



**UNIVERSITAS INDONESIA**

**SISTEM PENGENAL WAJAH BERBASIS ENSEMBLE  
NEURAL NETWORK UNTUK CITRA INFRA MERAH**

**SKRIPSI**

**STEPHEN ROY IMANTAKA  
0606074350**

**FAKULTAS TEKNIK  
PROGRAM STUDI TEKNIK ELEKTRO  
DEPOK  
JUNI 2010**



**UNIVERSITAS INDONESIA**

**SISTEM PENGENAL WAJAH BERBASIS ENSEMBLE  
NEURAL NETWORK UNTUK CITRA INFRA MERAH**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik**

**STEPHEN ROY IMANTAKA  
0606074350**

**FAKULTAS TEKNIK  
PROGRAM STUDI TEKNIK ELEKTRO  
DEPOK  
JUNI 2010**

**HALAMAN PERNYATAAN ORISINALITAS**

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

**Nama : Stephen Roy Imantaka**

**NPM : 0606074350**

**Tanda Tangan:**

**Tanggal : 1 Juni 2010**

## HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Stephen Roy Imantaka  
NPM : 0606073921  
Program Studi : Teknik Elektro  
Judul Skripsi : Sistem Pengenal Wajah Berbasis Neural Network Ensemble untuk Citra Infra Merah

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

### DEWAN PENGUJI

Pembimbing :

Prof. Dr.Eng. Drs. Benyamin Kusumoputro M.Eng. ( )

Penguji 1 :

Dr. Abdul Muis, S.T, M.Eng. ( )

Penguji 2 :

Dr. Ir. Dodi Sudiana, M.Eng. ( )

Ditetapkan di : Depok

Tanggal : 29 Juni 2010

## KATA PENGANTAR

Puji dan syukur ke hadirat Tuhan Yang Maha Esa, karena atas rahmat dan karunia-Nya penulis dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Jurusan Teknik Elektro pada Fakultas Teknik Universitas Indonesia.

Penulis menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi penulis untuk menyelesaikan skripsi ini. Oleh karena itu, penulis mengucapkan terima kasih kepada:

- 1) Prof. Dr.Eng. Drs. Benyamin Kusumoputro M.Eng., selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan penulis dalam penyusunan skripsi ini;
- 2) orang tua dan keluarga penulis yang telah memberikan bantuan dukungan material dan moral;
- 3) sahabat-sahabat yang telah menjadi responden pada pengambilan data (foto wajah) pada skripsi ini; dan
- 4) teman-teman yang baik secara langsung ataupun tidak langsung membantu penulis dalam menyelesaikan skripsi ini.

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

Depok, Juni 2010

Penulis

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

---

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

Nama : Stephen Roy Imantaka  
NPM : 0606074350  
Program Studi : Teknik Elektro  
Departemen : Teknik Elektro  
Fakultas : Teknik  
Jenis Karya : Skripsi

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:

**SISTEM PENGENAL WAJAH BERBASIS ENSEMBLE NEURAL  
NETWORK UNTUK CITRA INFRA MERAH**

beserta perangkat yang ada. Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir 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 : 1 Juni 2010

Yang menyatakan

(Stephen Roy Imantaka)

## ABSTRAK

Nama : Stephen Roy Imantaka  
Program Studi : Teknik Elektro  
Judul : Sistem Pengenal Wajah Berbasis Ensemble Neural Network  
untuk Citra Infra Merah

Dalam beberapa tahun ini, telah banyak penelitian yang berhubungan dengan pengenalan pola dilakukan untuk mengidentifikasi wajah manusia secara otomatis. Skripsi ini membahas sistem pengenal wajah berbasis jaringan syaraf tiruan tunggal dan ensemble. Kamera infra merah digunakan pada penelitian untuk mengatasi masalah pencahayaan yang ditemui pada kamera *visible light*. Tahapan sistem terdiri dari pra-pemrosesan, ekstraksi fitur oleh PCA, pelatihan, dan pengujian. Pada percobaan dengan jaringan syaraf tunggal, algoritma *backpropagation* diterapkan dan sejumlah parameter divariasikan untuk mencapai performa yang optimal. Pada percobaan dengan jaringan syaraf ensemble, sejumlah skema digunakan antara lain: algoritma *backpropagation*, algoritma NCL, dan parsialisasi data dengan kedua algoritma tersebut. Hasil yang ditinjau dari *recognition rate* menunjukkan jaringan syaraf ensemble, yang terdiri dari sejumlah jaringan syaraf, memberikan performa yang lebih baik dibandingkan sebuah jaringan syaraf tunggal. Bila dioptimalkan, jaringan syaraf ensemble dapat menghasilkan *recognition rate* sebesar 99.9%.

Kata kunci:

Jaringan syaraf tiruan, *Principal Component Analysis (PCA)*, *backpropagation*, *Negative Correlation Learning (NCL)*, *ensemble neural network*

## ABSTRACT

Name : Stephen Roy Imantaka  
Study Program: Electrical Engineering  
Title : Face Recognition System Based on Ensemble Neural Network for Infrared Images

People in pattern recognition have been working on automatic recognition of human faces for years. The focus of this thesis is a face recognition system based on both individual and ensemble neural network. An infrared camera is utilized to overcome the illumination matter encountered by visible light cameras. The procedure of the system consists of pre-processing, feature extraction by PCA, training and testing. In the experiment of individual neural network, back-propagation algorithm is applied and some parameters are varied to obtain the optimum performance. In the experiment of ensemble neural network, some schemes such as back-propagation algorithm, NCL algorithm and partition of data using both algorithms are particularly examined. According to the recognition rate, the results show that ensemble neural network, which is made up of several neural networks, have better performance than a single neural network does. An optimized ensemble neural network may reach up to 99.9% of recognition rate.

**Keywords:**

Artificial Neural Network, Principal Component Analysis (PCA), back-propagation, Negative Correlation Learning (NCL), Ensemble Neural Network

## DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERNYATAAN ORISINALITAS.....	ii
HALAMAN PENGESAHAN.....	iii
KATA PENGANTAR .....	iv
LEMBAR PERSETUJUAN PUBLIKASI KARYA ILMIAH .....	v
ABSTRAK .....	vi
ABSTRACT .....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	x
DAFTAR GAMBAR .....	xiii
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1.Latar Belakang.....	1
1.2.Pernyataan Masalah.....	2
1.3.Tujuan Penulisan .....	3
1.4.Batasan Masalah .....	4
1.5.Manfaat Penelitian.....	4
1.6.Tahapan Penelitian .....	5
1.7.Sistematika Penulisan Laporan.....	6
<b>BAB II TINJAUAN PUSTAKA.....</b>	<b>7</b>
2.1. Jaringan Syaraf Tiruan dengan Algoritma Backpropagation .....	7
2.2. Jaringan Syaraf Ensemble (Ensemble Neural Network).....	12
2.3. Algoritma Negative Correlation Learning.....	15
2.4. Principal Component Analysis .....	17
<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>25</b>
3.1. Skema Penelitian Jaringan Syaraf Tunggal .....	25
3.2. Parameter yang Digunakan sebagai Variabel pada Jaringan Syaraf Tunggal.....	29
3.2. Skema Penelitian Jaringan Syaraf Ensemble .....	30
<b>BAB IV ANALISIS PERFORMA JARINGAN SYARAF TUNGGAL .....</b>	<b>33</b>
4.1. Jaringan Syaraf Tiruan untuk Sistem Pengenal Wajah .....	33
4.2. Principal Component Analysis (PCA) untuk Mereduksi Data.....	44
4.3. Pengaruh Parameter-Parameter pada Algoritma Backpropagation.....	48
4.3.1. Inisialisasi Bobot Awal.....	48
4.3.2. Jumlah Unit Lapisan Tersembunyi.....	51
4.3.3. Banyaknya Iterasi yang Dilakukan.....	55
4.3.4. Besarnya Laju Pembelajaran ( $\alpha$ ).....	57
4.3.5. Penggunaan Momentum .....	60
4.4. Optimalisasi Jaringan Syaraf Tunggal.....	68

<b>BAB V ANALISIS PERFORMA JARINGAN SYARAF ENSEMBLE .....</b>	<b>72</b>
5.1. Skema Algoritma Backpropagation .....	72
5.2. Skema Parsialisasi Data dengan Algoritma Backpropagation .....	75
5.3. Skema Negative Correlation Learning (NCL) .....	82
5.4. Skema Parsialisasi Data dengan Algoritma NCL.....	87
5.5. Perbandingan Keempat Skema yang Digunakan .....	104
5.6. Optimalisasi Jaringan Syaraf Ensemble .....	109
<b>BAB VI KESIMPULAN DAN SARAN.....</b>	<b>113</b>
6.1. Kesimpulan.....	113
6.2. Saran .....	114
LAMPIRAN.....	115
DAFTAR REFERENSI .....	125

## DAFTAR TABEL

Tabel 4.1.	Recognition Rate dan Recognition Time dengan Parameter-Parameter Standar yang Ditetapkan .....	38
Tabel 4.2.	Recognition Rate per Kelas.....	39
Tabel 4.3.	Recognition Rate dan Recognition Time dengan Variansi Dimensi yang Direduksi PCA .....	44
Tabel 4.4.	Recognition Rate dan Recognition Time dengan Variasi Metode Inisialisasi Awal Bobot .....	49
Tabel 4.5.	Recognition Rate dan Recognition Time dengan Variasi Jumlah Unit Lapisan Tersembunyi.....	51
Tabel 4.6.	Recognition Rate dan Recognition Time dengan Variasi Banyaknya Iterasi yang Dilakukan .....	55
Tabel 4.7.	Recognition Rate dan Recognition Time dengan Variasi Besarnya Laju Pembelajaran.....	57
Tabel 4.8.	Recognition Rate dan Recognition Time dengan Variasi Momentum ( $\alpha = 0.5$ ).....	61
Tabel 4.9.	Perbandingan Parameter Default dan Parameter Optimalisasi .....	68
Tabel 4.10.	Perbandingan Recognition Rate dan Recognition Time Parameter Default dan Parameter Optimalisasi .....	61
Tabel 4.11.	Recognition Rate per Kelas pada Parameter Optimalisasi.....	61
Tabel 5.1.	Recognition Rate dan Recognition Time Skema 1 Algoritma Backpropagation .....	73
Tabel 5.2.	Recognition Rate dan Recognition Time Skema 2 untuk 1 Jaringan .....	75
Tabel 5.3.	Recognition Rate dan Recognition Time Skema 2 untuk 2 Jaringan .....	76
Tabel 5.4.	Recognition Rate dan Recognition Time Skema 2 untuk 3 Jaringan .....	76
Tabel 5.5.	Recognition Rate dan Recognition Time Skema 2 untuk 4 Jaringan .....	76
Tabel 5.6.	Recognition Rate dan Recognition Time Skema 2 untuk 5 Jaringan .....	76
Tabel 5.7.	Recognition Rate dan Recognition Time Skema 3 untuk 1 Jaringan .....	82
Tabel 5.8.	Recognition Rate dan Recognition Time Skema 3 untuk 2 Jaringan .....	83
Tabel 5.9.	Recognition Rate dan Recognition Time Skema 3 untuk 3 Jaringan .....	83
Tabel 5.10.	Recognition Rate dan Recognition Time Skema 3 untuk 4 Jaringan .....	83
Tabel 5.11.	Recognition Rate dan Recognition Time Skema 3 untuk 5 Jaringan .....	83
Tabel 5.12.	Recognition Rate dan Recognition Time Skema 4 untuk 1 Jaringan, 50% Data .....	88
Tabel 5.13.	Recognition Rate dan Recognition Time Skema 4 untuk 1 Jaringan, 60% Data .....	88

Tabel 5.14.	Recognition Rate dan Recognition Time Skema 4 untuk 1 Jaringan, 70% Data .....	88
Tabel 5.15.	Recognition Rate dan Recognition Time Skema 4 untuk 1 Jaringan, 80% Data .....	89
Tabel 5.16.	Recognition Rate dan Recognition Time Skema 4 untuk 1 Jaringan, 90% Data .....	89
Tabel 5.17.	Recognition Rate dan Recognition Time Skema 4 untuk 1 Jaringan, 100% Data .....	89
Tabel 5.18.	Recognition Rate dan Recognition Time Skema 4 untuk 2 Jaringan, 50% Data .....	89
Tabel 5.19.	Recognition Rate dan Recognition Time Skema 4 untuk 2 Jaringan, 60% Data .....	90
Tabel 5.20.	Recognition Rate dan Recognition Time Skema 4 untuk 2 Jaringan, 70% Data .....	90
Tabel 5.21.	Recognition Rate dan Recognition Time Skema 4 untuk 2 Jaringan, 80% Data .....	90
Tabel 5.22.	Recognition Rate dan Recognition Time Skema 4 untuk 2 Jaringan, 90% Data .....	90
Tabel 5.23.	Recognition Rate dan Recognition Time Skema 4 untuk 2 Jaringan, 100% Data .....	91
Tabel 5.24.	Recognition Rate dan Recognition Time Skema 4 untuk 3 Jaringan, 50% Data .....	91
Tabel 5.25.	Recognition Rate dan Recognition Time Skema 4 untuk 3 Jaringan, 60% Data .....	91
Tabel 5.26.	Recognition Rate dan Recognition Time Skema 4 untuk 3 Jaringan, 70% Data .....	91
Tabel 5.27.	Recognition Rate dan Recognition Time Skema 4 untuk 3 Jaringan, 80% Data .....	92
Tabel 5.28.	Recognition Rate dan Recognition Time Skema 4 untuk 3 Jaringan, 90% Data .....	92
Tabel 5.29.	Recognition Rate dan Recognition Time Skema 4 untuk 3 Jaringan, 100% Data .....	92
Tabel 5.30.	Recognition Rate dan Recognition Time Skema 4 untuk 4 Jaringan, 50% Data .....	92
Tabel 5.31.	Recognition Rate dan Recognition Time Skema 4 untuk 4 Jaringan, 60% Data .....	93
Tabel 5.32.	Recognition Rate dan Recognition Time Skema 4 untuk 4 Jaringan, 70% Data .....	93
Tabel 5.33.	Recognition Rate dan Recognition Time Skema 4 untuk 4 Jaringan, 80% Data .....	93
Tabel 5.34.	Recognition Rate dan Recognition Time Skema 4 untuk 4 Jaringan, 90% Data .....	93
Tabel 5.35.	Recognition Rate dan Recognition Time Skema 4 untuk 4 Jaringan, 100% Data .....	94
Tabel 5.36.	Recognition Rate dan Recognition Time Skema 4 untuk 5 Jaringan, 50% Data .....	94
Tabel 5.37.	Recognition Rate dan Recognition Time Skema 4 untuk 5 Jaringan, 60% Data .....	94

