwarna putih dengan objeknya berwarna hitam.

Keuntungan citra biner antara lain :

- kebutuhan memori kecil,
- waktu pemrosesan lebih cepat.

Citra biner didapatkan dengan cara meng-*threshold*-kan citra *grayscale*. Operasi *thresholding* akan mengelompokkan nilai derajat keabuan setiap pixel kedalam 2 kelas, yaitu hitam dan putih.

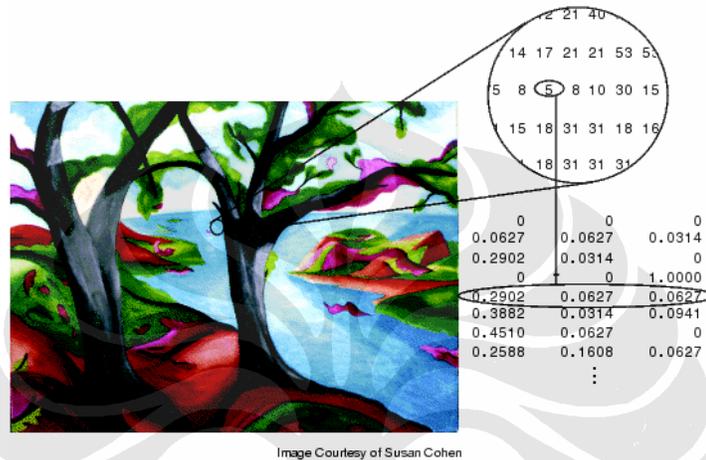
Dalam program untuk menghasilkan citra hitam putih dapat menggunakan fungsi `im2bw`.

#### e. Citra Terindex

Adalah citra yang harga-harga pixelnya direferensikan secara langsung ke peta warna RGB, sebuah citra terindex mengandung sebuah matrik data dan matrik peta warna. Matrik data bisa merupakan kelas `uint8`, `uint16` ataupun `double`. Matrik peta warnanya adalah sebuah array kelas `double` 'm x 3' yang mengandung harga floating point dalam range (0,1). Tiap-tiap baris dari peta menentukan komponen warna tersendiri dari merah, hijau dan biru.

Sebuah citra terindex menggunakan metode pemetaan langsung dari harga pixel ke harga peta warna. Warna dari setiap pixel citra di perkirakan dengan menggunakan harga yang bersangkutan dari data sebagai sebuah index menjadi

sebuah peta. Harga 1 menunjukkan baris pertama dari peta, harga 2 menunjukkan baris kedua dan seterusnya. Biasanya sebuah peta warna disimpan bersama citra terindex dan secara otomatis akan di *load* bersamaan saat citra dipanggil. Meskipun begitu, kita tidak terbatas hanya pada peta warna *default* saja akan tetapi kita dapat menggunakan peta warna yang kita pilih. Gambar 2.3 dibawah ini menunjukkan struktur dari citra terindex.



**Gambar 2.3** Struktur Citra Terindex

### 2.2.3 Operasi titik pada citra

Operasi titik pada citra melingkupi kecerahan (*Brightness*), peregangkan kontras (*Contrast Stretching*), *Average Filter*, *Median Filter*, *Filter Wiener*, *Order-Statistic Filter*, *Gaussian Filter*, *Laplacian on Gaussian (LoG) Filter*, *Laplacian Filter*, *Filter Unsharp*, *Sobel Filter*, dan *Prewitt Filter*.

#### a. *Brightness* (Kecerahan)

Pengubahan kecerahan gambar ini termasuk ke teknik domain spasial yang bertujuan untuk membuat citra lebih gelap atau lebih terang. Kecerahan (*brightness*) / kecermerlangan gambar dapat diperbaiki dengan menambahkan (atau mengurangi) sebuah konstanta kepada (atau dari) setiap pixel di dalam citra. Akibat dari operasi ini, histogram citra mengalami pergeseran. Secara sistematis operasional ini ditulliskan pada persamaan 2.3 seperti berikut :

$$f(x,y)' = f(x,y) + b \dots\dots\dots 2.3$$

Jika  $b$  positif, kecerahan gambar bertambah, sebaliknya jika  $b$  negatif kecerahan gambar berkurang.

b. Peregangan kontras (*Contrast Stretching*)

Peregangan kontras (seringkali disebut dengan normalisasi) merupakan teknik peningkatan mutu citra yang sederhana yang bertujuan untuk mengubah kontras dari sebuah citra dengan cara meregangkan/mengubah tingkat intensitas warna dari citra asli (misal  $f$ ) menjadi rentang intensitas baru dan menghasilkan citra hasil perubahan rentang tersebut yaitu  $J$ .

Sebelum proses peregangan kontras dapat dilakukan maka kita harus dapat menentukan nilai batasan atas dan bawah yang akan dikenakan kepada citra yang akan dinormalisasi. Terkadang batasan ini adalah hanya merupakan batasan maksimum dan minimum yang diperbolehkan oleh citra tersebut sebelum akhirnya terpotong (*clipped*). Sebagai contoh untuk citra 8 bit tingkat keabuan batasan nilai atas dan bawahnya bisa jadi dari 0 sampai 255.

Dalam program proses peregangan kontras ini dapat dilakukan salah satunya dengan menggunakan sintaks dan fungsi :

```
g = imadjust (f, [low_in; high_in], [low_out; high_out])
```

fungsi `imadjust` diatas akan menyesuaikan harga intensitas/peta warna citra asli ( $f$ ) menjadi citra baru ( $g$ ) dengan ketentuan nilai yaitu rentang nilai intensitas/peta warna citra asli ( $low\_in$  dan  $high\_in$ ) akan dipetakan menjadi nilai rentang citra baru yaitu  $low\_out$  dan  $high\_out$ . Harga intensitas dibawah  $low\_in$  dan diatas  $high\_in$  akan dipotong dimana  $low\_in$  yang telah terpotong menghasilkan  $low\_out$  dan  $high\_in$  yang telah terpotong akan menghasilkan  $high\_out$ .

Kita dapat menggunakan matriks kosong untuk menyatakan nilai rentang dari salah satunya ( $low\_in$   $high\_in$  ataupun  $low\_out$   $high\_out$ ), dan matriks kosong tersebut menyatakan nilai (0 1).

Perlu diketahui bahwa nilai  $low\_in$  tidak boleh lebih besar daripada nilai  $high\_in$  dan jika  $high\_out < low\_out$  maka citra barunya akan *direverse* dan memberikan hasil seperti proses negatif.

### c. Average Filter

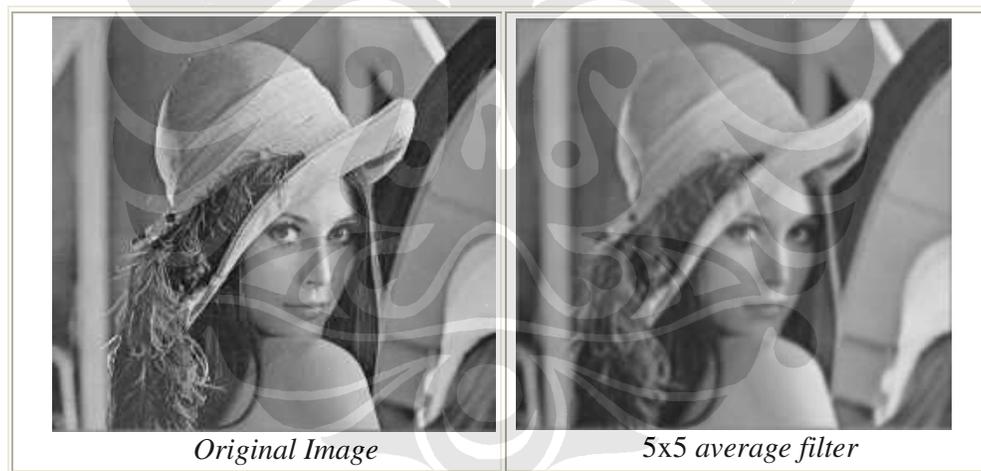
Pada *average filter*, pixel output yang dihasilkan merupakan hasil dari rata-rata dari harga pixel tersebut dan harga pixel-pixel tetangganya. *Filtering* ini merupakan hasil dari proses korelasi dan konvolusi.

Misalkan pada input pixel adalah  $f(2,2)$  maka output pixel pada  $g(2,2)$  bisa merupakan nilai rata-rata dari  $f(2,2)$  dan semua tetangga-tetangga dari  $f(2,2)$  dikoordinat  $f(1,1)$ ,  $f(1,2)$ ,  $f(1,3)$ ,  $f(2,1)$ ,  $f(2,3)$ ,  $f(3,1)$ ,  $f(3,2)$ ,  $f(3,3)$  tergantung berapa tetangga yang ingin dirata-ratakan yang dinyatakan dengan ukuran matriks, semisal *5x5 average filtering* pada contoh dibawah.

Salah satu cara untuk melakukan *average filtering* ini dalam program adalah dengan menggunakan fungsi *imfilter* ataupun dengan fungsi *fspecial*.

```
>> h = ones(5,5)/25 ;
>> output = imfilter(input,h) ;
```

Hasilnya adapat dilihat pada Gambar 2.4 berikut.



**Gambar 2.4** Average Filtering

Sintaks diatas akan mem-*filter* citra input dengan *filter* 5x5 yaitu pixel outputnya merupakan rata-rata pixel dari pixel sentralnya beserta 4 pixel disekelilingnya.

*Average filter* ini dapat digunakan untuk menghaluskan noise dalam suatu citra, *filter* ini juga memburamkan tepi, serta mengurangi kontras.

#### d. Median Filter

Cara penggunaan median *filter* adalah sama seperti saat menggunakan *average filter* yaitu setiap output yang dihasilkan merupakan hasil operasi pixel-pixel tetangga citra input, yang membedakannya adalah pada median *filter* harga pixel tetangga tersebut bukan dirata-ratakan melainkan dicari nilai mediannya.

Metode median ini mempunyai sensitivitas yang lebih 'kurang' daripada mean (rata-rata) jika berhadapan dengan nilai yang ekstrim (biasa disebut outliers) ataupun noise salt and peppers, oleh karena itu median *filter* lebih dapat untuk menghilangkan outliers dan noise tersebut tanpa menyebabkan pengurangan tingkat ketajaman dari citra.

1	2	2
1	9	3
4	6	5

Temukan nilai tengah dari 9 titik berdasarkan data diatas, maka :

1	2	2
1	3	3
4	6	3

↓

Digantikan dengan nilai tengah matriks yaitu dari 9 menjadi 3.

Pada program fungsi yang akan melakukan median ini adalah `medfilt2` yang akan melakukan *filtering* secara 2 dimensi.

```
>> B = medfilt2(A, [m n])
```

Akan melakukan median *filtering* dengan matriks A dalam 2 dimensi. Setiap pixel outputnya merupakan harga median dari tetangga matriks  $m \times n$  terhadap pixel yang diproses. `medfilt2` akan melapisi matriks citra dengan 0 pada sudutnya, sehingga nilai median terhadap suatu titik pada  $[m \ n]/2$  dari sudut akan terlihat terdistorsi

#### e. Filter Wiener

Wiener *filter* adalah *filter* yang akan melakukan penyaringan terhadap noise yang menyerang citra yang menyebabkan penurunan kualitas citra tersebut dengan menggunakan pendekatan statistik berdasarkan pixel-pixel tetangganya.

Pada program fungsi untuk melakukan operasi penyearangan wiener ini adalah wiener 2. Wiener2 akan memperkirakan mean lokal dan variasi yang terjadi di sekeliling tiap-tiap pixel, sesuai dengan  $m \times n$  tetangga-tetangga lokal dari tiap pixel pada citra seperti ditunjukkan oleh persamaan 2.4 dan 2.5 berikut;

$$\mu = \frac{1}{NM} \sum_{n_1, n_2 \in \eta} a(n_1, n_2) \dots\dots\dots 2.4$$

$$\sigma^2 = \frac{1}{NM} \sum_{n_1, n_2 \in \eta} a^2(n_1, n_2) - \mu^2 \dots\dots\dots 2.5$$

Wiener2 kemudian akan membuat suatu perhitungan dan perkiraan pixel baru yang akan terjadi menggunakan data-data tersebut dengan menggunakan rumus seperti pada persamaan 2.6 berikut:

$$b(n_1, n_2) = \mu + \frac{\sigma^2 - v^2}{\sigma^2} (a(n_1, n_2) - \mu) \dots\dots\dots 2.6$$

Dimana  $v^2$  adalah variasi noise yang terjadi. Jikalau variasi noise tidak diberikan, wiener2 akan menggunakan rata-rata dari dari semua perkiraan variasi lokal.

#### f. Order-Statistic Filter

Pada program, *order-statistic filtering* yang direpresentasikan oleh fungsi `ordfilt2` akan melakukan penyearangan berdasarkan order-statistik.

`>> B = ordfilt2(A,order,domain)` akan menggantikan setiap elemen pada A (citra input) dengan elemen susunan sesuai set tetangga yang tersusun yang ditentukan oleh elemen bukan nol pada domain.

Kelas dari A dapat berupa *logical*, *uint8*, *uint16*, atau *double*. B juga memiliki kelas yang sama dengan A, terkecuali bentuk *additive offset* dari `ordfilt2` digunakan.

Domain ekuivalen dengan elemen penyusun yang digunakan pada operasi citra biner. Domain adalah matriks yang hanya terdiri dari nilai 0 dan 1. nilai 1 mendefinisikan lingkungan dari operasi penyearangan.

g. *Gaussian Filter*

*Filter gaussian* adalah *filter* yang menerapkan metode masking untuk menimbulkan efek seperti *low pass filter* dan berfungsi untuk menghilangkan noise serta menghaluskan citra tersebut dengan rumusan seperti pada persamaan 2.7 dan 2.8 berikut :

$$h_g(n_1, n_2) = e^{-\frac{(n_1^2 + n_2^2)}{2\sigma^2}} \dots\dots\dots 2.7$$

$$h(n_1, n_2) = \frac{h_g(n_1, n_2)}{\sum_{n_1} \sum_{n_2} h_g} \dots\dots\dots 2.8$$

Pada program, untuk melakukan filterisasi secara gaussian ini dapat dilakukan dengan fungsi `imfilter` dan `fspecial` dengan parameter gaussian `hsize` dan `sigma` (standar deviasi) yang bernilai positif. Adapun sintaksnya adalah sebagai berikut :

```
h = fspecial('gaussian',hsize,sigma)
citra_output = imfilter (citra_input,h,'replicate')
```

`h size` bisa berupa vektor yang menyatakan jumlah baris dan kolom matriks `h`, ataupun bisa berupa nilai skalar, dengan asumsi `h` adalah matriks dengan ukuran barisxkolom yang sama. Harga *default* untuk fungsi `fspecial gaussian` adalah matriks `h 3x3` dengan `sigma 0.5`. Dengan `sigma` yang kecil dan ukuran mask, citra akan diburamkan (di-blur) dengan intensitas yang ringan, sedangkan untuk intensitas yang tinggi dapat diperoleh dengan mengubah `sigma` dan mask menjadi lebih besar.