Tabel 5.38.	Recognition Rate dan Recognition Time Skema 4 untuk 5 Jaringan, 70% Data .....	94
Tabel 5.39.	Recognition Rate dan Recognition Time Skema 4 untuk 5 Jaringan, 80% Data .....	95
Tabel 5.40.	Recognition Rate dan Recognition Time Skema 4 untuk 5 Jaringan, 90% Data .....	95
Tabel 5.41.	Recognition Rate dan Recognition Time Skema 4 untuk 5 Jaringan, 100% Data .....	95
Tabel 5.42.	Tabel Perbandingan Recognition Rate (Testing) untuk Keempat Skema .....	105
Tabel 5.43.	Tabel Perbandingan Training Time untuk Keempat Skema .....	107
Tabel 5.44.	Tabel Perbandingan Recognition Time untuk Keempat Skema .....	107
Tabel 5.45.	Recognition Rate dan Recognition Time pada Parameter Default .....	109
Tabel 5.46.	Recognition Rate dan Recognition Time pada Parameter Optimalisasi .....	109
Tabel 5.47.	Recognition Rate per Kelas pada Parameter Optimalisasi Jaringan Syaraf Ensemble .....	110

## DAFTAR GAMBAR

Gambar 2.1.	Struktur Jaringan Syaraf Tiruan dengan Lapisan Tersembunyi.....	8
Gambar 2.2.	Arsitektur Ensemble Neural Network.....	12
Gambar 2.3.	Skema Dependent Framework dan Independent Framework .....	14
Gambar 2.4.	Foto Berukuran 40x30 Pixel yang Dapat Dituliskan ke Dalam Bentuk Matriks.....	21
Gambar 2.5.	Matriks Gambar Berukuran 40x30 Pixel yang Ditata Menjadi Matriks Berukuran 1x1200 .....	21
Gambar 3.1.	Skema Penelitian.....	25
Gambar 3.2.	Kamera CCD Infrared Sketsa dan Disain .....	26
Gambar 4.1.	Arsitektur Jaringan Syaraf Tiruan.....	34
Gambar 4.2.	Sepuluh Kelas (Objek) yang Digunakan untuk Pelatihan.....	37
Gambar 4.3.	Grafik Error terhadap Epoch.....	42
Gambar 4.4.	Grafik Error terhadap Epoch Tanpa PCA dan Dengan PCA.....	46
Gambar 4.5.	Grafik Nilai Eigen Terbesar sampai Terkecil .....	47
Gambar 4.6.	Grafik Error terhadap Epoch Metode Acak dan Metode Nguyen- Widrow.....	50
Gambar 4.7.	Grafik Recognition Rte terhadap Jumlah Unit Lapisan Tersembunyi.....	52
Gambar 4.8.	Grafik Error terhadap Epoch dengan Jumlah Unit Lapisan Tersembunyi 10 Neuron, 100 Neuron, dan 500 Neuron.....	54
Gambar 4.9.	Grafik Error terhadap Epoch 100 Epoch, 1000 Epoch, 10000 Epoch.....	56
Gambar 4.10.	Grafik Recognition Rate vs. Alpha .....	57
Gambar 4.11.	Grafik Error terhadap Epoch $\alpha = 0.1$ , $\alpha = 0.5$ , dan $\alpha = 0.9$ .....	59
Gambar 4.12.	Grafik Recognition Rate vs. Momentum .....	61
Gambar 4.13.	Grafik Error terhadap Epoch untuk $\alpha = 0.5$ Tanpa Momentum $\mu = 0$ , $\mu = 0.5$ , dan $\mu = 0.9$ .....	63
Gambar 4.14.	Grafik Recognition Rate Training dengan Variasi $\alpha$ dan $\mu$ .....	64
Gambar 4.15.	Grafik Recognition Rate Testing dengan Variasi $\alpha$ dan $\mu$ .....	65
Gambar 4.16.	Grafik Error terhadap Epoch Parameter Default dan Parameter Optimalisasi .....	71
Gambar 5.1.	Grafik Recognition Rate dan Recognition Time terhadap Jumlah Network.....	73
Gambar 5.2.	Grafik Error terhadap Epoch Skema 1 untuk Jumlah Jaringan = 4 .....	74
Gambar 5.3.	Grafik Recognition Rate Training untuk Skema 2.....	77
Gambar 5.4.	Grafik Recognition Rate Testing untuk Skema 2 .....	77
Gambar 5.5.	Grafik Waktu Pembelajaran (Training Time) untuk Skema 2 .....	78
Gambar 5.6.	Grafik Waktu Pengenalan (Recognition Time) untuk Skema 2.....	78

Gambar 5.7. Grafik Error terhadap Epoch Skema 2 untuk Jumlah Jaringan = 3, Persentase 100%.....	81
Gambar 5.8. Grafik Recognition Rate Testing untuk Skema 3 .....	84
Gambar 5.9. Grafik Training Time untuk Skema 3 .....	84
Gambar 5.10. Grafik Recognition Time untuk Skema 3 .....	85
Gambar 5.11. Grafik Error terhadap Epoch Skema 3 untuk Jumlah Jaringan = 3 .....	87
Gambar 5.12. Grafik Recognition Rate (Training) Skema 4 untuk Jumlah Jaringan = 1 .....	96
Gambar 5.13. Grafik Recognition Rate (Testing) Skema 4 untuk Jumlah Jaringan = 1 .....	97
Gambar 5.14. Grafik Recognition Rate (Training) Skema 4 untuk Jumlah Jaringan = 2 .....	98
Gambar 5.15. Grafik Recognition Rate (Testing) Skema 4 untuk Jumlah Jaringan = 2 .....	99
Gambar 5.16. Grafik Recognition Rate (Testing) Skema 4 untuk Jumlah Jaringan = 3 .....	100
Gambar 5.17. Grafik Recognition Rate (Testing) Skema 4 untuk Jumlah Jaringan = 4 .....	101
Gambar 5.18. Grafik Recognition Rate (Testing) Skema 4 untuk Jumlah Jaringan = 5 .....	102
Gambar 5.19. Grafik Error terhadap Epoch Skema 4 untuk Jumlah Jaringan = 3, Persentase 100% $\lambda = 0.2$ dan $\lambda = 0.8$ .....	103
Gambar 5.20. Grafik Recognition Rate (Testing) untuk Keempat Skema .....	105

## BAB I PENDAHULUAN

Bab ini menjelaskan latar belakang penelitian yang dilakukan pada skripsi, permasalahan yang diteliti, tujuan penelitian, manfaat penelitian, batasan masalah penelitian, tahapan penelitian, serta sistematika penulisan laporan skripsi.

### 1.1. Latar Belakang

Semasa hidupnya, manusia mampu membedakan dan mengenali ribuan wajah yang berbeda. Hal ini disebabkan adanya syaraf-syaraf “*visual cortex*” pada otak manusia yang dapat memroses dan mempersepsikan bentuk wajah yang berbeda-beda. Seseorang dapat dengan mudah dan cepat mengenali wajah orang yang dikenalnya walaupun ekspresi dan kondisi wajah tersebut berbeda dari keadaan yang dikenalnya sebelumnya.<sup>1</sup>

Namun di masa modern ini, mesin dan komputer pun dirancang dengan suatu sistem cerdas agar dapat melakukan pengenalan wajah seperti halnya yang dapat dilakukan otak manusia. Teknologi pengenalan wajah semakin dimanfaatkan dan dikembangkan dengan aplikasi yang sangat luas cakupannya, seperti sistem keamanan, sistem yang dapat mengizinkan orang tertentu mengakses area terbatas, ataupun sistem absensi pegawai.

Pengenalan wajah secara umum mempunyai dua aplikasi utama yaitu verifikasi dan identifikasi. Verifikasi bertujuan untuk mencocokkan data baru seseorang dengan data yang ada di *database (one to one)* dan umumnya menghasilkan dua keadaan yaitu true atau false. Sedangkan identifikasi mengenali seseorang dengan keputusan berdasarkan tingkat kedekatan atau kemiripan. Kemampuan mengukur karakteristik fisik atau perilaku yang dapat digunakan untuk memverifikasi atau mengidentifikasi seseorang disebut dengan biometrik.<sup>2</sup>

---

<sup>1</sup> Alice O’Toole, *Handbook of Remote Biometrics, Advances in Pattern Recognition, Chapter 5* (London: Springer-Verlag), hal.111.

<sup>2</sup> Abdul Fadlil, *Sistem Pengenalan Wajah Manusia Secara Realtime Menggunakan Jaringan Syaraf Tiruan* (Seminar Nasional Aplikasi Teknologi Informasi 2008), hal. J-5

Sistem pengenalan wajah yang sifatnya realtime menjadi kebutuhan teknologi yang berorientasi ke masa depan. Teknologi di masa depan akan semakin cerdas, cepat, efisien dan praktis. Penelitian ini menggunakan jaringan syaraf tiruan tunggal dan ensemble dengan berbagai variabel untuk melakukan pengenalan wajah.

Sistem pengenalan wajah meliputi 4 tahap pemrosesan yaitu pra-pemrosesan (preprocessing), ekstraksi fitur, pembelajaran (training) dan pengujian (testing). Proses pra-pemrosesan dilakukan dengan menggunakan analisis proyeksi citra abu-abu (gray image), ekstraksi fitur menggunakan metode Principle Component Analysis (PCA), sedangkan training dan testing menggunakan jaringan syaraf tiruan (JST) dengan algoritma backpropagation.

Namun, pengolahan gambar yang diperoleh dari suatu kamera visible amatlah dipengaruhi oleh intensitas pencahayaannya. Hasil yang ditunjukkan akan berbeda jika gambar diambil pada keadaan yang terang dan keadaan yang gelap. Oleh karena itu dalam penelitian ini, penulis menggunakan kamera near infrared (infra merah) untuk mengambil gambar. Gambar yang dihasilkan kamera near infrared memiliki kualitas yang baik bahkan di tempat yang intensitas cahayanya sangat rendah sekalipun.

Pada penelitian seminar, penulis telah membuat sistem pengenalan wajah dengan jaringan syaraf tiruan (JST) tunggal dan algoritma backpropagation. Pada penelitian skripsi ini, penulis akan menitikberatkan penelitian pada metode terbaru yang merupakan perkembangan dari JST, yaitu jaringan syaraf ensemble (Ensemble Neural Networks – ENN). ENN mengkombinasikan beberapa neural network untuk mengambil keputusan.

Dari beberapa riset yang telah dilakukan, ENN dianggap memiliki kemampuan generalisasi yang lebih baik dibanding sekedar jaringan tunggal. Oleh karena itu, penulis melalui penelitian skripsi ini ingin mempelajari dan memahami sejauh mana ENN berpengaruh terhadap sistem pengenalan wajah.

## **1.2. Pernyataan Masalah**

Masalah yang akan dibahas pada laporan skripsi ini adalah penggunaan jaringan syaraf tunggal dan ensemble (ENN) dengan berbagai skema atau

variabel untuk melatih dan menguji suatu sistem pengenalan wajah dari gambar yang tertangkap suatu kamera near infrared. Selain itu penulis juga akan membahas penggunaan Principal Component Analysis (PCA) sebagai feature extraction untuk mereduksi data.

### 1.3. Tujuan Penulisan

Berikut ini adalah tujuan penelitian yang ingin dilakukan:

1. Memahami dan menganalisis bagaimana jaringan syaraf tiruan (JST) tunggal dengan algoritma backpropagation dapat digunakan untuk sistem pengenalan wajah (face recognition system) yang dapat mempelajari pola gambar wajah yang telah ada dan mengenali suatu gambar wajah yang lain.
2. Memahami dan menganalisis bagaimana PCA dapat digunakan untuk mereduksi dimensi suatu data yang berjumlah besar tanpa mengubah signifikansi data tersebut agar proses komputasi dapat berjalan lebih cepat.
3. Menganalisis bagaimana parameter-parameter pada JST tunggal dengan algoritma backpropagation berpengaruh terhadap hasil pelatihan dan pengujian.
4. Mengoptimalkan performa JST tunggal dengan algoritma backpropagation melalui perubahan parameter.
5. Memahami dan menganalisis bagaimana ensemble neural network (ENN) dengan skema algoritma backpropagation dapat digunakan untuk mengenali pola gambar wajah yang telah ada dan mengenali suatu gambar wajah yang lain.
6. Memahami dan menganalisis bagaimana ENN dengan skema parsialisasi data dan algoritma backpropagation dapat digunakan untuk mengenali pola gambar wajah yang telah ada dan mengenali suatu gambar wajah yang lain.
7. Memahami dan menganalisis bagaimana ENN dengan skema algoritma Negative Correlation Learning (NCL) dapat digunakan untuk mengenali pola gambar wajah yang telah ada dan mengenali suatu gambar wajah yang lain.

8. Memahami dan menganalisis bagaimana ENN dengan skema parsialisasi data dan algoritma NCL dapat digunakan untuk mengenali pola gambar wajah yang telah ada dan mengenali suatu gambar wajah yang lain.
9. Membandingkan performa ENN antara keempat skema yang dilakukan.
10. Mengoptimalkan performa ENN melalui perubahan parameter.

#### **1.4. Batasan Masalah**

Berikut ini adalah batasan-batasan masalah yang dilakukan pada penelitian:

1. Peneliti menggunakan kamera CCD yang memiliki sensor near infrared untuk merekam video dari objek dengan ekspresi yang berbeda-beda. Objek yang dijadikan sampel adalah sepuluh orang berusia 20-24 tahun dengan wajah berbeda, yang terdiri dari lima orang laki-laki dan lima orang perempuan. Dari video yang terekam, penulis secara manual memotong video tersebut untuk mendapatkan potongan gambar wajah yang diinginkan.
2. Jumlah kelas atau objek yang digunakan adalah sepuluh orang. Untuk proses training atau pelatihan, dari masing-masing kelas digunakan 20 gambar dan untuk proses testing atau pengujian, digunakan sepuluh gambar yang diambil dari database dan sepuluh gambar di luar database. Dengan demikian, database sistem akan memiliki data sebanyak  $10 \times 20 = 200$  gambar.