Secara umum, untuk gambar yang memiliki noise metode Gaussian merupakan langkah pertama kali yang dilakukan sebelum metode – metode atau proses – proses yang lebih jauh seperti *edge detection*.

h. *Laplacian on Gaussian (LoG) Filter*

Seperti halnya *laplace filter*, maka *filter LoG* ini juga berfungsi untuk melakukan penegasan wilayah-wilayah yang sebelumnya memiliki tekstur yang halus menjadi lebih tajam, dan *filter* ini digunakan untuk *edge detection*. Filterisasi LoG menerapkan rumusan seperti persamaan 2.9 dan 2.10 berikut dibawah kepada suatu citra input untuk menghasilkan citra output :

$$h_g(n_1, n_2) = e^{-(n_1^2 + n_2^2)/(2\sigma^2)} \dots\dots\dots 2.9$$

$$h(n_1, n_2) = \frac{(n_1^2 + n_2^2)h_g(n_1, n_2)}{2\pi\sigma^6 \sum_{n_1} \sum_{n_2} h_g} \dots\dots\dots 2.10$$

Pada program rumusan ini dijadikan sebuah sintaks dengan fungsi f spesial yaitu

```
h = fspecial('log', hsize, sigma)
```

yang akan menerapkan filterisasi LoG dengan menggunakan ukuran matrik h (hsize) yang memiliki standar deviasi sigma bernilai positif.

Nilai hsize bisa berupa sebuah vektor yang menyatakan ukuran baris dan kolom, ataupun bisa juga berupa skalar dengan asumsi akan menghasilkan matriks dengan ukuran baris dan kolom yang sama.

Nilai *default* yang digunakan oleh fungsi fspecial adalah matriks 5x5(hsize) dengan sigma sebesar 0.5.

#### i. Laplacian Filter

*Laplace* merupakan ukuran isotropic 2-D dari turunan spasial kedua dari suatu citra. Me-laplace-kan suatu citra akan menegaskan (*highlights*) wilayah-wilayah yang mengalami perubahan intensitas yang cepat dan oleh karena itu laplace ini digunakan untuk deteksi tepi (*edge detection*). Filterisasi laplace ini sering di kenakan kepada citra yang pertama-tama telah dihaluskan dengan menggunakan *filter* Gaussian yang bertujuan untuk menghilangkan noise di citra tersebut.

Filterisasi laplace terhadap sebuah citra mempunyai rumusan seperti pada persamaan 2.11 dan 2.12 sebagai berikut :

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \dots\dots\dots 2.11$$

$$\nabla^2 \approx \frac{4}{(\alpha+1)} \begin{bmatrix} \frac{\alpha}{4} & \frac{1-\alpha}{4} & \frac{\alpha}{4} \\ \frac{1-\alpha}{4} & 1 & \frac{1-\alpha}{4} \\ \frac{\alpha}{4} & \frac{1-\alpha}{4} & \frac{\alpha}{4} \end{bmatrix} \dots\dots\dots 2.12$$

Pada program, rumusan diatas diaplikasikan menggunakan fungsi fspesial dengan parameter laplacian dan alpha. Sintaksnya adalah sebagai berikut :

```
h = fspecial('laplacian',alpha)
citra_output = imfilter (citra_input,h,'replicate')
```

Rumusan diatas akan menyebabkan citra input akan di-*filter* menggunakan matriks 3x3 dengan cara memperkirakan bentuk dari operator laplace 2-dimensi. Parameter alpha akan mengontrol bentuk dari laplace dan nilai alpha tersebut harus berada pada range 0.0 s/d 1.0. dengan harga *default* yaitu 0.2.

#### j. *Filter Unsharp*

*Filter unsharp* ini merupakan “*edge enhancemen filter*“ yang akan menyebabkan citra input akan menjadi lebih jelas dan tajam tepiannya. Metode filterisasi ini biasanya dipergnakan diindustri fotografi ataupun pencetakan untuk semakin membuat tepian objek semakin “renyah”. Keuntungan dari *unsharp filter* ini adalah dapat lebih bisa dikontrol dan tidak akan menimbulkan efek buruk pada citra. *Filter* ini mempunyai rumusan seperti pada persamaan 2.13 berikut :

$$\frac{1}{(\alpha+1)} \begin{bmatrix} -\alpha & \alpha-1 & -\alpha \\ \alpha-1 & \alpha+5 & \alpha-1 \\ -\alpha & \alpha-1 & -\alpha \end{bmatrix} \dots\dots\dots 2.13$$

Pada program fungsi *filter unsharp* ini dilakukan dengan menggunakan fungsi imfilter dan fspesial dengan parameter *unsharp* dan alpha, sintaksnya adalah sebagai berikut :

```
filter = fspecial('unsharp',alpha) ;
citra_output = imfilter (citra_input,filter,'replicate')
```

Yang akan melakukan penyaringan dengan menggunakan *filter* peningkatan kontras *unsharp* dengan matriks 3x3. fspesial akan membuat *filter unsharp* dari dari citra negatif hasil *filter* laplace dengan parameter alpha. Alpha akan mengendalikan bentuk dari hasil laplace *filter* tersebut dan harus berkisa di range 0.0 s/d 1.0. Harga default untuk alpha adalah 0.2.

### k. Sobel *Filter*

Pada program untuk melakukan penyaringan dengan sobel *filter* dapat digunakan fungsi `imfilter` dan `fspesial` dengan parameter `sobel`, dengan sintaks adalah sebagai berikut :

```
Filter = fspesial('sobel')
Citra_output = imfilter(citra_input,Filter,'replicate')
```

*Filter* sobel pada fungsi `fspesial` tersebut akan melakukan *filtering* terhadap satu pixel dengan menggunakan matriks 3x3 seperti yang ditunjukkan dibawah, *filter* ini akan menegaskan tepian horisontal dengan memperkirakan kemiringan (gradien) vertikal yang ada. Jika kita ingin menegaskan tepian vertikal, maka transposekan matriks `h` dengan `h` seperti pada persamaan 2.14 berikut :

$$h = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \dots\dots\dots 2.14$$

### l. Prewitt *Filter*

Pada program untuk melakukan penyaringan dengan prewit *filter* ini, digunakan fungsi `imfilter` dan `fspesial` dengan parameter `prewitt`, dengan sintaks adalah sebagai berikut :

```
Filter = fspesial('prewitt')
Citra_output = imfilter(citra_input,Filter,'replicate')
```

*Filter* prewit pada fungsi `fspesial` tersebut akan melakukan *filtering* terhadap satu pixel dengan menggunakan matriks 3x3 seperti yang ditunjukkan dibawah, *filter* ini akan menegaskan tepian horisontal dengan memperkirakan kemiringan (gradien) vertikal yang ada. Jika kita ingin menegaskan tepian vertikal, maka transposekan matriks `h` dengan `h` seperti pada persamaan 2.15 berikut :

$$h = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \dots\dots\dots 2.15$$

## 2.2.4 Informasi Citra

Pada sebuah citra digital terdapat informasi-informasi yang perlu diketahui sebagai parameter sebuah citra. Informasi tersebut antara lain adalah informasi pixel dan histogram citra.

### a. Informasi Pixel

Informasi pixel akan memberikan keterangan tingkat keabuan dari suatu pixel penyusun citra yaitu koordinat dan tingkat intensitas atau warna dari pixel tersebut.

Dalam citra hitam putih maka informasi pixel keabuan hanya akan memberikan informasi nilai 0 atau 1. Jika citra merupakan citra RGB maka informasi pixel akan memberikan informasi tingkat warna merah, biru dan hijau pada koordinat pixel tersebut.

Dalam program informasi citra ini dapat diperoleh dengan menggunakan fungsi `pixval`.

### b. Histogram Citra

Informasi penting mengenai isi citra digital dapat diketahui dengan membuat histogram citra. Histogram citra adalah grafik yang menggambarkan penyebaran nilai – nilai intensitas pixel dari suatu citra atau bagian tertentu di dalam citra.

Dari sebuah histogram dapat diketahui frekuensi kemunculan nisbi (relative) dari intensitas pada citra tersebut. Histogram juga dapat menunjukkan banyak hal tentang kecerahan (*brightness*) dan kontras (*contrast*) dari sebuah gambar. Karena itu, histogram adalah alat bantu yang berharga dalam pekerjaan pengolahan citra baik secara kualitatif maupun kuantitatif.

Puncak histogram menunjukkan intensitas pixel yang menonjol. Lebar dari puncak menunjukkan rentang kontras dari gambar. Citra yang mempunyai kontras terlalu terang (*overexposed*) dan terlalu terang (*underexposed*) memiliki histogram yang sempit. Histogramnya terlihat hanya menggunakan setengah dari derajat keabuan secara penuh dengan distribusi yang merata pada setiap nilai intensitas pixel.

## 2.3 Pengenalan Wajah

Teknik-teknik pengenalan wajah yang dilakukan selama ini banyak yang menggunakan asumsi bahwa data wajah yang tersedia memiliki ukuran yang sama dan latar belakang yang seragam. Di dunia nyata, asumsi ini tidak selalu berlaku karena wajah dapat muncul dengan berbagai ukuran dan posisi di dalam citra dan dengan latar belakang yang bervariasi.

Pendeteksian wajah (*face detection*) adalah salah satu tahap awal yang sangat penting sebelum dilakukan proses pengenalan wajah (*face recognition*) [3]. Bidang-bidang penelitian yang berkaitan dengan pemrosesan wajah (*face processing*) adalah:

- a. Pengenalan wajah (*face recognition*) yaitu membandingkan citra wajah masukan dengan suatu database wajah dan menemukan wajah yang paling cocok dengan citra masukan tersebut.
- b. Autentikasi wajah (*face authentication*) yaitu menguji keaslian/kesamaan suatu wajah dengan data wajah yang telah diinputkan sebelumnya.
- c. Lokalisasi wajah (*face localization*) yaitu pendeteksian wajah namun dengan asumsi hanya ada satu wajah di dalam citra
- d. Penjejakan wajah (*face tracking*) yaitu memperkirakan lokasi suatu wajah di dalam video secara real time.
- e. Pengenalan ekspresi wajah (*facial expression recognition*) untuk mengenali kondisi emosi manusia.

Tantangan yang dihadapi pada masalah deteksi dan pengenalan wajah disebabkan oleh adanya faktor-faktor berikut [3]:

- a. Posisi wajah. Posisi wajah di dalam citra dapat bervariasi karena posisinya bisa tegak, miring, menoleh, atau dilihat dari samping.
- b. Komponen-komponen pada wajah yang bisa ada atau tidak ada, misalnya kumis, jenggot, dan kacamata.
- c. Ekspresi wajah. Penampilan wajah sangat dipengaruhi oleh ekspresi wajah seseorang, misalnya tersenyum, tertawa, sedih, berbicara, dan sebagainya.

- d. Terhalang objek lain. Citra wajah dapat terhalangi sebagian oleh objek atau wajah lain, misalnya pada citra berisi sekelompok orang.
- e. Kondisi pengambilan citra. Citra yang diperoleh sangat dipengaruhi oleh faktor-faktor seperti intensitas cahaya ruangan, arah sumber cahaya, dan karakteristik sensor dan lensa kamera.

Penelitian dari Yang [3] mengelompokkan metode deteksi wajah menjadi empat kategori, yaitu:

- a. ***Knowledge-based method***. Metode ini kebanyakan digunakan untuk lokalisasi wajah.
- b. ***Feature invariant approach***. Metode ini kebanyakan digunakan untuk lokalisasi wajah.
- c. ***Template matching method***. Metode ini digunakan untuk lokalisasi wajah maupun deteksi wajah.
- d. ***Appearance-based method***. Metode ini kebanyakan digunakan untuk deteksi wajah.

#### 2.4 Template Matching

Cara kerja metode *Template Matching* ini adalah melakukan *pattern recognition* pada karakter atau object yang ingin dikenali dan membandingkan antara *input pattern* dengan *template* yang disimpan [4].

*Template matching* yang disebut juga dengan *Pattern Matching* atau Matrix Matching merupakan suatu proses membandingkan suatu obyek karakter yang biasanya disebut sebagai glyph dengan sejumlah *template*. Glyph yang telah diekstrak akan dicocokkan dengan *template* yang telah disimpan dalam database *template*. Metode-metode *Template matching* yang digunakan meliputi XOR , WXOR, WAN dan CTM. Masing-masing metode memiliki satu atau dua parameter yang digunakan dalam pencocokan, dimana prosesnya memerlukan suatu *threshold* yang tepat. *Threshold* digunakan untuk menentukan batasan suatu glyph, apakah cocok atau berbeda dengan *template* yang sedang dibandingkan.

Induksi Decision Tree (ID3) dari machine learning digunakan untuk menentukan *threshold* dan urutan dari masing-masing metode secara otomatis. Metode ini disebut sebagai *Combined Template Matching Method*, yang mengkombinasikan *threshold* dan urutan pemakaian metode-metode di atas [5].

## 2.5 Jaringan Syaraf Tiruan

**Jaringan saraf tiruan (JST)** dalam Bahasa Inggris: *artificial neural network (ANN)*, atau juga disebut *simulated neural network (SNN)*, atau umumnya hanya disebut *neural network (NN)*, adalah jaringan dari sekelompok unit pemroses kecil yang dimodelkan berdasarkan jaringan saraf manusia. Secara sederhana, JST adalah sebuah alat pemodelan data statistik non-linier. JST dapat digunakan untuk memodelkan hubungan yang kompleks antara input dan output untuk menemukan pola-pola pada data [6].

Jaringan saraf tiruan (JST) merupakan salah satu algoritma pembelajaran mesin yang meniru cara kerja jaringan saraf makhluk hidup. Jaringan saraf tiruan (*artificial neural network*) merupakan jaringan dari banyak unit pemroses kecil (yang disebut neuron) yang masing-masing melakukan proses sederhana, yang ketika digabungkan akan menghasilkan perilaku yang kompleks. Jaringan saraf tiruan dapat digunakan sebagai alat untuk memodelkan hubungan yang kompleks antara masukan (*input*) dan keluaran (*output*) pada sebuah sistem untuk menemukan pola-pola pada data [7].