#### **1.5. Manfaat Penelitian**

Berikut ini adalah manfaat penelitian pada skripsi ini:

1. Penelitian diharapkan agar pembaca dapat memahami penggunaan jaringan syaraf tiruan tunggal dengan algoritma backpropagation dan jaringan syaraf ensemble (ENN) sebagai sarana atau tools untuk mengenali gambar wajah.
2. Penelitian dapat digunakan sebagai bahan pembelajaran untuk peneliti yang ingin mempelajari sistem pengenalan wajah secara umum ataupun dengan menggunakan kamera near infrared.

3. Penelitian dapat dijadikan acuan atau referensi untuk pengembangan aplikasi sistem pengenalan wajah dalam berbagai bidang, misalnya sistem sekuriti, absensi pegawai atau pelajar, dan sebagainya.

### 1.6. Tahapan Penelitian

Tahapan penelitian akan dimulai dari pengambilan sampel objek yaitu data foto gambar dari sepuluh orang dengan ekspresi yang berbeda-beda. Pengambilan data dilakukan dengan menggunakan kamera CCD infrared dengan jarak tertentu dan frontal terhadap objek. Setiap gambar yang diperoleh akan disesuaikan dengan memotong bagian wajahnya dan memperkecil ukuran atau dimensinya. Setelah itu data-data tersebut akan dimasukkan ke dalam database dengan mengubah setiap gambar menjadi matriks yang memiliki nilai intensitas RGB-nya dan kemudian mengubahnya ke format gray scale. Proses ini disebut dengan tahap *pre-processing*.

Untuk mempercepat komputasi, dilakukan proses Principal Component Analysis (PCA) untuk mereduksi dimensi input gambar. Dimensi gambar yang sebelumnya bernilai besar akan berkurang tanpa mengurangi signifikansi nilainya. Proses ini disebut dengan ekstraksi fitur.

Setelah mengumpulkan database berupa matriks gambar yang telah direduksi dimensinya, jaringan syaraf tiruan (baik tunggal maupun ensemble) digunakan untuk mengenali pola gambar-gambar tersebut, yang disebut dengan proses training. Setelah selesai melakukan proses training, dilakukan pengujian terhadap gambar-gambar uji baik yang ada di dalam database atau yang di luar database. Proses ini disebut dengan testing. Setelah itu, dilakukan perhitungan persentase kesesuaian antara gambar uji dengan target, yang disebut dengan recognition rate.

Semua komputasi jaringan syaraf tiruan backpropagation dan PCA dilakukan dengan membuat program dalam format *editor file* pada *software* bahasa pemrograman untuk komputasi numerik.

### 1.7. Sistematika Penulisan Laporan

Bab I, Pendahuluan, berisi tentang latar belakang penelitian, pernyataan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, dan tahapan penelitian.

Bab II, Tinjauan Pustaka, menjelaskan tentang teori jaringan syaraf tiruan dengan algoritma backpropagation, teori ENN, teori NCL, dan teori PCA yang akan digunakan pada sistem pengenalan wajah.

Bab III, Metodologi Penelitian, berisi tentang metode, skema, variabel-variabel, dan parameter-parameter yang digunakan dalam penelitian ini.

Bab IV, Analisis Performa Jaringan Syaraf Tunggal, yang merupakan pembahasan bagian I, menjelaskan secara detail hasil penelitian beserta analisisnya. Analisis ini mencakup analisis bagaimana JST dengan algoritma backpropagation dapat digunakan untuk sistem pengenalan wajah, analisis penggunaan PCA, analisis pengaruh parameter-parameter yang digunakan dalam penelitian, dan optimalisasi jaringan dengan memperhatikan recognition rate pada tahap pengujian dan grafik error terhadap epoch pada tahap pelatihan. Bab IV ini merupakan pembahasan pada laporan seminar yang telah diperbaiki.

Bab V, Analisis Performa Jaringan Syaraf Ensemble, yang merupakan pembahasan bagian II, menjelaskan hasil penelitian dan analisis ENN dengan skema algoritma backpropagation, parsialisasi data dengan algoritma backpropagation, algoritma NCL, dan parsialisasi data dengan algoritma NCL dapat digunakan untuk meningkatkan performa jaringan syaraf tunggal, perbandingan antara skema-skema yang digunakan, serta optimalisasi jaringan syaraf ensemble.

Bab VI, Kesimpulan dan Saran, berisi kesimpulan dari penelitian dan saran untuk pengembangan sistem pengenalan wajah ke depannya.

## BAB II

### TINJAUAN PUSTAKA

Bab ini menjelaskan tentang teori jaringan syaraf tiruan dengan algoritma backpropagation, teori ENN, teori NCL, dan teori PCA yang akan digunakan pada sistem pengenalan wajah.

#### 2.1. Jaringan Syaraf Tiruan dengan Algoritma Backpropagation

Sistem syaraf manusia merupakan suatu sistem biologis yang sangat canggih, di mana terdapat otak dengan strukturnya yang sangat kompleks dan neuron-neuron yang saling berhubungan dan menyampaikan informasi baik berupa rangsangan ataupun respon. Otak manusia mampu mengenali pola, melakukan perhitungan, dan mengontrol organ-organ tubuh.

Jaringan syaraf tiruan (JST) adalah sistem pemroses informasi yang memiliki karakteristik menyerupai jaringan syaraf manusia. Jaringan ini dapat dipakai atau dilatih untuk mengenali pola yang sudah sedikit berubah. Hal ini mirip dengan otak manusia yang masih mampu mengenali orang yang sudah lama tidak dijumpainya, dan mungkin wajahnya sudah berubah. Melalui pelatihan dengan menggunakan data matriks gambar wajah sebagai input dan vektor target yang telah ditentukan, jaringan syaraf tiruan dapat digunakan untuk mengenali gambar wajah seseorang.

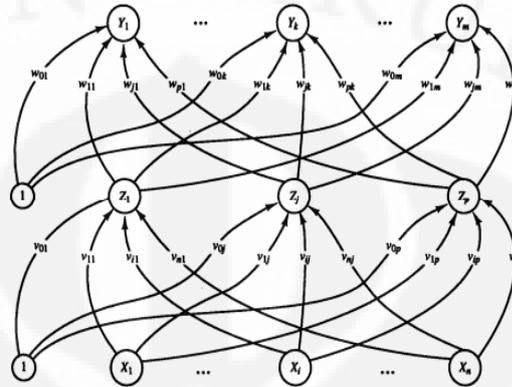
JST ini merupakan generalisasi dari pemodelan matematis dalam proses kognitif berdasarkan asumsi:

1. Pemrosesan informasi terjadi pada elemen sederhana yang dinamakan neuron.
2. Sinyal antar neuron berhubungan melalui saluran penghubung.
3. Setiap saluran penghubung mempunyai nilai bobot, dan melakukan operasi perkalian dengan sinyal yang ditransmisikan.
4. Setiap neuron memberlakukan fungsi aktivasi (biasanya tidak linier) pada masukan total untuk mendapatkan sinyal keluarannya.<sup>3</sup>

---

<sup>3</sup> Benyamin Kusumoputro, *Jaringan Neural Buatan* (Depok: Universitas Indonesia), hal.3-4.

Struktur jaringan syaraf tiruan yang digunakan terdiri dari tiga lapisan, yaitu lapisan input (input layer), lapisan tersembunyi (hidden layer), dan lapisan keluaran (output layer), seperti terlihat pada gambar di bawah ini.



**Gambar 2.1.** Struktur Jaringan Syaraf Tiruan dengan Lapisan Tersembunyi

Lapisan input terdiri dari nilai-nilai matriks yang membentuk pola gambar wajah, sedangkan lapisan output terdiri dari banyaknya kelas yang ingin diklasifikasikan dari semua gambar-gambar wajah yang ada, yang secara tak langsung terhubung pada siapa karakter di balik gambar wajah. Jumlah neuron pada lapisan tersembunyi, bobot-bobot yang menyatakan hubungan lapisan input dengan lapisan tersembunyi, dan bobot-bobot lapisan output dengan lapisan tersembunyi akan divariasikan pada saat pelatihan.

Algoritma pelatihan backpropagation merupakan jenis pelatihan yang paling sering digunakan pada jaringan syaraf tiruan. Pelatihan backpropagation ini terdiri dari tiga tahap, yaitu tahap propagasi maju (feedforward), propagasi balik (backward), dan perubahan bobot. Algoritma backpropagation menggunakan error output untuk mengubah nilai bobot-bobotnya dalam arah balik (backward). Untuk mendapatkan error ini, tahap propagasi maju harus dikerjakan terlebih dahulu.

Berdasarkan arsitektur jaringan pada gambar 2.1, selama proses propagasi maju, neuron input  $x_i$  ( $i=1,2,3,\dots,n$ ) menerima sinyal masukan dan mendistribusikannya ke tiap neuron pada lapisan tersembunyi. Pada lapisan tersembunyi, setiap neuron tersembunyi  $z_j$  ( $j=1,2,3,\dots,p$ ) akan menjumlahkan sinyal terbobot yang masuk dan menghitung sinyal aktivasi  $z_j$  yang dihasilkan.

Sinyal aktivasi ini kemudian dikirim sebagai masukan ke tiap neuron pada lapisan berikutnya, entah itu lapisan tersembunyi kedua (atau lapisan tersembunyi seterusnya) ataupun lapisan keluaran. Jika dikirim ke lapisan keluaran, maka setiap neuronnya  $y_k$  ( $k=1,2,3,\dots,m$ ) akan menjumlahkan sinyal terbobot yang masuk dan menghitung sinyal aktivasi  $y_k$  yang dihasilkan. Sinyal aktivasi pada lapisan keluaran ini adalah sinyal respon dari jaringan syaraf tiruan.

Secara teoretis, jaringan dengan sebuah lapisan tersembunyi sudah cukup bagi backpropagation untuk mengenali sembarang perkawanan antara masukan dan target dengan tingkat ketelitian yang ditentukan. Akan tetapi penambahan jumlah lapisan tersembunyi kadangkala membuat pelatihan lebih mudah.<sup>4</sup>

Selama proses training, tiap keluaran yang dihasilkan oleh neuron output akan dibandingkan dengan target atau respon yang dikehendaki untuk menentukan error yang akan dihasilkan. Error yang dihasilkan ini kemudian didistribusikan kembali ke lapisan-lapisan sebelumnya dan digunakan untuk memodifikasi bobot-bobot yang ada. Proses training melakukan perhitungan secara berulang-ulang sampai dihasilkan respon atau output jaringan syaraf tiruan yang sama dengan atau mendekati target dalam batas toleransi yang telah ditentukan atau jumlah iterasi pada proses ini melebihi jumlah iterasi yang diperkenankan.

Algoritma backpropagation dapat dituliskan seperti berikut:

1. Menentukan pola pada lapisan input ( $x_i$ ), menentukan target ( $t_k$ ) untuk tiap pola, dan menentukan berapa jumlah neuron lapisan tersembunyi dan laju pembelajaran.
2. Inisialisasi nilai-nilai bobot awal ( $w_{jk}$  dan  $v_{ij}$ ) pada lapisan tersembunyi dan lapisan keluaran.
3. Untuk setiap data pelatihan lakukan langkah 3 sampai 9.

**Propagasi maju**, di mana setiap neuron pada lapisan bawah menerima sinyal dan meneruskannya ke neuron pada lapisan di atasnya.

<sup>4</sup> Jong Jek Siang, *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan MATLAB*, (Yogyakarta: Penerbit Andi), hal.111

4. Menghitung semua output di lapisan tersembunyi  $z_j$  ( $j = 1, 2, 3, \dots, p$ ).

$$\text{keluaran: } z\_net_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (2.1)$$

$$\text{fungsi aktivasi: } z_j = f(z\_net_j) = \frac{1}{1 + e^{-z\_net_j}} \quad (2.2)$$

5. Menghitung semua output di lapisan keluaran  $y_k$  ( $k = 1, 2, 3, \dots, m$ ).

$$\text{keluaran: } y\_net_k = w_{0k} + \sum_{j=1}^p z_j w_{jk} \quad (2.3)$$

$$\text{fungsi aktivasi: } y_k = f(y\_net_k) = \frac{1}{1 + e^{-y\_net_k}} \quad (2.4)$$

### Propagasi balik

6. Menghitung faktor  $\delta$  neuron keluaran berdasarkan kesalahan di setiap neuron keluaran  $y_k$ .  $\delta_k$  merupakan unit kesalahan yang akan digunakan dalam perubahan bobot lapisan di bawahnya. Setelah itu, perubahan bobot  $w_{jk}$  dihitung dengan memasukkan faktor laju pembelajaran  $\alpha$ . Perubahan bobot ini akan digunakan untuk mengubah bobot  $w_{jk}$ .

$$\delta_k = (y_k - \hat{y}_k) f'(y\_net_k) = (y_k - \hat{y}_k) (-y_k) \quad (2.5)$$

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (k = 1, 2, 3, \dots, m \text{ dan } j = 0, 1, 2, 3, \dots, p) \quad (2.6)$$

7. Menghitung faktor  $\delta$  neuron tersembunyi berdasarkan kesalahan di setiap neuron tersembunyi.

$$\delta\_net_j = \sum_{k=1}^m \delta_k w_{jk} \quad (2.7)$$

$$\delta_j = \delta\_net_j f'(z\_net_j) = \delta\_net_j z_j (-z_j) \quad (2.8)$$

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (j = 1, 2, 3, \dots, p \text{ dan } i = 0, 1, 2, 3, \dots, n) \quad (2.9)$$

### Perubahan bobot

8. Menghitung semua perubahan bobot yang menuju ke lapisan keluaran dan lapisan tersembunyi.

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (k = 1, 2, 3, \dots, m \text{ dan } j = 0, 1, 2, 3, \dots, p) \quad (2.10)$$

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (j = 1, 2, 3, \dots, p \text{ dan } i = 0, 1, 2, 3, \dots, n) \quad (2.11)$$