Jaringan saraf tiruan dibuat berdasarkan model biologis otak manusia. Kemampuan komputer sudah melampaui otak manusia dalam hal komputasi numerik, tetapi otak manusia dapat mengerjakan persoalan lainnya secara lebih cepat dan akurat, misalnya pada persoalan pengenalan wajah, persoalan klasifikasi, dan persoalan penarikan keputusan. Oleh karena itu, dilakukanlah riset yang mencoba memodelkan proses yang terjadi di otak manusia. Riset-riset tersebut menghasilkan sebuah model matematis yang disebut jaringan saraf tiruan (*artificial neural network*) atau sering juga disebut *simulated neural network* atau hanya jaringan saraf (*neural network*).

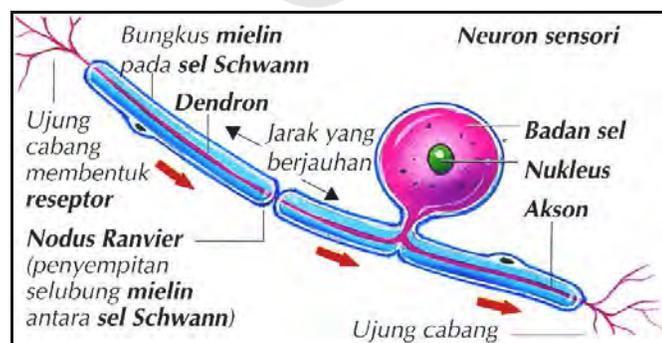
Otak manusia terdiri dari lebih dari  $10^{11}$  sel saraf yang disebut neuron [1]. Masing-masing sel saraf ini terdiri atas empat komponen dasar, yaitu nukleus (inti), dendrit, soma (badan sel) dan akson. Terdapat tiga macam neuron, yaitu neuron sensori, asosiasi dan motor.

Masing-masing neuron terhubung dengan 200.000 hubungan melalui dendrit ke neuron lain pada otak. Dendrit menerima impuls dari neuron lain atau dari reseptor (bagian dari sistem pengindra yang mengirimkan impuls ketika dirangsang).

Sebagian besar neuron memiliki beberapa dendrit pendek, kecuali neuron sensori yang hanya memiliki satu dendrit panjang. Dendrit panjang tersebut disebut dendron. Badan sel adalah bagian dari neuron yang di dalamnya terdapat nukleus (inti sel) dan sebagian besar sitoplasma sel.

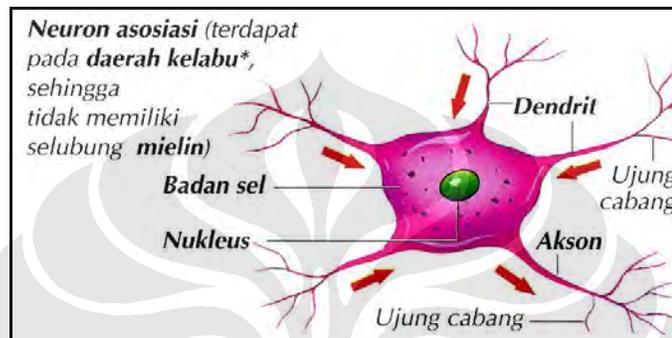
Masing-masing neuron mempunyai akson, yaitu serabut saraf tunggal panjang yang membawa impuls dari badan sel. Akson akan menyampaikan impuls tersebut ke efektor (otot atau kelenjar) atau ke dendrit dari neuron lain. Daerah sempit tempat bertemunya akson dengan dendrit neuron lain disebut sinapsis. Pada celah sinapsis ini, impuls diteruskan ke dendrit sel lain dengan menggunakan zat kimia yang disebut neurotransmitter.

Neuron sensori atau disebut juga neuron aferen merupakan neuron yang membawa impuls saraf. Ujung-ujung dendron (dendrit panjang) beberapa neuron sensori membentuk reseptor-reseptor di seluruh tubuh, yang mengirimkan impuls ke neuron ketika dirangsang. Kemudian reseptor akan terhubung dan mengantarkan impuls ke otak manusia melalui beberapa neuron sensori. Neuron sensori pada manusia ditunjukkan pada Gambar 2.5 berikut



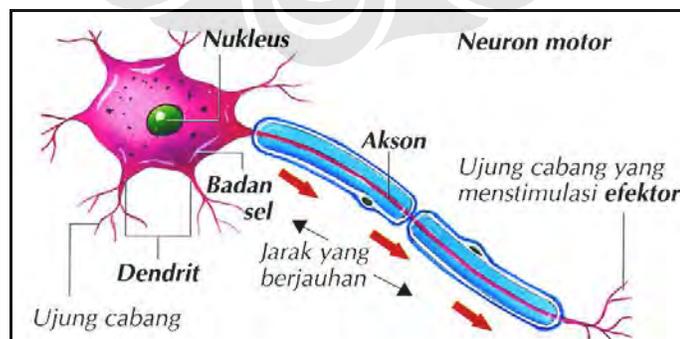
Gambar 2.5 Neuron Sensori [7]

Neuron asosiasi, yang disebut juga neuron penghubung atau interneuron merupakan neuron dengan jumlah paling banyak di otak dan sumsum tulang belakang. Neuron asosiasi ini terlibat dalam penerimaan input dari neuron sensori, penerjemahan input tersebut menjadi informasi serta penyampaian impuls ke neuron motor untuk melakukan gerakan/aksi. Neuron asosiasi pada manusia ditunjukkan pada Gambar 2.6 berikut



**Gambar 2.6** Neuron Asosiasi [7]

Neuron motor, disebut juga neuron aferen merupakan neuron yang membawa impuls saraf dari otak dan sumsum tulang belakang. Ujung akson neuron motor membentuk sambungan dengan otot atau kelenjar. Impuls yang dibawa oleh neuron motor dari neuron asosiasi akan merangsang organ-organ tubuh ini untuk bekerja. Neuron motor pada manusia ditunjukkan pada Gambar 2.7 berikut

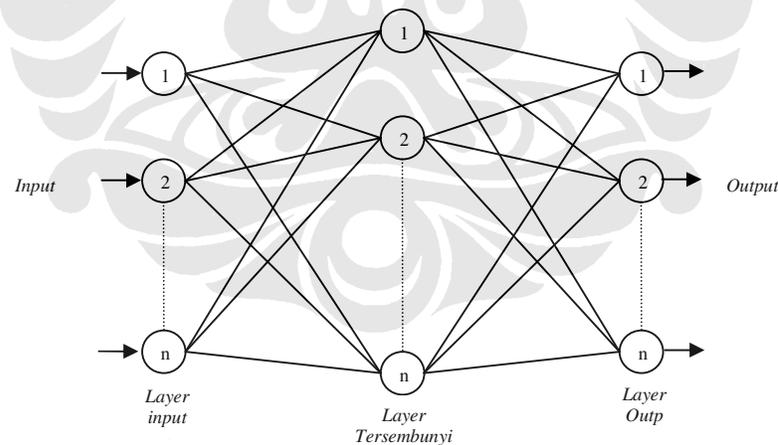


**Gambar 2.7** Neuron Motor [7]

Otak manusia bekerja dengan meneruskan impuls yang didapat dari neuron sensori ke neuron asosiasi dan kemudian menuju ke neuron motor. Hubungan antara neuron neuron yang sangat banyak itu dapat menguatkan ataupun melemahkan, bahkan dapat terbentuk hubungan baru ataupun putus hubungan antara neruon-neuron tersebut, sesuai dengan aktivitas pada otak manusia.

Analogi JST ini seperti pada awal pertumbuhan anak, perubahan yang terjadi pada keterhubungan neuron tersebut sangat banyak, seiring dengan pembelajaran yang dilakukan anak terhadap lingkungan sekitarnya. Saat manusia berkembang menjadi dewasa, hubungan antar neuron-neuron pada otaknya telah mampu membuatnya dapat berjalan, membaca, mendengar, mengambil keputusan, dan hal lain yang dapat dilakukan oleh manusia dewasa.

Sebuah jaringan saraf tiruan terdiri dari banyak neuron yang saling terhubung dalam jaringan yang rumit. Pada jaringan saraf tiruan juga terdapat tiga jenis neuron, yaitu neuron masukan, neuron tersembunyi, serta neuron keluaran. Cara kerja atau alur dari jaringan saraf tiruan ini bekerja menirukan jaringan saraf manusia digambarkan pada diagram pada Gambar 2.8 berikut.



**Gambar 2.8** Model Generik Multilayer Perception. [3]

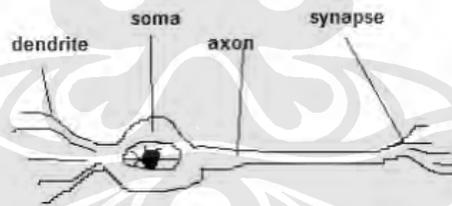
Neuron masukan merupakan neuron yang berfungsi seperti neuron sensori, yaitu menerima nilai masukan. Biasanya sebuah neuron masukan hanya merepresentasikan sebuah input ke jaringan saraf tiruan. Tugas utama neuron masukan adalah memastikan bahwa nilai masukan yang masuk pada jaringan

saraf tiruan merupakan nilai yang dapat diterima dan diproses oleh neuron lainnya, baik neuron tersembunyi maupun neuron keluaran.

Neuron tersembunyi berfungsi seperti neuron asosiasi. Neuron jenis ini berfungsi menerima nilai masukan dari neuron-neuron masukan atau neuron tersembunyi lainnya, memproses nilai-nilai masukan tersebut, dan kemudian meneruskan hasilnya ke neuron keluaran atau ke neuron tersembunyi lainnya. Kemampuan jaringan saraf tiruan untuk mengaproksimasi berbagai macam fungsi target sebagian besar disumbang oleh keberadaan neuron tersembunyi ini.

Neuron keluaran merupakan neuron yang memiliki fungsi seperti neuron motor. Neuron jenis ini berfungsi menerima output dari neuron-neuron tersembunyi, kemudian memprosesnya sehingga menghasilkan nilai keluaran. Nilai yang dikeluarkan oleh neuron keluaran menjadi nilai output dari jaringan saraf tiruan. Nilai keluaran tersebut dapat disimpan atau dapat juga dihubungkan dengan sistem lain yang akan melakukan aksi tertentu.

Proses pembelajaran yang dilakukan terhadap jaringan saraf tiruan yang terdiri dari banyak neuron merupakan pengembangan dari pembelajaran satu neuron. Gambaran umum neuron penyusun saraf manusia ditunjukkan pada Gambar 2.9 sebagai berikut :



**Gambar 2.9** Neuron pada saraf manusia.

Empat komponen utama dari *neuron* pada saraf manusia adalah [7]

a. *Dendrite*

berbentuk seperti rambut, dan merupakan perpanjangan dari *neuron*. Setiap dendrit dapat membawa *input* ke *neuron* yang berasal dari *layer* sebelumnya. *Input* tersebut diberikan kepada *soma*.

b. *Soma*

berfungsi untuk memproses *input*, dan *output* diberikan ke *neuron* lain melalui *axon* dan *synapsis*.

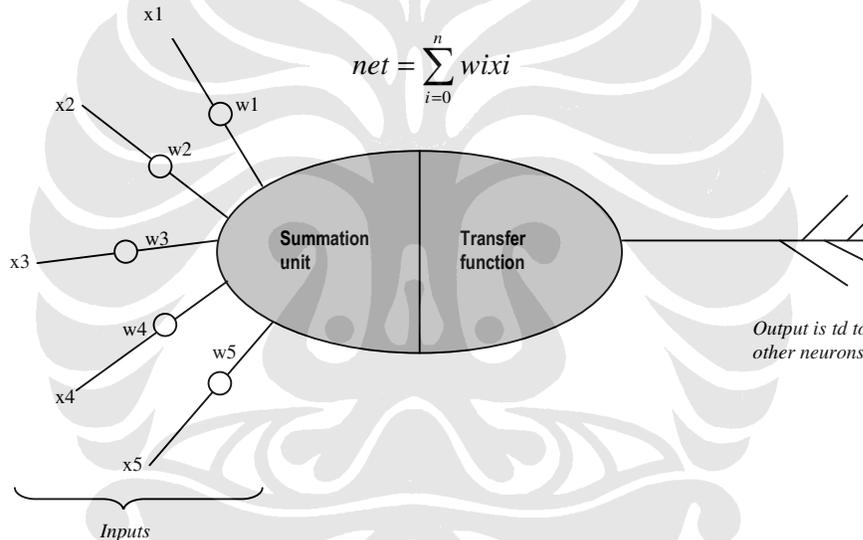
c. *Axon*

berfungsi untuk membawa *output* dari *soma* ke *neuron* lain melalui *synapsis*.

d. *Synapsis*

terhubung dengan dendrit dari *neuron* lain pada *layer* berikutnya. Koneksi antar *neuron* terjadi akibat adanya *synapse* dan *dendrite*

Sebuah *neuron* akan terhubung dengan *hampir* semua *neuron* yang ada di *layer* berikutnya dan dapat menerima *input* lebih dari satu *neuron* dari *layer* sebelumnya. Diagram dari sebuah *neuron* pada JST ditunjukkan pada Gambar 2.10 sebagai berikut



**Gambar 2.10** Neuron pada JST [8]

JST memiliki berbagai macam *input*, dan sebagai pengganti *soma* dan *axon*, maka JST memiliki *summation unit* dan *transfer function unit*. *Output* dari sebuah *neuron* dapat diberikan sebagai *input* ke banyak *neuron*. Setiap *input* memiliki bobot masing-masing.

*Summation Unit* berfungsi sama seperti *soma*, yang berfungsi untuk memproses *input*. Hal yang pertama kali dilakukan adalah menghitung *net value*, yaitu jumlah dari *setiap* nilai *input* dikalikan dengan bobotnya. *Neuron* juga memiliki bias yang ditetapkan secara acak pada saat inisialisasi JST. Secara garis

besar, JST akan merubah bobot dan bias selama fase *training* sehingga JST dapat berfungsi secara benar.

*Transfer Function Unit* adalah sebuah fungsi sederhana untuk menghasilkan *output* dari *net value* serta membawa *output* ke *neuron* di *layer* berikutnya.

Ada beberapa jenis *transfer function*, diantaranya sebagai berikut :

a. *Hard Limit / Step Transfer Function*

Fungsi yang akan menghasilkan *output* 1 atau 0 berdasarkan kondisi tertentu dari *net value*. Gambar 2.11 berikut menunjukkan fungsi *Step Transfer*.



**Gambar 2.11** *Step Transfer Function*

b. *Sigmoid Transfer Function*

Fungsi ini akan mengambil *net value* sebagai *input* dan menghasilkan *output* antara 0 dan 1. Gambar 2.11 berikut menunjukkan *Sigmoid Transfer Function*.



**Gambar 2.12** *Sigmoid Transfer Function*

### 2.5.1 Supervised Learning

Supervised learning atau belajar arahan bertujuan untuk menentukan nilai bobot-bobot koneksi di dalam jaringan sehingga jaringan dapat melakukan pemetaan (*mapping*) dari input ke output sesuai dengan yang diinginkan. Pemetaan ini ditentukan melalui satu set pola contoh atau data pelatihan (*training data set*).

Setiap pasangan pola  $p$  terdiri dari vektor input  $x_p$  dan vektor target  $t_p$ . Setelah selesai pelatihan, jika diberikan masukan  $x_p$  seharusnya jaringan menghasilkan nilai output  $t_p$ . Besarnya perbedaan antara nilai vektor target dengan output aktual diukur dengan nilai error yang disebut juga dengan *cost function* seperti yang ditunjukkan pada Persamaan 2.16 di mana  $n$  adalah banyaknya unit pada *output layer*.

$$E = \frac{1}{2} \sum_{p \in P} \sum_n (t_n^p - s_n^p)^2 \dots\dots\dots 2.16$$

Tujuan dari *training* ini pada dasarnya sama dengan mencari suatu nilai minimum global dari  $E$ .

### 2.5.2 Multi-Layer Perceptron

Multi-Layer Perceptron adalah jaringan syaraf tiruan *feed-forward* yang terdiri dari sejumlah neuron yang dihubungkan oleh bobot-bobot penghubung. Neuron-neuron tersebut disusun dalam lapisan-lapisan yang terdiri dari satu lapisan input (*input layer*), satu atau lebih lapisan tersembunyi (*hidden layer*), dan satu lapisan output (*output layer*). Lapisan input menerima sinyal dari luar, kemudian melewatkannya ke lapisan tersembunyi pertama, yang akan diteruskan sehingga akhirnya mencapai lapisan output.

Setiap neuron  $i$  di dalam jaringan adalah sebuah unit pemrosesan sederhana yang menghitung nilai aktivasinya yaitu  $s_i$  terhadap input eksitasi yang juga disebut net input  $net_i$ . Hal ini dipresentasikan oleh Persamaan 2.17 sebagai berikut

$$net_i = \sum_{j \in pred(i)} s_j w_{ij} - \theta_i \dots\dots\dots 2.17$$

dimana  $pred(i)$  melambangkan himpunan predesesor dari unit  $i$ ,  $w_{ij}$  melambangkan bobot koneksi dari unit  $j$  ke unit  $i$ , dan  $\theta_i$  adalah nilai bias dari unit  $i$ . Untuk membuat representasi menjadi lebih mudah, seringkali bias digantikan dengan suatu bobot yang terhubung dengan unit bernilai 1. Dengan demikian bias dapat diperlakukan secara sama dengan bobot koneksi.

Aktivasi dari unit  $i$ , yaitu  $s_i$ , dihitung dengan memasukkan net input ke dalam sebuah fungsi aktivasi non-linear. Biasanya digunakan fungsi logistik sigmoid dan ditunjukkan oleh Persamaan 2.18 berikut:

$$s_i = f_{\log}(net_i) = \frac{1}{1 + e^{-net_i}} \dots\dots\dots 2.18$$

Salah satu keuntungan dari fungsi ini adalah memiliki derivatif yang mudah dihitung dengan Persamaan 2.19 berikut:

$$\frac{\partial s_i}{\partial net_i} = f'_{\log}(net_i) = s_i * (1 - s_i) \dots\dots\dots 2.19$$

Nilai dari fungsi sigmoid di atas memiliki nilai output antara 0 dan 1. Jika diinginkan nilai output antara  $-1$  dan  $1$ , dapat digunakan fungsi bipolar sigmoid seperti pada Persamaan 2.20 berikut:

$$s_i = g_{\log}(net_i) = \frac{2}{1 + e^{-net_i}} - 1 \dots\dots\dots 2.20$$

Derivatif dari fungsi tersebut adalah seperti ditunjukkan pada Persamaan 2.21 berikut:

$$\frac{\partial s_i}{\partial net_i} g'_{\log}(net_i) = \frac{1}{2} (1 + s_i) * (1 - s_i) \dots\dots\dots 2.21$$

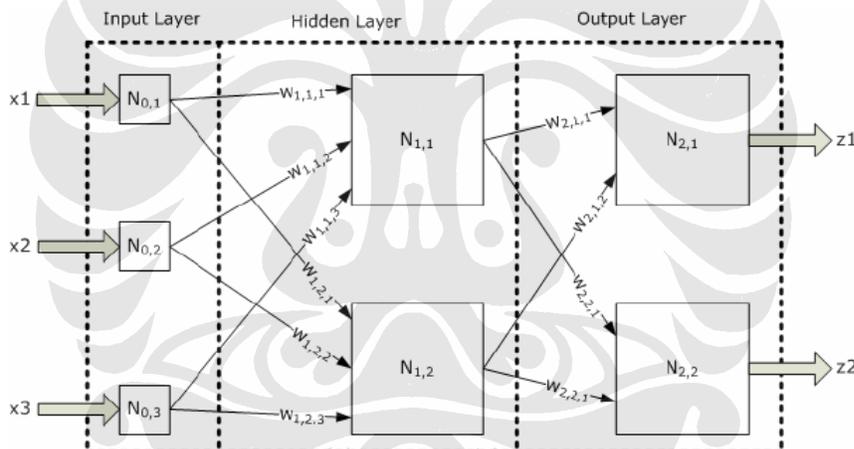
### 2.5.3 Algoritma Backpropagation

Algoritma ini dibuat dengan menggeneralisasi Widrow-Hoff *learning rule* untuk *multiple-layer network* dan fungsi *transfer* differensial nonlinear. Vektor *input* dan vektor target digunakan untuk melatih JST hingga dapat menghasilkan sebuah fungsi yang diinginkan.

Backpropagation yang standar adalah sebuah algoritma gradient descent, sama seperti di Widrow Hoff, dimana bobot diubah sesuai dengan gradient negatif dari fungsi. Dinamakan backpropagation karena mengacu pada bagaimana gradient dihitung untuk *nonlinear multilayer network*.

Jumlah *neuron* pada *input layer* tergantung dengan jumlah *input* yang kita miliki, sedangkan jumlah *neuron* pada *output layer* tergantung pada jumlah *output* yang kita inginkan. Jumlah *hidden layer* serta jumlah *neuron* pada *hidden layer* tidak dapat ditentukan secara pasti berapa yang paling optimal, namun yang pasti adalah dengan menambah jumlah *hidden layer* akan memungkinkan JST untuk mempelajari *pattern* yang lebih kompleks, namun akan menurunkan kinerjanya. Kita dapat menggunakan 1 *hidden layer* pada saat pertama kali, dan menambah jumlahnya apabila JST tidak bekerja sebagaimana kita inginkan. Dengan kata lain, kita dapat merubah konfigurasi dari JST untuk mendapat hasil yang paling optimal.

Sebagai contoh bagaimana backpropagation bekerja, maka digunakan contoh sebuah JST yang memiliki 3 *neuron* di *input layer*, 2 *neuron* di *hidden layer*, dan 2 *neuron* di *output layer* seperti Gambar 2.13 di bawah ini:



**Gambar 2.13** Algoritma Propagasi Balik [3]

Proses kerja dari backpropagation adalah [8]:

- Inisialisasi bobot tiap *neuron* dengan nilai acak antara 0 sampai 1
- Masukkan vektor *input* pada *input layer* JST
- Hitung *output*
- Bandingkan antara *output* yang dihasilkan dengan target yang diinginkan. Selisihnya dinamakan *error*.
- Ubah bobot dan bias pada *setiap neuron* untuk meminimalisasi *error*

- f. Ulang proses sampai *error* turun hingga nilai yang kita tentukan (*training* sukses) atau hingga iterasi maksimum (*epoch*) tercapai yang menandakan *training* JST tersebut gagal.

Kesulitan utama dari algoritma ini adalah mengubah bobot dan bias pada setiap iterasi untuk menurunkan tingkat *error* yang terjadi.

Proses perubahan bobot dan bias untuk *neuron* di *hidden layer* berbeda dengan *neuron* pada *output layer*. Ada 2 besaran yang digunakan dalam proses *training* tersebut :

- a.  $\lambda$  (*Lambda*) / *Learning Rate* : konstanta untuk *neuron* (biasanya 0.2 untuk *neuron* di *output layer* dan 0.15 untuk *neuron* di *hidden layer*).
- b.  $\Delta$  (*Delta*) : menunjukkan selisih atau perubahan nilai yang terjadi.

## 2.6 Sistem Autentikasi

Saat ini, autentikasi dengan menggunakan *password* adalah teknik yang paling umum digunakan pada berbagai layanan untuk membuktikan atau mengkonfirmasi bahwa identitas seseorang adalah benar-benar milik orang yang sah. Namun, jika metode ini digunakan pada saluran komunikasi yang tidak aman, metode ini rentan terhadap berbagai serangan, misalnya penyadapan *password* oleh pihak yang tidak berhak. Keamanan data merupakan bagian yang tak terhindarkan pada kehidupan sehari-hari saat ini. Untuk itu, setiap orang berupaya untuk melindungi datanya dengan berbagai cara. Salah satu teknik perlindungan data adalah dengan menggunakan autentikasi terhadap pengguna. Dengan menggunakan autentikasi, maka identitas pengguna dapat diketahui, sehingga sistem dapat menentukan hak akses yang sesuai bagi pengguna tersebut.

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. Pada suatu sistem komputer, autentikasi biasanya

terjadi pada saat login atau permintaan akses. Dikenal pula istilah otorisasi, yaitu proses untuk memverifikasi bahwa seseorang atau sesuatu memiliki wewenang untuk melakukan suatu aksi atau kegiatan.

Otorisasi biasanya didahului dengan autentikasi. Contoh yang paling nyata dari autentikasi adalah untuk keperluan kontrol akses (*access control*). Sebuah sistem komputer biasanya hanya diizinkan untuk diakses oleh pihak yang berwenang, namun tidak diizinkan kepada pihak lain. Sehingga, akses kepada sistem biasanya diawali dengan prosedur autentikasi untuk menentukan identitas seorang pengguna. Kemudian baru dapat dilakukan pemberian hak akses kepada pengguna yang telah terotorisasi.

Contoh yang sering ditemukan pada kehidupan sehari-hari adalah sebagai berikut:

- a. Login ke komputer lain melewati jaringan internet
- b. Mengambil uang dari ATM
- c. Menggunakan *internet banking*

Autentikasi yang dianggap kuat (*strong authentication*) didefinisikan sebagai pendekatan autentikasi yang berlapis-lapis dan tergantung kepada dua atau lebih faktor autentikasi untuk menentukan identitas penerima atau pengirim informasi.

### 2.6.1 Faktor Autentikasi

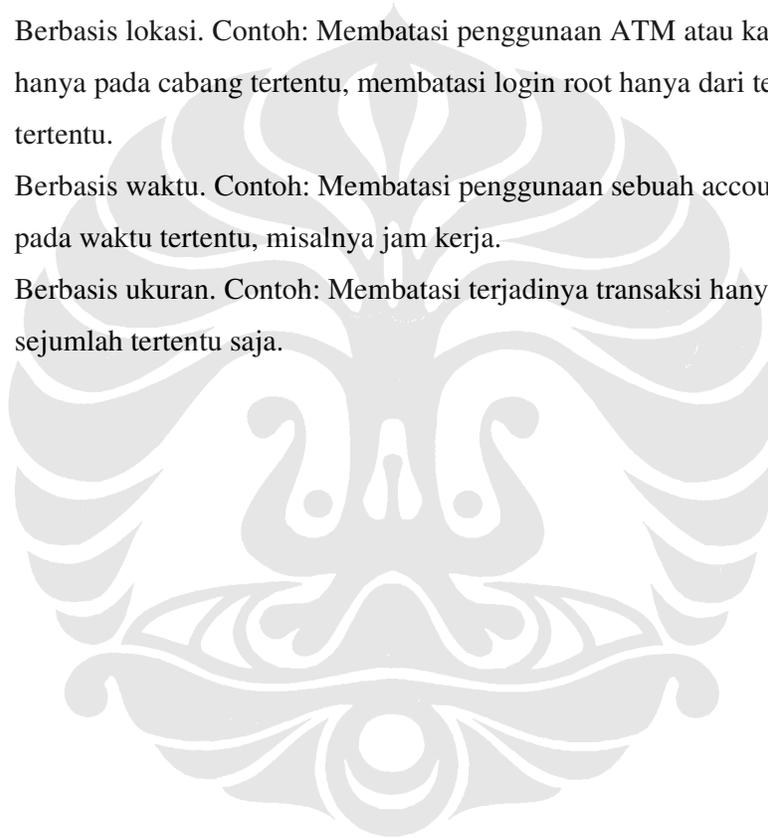
Faktor autentikasi adalah sebuah informasi yang digunakan untuk memverifikasi identitas seseorang untuk kepentingan keamanan.

Tiga jenis faktor autentikasi yang umum digunakan adalah:

- a. Sesuatu yang diketahui oleh pengguna  
Contoh: *password*, *passphrase*, dan PIN (*Personal Identification Number*)
- b. Sesuatu yang dimiliki oleh pengguna  
Contoh: *ID card*, kartu kredit, telepon seluler, dan perangkat token
- c. Sesuatu yang 'ada' pada pengguna  
Contoh: sidik jari, DNA, suara, pola retina, atau aspek biometrik lain.

Sedangkan, beberapa faktor autentikasi lain yang lebih jarang digunakan adalah:

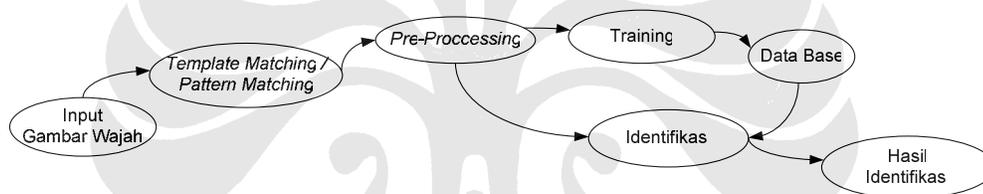
- a. Berbasis pengenalan (*recognition*) atau autentikasi *cognometric*, yaitu sesuatu yang dikenal oleh pengguna. Contoh: Pengguna harus mengenali dari beberapa wajah yang dirahasiakan.
- b. Berbasis *cybermetric*, yaitu sesuai yang ada pada komputer. Contoh: Membatasi akses hanya dari komputer yang memiliki kombinasi unik hardware dan software tertentu.
- c. Berbasis lokasi. Contoh: Membatasi penggunaan ATM atau kartu kredit hanya pada cabang tertentu, membatasi login root hanya dari terminal tertentu.
- d. Berbasis waktu. Contoh: Membatasi penggunaan sebuah account hanya pada waktu tertentu, misalnya jam kerja.
- e. Berbasis ukuran. Contoh: Membatasi terjadinya transaksi hanya pada sejumlah tertentu saja.