9. Jika jumlah iterasi belum terpenuhi atau nilai Mean Square Error (MSE) belum dicapai, kembali ke langkah 3.

Bobot awal akan mempengaruhi apakah jaringan mencapai titik minimum lokal atau global, dan seberapa cepat konvergensinya. Bobot yang menghasilkan nilai turunan aktivasi yang kecil sedapat mungkin dihindari karena akan menyebabkan perubahan bobotnya menjadi sangat kecil. Demikian pula nilai bobot awal tidak boleh terlalu besar karena nilai turunan fungsi aktivasinya menjadi sangat kecil juga. Oleh karena itu dalam standar backpropagation, bobot dan bias diisi dengan bilangan acak kecil.<sup>5</sup>

Nguyen dan Widrow mengusulkan cara membuat inisialisasi bobot dan bias ke unit tersembunyi sehingga menghasilkan iterasi lebih cepat. Misal:

$n$  = jumlah unit masukan

$p$  = jumlah unit tersembunyi

$\beta$  = faktor skala =  $0.7 \sqrt{p}$

Algoritma inisialisasi Nguyen-Widrow adalah sebagai berikut:

1. Inisialisasi semua bobot ( $v_{ij}$  (lama)) dengan bilangan acak dalam interval  $[-0.5, 0.5]$

2. Hitung  $\|v_j\| = \sqrt{v_{1j}^2 + v_{2j}^2 + \dots + v_{nj}^2}$  (2.12)

3. Bobot yang dipakai sebagai inisialisasi,  $v_{ij} = \frac{\beta v_{ij}(\text{lama})}{\|v_j\|}$  (2.13)

4. Bias yang dipakai sebagai inisialisasi,  $v_{0j} =$  bilangan acak antara  $-\beta$  dan  $\beta$

Modifikasi bobot dapat dilakukan dengan penambahan momentum. Jika ada penambahan momentum, bobot baru pada waktu ke- $(t+1)$  didasarkan bobot pada waktu  $t$  dan  $(t-1)$ . Jika  $\mu$  adalah konstanta ( $0 < \mu < 1$ ) yang menyatakan parameter momentum maka bobot baru dihitung berdasarkan persamaan:

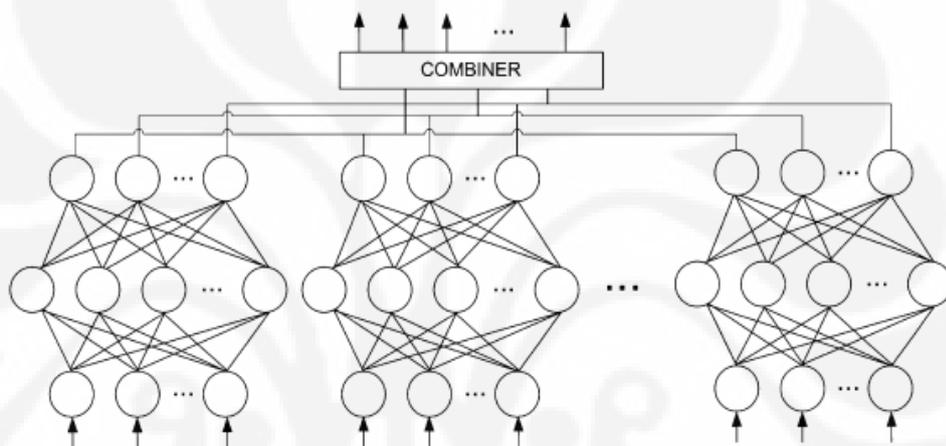
$$w_{jk}(t+1) = w_{jk}(t) + \alpha \delta_k z_j + \mu (w_{jk}(t) - w_{jk}(t-1)) \quad (2.14)$$

$$v_{ij}(t+1) = v_{ij}(t) + \alpha \delta_j x_i + \mu (v_{ij}(t) - v_{ij}(t-1)) \quad (2.15)$$

<sup>5</sup> Ibid, hal.109

## 2.2. Jaringan Syaraf Ensemble (Ensemble Neural Network)

Konsep jaringan syaraf ensemble atau *Ensemble Neural Network* (ENN) ini diajukan pada tahun 1990 oleh Lars Kai Hansen dan Peter Salamon. ENN mengkombinasikan beberapa jaringan syaraf tiruan (individual neural network) untuk dilatih bersamaan dan mengambil keputusan bersama. Sama seperti jaringan syaraf tunggal, setiap jaringan memiliki sebuah lapisan masukan, lapisan tersembunyi, dan lapisan keluaran. Sementara itu, pada ENN terdapat sebuah *combiner* yang berfungsi menentukan keputusan akhir. Berikut ini adalah arsitektur ENN.



Gambar 2.2. Arsitektur Ensemble Neural Network

Sama seperti prinsip jaringan syaraf individual, ENN juga merupakan *supervised learning* atau pembelajaran dengan pengarah. Tujuan utama dari supervised learning adalah untuk mengelompokkan pola (yang disebut dengan *instances*) ke dalam suatu kategori yang disebut kelas atau label.

Biasanya, suatu set instances diberikan, yang disebut juga dengan *training set*. Selain itu, label dari training set juga diketahui. Dengan demikian, jaringan dibuat sedemikian hingga tercipta model baru yang dapat melabelkan instances-instances baru. Algoritma untuk membangun model ini disebut dengan *inducer*, dan suatu instances dari suatu inducer pada suatu training set disebut dengan *classifier*.<sup>6</sup>

<sup>6</sup> Lior Rokach, *Ensemble-based Classifiers* (Springer Science and Business Media B.V. 2009), hal 1.

Konsep utama dari pembelajaran ensemble (*ensemble learning*) adalah menentukan bobot-bobot pada tiap classifier individual dan mengkombinasikan keluarannya sehingga hasil akhir yang diperoleh akan lebih baik daripada sekedar keluaran classifier individual.

Dalam bukunya yang berjudul “*The Wisdom of Crowds: Why the Many are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations*” tahun 2004, James Michael Surowiecki mengungkapkan bahwa dalam kondisi-kondisi tertentu, penggabungan informasi dari beberapa sumber akan menghasilkan keputusan yang lebih superior/tepat dibandingkan dengan keputusan yang dibuat sendiri, bahkan oleh seorang ahli.

Tidak semua kumpulan (*crowds*) itu bijaksana. Surowiecki menyatakan untuk membuat kumpulan tersebut menjadi bijaksana, ada empat kriteria yang harus dipenuhi setiap kumpulan<sup>7</sup>, yaitu:

1. Perbedaan opini (*diversity of opinion*), di mana setiap anggota memiliki informasi pribadi yang berbeda-beda, bahkan yang ekstrim dari kenyataan.
2. Ketidakterkaitan (*independence*), di mana opini setiap anggota tidak ditentukan oleh opini lain di sekitar mereka.
3. Desentralisasi (*decentralization*), di mana setiap anggota mampu mengelompokkan dan menentukan keputusan berdasarkan pengetahuan yang dimilikinya.
4. Penggabungan (*aggregation*), yaitu adanya suatu mekanisme yang menggabungkan keputusan/pertimbangan setiap anggota menjadi sebuah keputusan bersama.

Dari penjelasan di atas, maka dapat disimpulkan bahwa metode pengelompokan secara ensemble pada ENN membutuhkan:

1. *Training Set*, yaitu set data yang berisi pola dan target yang telah ditentukan.

$$A = [a_1, \dots a_i, \dots a_n] ; y = \text{kelas/target}$$

2. *Base Inducer*, yaitu algoritma yang dapat melakukan pengelompokan.

$$M = I(S)$$

---

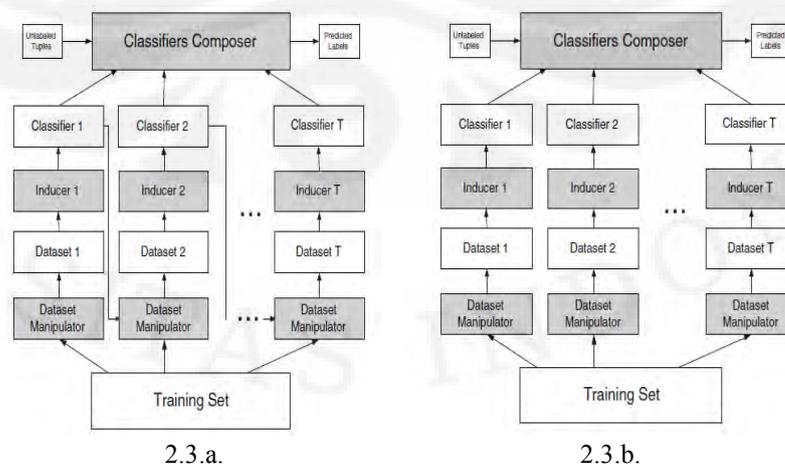
<sup>7</sup> Ibid, hal. 2.

Persamaan di atas dapat dinyatakan sebagai berikut: suatu classifier  $M$  diinduksikan inducer  $I$  pada training set  $S$ .

3. *Diversity Generator*, yaitu komponen untuk menghasilkan classifier yang bervariasi.
4. *Combiner*, yaitu suatu langkah yang dapat menggabungkan keputusan-keputusan dari setiap classifier yang bervariasi.

Secara umum, terdapat dua jenis jaringan ensemble, yaitu *dependent framework* dan *independent framework*. Pada *dependent framework*, output dari suatu classifier digunakan untuk membangun classifier berikutnya. Dengan kata lain, pengetahuan yang didapat dari iterasi sebelumnya digunakan untuk iterasi selanjutnya. Contoh algoritma yang sering diterapkan pada *dependent framework* antara lain *boosting* dan turunannya, *adaboost* (*adaptive boosting*).

Sedangkan pada *independent framework*, set data asli diubah menjadi beberapa set data dan dilatih pada classifier yang berbeda-beda. Set data yang baru mungkin saling berbeda (*mutually exclusive*) atau bisa saja memiliki irisan diantaranya (*overlapping*). Karena output dari suatu classifier tidak digunakan untuk membangun classifier lainnya, maka jaringan seperti ini tidak memiliki ketergantungan, sehingga kombinasi dari beberapa inducer dapat diterapkan di sini. Contoh algoritma yang sering digunakan pada *independent framework* antara lain *bagging* (*bootstrap aggregating*), *wagging*, *random forest*, dan *cross-validated committees* (CVC). Di bawah ini adalah skema dari *dependent* dan *independent framework*.



**Gambar 2.3.** Skema (a) *dependent framework* dan (b) *independent framework*

Untuk mengambil keputusan bersama, ada beberapa metode yang umum digunakan sebagai combiner, antara lain:

1. Metode rata-rata

$$H(x) = \max_{k=1,2,\dots,C} \left[ \frac{1}{N} \sum_{j=1}^N Y_j(x) \right] \quad (2.16)$$

2. Metode nilai maksimum

$$H(x) = \max_{k=1,2,\dots,C} \left[ \max \left[ Y_1(x), Y_2(x), \dots, Y_N(x) \right] \right] \quad (2.17)$$

3. Metode perkalian produk

$$H(x) = \max_{k=1,2,\dots,C} \left[ \prod_{j=1}^N Y_j(x) \right] \quad (2.18)$$

4. Metode voting

$$H(x) = \max_{k=1,2,\dots,C} \left[ \sum_{j=1}^N B_j(x) \right] \quad (2.19)$$

Di mana B adalah jumlah kelas yang diurutkan di bawah kelas k pada jaringan ke-j. Perhitungan ini disebut dengan *Borda Count*.<sup>8</sup>

### 2.3. Algoritma Negative Correlation Learning

Seperti yang telah disebutkan di atas, salah satu kriteria *wisdom of crowds* adalah *diversity of opinion*, di mana untuk menghasilkan suatu keputusan bersama yang tepat maka diperlukan perbedaan opini di antara anggotanya. Sama seperti halnya ENN, untuk membuat keputusan akhir yang akurat, diperlukan classifier yang bervariasi, yang dihasilkan oleh suatu *diversity generator*. Liu dan Yao (1999) mengajukan sebuah metode untuk mendiversifikasi pelatihan pada ENN yaitu dengan Negative Correlated Learning (NCL).

Konsep di balik NCL bertujuan untuk melatih setiap jaringan individual berbeda di dalam ENN dengan bagian atau aspek yang berbeda dari suatu training set sehingga jaringan ensemble dapat melakukan proses pembelajaran dengan lebih baik. Pada NCL, semua jaringan individual pada jaringan ensemble dilatih secara berkesinambungan dengan suatu *penalty term*

<sup>8</sup> Lee Heesung et.al, *Neural Network Ensemble with Probabilistic Fusion and Its Application to Gait Recognition* (Elsevier BV. 2008), hal. 1558.

yang terdapat pada fungsi errornya. NCL mencoba melatih dan menggabungkan jaringan-jaringan individual pada suatu proses pelatihan yang sama.<sup>9</sup>

Berikut ini adalah algoritma NCL.<sup>10</sup> Pada awalnya, anggap sebuah *training set* yang didefinisikan sebagai berikut:

$$D = \left\{ \left( \underset{1}{x}, \underset{1}{d} \right), \left( \underset{2}{x}, \underset{2}{d} \right), \dots, \left( \underset{N}{x}, \underset{N}{d} \right) \right\} \quad (2.20)$$

di mana  $x \in R^p$  adalah input jaringan yang berupa vektor,  $d$  adalah output yang diharapkan (target) yang berupa skalar.  $N$  adalah jumlah set data yang digunakan dalam training set. Target dibuat berupa skalar sekedar untuk penyederhanaan saja. Setiap bobot pada masing-masing jaringan individual diberikan secara acak dan berbeda. Setiap jaringan individual ke- $i$  memiliki keluaran  $F_i(n)$  dan output dari keseluruhan jaringan pada set data ke- $n$  disebut dengan  $F(n)$ , yang dinyatakan sebagai berikut:

$$F(n) = \frac{1}{M} \sum_{i=1}^M F_i(n) \quad (2.21)$$

Pada persamaan (2.21),  $M$  menyatakan jumlah jaringan individual yang digunakan.

Seperti yang telah disinggung sebelumnya, tujuan NCL adalah menghasilkan jaringan yang bervariasi dengan menambahkan suatu penalty term pada fungsi error masing-masing jaringannya. Fungsi error tersebut dinyatakan dengan:

$$E_i = \frac{1}{N} \sum_{n=1}^N E_i(n) = \frac{1}{N} \sum_{n=1}^N \frac{1}{2} \left( F_i(n) - d(n) \right)^2 + \frac{1}{N} \sum_{n=1}^N \lambda p_i(n) \quad (2.22)$$

di mana  $E_i(n)$  adalah nilai fungsi error jaringan ke- $i$  pada pola pelatihan ke- $n$ . Suku pertama pada fungsi error di atas adalah fungsi *empirical risk* dari jaringan ke- $i$ . Suku kedua mengandung  $p_i$  yang adalah fungsi penalti. Tujuan meminimalkan  $p_i$  adalah untuk mengkorelasikan error masing-masing jaringan secara negatif terhadap error jaringan-jaringan lainnya. Parameter  $\lambda$  yang berkisar antara  $0 \leq \lambda \leq 1$  berfungsi untuk mengatur kekuatan penalti. Fungsi penalti sendiri dapat dinyatakan sebagai berikut:

<sup>9</sup> Liu Y. dan Yao X., *Ensemble Learning via Negative Correlation*, (Elsevier Science Ltd.), hal.1399.

<sup>10</sup> Fernanda Li Minku et.al., *Negative Correlation in Incremental Learning*, (Springer Science and Business Media B.V. 2007), hal.292.

$$p_i(n) = \left( F_i(n) - F(n) \sum_{j \neq i}^M \left( F_j(n) - F(n) \right) \right) \quad (2.23)$$

Sedangkan turunan parsial dari  $E_i(n)$  terhadap keluaran jaringan ke- $i$  pada pola ke- $n$  dinyatakan sebagai berikut:

$$\frac{\partial E_i(n)}{\partial F_i(n)} = F_i(n) - d(n) + \lambda \frac{\partial p_i(n)}{\partial F_i(n)} \quad (2.24a)$$

$$\begin{aligned} &= F_i(n) - d(n) + \lambda \sum_{j \neq i}^M \left( F_j(n) - F(n) \right) \\ &= F_i(n) - d(n) + \lambda \left( MF(n) - F_i(n) - (M-1)F(n) \right) \\ &= F_i(n) - d(n) + \lambda \left( F(n) - F_i(n) \right) \\ &= F_i(n) - d(n) - \lambda \left( F_i(n) - F(n) \right) \\ &= \left( -\lambda \left( F_i(n) - d(n) \right) + \lambda \left( F(n) - d(n) \right) \right) \end{aligned} \quad (2.24b)$$

Selanjutnya, algoritma backpropagation digunakan untuk memodifikasi bobot pada setiap jaringan dengan mengacu pada persamaan 2.24. Penurunan rumus untuk perubahan bobot  $v_{ij}$  dan  $w_{jk}$  tiap jaringan terdapat pada lampiran A. NCL dari persamaan tersebut tidak lain adalah perluasan sederhana dari algoritma backpropagation, yang hanya memerlukan perhitungan suku tambahan  $\lambda \left( F(n) - F_i(n) \right)$  pada jaringan ke- $i$ .

#### 2.4. Principal Component Analysis (PCA)

*Principal Component Analysis* (PCA) adalah suatu metode statistik yang dapat mengidentifikasi pola suatu data berdimensi tinggi/banyak, dan menyatakannya dengan memperhatikan kesamaan dan ketidaksamaannya. Pola-pola pada data yang berdimensi tinggi kadang sulit ditemukan dan grafisnya sulit digambarkan, oleh sebab itu PCA digunakan sebagai alat yang sangat bermanfaat untuk menganalisis data.

Konsep utama dari PCA adalah mereduksi dimensi dari suatu set data yang terdiri dari banyak variabel, di mana kemungkinan-kemungkinan variasi sebisa mungkin dipertahankan. Hal ini dicapai dengan mentransformasi set data tersebut menjadi suatu set variabel yang satu sama lain tidak berhubungan, yang disebut *Principal Components* (PCs). PC ini merupakan

kumpulan beberapa variasi yang diurutkan dan paling dipilih dari semua variabel asli.<sup>11</sup>

Secara matematis, PCA melakukan transformasi linear dari data  $X$  yang berdimensi  $D$  ke suatu data baru  $Y$  yang berdimensi  $d$  dan tidak berkorelasi, di mana  $d \leq D$ .<sup>12</sup> Keuntungan PCA adalah ketika pola pada data telah ditemukan dan data telah dimampatkan dengan memperkecil jumlah dimensinya, informasi yang terkandung di dalamnya tidak akan mengalami perubahan yang signifikan. Metode ini sering digunakan dalam image compression.

Sebelum memahami langkah-langkah dalam melakukan PCA, diperlukan pengetahuan tentang konsep matematis/statistik terlebih dahulu. Berikut ini adalah beberapa konsep yang digunakan dalam PCA.

a. Mean

Mean adalah nilai rata-rata dari suatu sampel, yang dinyatakan sebagai berikut:

$$\bar{X} = \frac{\sum_{i=1}^n X_i}{n} \quad (2.25)$$

b. Standar deviasi

Nilai mean dapat menunjukkan nilai atau kondisi tengah dari suatu sampel, tetapi tidak dapat menunjukkan persebaran di dalamnya. Untuk itu, diperlukan suatu standar deviasi, yaitu nilai jarak rata-rata dari mean terhadap sebuah nilai dalam suatu data set. Standar deviasi dapat dinyatakan sebagai berikut:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n-1}} \quad (2.26)$$

c. Nilai z

Dalam ilmu statistik, nilai  $z$  menunjukkan seberapa banyak standar deviasi berada di atas atau di bawah nilai rata-rata/mean. Konversi data

<sup>11</sup> I.T. Jolliffe, *Principal Component Analysis second edition* (Springer Series in Statistics), hal.1.

<sup>12</sup> Samarasena Buchala, *Principal Component Analysis of Gender, Ethnicity, Age, and Identity of Face Images*.

ini sering disebut normalisasi atau standarisasi data dan dinyatakan sebagai berikut:

$$z = \frac{X - \bar{X}}{s} \quad (2.27)$$

d. Varians dan Kovarians

Varians adalah nilai yang juga dapat menunjukkan persebaran suatu data, dan dinyatakan sebagai berikut:

$$s^2 = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n-1} \quad (2.28)$$

atau

$$\text{var}(X, X) = \frac{\sum_{i=1}^n (X_i - \bar{X})(X_i - \bar{X})}{n-1} \quad (2.29)$$

Dapat dilihat bahwa standar deviasi merupakan akar kuadrat dari varians suatu sampel data. Baik varians maupun standar deviasi dapat digunakan untuk mengukur penyebaran data, namun masih terbatas pada satu dimensi suatu set data saja, misalnya tinggi badan mahasiswa atau nilai IQ mahasiswa suatu universitas.

Namun, banyak data set yang terdiri dari lebih dari satu dimensi, misalnya set data yang terdiri dari tinggi badan dan nilai IQ mahasiswa suatu universitas. Kita ingin melihat keterkaitan atau hubungan dari antara kedua dimensi ini. Kovarians adalah parameter yang digunakan untuk mengukur seberapa jauh persebaran nilai suatu dimensi terhadap mean dari dimensi lain. Kovarians dimensi X terhadap dimensi Y (dua dimensi) dapat dinyatakan sebagai berikut:

$$\text{cov}(X, Y) = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{n-1} \quad (2.30)$$

Nilai kovarians ini menyerupai nilai varians, hanya saja nilai  $(Y_i - \bar{Y})$  pada kovarians digantikan dengan  $(X_i - \bar{X})$  pada varians. Perlu diketahui bahwa nilai  $\text{cov}(X, Y)$  sama dengan  $\text{cov}(Y, X)$  dan kovarians hanya dapat diukur antara dua dimensi saja.

Untuk mengukur kovarians yang lebih dari dua dimensi, maka terdapat lebih dari satu nilai kovarians yang dapat dicari. Untuk set data berdimensi tiga (misalnya dimensi  $X$ ,  $Y$ , dan  $Z$ ), terdapat tiga nilai kovarians yang dapat dicari yaitu  $cov(X,Y)$ ,  $cov(X,Z)$  dan  $cov(Y,Z)$ . Secara umum bila terdapat  $n$  dimensi, maka banyaknya nilai kovarians yang dapat dicari adalah  $\frac{n!}{(n-2)! \times 2}$ .<sup>13</sup>

Untuk memudahkan perhitungan selanjutnya, nilai-nilai kovarians dapat dimasukkan menjadi suatu matriks yang disebut dengan matriks kovarians. Matriks kovarians dapat didefinisikan sebagai berikut:

$$C^{n \times n} = \begin{bmatrix} c_{1,1} & c_{1,2} & \dots & c_{1,n} \\ c_{2,1} & c_{2,2} & \dots & c_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n,1} & c_{n,2} & \dots & c_{n,n} \end{bmatrix} \quad \text{atau} \quad C = \begin{bmatrix} cov(Dim_1, Dim_1) & cov(Dim_1, Dim_2) & \dots & cov(Dim_1, Dim_n) \\ cov(Dim_2, Dim_1) & cov(Dim_2, Dim_2) & \dots & cov(Dim_2, Dim_n) \\ \vdots & \vdots & \ddots & \vdots \\ cov(Dim_n, Dim_1) & cov(Dim_n, Dim_2) & \dots & cov(Dim_n, Dim_n) \end{bmatrix} \quad (2.32)$$

Dari persamaan di atas dapat kita lihat bahwa nilai diagonal dari matriks kovarians adalah nilai varians dari masing-masing dimensinya dan matriks ini adalah matriks persegi yang elemen-elemennya simetris terhadap diagonalnya.

#### e. Vektor Eigen dan Nilai Eigen

Jika  $A$  adalah suatu matriks persegi  $n \times n$ , maka vektor tak-nol  $\mathbf{x}$  pada suatu ruang vektor  $R^n$  disebut sebagai suatu vektor eigen dari  $A$  jika  $A\mathbf{x}$  adalah suatu penggandaan skalar dari  $\mathbf{x}$  yaitu:

$$A\mathbf{x} = \lambda\mathbf{x} \quad \text{atau ekuivalen dengan} \quad (I - A)\mathbf{x} = \mathbf{0} \quad (2.33)$$

untuk suatu skalar  $\lambda$ . Skalar  $\lambda$  disebut sebagai nilai eigen dari  $A$ , dan  $\mathbf{x}$  disebut suatu vektor eigen dari  $A$  yang berpadanan dengan  $\lambda$ .<sup>14</sup>

Setelah mengetahui konsep-konsep matematis yang digunakan dalam PCA, berikut ini adalah penjelasan mengenai algoritma PCA pada sistem

<sup>13</sup> Lindsay I Smith, *A Tutorial on Principal Component Analysis*, hal.6.

<sup>14</sup> Howard Anton, *Dasar-Dasar Aljabar Linear Jilid2* (Batam: Interaksara), hal.100.

pengenal wajah. Pada penelitian ini, PCA hanya dilakukan pada tahap ekstraksi fitur, yaitu untuk memampatkan dimensi dari input gambar yang jumlahnya besar agar menjadi lebih kecil sehingga mempercepat komputasi.

### 1. Memperoleh data

Dalam sistem pengenal wajah ini, data yang diperoleh merupakan intensitas gray scale dari suatu gambar foto yang berukuran 40x30 pixel. Intensitas-intensitas ini dijadikan suatu matriks yang berukuran 40x30. Jadi, terdapat  $40 \times 30 = 1200$  elemen dalam matriks yang akan menjadi data input untuk satu gambar.



$$= \begin{bmatrix} x_{1,1} & \cdots & x_{1,30} \\ \vdots & \ddots & \vdots \\ x_{40,1} & \cdots & x_{40,30} \end{bmatrix}$$

**Gambar 2.4.** Foto Berukuran 40x30 Pixel yang Dapat Dituliskan ke Dalam Bentuk Matriks

Untuk memudahkan perhitungan, kita menata ulang elemen-elemen tersebut sehingga menjadi satu baris yang memiliki 1200 kolom (atau bisa saja satu kolom yang memiliki 1200 baris), seperti yang ditunjukkan di bawah ini:

$$\begin{bmatrix} x_{1,1} & \cdots & x_{1,30} \\ \vdots & \ddots & \vdots \\ x_{40,1} & \cdots & x_{40,30} \end{bmatrix} = [x_{1,1} \cdots x_{1,30} x_{2,1} \cdots x_{2,30} \cdots x_{40,1} \cdots x_{40,30}]$$

$$= [x_{1,1} \cdots x_{1,1200}]$$

**Gambar 2.5.** Matriks Gambar Berukuran 40x30 Pixel yang Ditata Menjadi Matriks Berukuran 1x1200

Dengan demikian, sebuah data gambar memiliki 1200 elemen atau secara matematis kita sebut dimensi. Katakan jumlah target (cluster) adalah  $m$  dan jumlah gambar tiap target untuk pelatihan adalah  $n$ , dan  $t = m \times n$ . Jika seluruh database dituliskan dalam suatu matriks, matriks tersebut akan berukuran  $t \times 1200$  (atau bisa juga  $1200 \times t$ ), seperti berikut:

$$database = \begin{bmatrix} x_{1,1} & \cdots & x_{1,1200} \\ x_{2,1} & \cdots & x_{2,1200} \\ \vdots & \ddots & \vdots \\ x_{t,1} & \cdots & x_{t,1200} \end{bmatrix} \quad (2.34)$$

## 2. Modifikasi nilai database

Awalnya, kita menghitung nilai mean atau rata-rata dari dimensi database. Karena data setiap gambar memiliki 1200 dimensi, maka akan terdapat 1200 nilai rata-rata. Supaya PCA dapat bekerja dengan baik, maka setiap elemen data di dalam database dimodifikasi nilainya sedemikian hingga nilai rata-ratanya sama dengan nol.<sup>15</sup> Ada dua cara melakukan modifikasi ini, yaitu dengan:

### a. Mengurangi setiap data dengan nilai mean-nya

Dengan mengurangi setiap data dengan nilai rata-ratanya, maka akan terbentuk database baru yang nilainya telah termodifikasi dan memiliki nilai rata-rata baru yang sama dengan nol. Cara ini merupakan cara yang paling sederhana.

$$\text{database} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,1200} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,1200} \\ \vdots & \vdots & \ddots & \vdots \\ x_{t,1} & x_{t,2} & \cdots & x_{t,1200} \end{bmatrix} \Rightarrow N = \begin{bmatrix} \overbrace{x_{1,1} - \bar{x}_1} & \overbrace{x_{1,2} - \bar{x}_2} & \cdots & \overbrace{x_{1,1200} - \bar{x}_{1200}} \\ \overbrace{x_{2,1} - \bar{x}_1} & \overbrace{x_{2,2} - \bar{x}_2} & \cdots & \overbrace{x_{2,1200} - \bar{x}_{1200}} \\ \vdots & \vdots & \ddots & \vdots \\ \overbrace{x_{t,1} - \bar{x}_1} & \overbrace{x_{t,2} - \bar{x}_2} & \cdots & \overbrace{x_{t,1200} - \bar{x}_{1200}} \end{bmatrix} \quad (2.35)$$

### b. Normalisasi nilai z

Dengan mengubah setiap elemen database ke dalam nilai z, maka akan terbentuk sebuah database baru yang juga memiliki nilai rata-rata baru yang sama dengan nol. Matriks nilai z dapat dituliskan sebagai berikut:

$$\text{database} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,1200} \\ x_{2,1} & \cdots & x_{2,1200} \\ \vdots & \ddots & \vdots \\ x_{t,1} & \cdots & x_{t,1200} \end{bmatrix} \Rightarrow Z = \begin{bmatrix} z_{1,1} & \cdots & z_{1,1200} \\ z_{2,1} & \cdots & z_{2,1200} \\ \vdots & \ddots & \vdots \\ z_{t,1} & \cdots & z_{t,1200} \end{bmatrix} \quad (2.36)$$

## 3. Menghitung matriks kovarians

Setelah mendapatkan database baru yang nilainya telah dimodifikasi, langkah selanjutnya adalah mencari nilai-nilai kovarians dari setiap dimensi yang satu dengan dimensi yang lainnya. Nilai-nilai kovarians ini dituliskan ke dalam suatu matriks kovarians.