## BAB 3

### PERANCANGAN SISTEM

Sistem pengenalan wajah ini di desain untuk identifikasi wajah untuk sistem autentikasi yang dibuat berdasarkan penggabungan dua metode yakni metode jaringan saraf tiruan (JST) atau *neural network* dan metode pencocokan pola atau *Template matching*, biasa disebut juga *pattern matching*. Dengan alat yang digunakan sebagai sensor untuk mengumpulkan data-data gambar wajah adalah webcam. Penggabungan kedua metode tersebut dapat dilihat pada diagram rancangan sistem yang ditunjukkan pada Gambar 3.1.

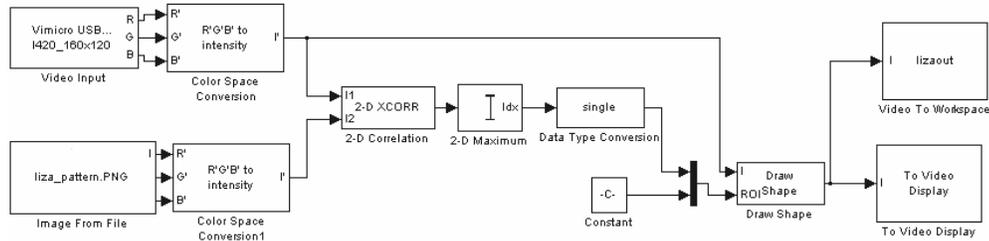


Gambar 3.1 Diagram Sistem Identifikasi Wajah

Seperti yang terlihat pada diagra di atas, sistem Identifikasi wajah ini dibuat dengan menggabungkan dua buah metode yang berbeda, yakni *pattern matching* dan *neural network* dengan tujuan membuat sistem menjadi lebih presisi. Karena output dari metode pencocokan pola pada sistem pendeteksi wajah akan menjadi input dari sistem pengenalan wajah nantinya. Sehingga pengenalan akan lebih fokus pada citra wajah saja tanpa harus memperhitungkan *background* citra wajah.

#### 3.1 Pendeteksi Wajah.

Pendeteksi wajah ini menggunakan metode pencocokan pola atau *pattern matching* dibangun dengan blok set simulink. Susunan blok set pada program untuk dapat melakukan pengenalan wajah dapat dilihat pada Gambar 3.2 berikut.



Gambar 3.2. Diagram Blok Sistem Pendeteksi Wajah

Sistem pendeteksi wajah ini terdiri dari beberapa blok set yang diambil dari simulink dan disusun berdasarkan contoh pelacakan objek menggunakan blok korelasi. Namun secara garis besar terdapat 3 blok set yang mempunyai peranan penting untuk mendukung sistem Pendeteksi Wajah ini.

a. Parameter Blok fungsi: Korelasi 2 Dimensi

Korelasi yang dimaksudkan disini adalah korelasi antara dua buah citra dua dimensi. Korelasi inilah inti dari metode pencocokan pola atau *pattern matching*. Pada saat citra masukan dikorelasikan dengan citra pola atau *template* wajah, maka matriks pada gambar akan dicocokkan sesuai dengan matriks pola yang ada. Blok korelasi 2 dimensi ini ditunjukkan pada Gambar 3.3 berikut.



Gambar 3.3 Blok korelasi 2 dimensi

b. Blok Maksimum

Untuk menemukan indeks nilai maksimum pada tiap output matriks citra 2 dimensi dari proses korelasi. Dengan demikian akan lebih memudahkan sistem mencocokkan citra input dengan citra pola wajah. Pengesetan dilakukan di *Mode parameter* menjadi "index". Lambang bloksetnya terlihat pada Gambar 3.4 berikut.



Gambar 3.4 Maximum block

c. *Draw Shapes*

Penggunaan blok set *draw shape* ini adalah untuk menggambarkan persegi empat yang akan bergerak mengarah ke matriks yang paling sesuai dengan matriks pattern yang masukan. Peranan *Drawshape* inilah yang menyetting sebesar apa kotak yang akan dimunculkan pada video display. Hal ini juga menentukan seberapa jauh jarak wajah dari kamera yang menghadap ke wajah. Lambangnya diperlihatkan pada Gambar 3.5 sebagai berikut.



Gambar 3.5. *Draw Shape block*

Sistem pendeteksi wajah adalah sebuah sistem yang dirancang untuk mengklasifikasikan mana yang bagian wajah dan mana yang bagian bukan wajah. Dengan demikian pengambilan citra wajah akan lebih fokus, tanpa harus memperhitungkan *background* atau latar belakang citra digital. Tantangan yang diperoleh pada sistem pengenalan wajah ini adalah pengambilan input citra digital secara langsung atau *real time* melalui kamera webcam. Dengan demikian pendeteksi wajah diharuskan dapat mengikuti gerak dari objek yang terdapat pada input gambar video.

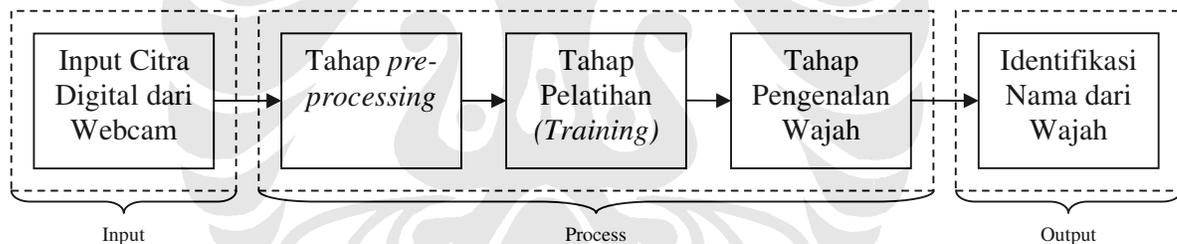
Dengan metode *pattern matching* atau pencocokan pola, data input yang berupa citra digital video, akan dikorelasikan dengan sebuah citra wajah sebagai pola atau *template* yang mencirikan bentuk wajah seseorang itu seperti apa. Permasalahannya adalah proses korelasi yang terjadi tidaklah semudah membandingkan antara dua buah citra digital dan menemukan titik-titik pixel yang sama diantara kedua buah citra tersebut, karena dengan perbandingan saja tidak akan cukup. Disinilah peran dari blok set maksimum yang mampu mengolah nilai – nilai pada matriks citra digital sehingga dapat menonjolkan nilai-nilai maksimum dari hasil korelasi.

Pengklasifikasian wajah dan non wajah nantinya akan ditandai dengan sebuah kotak yang akan mengarah ke citra wajah, jika pada citra digital masukan

terdapat gambar wajah atau ayng mirip dengan wajah. Kotak tersebut berasal atau di-generate oleh blok *draw shape*.

### 3.2 Pengenalan Wajah

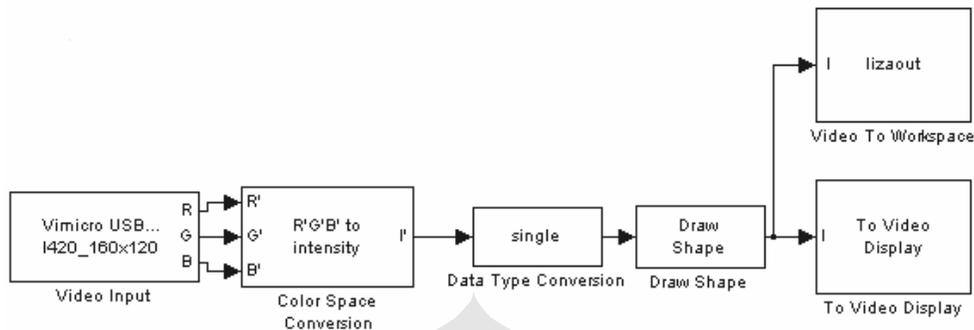
Pada setiap sistem pada umumnya terdiri dari 3 bagian utama; bagian input, bagian proses dan bagian ouput (hasil). Bagian input adalah keluaran dari sistem deteksi wajah. Masuk kepada bagian proses, secara garis besar, proses pengenalan wajah ini dirancang menjadi 3 tahap yaitu proses pengolahan citra, *Training*, dan proses Identifikasi seperti yang ditunjukkan pada Gambar 3.6. Tahap pertama, pengolahan citra, adalah pada saat pertama kali data diambil menggunakan webcam. Setelah itu masuk ke tahap kedua, yakni data-data yang sudah ada dilatih (*training*) sesuai dengan parameter-parameter yang sudah ditentukan. Tahap ketiga barulah proses identifikasi atau pengenalan dilakukan.



Gambar 3.6. Diagram Blok Sistem Pengenalan Wajah

Input untuk sistem ini adalah citra digital yang diambil dari webcam dan dimasukkan kedalam proses pengambilan gambar wajah menggunakan blokset simulink sebagai alat bantu mensimulasikan pengambilan citra dari webcam secara *real time* agar bisa dikenali oleh sistem pengenalan wajah. Pada sistem pendeteksi wajah menggunakan *pattern matching* kotak yang dibuat bergerak dinamis mengikurti arahnya posisi wajah. Pada saat pengambilan wajah pertama kali untuk data training diperlukan pengambilan wajah yang tepat sesuai dengan ukuran citra wajah yang diinginkan. Oleh karena itu beberapa blokset dari sistem pendeteksi wajah digunakan untuk mengambil input citra wajah, hanya saja pada

kali ini pengguna yang mengarahkan wajahnya kedalam kotak yang statis. Adapun susunan blok set-nya adalah seperti Gambar 3.7 berikut.



Gambar 3.7. Diagram Blok Proses Input Wajah Dari Webcam dengan Kotak pembatas.

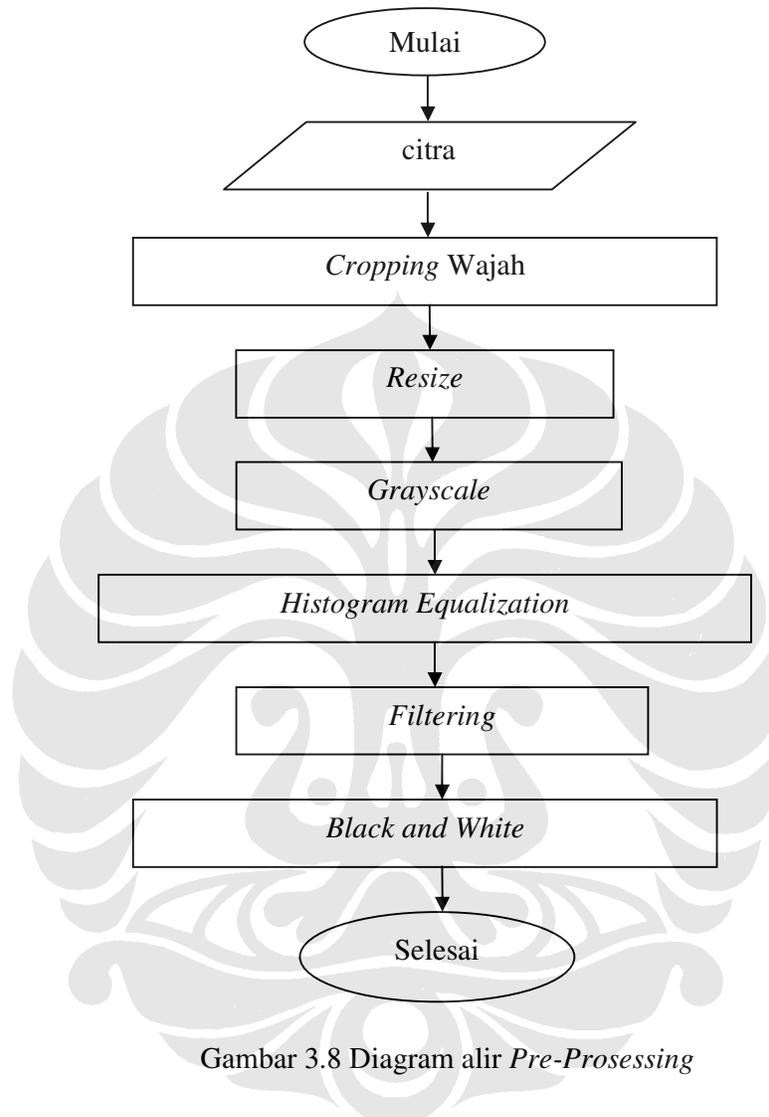
Seperti yang dibahas pada paragraf sebelumnya, pada proses memasukan input wajah secara realtime dari langsung dari webcam ini adalah sebagian dari proses pendeteksi wajah menggunakan metode *pattern matching* atau pencocokan pola. Hanya saja pada proses input dengan hanya menggunakan kotak pembatas ini, tidak lagi menggunakan blokset korelasi dua dimensi dan blok maksimum. Jadi hasilnya hanya gambar input dari webcam dengan kotak (sebagai pembatas) agar user atau pengguna sistem dapat menyesuaikan atau menempatkan wajah di tempat yang telah di set untuk proses cropping, proses cropping ini akan dijelaskan pada subbab *pre-processing*. Dengan demikian hanya bagian wajah atau citra yang diperlukan saja yang diambil untuk di proses di tahapan selanjutnya dan tidak perlu memperhitungkan *background* citra. Sehingga perhitungan algoritma jaringan saraf tiruan diharapkan akan lebih akurat.

### 3.2.1 Pengolahan Citra (*Pre-processing*)

Data gambar wajah yang diambil dari webcam perlu di proses lebih lanjut untuk menyeragamkan data yang kelak akan menjadi data latih untuk proses pelatihan atau *training*.

Adapun *pre-processing* pengolahan citra sebelum data dilatih ada 6 proses, ke-enam proses tersebut adalah sebagai berikut; *cropping*, *resizing*, *gray scaling*,

*histogram equalization, filtering, dan terakhir black and white scale.* Proses ini dapat dilihat secara garis besar pada diagram alir Gambar 3.8 berikut.



Gambar 3.8 Diagram alir *Pre-Processing*

a. *Cropping* - pemotongan gambar.