<sup>15</sup> Lindsay I Smith, Op.Cit, hal. 12.

Apabila database baru diperoleh dengan mengurangi setiap data dengan mean-nya, maka matriks kovarians dapat diperoleh dengan:

$$C_N = \frac{N^T N}{(n-1)} \quad (2.37)$$

Namun apabila database baru diperoleh melalui normalisasi nilai  $z$ , maka matriks kovarians dapat diperoleh dengan:

$$C_Z = \frac{Z^T Z}{(n-1)} \quad (2.38)$$

Perbedaan  $C_N$  dan  $C_Z$  terletak pada nilai entri diagonalnya, di mana  $C_Z$  memiliki nilai entri diagonal atau varians-variens yang sama dengan satu. Hal ini disebabkan karena berlakunya hubungan:

$$C_{Z(i,j)} = \frac{C_{N(i,j)}}{\text{var}(\text{database}_{(i,j)})} \quad (2.39)$$

Pada penelitian, baik  $C_N$  ataupun  $C_Z$  dapat digunakan dalam proses PCA.

#### 4. Menghitung vektor eigen dan nilai eigen dari matriks kovarians

Karena matriks kovarians berbentuk persegi, maka kita dapat mencari semua vektor-vektor eigen dan nilai eigen yang berpasangan dengan vektornya. Nilai dan vektor eigen ini merupakan informasi yang berguna untuk data.

Semua vektor eigen suatu matriks adalah tegak lurus satu sama lain, tak peduli berapapun banyaknya dimensi yang dimiliki. Umumnya pada komputasi software apapun, semua vektor-vektor eigen yang dicari akan memiliki panjang atau norma sama dengan satu (*unit eigenvectors*). Secara matematis, keadaan seperti ini disebut orthonormal. Hal ini penting karena ini berarti bahwa kita dapat mengekspresikan data tersebut dalam vektor-vektor eigen, alih-alih mengekspresikannya dalam sumbu-sumbu koordinat.

#### 5. Memilih komponen dan membentuk sebuah matriks transformasi.

Pada langkah sebelumnya, kita telah menentukan nilai eigen dan vektor eigen. Dari hasil tersebut, nilai-nilai eigen memiliki nilai yang

bervariasi. Vektor eigen yang memiliki nilai eigen terbesar merupakan komponen utama (principal component) dari suatu set data. Jadi, begitu nilai dan vektor eigen telah diperoleh, langkah berikutnya adalah mengurutkannya dari nilai tertinggi sampai nilai terendah. Ini akan memberikan komponen-komponen secara terurut dan signifikan.

Selanjutnya, komponen-komponen yang memiliki nilai terkecil dapat dibuang. Hal ini memang akan menyebabkan kurangnya informasi data, namun komponen yang nilainya kecil tidak berpengaruh besar sehingga dapat diabaikan. Dengan demikian, set data akhir akan memiliki dimensi yang lebih kecil daripada set data awal. Jika set data awal memiliki  $n$  dimensi dan kita hanya memilih  $p$  vektor eigen terbesar, maka set data akhir pun hanya akan memiliki  $p$  dimensi.

Apa yang perlu dilakukan selanjutnya adalah membentuk matriks transformasi atau vektor fitur (*feature vector*) yang dibentuk dari vektor-vektor eigen yang dipilih berdasarkan besarnya nilai eigennya.

$$Feature\ vector = \begin{bmatrix} eig_1 & eig_2 & \dots & eig_n \end{bmatrix} \quad (2.40)$$

#### 6. Menurunkan set data baru

Setelah mendapatkan vektor-vektor eigen yang akan digunakan, set data baru dapat ditentukan dengan mengalikannya dengan set data awal yang telah dikurangi rata-rata.

$$FinalData = RowFeatureVector \times RowDataAdjust \quad (2.41)$$

di mana *RowFeatureVector* adalah matriks feature vector yang telah ditranspose dan *RowDataAdjust* adalah database yang telah dimodifikasi nilainya dan kemudian di-transpose ( $N^T$  atau  $Z^T$ ). *FinalData* adalah set data terakhir atau database terbaru yang telah berhasil dimampatkan dimensinya. Dari hasil ini, kita akan mendapatkan data asli yang hanya terdiri dari vektor-vektor yang kita pilih.

## BAB III METODOLOGI PENELITIAN

Bab ini memaparkan skema penelitian jaringan syaraf tunggal dan ensemble, serta parameter-parameter yang digunakan sebagai variabel dalam penelitian ini.

### 3.1. Skema Penelitian Jaringan Syaraf Tunggal

Sistem pengenalan wajah yang dikerjakan pada penelitian ini terbagi menjadi 5 tahapan, yang dapat dijabarkan sebagai berikut:



Gambar 3.1. Skema Penelitian

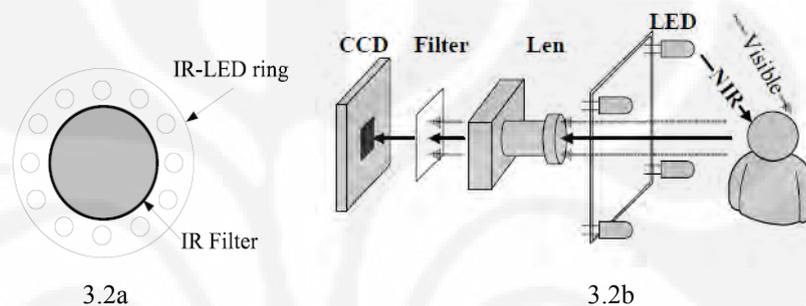
Proses pengambilan data dilakukan dengan peralatan berupa kamera CCD near infrared yang terhubung ke sebuah TV Box dan komputer, dan menggunakan *software* TVHomeMedia sebagai *real-time display* pada monitor. Sedangkan pada proses pre-processing sampai testing, peneliti menggunakan program yang disimpan dalam format *editor file* pada software bahasa pemrograman untuk komputasi numerik.

#### 1. Pengambilan data

Pada tahap ini, data berupa gambar wajah diambil dari sample. Peneliti menggunakan sepuluh orang objek sebagai sample, yang terdiri dari lima orang laki-laki dan lima orang perempuan, dengan usia 20-24 tahun. Gambar diambil melalui rekaman video yang diperoleh dengan menggunakan kamera CCD near infrared (CCTV) dan kamera diletakkan persis berhadapan frontal terhadap wajah objek dengan jarak  $\pm 50$  cm.

Kamera CCD near infrared ini memiliki cincin iluminasi (*illuminating ring*) yang terdiri dari 12 Infrared LED, yang letaknya mengelilingi kamera. Cincin ini bertujuan untuk menghasilkan sumber

cahaya infra merah ketika pencahayaan luar gelap. Panjang gelombang yang dihasilkan LED ini antara 780-1100 nm.<sup>16</sup> Supaya memperoleh kondisi pencahayaan yang stabil, kamera CCD ini juga dilengkapi dengan Infrared Filter untuk menghalangi cahaya tampak.<sup>17</sup> Jika dalam keadaan yang terang, gambar yang ditangkap kamera ini berwarna seperti layaknya kamera visible biasa, namun ketika keadaan gelap, gambar yang ditangkap akan hitam-putih. Keunggulan kamera ini adalah dapat menangkap gambar meskipun kondisi pencahayaan di luar gelap total. Berikut adalah sketsa dan desain kamera CCD Infrared.



**Gambar 3.2.** Kamera CCD Infrared (a) Sketsa (b) Desain

Objek diminta untuk melakukan empat ekspresi yaitu netral, sedih, senang dan bebas. Kondisi pencahayaan untuk setiap pengambilan video adalah gelap total dengan latar belakang polos. Hal ini berlaku sama pada setiap objek. Untuk selanjutnya, istilah objek akan digantikan dengan sebutan target atau kelas atau kelompok.

Setelah mendapatkan rekaman video, dilakukan segmentasi video sehingga diperoleh gambar-gambar yang diinginkan. Untuk setiap objek atau target, diambil sebanyak 30 gambar, di mana 20 gambar akan dimasukkan ke dalam database training dan sepuluh gambar lainnya akan menjadi gambar uji (testing). Peneliti secara manual memotong (cropping) bagian wajah dan memampatkannya (resizing) menjadi gambar yang hanya berukuran 30 x 30 pixel. Setiap gambar disimpan dalam format *.jpg*.

<sup>16</sup> Zhang Xingming, *A Biological Intelligent Access Control System Based on DSP and NIR Technology* (Berlin: Springer-Verlag ICIC 2008, LNCS 5226), hal.58.

<sup>17</sup> Zhao Shuyan, *An Automatic Face Recognition System in the Near Infrared Spectrum* (Leipzig: MLDLM 2005), hal.438.

Secara total, terdapat  $10 \times 30 = 300$  gambar foto yang akan digunakan dalam penelitian.

## 2. Pre-processing

Pada tahap pre-processing ini, gambar yang telah diperoleh dijadikan dalam bentuk matriks di mana setiap elemen dalam matriksnya memiliki nilai intensitas RGB (Red, Green, Blue). Namun karena intensitas RGB memiliki tiga dimensi nilai maka nilai-nilai tersebut dikonversikan ke intensitas Gray Scale yang hanya memiliki satu dimensi nilai. Dengan demikian setiap gambar akan berubah menjadi matriks berukuran  $40 \times 30$  di mana setiap elemen matriksnya adalah nilai input gambar.

Seperti yang telah dijelaskan pada tinjauan pustaka mengenai PCA, untuk memudahkan komputasi berikutnya maka setiap matriks yang berukuran  $40 \times 30$  tersebut disusun ulang menjadi sebuah matriks yang berukuran  $1 \times 1200$  tanpa mengubah nilai-nilai di dalamnya. Dengan demikian, semakin terlihat jelas bahwa setiap gambar memiliki 1200 nilai inputan atau dimensi.

## 3. Ekstraksi Fitur

Tahap ekstraksi fitur merupakan proses di mana dimensi input yang besar direduksi menjadi dimensi yang lebih kecil dengan menggunakan PCA. Hal ini bertujuan untuk menyederhanakan komputasi dan mempercepat waktu pelatihan. Dengan demikian, database yang pada awalnya memiliki 900 dimensi dapat direduksi menjadi 20, 50, ataupun 100 dimensi.

## 4. Training

Setelah melalui ekstraksi fitur, tahap selanjutnya yang merupakan bagian utama sistem pengenalan wajah adalah pelatihan pola atau training. Sistem dilatih untuk mengenali pola input database agar menghasilkan output yang sesuai dengan target tertentu. Karena telah diberi pengarahan akan target, pelatihan ini disebut pula supervised learning (pembelajaran dengan supervisi).

Pada penelitian ini, terdapat 200 database gambar training dengan sepuluh target atau kelas yang telah ditentukan. Pelatihan ini dilakukan dengan menggunakan algoritma JST backpropagation yang hanya memiliki sebuah lapisan tersembunyi. Pada setiap pelatihan, terdapat tiga tahapan, yaitu propagasi maju, propagasi balik, dan perubahan nilai bobot. Nilai bobot antara setiap neuron akan dimodifikasi terus-menerus sampai sistem ini menemukan bobot akhir yang menggeneralisasi nilai inputan apapun dan membawa hasil akhir komputasi kepada salah satu kelas yang ada. Dalam penelitian ini, satu kali pelatihan dinyatakan sebagai satu *epoch*.

#### 5. Testing

Tahap pengujian ini berupa verifikasi suatu gambar akan ditempatkan dalam kelas yang mana. Pada penelitian, tahap pengujian ini dilakukan sebanyak dua kali, yang pertama dengan menggunakan 100 gambar dari dalam database dan yang kedua dengan menggunakan 100 gambar di luar database yang telah disiapkan. 100 gambar yang diambil dari dalam database merupakan sepuluh gambar dari masing-masing target (yang berjumlah sepuluh) dan diambil secara acak. Gambar-gambar uji tersebut akan selalu digunakan untuk setiap percobaan.

Dalam program testing, setiap gambar uji akan dijadikan ke bentuk matriks seperti tahap pre-processing, kemudian direduksi dimensinya dengan PCA, dan dijalankan ke dalam jaringan backpropagation tetapi hanya sampai pada tahap propagasi maju saja dengan menggunakan nilai bobot akhir yang disimpan sebelumnya. Kemudian vektor gambar hasil perhitungan akhir dibandingkan dengan tiap target dengan menghitung jarak masing-masing, yang disebut jarak Euclidean, seperti pada persamaan di bawah ini:

$$d = \sqrt{\sum_{k=1}^M (q_k - t_k)^2} \quad (3.1)$$

Keputusan diambil berdasarkan nilai minimum jarak-jarak tersebut. Untuk satu kali testing, persentase kesesuaian terhadap target, yang disebut recognition rate, pun dihitung.

### 3.2. Parameter yang Digunakan sebagai Variabel pada Jaringan Syaraf

#### Tunggal

Parameter yang digunakan dalam penelitian jaringan syaraf tunggal ini dan sekaligus menjadi variabel bebas, antara lain:

a. Dimensi baru dengan PCA

Peneliti mencoba membandingkan hasil yang diperoleh ketika PCA tidak digunakan dan ketika dimensi database disederhanakan dengan PCA menjadi tiga keadaan, yaitu dimensi baru yang bernilai 20, 50, 100.

b. Inisialisasi bobot

Peneliti membuat dua keadaan yaitu ketika inisialisasi bobot ditentukan secara random dan metode Nguyen-Widrow.

c. Banyaknya unit pada lapisan tersembunyi

Peneliti mencoba membandingkan hasil yang diperoleh ketika jumlah neuron atau unit pada hidden layer berjumlah 1, 10, 50, 100, 200, 300, 400, dan 500.

d. Besarnya epoch yang ditentukan

Peneliti mencoba membandingkan hasil yang diperoleh ketika epoch yang ditentukan antara lain 100, 1000, dan 10000

e. Besarnya laju pembelajaran ( $\alpha$ )

Peneliti mencoba memperhatikan pengaruh besarnya  $\alpha$  dengan nilai 0.1, 0.2, 0.3, dan seterusnya sampai 1.0.

f. Momentum ( $\mu$ )

Peneliti mencoba membandingkan hasil yang diperoleh ketika digunakannya momentum dan tidak digunakannya momentum pada tahap training. Nilai momentum yang digunakan pun divariasikan 0.1, 0.2, 0.3, dan seterusnya sampai 1.0.

Selama penelitian dilakukan untuk masing-masing variabel bebas di atas, peneliti menggunakan parameter di luar masing-masing poin sebagai variabel kontrol default, yaitu:

1. Dimensi baru bernilai 100
2. Inisialisasi bobot menggunakan metode Nguyen-Widrow
3. Jumlah unit hidden neuron sebanyak 100
4. Epoch sebesar 1000
5. Nilai  $\alpha$  sebesar 0.5
6. Momentum tidak digunakan

Jadi, misalnya ketika peneliti melakukan percobaan untuk variabel bebas “besarnya nilai epoch” (parameter d), variabel kontrol yang digunakan adalah 1, 2, 3, 5, dan 6. Khusus untuk parameter momentum, peneliti juga mencoba eksperimen dengan nilai  $\alpha = 0.1, 0.2, 0.3, 0.4, 0.6, 0.7, 0.8, 0.9,$  dan 1.0.

Sementara itu, variabel terikat yang digunakan pada penelitian ini adalah nilai recognition rate pada tahap pengujian, baik itu data dalam database (training) ataupun dari luar database (testing), waktu pelatihan (training time), waktu pengenalan (recognition time), dan grafik error terhadap epoch. Berdasarkan hasil tersebut, akan ditentukan parameter-parameter yang dapat mengoptimalkan performa jaringan, yang selanjutnya disebut parameter optimalisasi.

### 3.3. Skema Penelitian Jaringan Syaraf Ensemble

Sama seperti pada penelitian Jaringan Syaraf Tunggal, sistem pengenalan wajah dengan jaringan syaraf ensemble juga terbagi ke dalam lima tahapan, yaitu pengambilan data, pra-pemrosesan, ekstraksi fitur, training, dan testing. Hanya saja pada penelitian ini, peneliti menggunakan empat skema yang berbeda, yaitu:

1. Skema algoritma backpropagation dan simple averaging

Pada skema ini, peneliti melakukan eksperimen dengan menggunakan jumlah jaringan yang bervariasi antara 1-5. Setiap jaringan diberi bobot awal yang berbeda-beda dan dilatih secara terpisah (independen) hingga terbentuk bobot akhir yang berbeda-beda. Pengambilan keputusan

dilakukan dengan metode rata-rata (simple averaging). Setelah proses pelatihan selesai, jaringan ensemble digunakan untuk mengenali gambar uji.

2. Skema parsialisasi data dengan algoritma backpropagation dan simple averaging

Pada skema ini, jumlah jaringan divariasikan antara 1-5. Pada setiap jumlah jaringan, peneliti membuat variabel bebas yaitu parsialisasi data dengan persentase 50%, 60%, 70%, 80%, 90%, dan 100%. Dengan jumlah pola pada database sebanyak 200, maka parsialisasi data 50% berarti hanya 100 pola yang diambil secara acak dari database untuk dilatih. Perlu diketahui bahwa meskipun data diambil secara acak, jumlah pola untuk setiap target adalah sama. Karena terdapat 10 target pada penelitian, maka akan ada 10 pola dari masing-masing target yang akan digunakan.

Dengan persentase yang sama, maka setiap jaringan akan memiliki jumlah masukan pola yang sama, tetapi urutannya berbeda. Setiap jaringan diberi bobot awal yang berbeda-beda dan dilatih secara terpisah (independen) dengan algoritma backpropagation hingga diperoleh bobot akhir tiap jaringan yang masing-masing unik. Setelah proses pelatihan selesai, jaringan ensemble yang telah memiliki bobot pada masing-masing jaringannya siap digunakan untuk mengenali gambar uji. Pengambilan keputusan ensemble dilakukan dengan metode rata-rata (simple averaging).

3. Skema algoritma NCL

Sama seperti pada skema-skema sebelumnya, jumlah jaringan divariasikan antara 1-5. Pada algoritma NCL, setiap jaringan awalnya diberi bobot yang berbeda-beda, dan pelatihan pada jaringan tertentu dilakukan dengan memperhatikan tidak hanya keluaran jaringan tersebut tetapi juga keluaran jaringan ensemble. Algoritma NCL ini telah dibahas

pada bab II. Pada skema ini, peneliti menentukan variabel bebas berupa parameter nilai  $\lambda$  (penalty term) sebesar 0, 0.2, 0.4, 0.6, 0.8, dan 1.

#### 4. Skema parsialisasi data dengan algoritma NCL

Skema ini menggabungkan skema 2 dan skema 3. Data yang digunakan sebagai masukan pelatihan diambil sebagian tertentu (50%, 60%, 70%, 80%, 90%, dan 100%) secara acak. Jumlah jaringan divariasikan antara 1-5. Selanjutnya, algoritma NCL digunakan pada pelatihan. Setiap jaringan awalnya diberi bobot yang berbeda-beda. Pada skema ini, peneliti menentukan variabel bebas berupa parameter nilai  $\lambda$  (penalty term) sebesar 0, 0.2, 0.4, 0.6, 0.8, dan 1.

Perlu diketahui bahwa semua skema menggunakan parameter default, di mana dimensi gambar database direduksi dari 1200 menjadi 100 oleh PCA, inisialisasi bobot awal menggunakan Nguyen-Widrow Initialization, jumlah neuron lapisan tersembunyi sebanyak 100, laju pembelajaran  $\alpha$  sebesar 0.5, epoch maksimal sebesar 1000, dan tidak menggunakan momentum.

Hasil yang diperoleh dari setiap skema akan dibandingkan satu sama lain. Setelah itu, berdasarkan parameter optimalisasi pada jaringan syaraf tunggal, dilakukan pula optimalisasi pada jaringan syaraf ensemble, mengacu kepada hasil terbaik dari setiap skema.

## BAB IV

### ANALISIS PERFORMA JARINGAN SYARAF TUNGGAL

Pada bab ini penulis akan membahas pengamatan dan hasil percobaan dari penelitian pada jaringan syaraf tunggal. Secara garis besar, pembahasan akan dibahas menjadi tiga bagian. Pada bagian pertama, akan dibahas bagaimana jaringan syaraf tiruan (JST) tunggal dengan algoritma backpropagation dapat digunakan untuk sistem pengenalan wajah yang dapat mempelajari pola gambar wajah yang telah ada dan mengenali suatu gambar wajah yang lain.

Pada bagian kedua, akan dibahas bagaimana Principal Component Analysis (PCA) dapat digunakan untuk mereduksi dimensi suatu data yang berjumlah besar tanpa mengubah signifikansi data tersebut agar proses komputasi dapat berjalan lebih cepat. Besarnya dimensi baru yang direduksi oleh PCA akan menjadi parameter dalam pembahasan ini.

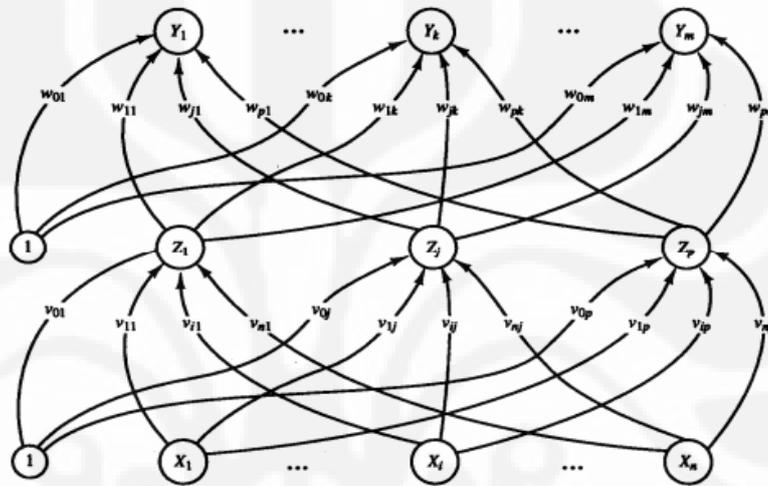
Pada bagian ketiga, akan dibahas bagaimana parameter-parameter pada JST backpropagation berpengaruh terhadap hasil pelatihan dan pengujian. Parameter-parameter tersebut antara lain adalah modifikasi inisialisasi bobot awal, nilai laju pembelajaran ( $\alpha$ ), banyaknya unit lapis tersembunyi yang digunakan, jumlah epoch pelatihan, dan penggunaan nilai momentum ( $\mu$ ).

#### 4.1. Jaringan Syaraf Tiruan untuk Sistem Pengenal Wajah

Pada penelitian sistem pengenalan wajah ini, penulis menggunakan jaringan syaraf tiruan (JST) atau neural network dengan algoritma back-propagation. Seperti yang telah dijelaskan pada bab II, JST adalah jaringan yang prinsip kerjanya menyerupai jaringan syaraf manusia, di mana jaringan ini dapat mempelajari suatu pola yang diberikan dengan target tertentu. Dari pola-pola tersebut, proses pembelajaran dilakukan dengan memperbarui bobot-bobot atau koefisien-koefisien di antara unit-unitnya secara berulang-ulang sehingga pada akhirnya akan mendapatkan suatu tetapan yang default, yang dapat digunakan untuk mengidentifikasi suatu pola baru.

JST yang digunakan pada penelitian ini terdiri dari tiga lapisan yaitu: lapisan masukan  $X_i$  (input layer), lapisan tersembunyi  $Z_j$  (hidden layer), dan

lapisan keluaran  $Y_k$  (output layer). Lapisan masukan  $X_i$  ( $i = 1, 2, \dots, n$ ) berisi nilai-nilai atau informasi suatu pola, yang dalam hal ini adalah intensitas setiap pixel pada setiap gambar yang digunakan pada pelatihan. Karena setiap gambar berukuran  $40 \times 30$  pixel, maka akan terdapat 1200 nilai inputan untuk sebuah gambar, yang disebut juga dimensi input. Oleh karena itu, jumlah unit atau neuron pada lapisan ini ( $n$ ) adalah 1200. Namun jika PCA digunakan, dimensi input tersebut akan direduksi sehingga jumlah unit yang digunakan pun akan disesuaikan.



dalam  $v_{ij}$  sedangkan bobot-bobot yang menghubungkan lapisan tersembunyi dengan lapisan keluaran dinotasikan dengan  $w_{jk}$ . Pada awal-awal, bobot-bobot tersebut diberi nilai atau diinisialisasi secara acak atau dengan metode Nguyen-Widrow.

JST yang digunakan pada penelitian ini adalah jaringan sederhana yang hanya memiliki sebuah lapisan tersembunyi. Namun pada jaringan yang lebih kompleks, tidak menutup kemungkinan bahwa jumlah lapisan tersembunyi dapat lebih dari satu. JST ini menggunakan algoritma backpropagation, yang terdiri dari proses propagasi maju untuk menentukan keluaran dan propagasi balik untuk memperbaiki nilai bobot.

Proses propagasi maju dapat dijelaskan sebagai berikut. Lapisan tersembunyi  $Z_j$  ( $j = 1, 2, \dots, p$ ) berisi nilai-nilai keluaran pertama yang merupakan hasil komputasi antara nilai setiap unit pada lapisan masukan dengan bobot-bobot yang berkesesuaian dengannya ( $v_{ij}$ ). Kemudian nilai-nilai pada lapisan tersembunyi ini akan dikomputasi dengan bobot-bobot yang berkesesuaian dengannya ( $w_{jk}$ ) sehingga menghasilkan nilai pada unit-unit lapisan keluaran  $Y_k$  ( $k = 1, 2, \dots, m$ ). Banyaknya unit lapisan tersembunyi ( $p$ ) dan lapisan keluaran ( $m$ ) dapat ditentukan sendiri. Rumus untuk menghitung nilai  $Z_j$  ini terdapat pada persamaan 2.1 – 2.4.

Dari persamaan-persamaan tersebut, dapat dilihat bahwa setiap keluaran, baik pada lapisan tersembunyi maupun lapisan keluaran harus diaktivasi terlebih dahulu. Dalam backpropagation, fungsi aktivasi yang dipakai harus memenuhi beberapa syarat yaitu : kontinu, terdiferensial dengan mudah dan merupakan fungsi yang tidak turun. Salah satu fungsi yang memenuhi ketiga syarat tersebut dan dipakai pada penelitian ini adalah fungsi sigmoid biner (persamaan 2.2 dan 2.4) yang memiliki range (0,1).<sup>18</sup>

Berdasarkan cara memodifikasi bobotnya, pelatihan pola dengan algoritma backpropagation digolongkan dalam pembelajaran dengan arahan (*supervised learning*). Dalam pembelajaran dengan arahan, terdapat sejumlah pasangan data (masukan – target keluaran) yang dipakai untuk melatih jaringan hingga diperoleh bobot yang diinginkan. Karena fungsi sigmoid

---

<sup>18</sup> Jong Jek Siang, Loc.Cit, hal.99.

yang digunakan sebagai fungsi aktivasi memiliki range (0,1), maka target yang ditentukan harus memiliki range yang sama (0,1). Pasangan data tersebut berfungsi sebagai “guru” untuk melatih jaringan hingga diperoleh bentuk yang terbaik.<sup>19</sup>

Setelah memperoleh keluaran pada proses propagasi maju, hasil tersebut ( $Y_k$ ) dibandingkan nilainya dengan target ( $T_k$ ) yang telah ditentukan. Selisih antara keluaran jaringan dengan target (keluaran yang diinginkan) merupakan kesalahan yang terjadi. Jaringan akan memodifikasi bobot sesuai dengan kesalahan tersebut. Proses yang dimulai dari menentukan kesalahan sampai modifikasi bobot ini disebut dengan propagasi balik, dan algoritmanya dapat dilihat pada bab II.