Pemotongan gambar atau *cropping* adalah suatu proses untuk membersihkan bagian sebelah luar gambar untuk memperbaiki atau membingkai kembali suatu citra atau gambar digital. *Cropping* ini diperlukan untuk memisahkan bagian wajah saja, sehingga gambar latar belakang tidak mempengaruhi proses pengenalan. Pada sistem pengenalan wajah yang saya buat ini *Cropping* atau pemotongan gambar ini dilakukan secara otomatis dengan menggunakan bantuan blok set simulink seperti

yang digambarkan pada Gambar 3.7. Gambar 3.9 berikut ini adalah gambar hasil dari pemotongan secara otomatis ke ukuran 120 x 110 dari koordinat 110 x 100 dengan perintah pada program sebagai berikut;

```
I2 = imcrop(frame,[100 110 110 120]);  
figure;  
imshow(I2);
```



(a)

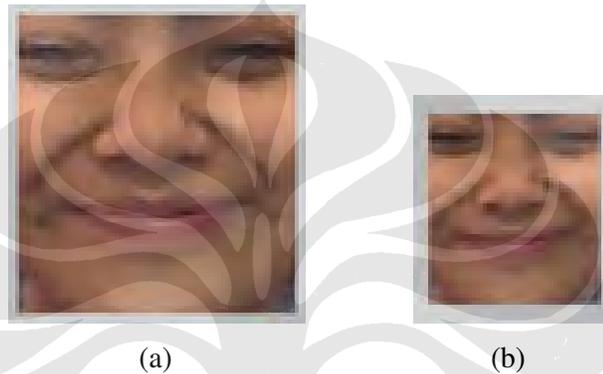


(b)

Gambar 3.9. Citra Digital Sebelum di-Crop(a); Citra Digital Sesudah di-Crop(b).

b. *Resizing* - perubahan ukuran gambar.

Perubahan ukuran gambar ini sangat penting peranannya untuk proses *training*, karena proses *training* atau pelatihan pada sistem pengenalan wajah membutuhkan data gambar yang seragam ukurannya. Ukuran yang dipakai disini adalah 60 x 55. Pada Gambar 3.10 Dapat dilihat bahwa citra digital setelah dipotong dengan ukuran 110 x 120 di susutkan menjadi 60 x 55. Perlu diketahui bahwa *resize* tidak hanya dapat menyusutkan ukuran citra namun juga dapat membesarkan ukuran citra.



Gambar 3.10 Citra Digital Sebelum di-*Resize* (a); Citra Digital Setelah di-*Resize*(b).

c. *Gray Scale* - Skala keabu-Abuan

Pada tahap ini, citra berwarna diubah menjadi *grayscale*, dengan memperoleh informasi intensitas dari gambar tersebut, gambar dapat di sortir secara eksklusif mulai dari hitam untuk intensitas yang paling lemah sampai dengan putih untuk intensitas yang paling kuat. Skala keabu-abuan disini merupakan suatu step yang perlu dilalui citra sebelum diolah ke proses selanjutnya, yakni *histogram equalization*. Seperti yang sudah dijelaskan pada teori dasar, citra berwarna terdiri dari 3 layer matriks yaitu R-layer, G-layer, B-layer. Untuk melakukan proses mengubah citra warna menjadi citra *grayscale* perlu diperhatikan 3 layer diatas, Jika setiap proses perhitungan dilakukan menggunakan 3 layer maka dilakukan 3 perhitungan yang sama. Mengubah 3 layer diatas menjadi 1 layer matrik *grayscale*. Dalam citra ini tidak ada lagi warna, yang ada adalah derajat keabuan. Untuk mengubah citra berwarna yang mempunyai nilai matrik

masing – masing  $r$ ,  $g$ ,  $b$  menjadi citra grayscale dengan nilai  $s$ . Konversi dapat dilakukan dengan mengambil rata – rata dari nilai  $r$ ,  $g$ , dan  $b$  sehingga dapat dituliskan pada persamaan 3.1

$$s = \frac{r + g + b}{3} \dots\dots\dots 3.1$$

Dengan:

$S$  : Nilai derajat keabuan

$r$  : Nilai warna merah

$g$  : Nilai warna hijau

$b$  : Nilai warna biru

Untuk melakukan konversi citra dari RGB menjadi *grayscale* didalam program dapat dilakukan dengan mempergunakan fungsi `rgb2gray`. Hasil yang didapatkan dari perintah tersebut dapat pada Gambar 3.11.



Gambar 3.11. Citra digital sebelum di-*greyscale* (a); Citra digital setelah di-*greyscale* (b).

#### d. *Histogram Equalization (Histeq)*

*Histogram Equalization* biasa digunakan pada citra digital untuk untuk memperbaiki kualitas gambar (*image enhancement*). Dengan *Histogram Equalization*, suatu teknik untuk meratakan distribusi terang/gelap sehingga gambar kelihatan lebih jelas. *Histogram Equalization* adalah suatu proses untuk meratakan histogram agar derajat keabuan dari yang paling rendah (0) sampai dengan yang paling tinggi (255) mempunyai kemunculan yang rata. Pada program, *Histogram Equalization* ini dilakukan dengan perintah pada program sebagai berikut;

```
I21 = histeq(I2);
axes(handles.axes1);
imshow(I21)
```

Dengan *histogram equalization* hasil gambar yang memiliki histogram yang tidak merata atau distribusi kumulatif yang banyak loncatan gradiasinya akan menjadi gambar yang lebih jelas karena derajat atau tingkatan keabuannya tidak dominan gelap atau dominan terang seperti yang ditunjukkan pada Gambar 3.12 berikut.



Gambar 3.12 Citra digital sebelum di-*histeq* (a); Citra digital setelah di-*histeq*(b).

e. *Filtering* - Penyaringan

*Filtering* atau penyaringan dibutuhkan untuk menguraingi derau atau *noise* yang ada pada citra digital. *Filter* yang digunakan pada *pre-processing* sistem pengenalan wajah ini menggunakan *filter "unsharp"*. *Filtering* ini dilakukan dengan perintah pada program sebagai berikut;

```
h = fspecial('unsharp');
I22 = imfilter(I21,h);
axes(handles.axes1);
imshow(I22)
```

Seperti yang sudah disebutkan pada Bab 2 di Teori Dasar, *filter unsharp* ini merupakan "*edge enhancemen filter*" yang akan menyebabkan citra input akan menjadi lebih jelas dan tajam tepiannya. Metode filterisasi ini biasanya dipergnakan diindustri fotografi ataupun pencetakan untuk semakin membuat tepian objek semakin "renyah". Hal ini seperti ditunjukkan pada Gambar 3.13.



Gambar 3.13 Citra digital sebelum di-*Filter* (a); Citra digital setelah di-*Filter*(b).

f. *Black and White* – hitam putih

Tahap ini adalah tahap terakhir sebelum akhirnya data dilatih, diolah dan dicocokkan nantinya. Gambar atau citra digital diubah menjadi citra hitam putih untuk menyederhanakan proses pengenalan pola wajah pada tahap selanjutnya. Perubahan citra digital menjadi *Black and White* ini dilakukan dengan perintah pada program sebagai berikut;

```
U = im2bw(I22);
axes(handles.axes1);
imshow(U);
```

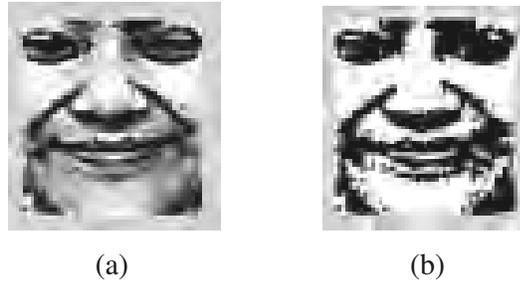
Hasilnya seperti ditunjukkan pada Gambar 3.14. Dengan ini nilai yang ada pada citra digital hanya ada 2 kemungkinan, yakni satu atau nol. Maka dari itu, *black and white* seringkali juga disebut sebagai citra biner atau *binary image*, karena hanya mempunyai 2 macam nilai.

Seperti yang telah diterangkan citra biner hanya memiliki 2 derajat keabuan, pixel-pixel objek bernilai 1 dan pixel-pixel latar belakang bernilai 0. Pada waktu menampilkan gambar, 0 adalah putih dan 1 adalah hitam. Jadi pada citra biner, latar belakang berwarna putih dengan objeknya berwarna hitam.

Keuntungan citra biner antara lain :

- Kebutuhan memori kecil
- Waktu pemrosesan lebih cepat

Citra biner didapatkan dengan cara meng-*threshold*-kan citra *grayscale*. Operasi *thresholding* akan mengelompokkan nilai derajat keabuan setiap pixel kedalam 2 kelas, yaitu hitam dan putih.



Gambar 3.14 Citra digital sebelum diubah ke *Binary Image* (a); Citra digital setelah diubah ke *Binary Image* (b).

### 3.2.2 Pelatihan

Proses *training* dilakukan dengan metode *neural network*, dengan diagram alir pada Gambar 3.15 sebagai berikut. Pembahasan lebih lanjut mengenai fase *training* dan uji cobanya dapat dilihat pada Bab 4.

*Training* atau pelatihan dari sistem ini menggunakan syntax dari *neural network toolbox* sebagai berikut:

1. Inisialisasi;

```
[wajah,t] = filewajah;
[R,Q] = size(wajah);
[S2,Q] = size(t);
S1 = 10;
```

2. Menentukan Input untuk *training*.

```
P = wajah;
```

3. Menentukan target set dari jaringan

```
T = t;
```

4. Membangun jaringan dan menetapkan banyaknya neuron tiap lapisan dan fungsi-fungsi aktivasi yang digunakan :

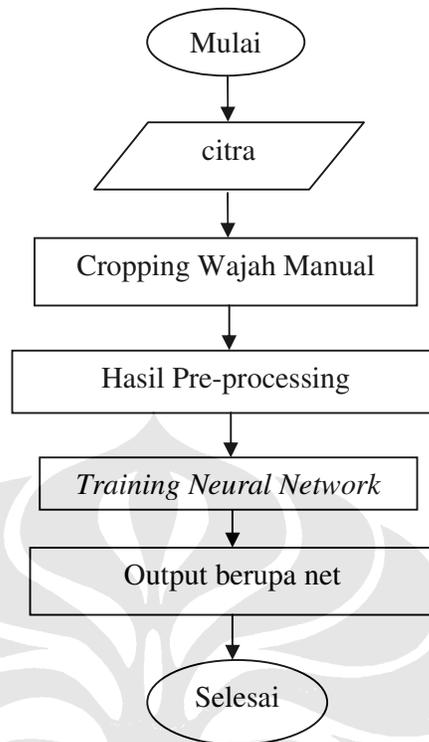
```
net = newff(minmax(wajah), [S1 S2], {'logsig'
'logsig'}, 'traingdx');
```

5. Selanjutnya menentukan parameter-parameter *neural network backpropagation*

```
net.LW{2,1} = net.LW{2,1}*0.01;
net.b{2} = net.b{2}*0.01;
net.performFcn = 'sse';
net.trainParam.goal = 0.00001;
net.trainParam.show = 20;
net.trainParam.epochs = 5000;
net.trainParam.mc = 0.95;
```

6. melakukan pembelajaran (*training*)

```
[net,tr] = train(net,P,T);
```



Gambar 3.15 Diagram alir proses *training*

Metode jaringan saraf tiruan propagasi balik atau *neural network back propagation* pada umumnya yang menggunakan 2 layer pada fase yakni *hidden layer taining* dan *output layer training*. Pada kedua layer tersebut digunakan fungsi transfer 'logsig'. Fungsi transfer logsig ini merupakan fungsi transfer unipolar atau logaritmik signoid.[1]. *Hidden layer* yang digunakan adalah sebanyak 10 *hidden layer* yang di set pada S1 angka 10 ini adalah hasil dari uji coba sistem (dibahas lebih lanjut pada Bab 4).

Sementara itu juga terdapat *traingdx (gradient Descent)* sebagai fungsi *training* yang digunakan. *Traingdx* adalah salah satu jenis pelatihan pada *neural network* dengan momentum dan *adaptife learning* sehingga pada saat proses *training* sistem dapat beradaptasi atau menyesuaikan terhadap input yg ada. *Training* yang dilakukan disini menggunakan algoritma *back propagation* yang juga ditunjukkan pada penggunaan komponen "newff" sebagai fungsi yang akan membuat jaringan umpan maju (*feed forward*).

Adapun data-data wajah yang diambil untuk proses pelatihan ini ada 12 citra wajah, yang dapat dilihat pada Tabel 3.1.

Tabel 3.1. Sampel data untuk proses pelatihan

Citra Wajah	Nama File
	Nurina.jpg
	irma.jpg
	uli.jpg
	iramaya.jpg
	edit.jpg
	liza.jpg
	ami.jpg
	citra.jpg
	ira.jpg
	astri.jpg
	uji.jpg
	nia.jpg

### 3.2.3 Identifikasi

Proses identifikasi adalah suatu proses yang menunjukkan identitas seseorang . Identitas yang dimaksud disini adalah nama dari wajah citra digital. Pada tahap ini sistem sudah melakukan pembelajaran dan setelah itu diharapkan dapat mengenali atau mengidentifikasi suatu citra wajah. Proses pengenalan diuraikan seperti berikut:

1. Memasukan file citra wajah yang akan diidentifikasi

```
global lizaout
z = imcrop(lizaout,[43 23 72 73]);
axes(handles.axes2);
imshow(z)
hold on
```

2. Mensimulasikan file tersebut kedalam jaringan yang telah di-*training* untuk mendapatkan output.

```
A2 = sim(net,U7);
```

3. Menentukan hasil identifikasi dengan *competitive function* atau 'compet' yang merupakan salah satu fungsi transfer pada *neural network toolbox*. Sintax lengkapnya adalah sebagai berikut.

```
A2 = sim(net,U7);
A2 = compet(A2);
answer = find(compet(A2) == 1);
if answer == 1
    s1='Arthania';
    set(handles.edit2,'string',s1);
    axes(handles.axes3);
    imshow('nia.jpg');
    hold on
elseif answer == 2
    s2='Iramaya';
    set(handles.edit2,'string',s2);
    axes(handles.axes3);
    imshow('iramaya.jpg');
    hold on
elseif answer == 3
    s3='Puji';
    set(handles.edit2,'string',s3);
    axes(handles.axes3);
    imshow('uji.jpg');
    hold on
elseif answer == 4
    s4='Uli';
    set(handles.edit2,'string',s4);
    axes(handles.axes3);
```

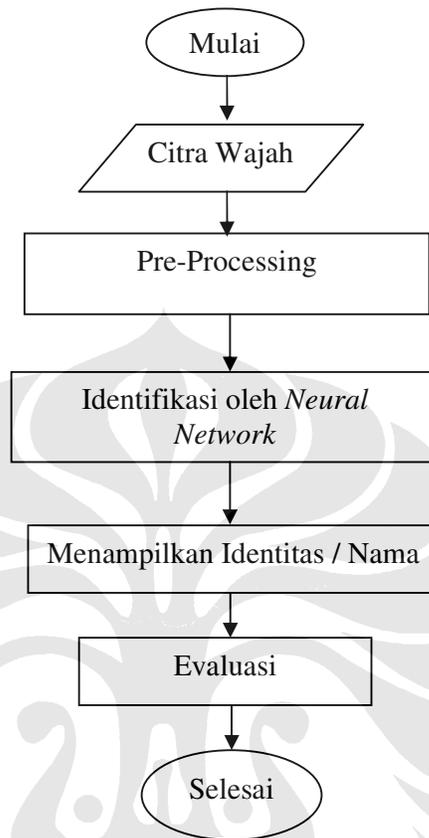
```

        imshow('uli.jpg');
        hold on
elseif answer == 5
    s5='Irma';
    set(handles.edit2,'string',s5);
    axes(handles.axes3);
    imshow('irma.jpg');
    hold on
elseif answer == 6
    s6='Nurina';
    set(handles.edit2,'string',s6);
    axes(handles.axes3);
    imshow('nurina.jpg');
    hold on
elseif answer == 7
    s7='Edit';
    set(handles.edit2,'string',s7);
    axes(handles.axes3);
    imshow('edit.jpg');
    hold on
elseif answer == 8
    s8='Eliza';
    set(handles.edit2,'string',s8);
    axes(handles.axes3);
    imshow('liza.jpg');
    hold on
elseif answer == 9
    s9='Astri';
    set(handles.edit2,'string',s9);
    axes(handles.axes3);
    imshow('astri.jpg');
    hold on
elseif answer == 10
    s10='Ira';
    set(handles.edit2,'string',s10);
    axes(handles.axes3);
    imshow('ira.jpg');
    hold on
elseif answer == 11
    s11='Ami';
    set(handles.edit2,'string',s11);
    axes(handles.axes3);
    imshow('ami.jpg');
    hold on
elseif answer == 12
    s12='Citra';
    set(handles.edit2,'string',s12);
    axes(handles.axes3);
    imshow('citra.jpg');
    hold on
else
    disp('tidak dikenali');
end

```

Pada Gambar 3.16 dapat dilihat diagram alir proses identifikasi secara garis besar pada sistem pengenalan wajah. Sebelum masuk ke proses identifikasi,

citra wajah digital juga terlebih dahulu melalui proses *pre-processing*, sehingga data yang diolah pada tahap pengenalan wajah ini juga merupakan citra biner.

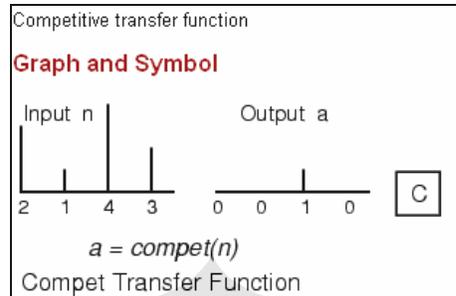


Gambar 3.16 Diagram alir proses identifikasi

Pengenalan dengan metode *neural network* ini disimulasikan dengan menggunakan syntax  $A2 = \text{sim}(\text{net}, U7)$ ; dimana *sim* artinya mensimulasikan jaringan saraf tiruan. Jaringan tersebut sudah dibangun pada saat pelatihan berupa *net*. Sementara *U7* adalah data matriks biner sebagai gambar masukan yang sudah melalui proses *pre-processing*.

Proses pengenalan kemudian dilakukan dengan menggunakan salah satu transfer function yang sudah di sediakan di *neural network* toolbox pada PROGRAM yakni "compet". Fungsi transfer ini mengkalkulasikan lapisan keluaran atau output layer dari input net-nya. Dengan fungsi ini hasil perhitungan matriks pada simulasi yang sudah dilakukan sebelumnya *neural network* di bandingkan satu sama lain lalu di pilih yang aling tinggi nilainya, atau yang paling menonjol. Nilai yang paling menonjol tersebut nilainya akan sama dengan satu,

sementara yang lainnya nol. Compet berasal dari kata *Competitive*, cara kerja dan lambang fungsi compet ini dapat digambarkan pada Gambar 3.17 berikut.



Gambar 3.17 Fungsi Transfer *Competitive*

Tampilan program pengenalan citra wajah ini di buat dengan gui agar terlihat lebih *user friendly* seperti pada tampilan Gambar 3.18 berikut.



Gambar 3.18 Tampilan Program Pengenalan Wajah.

## BAB 4

### PENGUJIAN DAN ANALISA

Pengujian dan analisa pada Bab 4 ini pada intinya adalah untuk mengetahui tingkat keberhasilan suatu sistem dengan metode yang sudah ditentukan. Masing masing metode mempunyai kelemahan dan kelebihan sendiri-sendiri. Sistem identifikasi wajah ini terdiri dari 2 sistem yakni sistem pendeteksi wajah dan sistem pengenalan wajah. Masing-masing sistem diuji dengan cara yang berbeda karena menggunakan metode yang berbeda pula.

#### 4.1. Pendeteksi wajah

Pada awal sistem sbelum *pre-processing* pada sistem pengenalan wajah ada sebuah sistem lain yakni sistem pendeteksi wajah dengan metode *pattern matching* atau pencocokan pola. Pendeteksi wajah ini yang membedakan bagian wajah dan bagian bukan wajah pada citra digital sehingga mempercepat proses pengenalan wajah nantinya, karena tidak perlu memperhitungkan komponen citra digital yang bukan wajah. Dengan demikian sistem pengenalan wajah diharapkan dapat melaukan proses lebih cepat dan akurat karena sistem pendeteksi wajah ini akan secara otomatis membuang bagian bukan wajah..

##### 4.1.1. Penentuan Pola pada Sistem Pendeteksi Wajah

Input dari sistem pendeteksi wajah ini adalah berupa Video input yang telah di set *frame rate* nya sebesar 15 fps (*frame per second*), untuk membuktikan bahwa dengan kamera digital sederhana sekalipun sistem ini dapat berjalan. Setelah mendapatkan parameter yang tepat dan juga susunan diagram blok set yg tepat pada program, sistem pendeteksi wajah dapat dikatakan berhasil. Tingkat keberhasilannya adalah 85.1%.

Adapun komponen yang paling menentukan dalam sistem pendeteksi wajah menggunakan metode *pattern matching* atau yang disebut pencocokan pola disini adalah pola wajah yang digunakan. Berikut ini adalah uji coba yang dilakukan untuk memperoleh parameter ukuran pola dan *template* wajah.

Perbedaan antara pola wajah dan *template* wajah adalah dimana pola wajah merupakan citra wajah digital hasil *cropping* manual dari pengambilan gambar dengan webcam, seperti yang ditunjukkan pada Gambar 4.1



Gambar 4.1 Citra Pola Wajah

Sementara itu *template* wajah adalah citra *template* yang dibuat manual seperti yang ditunjukkan Gambar 4.2



Gambar 4.2 Citra Template Wajah

Percobaan dilakukan dengan membuat variasi ukuran pola wajah dan *template* wajah. Hasil percobaan untuk mendapatkan pola yang cocok dan sesuai dengan input video yang digunakan sistem deteksi wajah, sehingga diharapkan akan mendapatkan hasil yang maksimal.

Pendeteksi wajah ini dirancang untuk dapat diintegrasikan pada sistem pengenalan wajah. Pendeteksi wajah itu sendiri merupakan sebuah sistem yang mempunyai input-proses-output. Input sistem berupa Gambar video yang langsung di ambil dari webcam atau *realtime*. Progresnya adalah berupa korelasi antara dua buah citra yang sesuai, yang ditunjukkan nilai maksimumnya, dan diklasifikasikan menjadi dua bagian, yakni wajah dan non wajah dengan membuat kotak pembatas yang mengarah kepada objek wajah. Hasil atau output-nya adalah sebagai berikut pada Gambar 4.3



Gambar 4.3. Beberapa sample frame video input 160 x 120 hasil deteksi wajah.

Untuk mendapatkan korelasi yang sesuai antara input yang akan di deteksi wajah berupa video dengan pola wajah, maka dilakukan ujicoba dengan memasukan pola wajah yang berbeda ukuran. Berikut hasil ujicoba pada Tabel 4.1 dan 4.2.

Tabel 4.1 Hasil Percobaan Deteksi Wajah dengan Menentukan Ukuran Pola Wajah

Ukuran Pola Wajah	Input Video	Terdeteksi Sempurna
20 x 18	160 x 120	0%
30 x 27	160 x 120	0%
40 x 36	160 x 120	0%
50 x 45	160 x 120	50%
60 x 54	160 x 120	100%
70 x 63	160 x 120	100%
80 x 72	160 x 120	100%
90 x 81	160 x 120	100%
100 x 90	160 x 120	50%
110 x 100	160 x 120	0%
120 x 109	160 x 120	50%
20 x 18	174 x 144	0%
30 x 27	174 x 144	0%
40 x 36	174 x 144	0%
50 x 45	174 x 144	50%
60 x 54	174 x 144	100%
70 x 63	174 x 144	50%
80 x 72	174 x 144	100%
90 x 81	174 x 144	100%
100 x 90	174 x 144	50%
110 x 100	174 x 144	0%
120 x 109	174 x 144	0%

Berdasarkan pada percobaan pertama dengan memvariasikan ukuran pola wajah sebagai data citra yang akan dikorelasikan dengan input video dari webcam secara *real time*, pada ukuran citra input video dengan ukuran frame yang lebih besar hasilnya tidak lebih baik dari pada citra input video dengan ukuran frame

yang lebih kecil dan frame yang lebih besar juga memakan waktu yang lebih lama.. Hal ini kemungkinan dikarenakan oleh bertambahnya perhitungan. Maka pada percobaan dengan *template* selanjutnya hanya dilakukan pada ukuran frame 160 x 120 saja untuk dapat menentukan ukuran dari pola wajah yang akan dikorelasikan dengan input video.

Tabel 4.2 Hasil Percobaan Deteksi Wajah dengan Menentukan Ukuran Template Wajah

Ukuran Template Wajah	Input Video	Terdeteksi Sempurna
38 x 40	160 x 120	0%
40 x 50	160 x 120	0%
58 x 60	160 x 120	50%
67 x 70	160 x 120	100%
76 x 80	160 x 120	100%
86 x 90	160 x 120	100%

Dari percobaan di atas, pola wajah yang paling sesuai dengan input video 160 x 120 adalah berkisar antara 70 x 63 pada penggunaan pola wajah, dan 67 x 70 pada penggunaan *template* wajah. Hasil uji coba menunjukkan pada range nilai frame seperti itulah yang paling sesuai untuk melakukan korelasi dengan input video 160 x 120, sementara dengan ukuran pola yang lebih besar lagi akan membuat gambar menjadi terkesan patah-patah.

#### 4.1.2. Pendeteksi Wajah pada Posisi Wajah yang Berbeda-beda

Percobaan selanjutnya adalah uji coba keberhasilan deteksi wajah secara real time pada objek bergerak yang langsung diambil dari webcam, dengan berbagai pergerakan atau perubahan posisi. Pada saat pengujian jarak antara lensa webcam dengan objek wajah adalah 44 cm. Hasil uji coba sistem pendeteksi wajah ini ditunjukkan pada Tabel 4.3. Percobaan ini dilakukan dengan cara menggerakkan objek ke segala arah untuk memastikan wajah yang bergerak dapat berhasil terdeteksi. Hasil menunjukkan selama wajah lengkap berada dalam jangkauan kamera, maka 100% dapat terdeteksi, hanya saja gerakan di sebelah

kiri gambar dapat lebih cepat terdeteksi, hal ini dikarenakan jarak koordinat yang lebih dekat ke pusat.

Tabel 4.3 Ujicoba Deteksi wajah pada Posisi Kordinat (x,y) yang berbeda.

Posisi Wajah	Wajah Terdeteksi	Keberhasilan
Bergerak ke kiri	Ya	100%
Bergerak ke kanan	Ya	100%
Bergerak ke atas	Ya	100%
Bergerak ke bawah	Ya	100%
Bergerak ke depan	Ya	100%
Bergerak ke belakang	Ya	100%

Sementara itu proses pengambilan wajah dengan menggunakan metode *pattern matching* ini mempunyai beberapa kelemahan terutama jika objek wajah miring, tidak sampai 45 derajat kemiringan wajah, sistem sudah tidak mamu lagi mendeteksi wajah. Sementara pada kondisi wajah menoleh, sistem masih mampu mendeteksi sampai dengan sudut 90 derajat kekiri dan ke kanan. Hal ini dirangkum pada tabel 4.4 berikut.

Tabel 4.4. Hasil Uji Coba Deteksi Wajah pada Posisi Wajah yang Berbeda-beda

Kemiringan Tabel Wajah	Wajah Terdeteksi	Keberhasilan
45 derajat menoleh ke kiri	Ya	100%
90 derajat menoleh ke kiri	Ya	50%
45 derajat menoleh ke kanan	Ya	100%
90 derajat menoleh ke kanan	Ya	50%
15 derajat miring kiri	Ya	100%
30 derajat miring kiri	Ya	100%
45 derajat miring kanan	Tidak	0%
15 derajat miring kanan	Ya	100%
30 derajat ke arah kanan	Ya	100%
45 derajat ke arah kanan	Tidak	0%

Tingkat keberhasilan percobaa pertama saat sistem mendeteksi wajah secara posisi berdasarkan koordinat adalah 100% selama wajah yang utuh masih berada di dalam jangkauan kamera. Pada percobaan kedua, pendeteksian wajah pada tingkat derajat kemiringan yang berbeda-beda tingkat keberhasilannya adalah 70%.

Penggabungan nilai rata-rata kedua ujicoba tersebut adalah tingkat keberhasilan sistem pendeteksi wajah yakni 85%.

## 4.2. Pengenalan wajah

Masuk ke sistem pengenalan wajah, metode yang digunakan adalah metode jaringan saraf tiruan atau *neural network*. Karena perbedaan metode ini maka uji coba yang dilakukan juga berbeda-beda. Pada proses pengujian sistem ini akan diuji sejauh mana kehandalan proses pengenalan wajah menggunakan *neural network* yang parameternya telah diujicoba, terlebih dahulu.

### 4.2.1 Pengujian Parameter Sistem Pengenalan Wajah

*Hidden layer* yang digunakan pada sistem pengenalan wajah ini adalah 100 layer, hal ini berdasarkan pengujian saat menggunakan jumlah *hidden layer* yang berbeda, terjadi penurunan tingkat keberhasilan saat melakukan ujicoba pengenalan wajah. Untuk lebih lengkapnya dapat dilihat di Tabel 4.5, Tabel 4.6, dan Tabel 4.7 sebagai berikut.

Tabel 4.5 Uji *Hidden Layer* 10 *Neural Network* pada pengenalan Wajah

Data Latih sebagai DataMasukan	Data Latih yang Dikenal	<i>Hidden Layer</i>	Tingkat Keberhasilan
liza.jpg	liza.jpg	10	100%
uji.jpg	edit.jpg	10	0%
iramaya.jpg	iramaya.jpg	10	100%
uli.jpg	uli.jpg	10	100%
irma.jpg	irma.jpg	10	100%
nurina.jpg	Nurina.jpg	10	100%
edit.jpg	Nurina.jpg	10	0%
nia.jpg	Nurina.jpg	10	0%
citra.jpg	citra.jpg	10	100%
ira.jpg	ira.jpg	10	100%
astri.jpg	astri.jpg	10	100%
ami.jpg	ami.jpg	10	100%

\*Tingkat kegagalan =  $3/12 * 100 = 25\%$

*Adjustment* atau penyetingan dilakukan pada *hidden layer* 40 dan 100 setelah menggunakan *hidden layer* 10 tingkat keberhasilannya masih kecil. Tabel 4.6 menunjukkan pengujian pengenalan data latih pada *hidden layer* 40 sementara Tabel 4.7 menunjukkan pengujian pengenalan data latih pada *hidden layer* 100.

Tabel 4.6 Uji *Hidden Layer 40 Neural Network* pada pengenalan Wajah

Data Latih sebagai DataMasukan	Data Latih yang Dikenal	Hidden Layer	Tingkat Keberhasilan
liza.jpg	liza.jpg	40	100%
uji.jpg	uji.jpg	40	100%
iramaya.jpg	iramaya.jpg	40	100%
uli.jpg	uli.jpg	40	100%
irma.jpg	irma.jpg	40	100%
nurina.jpg	nurina.jpg	40	100%
edit.jpg	edit.jpg	40	100%
nia.jpg	nia.jpg	40	100%
citra.jpg	citra.jpg	40	100%
ira.jpg	ira.jpg	40	100%
astri.jpg	astri.jpg	40	100%
ami.jpg	ami.jpg	40	100%

\*Tingkat kegagalan = 0%

Hasil uji coba yang ditunjukkan pada Tabel 4.5 , Tabel 4.6 maupun Tabel 4.7 merupakan pengujian terhadap data latih hingga keberhasilannya mencapai 100%.

Tabel 4.7 Uji *Hidden Layer 100 Neural Network* pada pengenalan Wajah

Data Latih sebagai DataMasukan	Data Latih yang Dikenal	Hidden Layer	Tingkat Keberhasilan
liza.jpg	liza.jpg	100	100%
uji.jpg	uji.jpg	100	100%
iramaya.jpg	iramaya.jpg	100	100%
uli.jpg	uli.jpg	100	100%
irma.jpg	irma.jpg	100	100%
nurina.jpg	nurina.jpg	100	100%
edit.jpg	edit.jpg	100	100%
nia.jpg	nia.jpg	100	100%
citra.jpg	citra.jpg	100	100%
ira.jpg	ira.jpg	100	100%
astri.jpg	astri.jpg	100	100%
ami.jpg	ami.jpg	100	100%

\*Tingkat kegagalan = 0%

#### 4.2.2 Pengujian Tingkat keberhasilan Sistem Pengenalan Wajah

Pengujian sistem pengenalan wajah dilakukan dengan mengidentifikasi nama dari data wajah yang sudah di *training* dan tingkat kegagalan paling rendah adalah di *hidden layer* 100, karena dari percobaan dengan menggunakan *hidden layer* yang lainnya tingkat keberhasilannya lebih rendah dari 76%. Percobaan dilakukan sebanyak 696 kali pada *hidden layer* yang berbeda-beda.

Tabel 4.8. Hasil identifikasi 12 data *training*,

Data Latih sebagai DataMasukan	Data Latih yang Dikenal	Tingkat Keberhasilan
liza.jpg	liza.jpg	100%
uji.jpg	uji.jpg	100%
iramaya.jpg	iramaya.jpg	100%
uli.jpg	uli.jpg	100%
irma.jpg	irma.jpg	100%
nurina.jpg	nurina.jpg	100%
edit.jpg	edit.jpg	100%
nia.jpg	nia.jpg	100%
citra.jpg	citra.jpg	100%
lra.jpg	ira.jpg	100%
astri.jpg	astri.jpg	100%
ami.jpg	ami.jpg	100%

\*Tingkat kegagalan = 0%

Percobaan pengenalan dibagi menjadi 2 yakni percobaan pengenalan yang dilakukan pada ke-duabelas data latih dan percobaan pengenalan pada data gambar selain data latih. Masih sama seperti percobaan sebelumnya, tingkat keberhasilan adalah 100% setelah dua kali diuji dan di *training* ulang pada data yang sama dan *hidden layer* yang sama yakni 100. Namun pada *hidden layer* 10 tingkat keberhasilannya hanya 75%

Pada *hidden layer* 10 dengan data *training* masih hanya berjumlah 6 data, keberhasilan yang dicapai juga mencapai 100% saat mencoba mengenali data-data yang sudah pernah di-*training*, seperti Tabel 4.9 berikut.

Tabel 4.9. Hasil identifikasi 6 data *training*

Data Masukan	Data Latih yang Dikenal	Tingkat Keberhasilan
Nurina.jpg	Nurina.jpg	100%
Irma.jpg	Irma.jpg	100%
Uli.jpg	Uli.jpg	100%
Ira.jpg	Ira.jpg	100%
Edit.jpg	Edit.jpg	100%
Liza.jpg	Liza.jpg	100%

\*Tingkat kegagalan = 0%

Hal ini diperkirakan terjadi karena dengan bertambahnya data latih pada metode jaringan saraf tiruan maka bertambah pula kompleksitas perhitungan data, sehingga dengan bertambahnya kompleksitas, peluang terjadinya kesalahan menjadi lebih besar.

Pada percobaan identifikasi data wajah yang bukan data *training* menggunakan *hidden layer* 100, dari total kedua dua puluh tujuh, tingkat keberhasilannya 79% yang dapat dikenali sebagai wajah dengan pemetaan nama yang benar. Hal ini dikarenakan mimik wajah dan pencahayaan yang hampir masih belum sempurna dikenali oleh sistem. Salah satu contoh kesalahan karena mimik wajah tersenyum adalah seperti yang ditunjukkan pada Gambar 4.4. Sementara contoh pengenalan yang benar ditunjukkan pada Gambar 4.5 Untuk lebih detailnya data uji dapat dilihat pada Tabel 4.10 pada halaman berikutnya.



Gambar 4.4. Kesalahan pada proses pengenalan karena mimik wajah.



Gambar 4.5. Contoh pengenalan wajah yang berhasil.

Tabel 4.10 Hasil identifikasi 12 data *training* pada data wajah selain data *training*

Data Masukan (non data latih)	Data Latih yang Dikenal	Tingkat Keberhasilan
ami2.jpg	liza.jpg	0%
astri2.jpg	astri.jpg	100%
citra2.jpg	citra.jpg	100%
edit2.jpg	edit.jpg	100%
ira2.jpg	ira.jpg	100%
iramaya2.jpg	iramaya.jpg	100%
irma2.jpg	irma.jpg	100%
liza2.jpg	liza.jpg	100%
nia2.jpg	citra.jpg	0%
nurina2.jpg	nurina.jpg	100%
uji2.jpg	uji.jpg	100%
uli2.jpg	uli.jpg	100%
ami3.jpg	ami.jpg	100%
astri3.jpg	astri.jpg	100%
citra3.jpg	nia.jpg	0%
edit3.jpg	edit.jpg	100%
ira3.jpg	ira.jpg	100%
iramaya3.jpg	iramaya.jpg	100%
irma3.jpg	citra.jpg	0%
liza3.jpg	liza.jpg	100%
nia3.jpg	citra.jpg	0%
nurina3.jpg	nurina.jpg	100%
uji3.jpg	uji.jpg	100%
uli3.jpg	uli.jpg	100%

\*Tingkat kegagalan =  $5/27 * 100 = 21\%$

Dengan demikian dapat diambil rata tingkat Keberhasilan adalah 79% pada data non latih. Hal ini merupakan peningkatan dibandingkan dengan uji coba

sebelumnya ke sepuluh data selain data *training*, dengan 6 data *training* pada Tabel 4.9. Kesepuluh data non-*training* tersebut dengan tingkat keberhasilannya yang hanya 60 % ditunjukkan di Tabel 4.11 berikut.

Tabel 4.11 Hasil identifikasi 6 data *training* pada data wajah selain data *training*

Data Masukan	Data Latih yang Dikenal	Tingkat Keberhasilan
Nurina2.jpg	Nurina.jpg	100%
Irma2.jpg	Irma.jpg	100%
Uli2.jpg	Irma.jpg	0%
Ira2.jpg	Ira.jpg	100%
Edit2.jpg	Edit.jpg	100%
Liza1.jpg	Uli.jpg	0%
Liza2.jpg	Liza.jpg	100%
Citra.jpg	Irma.jpg	0%
Nia.jpg	Irma.jpg	0%
Irma1.jpg	Irma.jpg	100%

\*Tingkat kegagalan =  $4/10 * 100 = 40\%$

Dengan demikian dapat diambil kesimpulan bahwa dengan menambahkan data *training*, maka peluang untuk mengenali wajah – wajah yang belum non *training* menjadi lebih besar.

Sementara itu jika keseluruhan data wajah digabungkan diuji dan dihitung persentase keberhasilannya mencapai 86% dengan *hidden layer* 100. Hal ini ditunjukkan pada Tabel 4.12 berikut.

Tabel 4.12 Hasil identifikasi seluruh data wajah

Input Data	Data Latih yang Dikenal	Hidden Layer	Tingkat Keberhasilan
ami.jpg	ami.jpg	100	100%
astri.jpg	astri.jpg	100	100%
citra.jpg	uji.jpg	100	100%
edit.jpg	edit.jpg	100	100%
ira.jpg	liza.jpg	100	100%
iramaya.jpg	irma.jpg	100	100%
irma.jpg	irma.jpg	100	100%
liza.jpg	liza.jpg	100	100%
nia.jpg	nia.jpg	100	100%

Tabel 4.12 Hasil identifikasi seluruh data wajah (lanjutan)

Input Data	Data Latih yang Dikenal	Hidden Layer	Tingkat Keberhasilan
Nurina.jpg	uji.jpg	100	100%
uji.jpg	nia.jpg	100	100%
uli.jpg	uli.jpg	100	100%
ami2.jpg	liza.jpg	100	0%
astri2.jpg	astri.jpg	100	100%
citra2.jpg	citra.jpg	100	100%
edit2.jpg	edit.jpg	100	100%
ira2.jpg	ira.jpg	100	100%
iramaya2.jpg	iramaya.jpg	100	100%
irma2.jpg	irma.jpg	100	100%
liza2.jpg	liza.jpg	100	100%
nia2.jpg	citra.jpg	100	0%
Nurina2.jpg	nurina.jpg	100	100%
uji2.jpg	uji.jpg	100	100%
uli2.jpg	uli.jpg	100	100%
ami3.jpg	ami.jpg	100	100%
astri3.jpg	astri.jpg	100	100%
citra3.jpg	nia.jpg	100	0%
edit3.jpg	edit.jpg	100	100%
ira3.jpg	ira.jpg	100	100%
iramaya3.jpg	iramaya.jpg	100	100%
irma3.jpg	citra.jpg	100	0%
liza3.jpg	liza.jpg	100	100%
nia3.jpg	citra.jpg	100	0%
Nurina3.jpg	nurina.jpg	100	100%
uji3.jpg	uji.jpg	100	100%
uli3.jpg	uli.jpg	100	100%
Tingkat Keberhasilan Rata-rata			86%

\*Tingkat kegagalan =  $5/36 * 100 = 14\%$

## BAB 5

### KESIMPULAN

Dari uraian teori, perancangan dan pengembangan metodologi, serta uji coba yang telah dilakukan pada system identifikasi wajah ini, maka dapat dibuat kesimpulan bahwa:

- tingkat keberhasilan sistem pendeteksi wajah menggunakan metode *pattern matching* mencapai 85,1%,
- pada parameter yang di set *hidden layer* 100, proses pengenalan pada data non-latih mencapai tingkat akurasi yang paling tinggi yakni 79%,
- tingkat keberhasilan secara keseluruhan pada proses pengenalan menggunakan metode *Neural Network* algoritma *backpropagation* yang dilakukan 86%,
- proses pengenalan wajah dengan metode jaringan saraf tiruan dapat dilakukan dengan bantuan pengolahan citra yang tepat pada *pre-processing*,
- proses pendeteksi wajah dengan pencocokan pola akan lebih cepat mendeteksi dengan ukuran pola yang kecil,
- penentuan parameter pada komponen pendukung masing-masing metode sangat mempengaruhi tingkat keberhasilan yang dicapai.

## DAFTAR REFERENSI

- [1] Budi Setiawan, *Pengantar Jaringan Saraf Tiruan*, Perpustakaan FTUI, 2003, hal 1.
- [2] Rafael Gonzalez, Richard Woods, *Digital Image Processing with MATLAB*, Addison Wesley Publishing co., USA, 1993, hal 12
- [3] Setyo Nugroho, *Sistem Pendeteksi Wajah Manusia pada Citra Digital*, Tesis Ilmu Komputer, UGM, 2004, hal 7-18.
- [4] Resmana Lim, *APLIKASI OCR DENGAN METODE TEMPLATE MATCHING UNTUK PENDUKUNG BERMAIN GITAR*, Universitas Kristen Petra, hal 2.
- [5] Cristine Siandawati Lukmanto, *Optical Character Recognition Berbasis Combined Template Matching*, 2001, diakses dari internet April 2008.
- [6] [www.wikipedia.org](http://www.wikipedia.org) dengan key word "JST", diakses dari internet Maret 2008.
- [7] Ali Akbar, *JARINGAN SARAF TIRUAN UNTUK MENILAI ARANSEMEN MUSIK*, Tugas Akhir ITB, 2007, hal II-1 sampai II-4
- [8] Johannes Andria, *Penerapan Teknik E-mail Filtering Berbasis Ekstraksi Ciri dan Jaringan Saraf Tiruan Propagasi Balik*, Tugas Akhir Fasilkom UI 2006, hal 16 - 25