Proses propagasi maju dan balik tersebut dilakukan berulang-ulang untuk semua pola yang ada hingga jaringan dapat mengenali polanya. Satu siklus pelatihan ini disebut dengan epoch. Proses iterasi ini terus-menerus hingga kondisi penghentian dipenuhi. Kondisi penghentian yang umum digunakan adalah jumlah iterasi (epoch) dan kesalahan (apakah kesalahan yang terjadi sudah lebih kecil dari batas toleransi yang diizinkan). Pada penelitian ini, kondisi penghentian yang digunakan adalah jumlah iterasi.

Setelah pelatihan selesai dilakukan, jaringan yang kini memiliki bobot-bobot akhir yang telah mantap dapat digunakan untuk pengenalan pola, entah itu pola lama yang diambil dari database ataupun pola baru yang diambil dari luar database. Proses yang dilakukan untuk pengenalan pola ini hanyalah propagasi maju.

Pada penelitian ini, terdapat sepuluh kelas atau cluster, di mana setiap kelas mewakili objek yang berbeda, yaitu orang-orang yang digunakan dalam pelatihan. Nilai yang unik (spesifik) dan berbeda-beda diberikan kepada kesepuluh kelas tersebut dengan range (0,1). Untuk memudahkan perhitungan pada program, nilai sebuah target dinyatakan dalam matriks yang berukuran 1x10. Misalnya target ke-1 bernilai [0.5 0.3 0.3 0.3 0.3 0.3 0.3 0.3 0.3 0.3], target ke-2 bernilai [0.3 0.5 0.3 0.3 0.3 0.3 0.3 0.3 0.3 0.3], dan

---

<sup>19</sup> Ibid, hal.28.

seterusnya hingga target ke-10 yang bernilai [0.3 0.3 0.3 0.3 0.3 0.3 0.3 0.3 0.3 0.5].



**ann – dau – cin – roy – dev – chr – pel – rud – eni - pep**

**Gambar 4.2.** Sepuluh Kelas (Objek) yang Digunakan untuk Pelatihan

Database pelatihan jaringan terdiri dari 200 gambar dengan berbagai ekspresi wajah yang masing-masing dijadikan input pada lapisan masukan. Setiap gambar memiliki 1200 dimensi input dan sebuah target yang berdimensi 10. Dengan demikian, jumlah unit lapisan masukan  $n$  adalah 1200 (jika tidak menggunakan PCA) sedangkan jumlah unit masukan keluaran  $m$  adalah 10, dan agar seragam pada pelatihan terdapat 20 gambar berbeda yang memiliki target yang sama. Gambar-gambar yang digunakan untuk database pelatihan ini dapat dilihat pada lampiran.

Agar jaringan dapat mempelajari pola dengan baik, ke-20 gambar dengan target yang sama sebaiknya tidak dilatih secara berurutan, melainkan silih berganti antara gambar dengan target ke-1, dilanjutkan dengan gambar dengan target ke-2, dan seterusnya sampai gambar dengan target ke-10. Setelah itu, barulah pelatihan dilanjutkan dengan gambar dengan target ke-1 kembali hingga seluruh 200 gambar selesai dilatih.

Pada penelitian ini, parameter-parameter standar yang ditetapkan adalah penggunaan PCA dengan reduksi dimensi input dari 1200 menjadi 100, inisialisasi bobot awal dengan metode Nguyen-Widrow, jumlah lapis tersembunyi ( $p$ ) sebanyak 100, jumlah iterasi sebagai syarat henti adalah 1000, laju pembelajaran ( $\alpha$ ) sebesar 0.5, dan tidak menggunakan momentum. Untuk selanjutnya, skema ini disebut dengan skema *default*.

Setelah pelatihan dilakukan, dilakukan pengenalan wajah dengan menggunakan 100 gambar. Pertama-tama, digunakan 100 gambar yang dipilih secara acak dari database pelatihan, sedangkan yang kedua digunakan 100 gambar yang diambil di luar database pelatihan. Meskipun diambil

secara acak, agar seragam diambil gambar dengan jumlah sama untuk setiap target, yaitu 10. Dengan demikian, 100 gambar pertama merupakan setengah dari database pelatihan, sedangkan 100 gambar kedua merupakan gambar lain untuk menguji seberapa “pandai” jaringan yang telah terlatih untuk mengenali pola baru. Pada proses pengenalan gambar ini, sistem akan memberi jawaban terhadap setiap pola gambar, masuk ke kelas yang mana dari 10 kelas yang ada. Gambar-gambar yang digunakan untuk pengujian ini dapat dilihat pada lampiran.

Apabila jawaban yang diberikan sistem sesuai dengan target, maka jaringan dianggap telah mengenali gambar dengan benar. Namun bila jawaban yang diberikan tidak sesuai dengan target, maka jaringan dinilai salah dalam mengenali gambar. Sistem tidak dapat menghasilkan jawaban bahwa suatu pola berada di luar target yang ditentukan. Dengan demikian hanya ada dua pilihan, yaitu benar atau salah. Persentase jawaban yang benar terhadap seluruh pertanyaan mengacu kepada *recognition rate*.

Berikut ini adalah *recognition rate*, waktu pra-pemrosesan (*pre-processing time*), waktu pembelajaran (*training time*), dan waktu pengenalan (*recognition time*) yang diperoleh dari proses pengenalan wajah pada penelitian dengan standar yang telah ditentukan di atas. Percobaan ini dilakukan sebanyak sepuluh kali.

**Tabel 4.1.** Recognition Rate dan Recognition Time dengan Parameter-Parameter Standar yang Ditetapkan

Recognition Rate		Waktu (detik)			
Training	Testing	Pre-Processing	Training	Testing (1 gambar)	Total
100%	96.4%	32.06	36.97	0.0107	69.04

Pada tabel 4.1, *recognition rate-training* menunjukkan *recognition rate* untuk gambar-gambar dari database, sedangkan *recognition rate-testing* menunjukkan *recognition rate* untuk gambar-gambar dari luar database. Waktu-pre-processing menunjukkan waktu yang dibutuhkan sistem untuk mengkonversi gambar pada database menjadi matriks dan waktu eksekusi proses PCA. Waktu-training menunjukkan lama waktu yang dibutuhkan

jaringan untuk mempelajari pola yang ada, sedangkan waktu-testing menunjukkan waktu yang dibutuhkan jaringan untuk mengenali gambar uji. Waktu-total adalah waktu keseluruhan yang diperlukan jaringan, mulai dari tahap pre-processing sampai pengujian.

Dari tabel 4.1, recognition rate untuk gambar-gambar di dalam database mencapai 100%, berarti sistem berhasil mengenali semua gambar di dalam database dengan benar. Recognition rate untuk gambar-gambar di luar database mendekati 96%, artinya sistem berhasil mengenali 96 gambar dengan benar dan 4 gambar salah dikenali.

Perlu diingat bahwa semua gambar pada pengenalan kedua berasal dari luar database, nilai 96% ini cukup tinggi dan menunjukkan bahwa sistem sudah baik dalam mengenali wajah setelah dilatih dengan algoritma backpropagation dengan parameter-parameter standar yang ditentukan. Hal ini dapat diibaratkan dengan otak manusia yang dapat mengenali wajah seseorang meskipun dalam keadaan yang berbeda dari yang dikenalnya sebelumnya.

Namun bila recognition rate testing dikaji per kelas, maka dari sepuluh kali pengambilan data ini akan diperoleh hasil sebagai berikut:

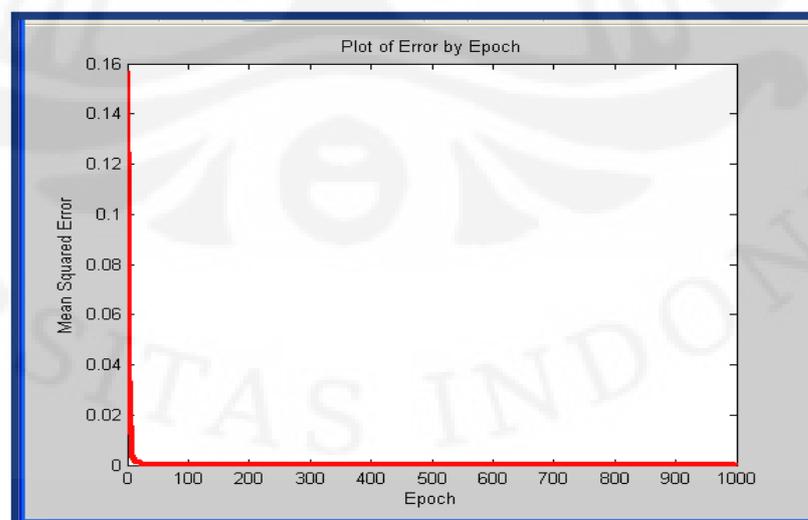
**Tabel 4.2.** Recognition Rate per Kelas

No	Kelas	Recognition Rate (Testing)	Gambar Salah Uji
1	ann 	98%	2,7
2	dau 	89%	13,16,17,18,20
3	cin 	100%	-
4	roy 	94%	32,34,39
5	dev 	100%	-
6	chr 	100%	-
7	pel 	87%	65,69,70
8	rud 	100%	-
9	eni 	96%	87
10	pep 	100%	-

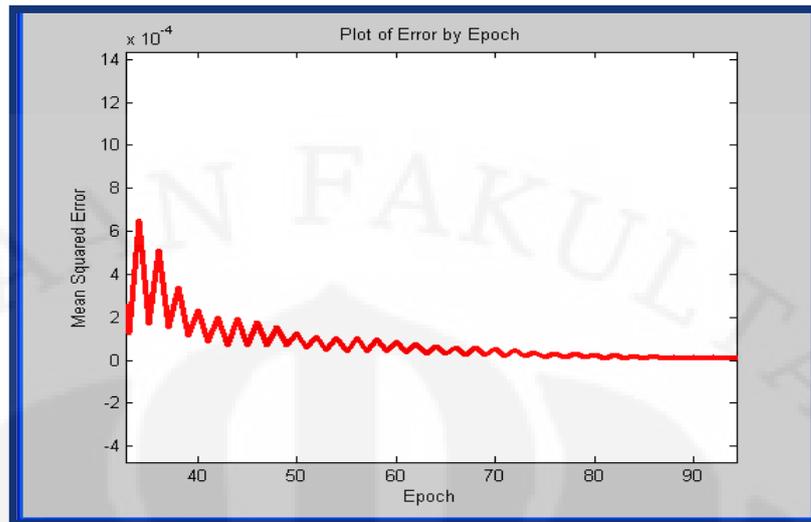
Dari tabel 4.2, dapat dilihat adanya perbedaan recognition rate antara kelas yang satu dengan kelas yang lainnya. Gambar salah uji menunjukkan gambar-gambar yang salah diidentifikasi oleh sistem dari sepuluh kali percobaan. Ilustrasi gambar-gambar tersebut dapat dilihat pada lampiran, di mana nomor gambar telah diurutkan. Secara umum, kelas ke-3, 5, 6, 8, dan 10 selalu mencapai recognition rate 100%, sedangkan kelas-kelas lainnya tidak dapat mencapai 100%. Kelas yang memiliki recognition rate paling rendah adalah kelas ke-2 dan 7, dan gambar-gambar yang paling sering salah diidentifikasi (lebih dari tiga kali) adalah gambar ke-13, 69, 70, dan 87.

Jika diperhatikan lebih seksama, ada beberapa faktor yang menyebabkan suatu gambar uji salah diidentifikasi oleh sistem. Perlu diketahui, seratus gambar uji tersebut diambil dari luar database pelatihan sehingga sistem tidak mempelajari pola gambar-gambar tersebut. Perbedaan ekspresi (mimik), posisi, ukuran, dan kemiringan wajah, serta pencahayaan luar (*ambient light*) menjadi pengaruh utama yang menyebabkan suatu sistem salah mengidentifikasi pola.

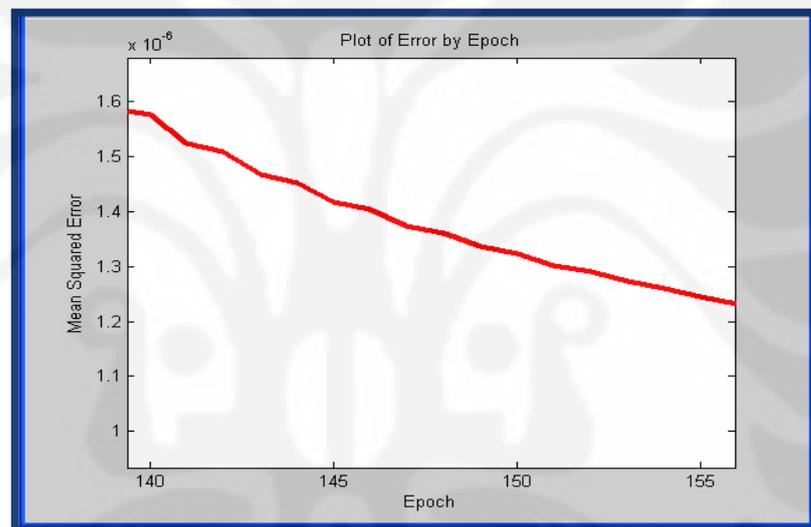
Dapat dilihat pada lampiran, gambar 13 menunjukkan bagian mata yang lebih besar, gambar 69 dan 70 menunjukkan bagian mulut yang terbuka lebar, dan gambar 87 yang memperlihatkan gigi. Gambar-gambar tersebut memang tampak lebih berbeda jika dibandingkan dengan gambar-gambar pada kelas yang sama untuk pelatihan.



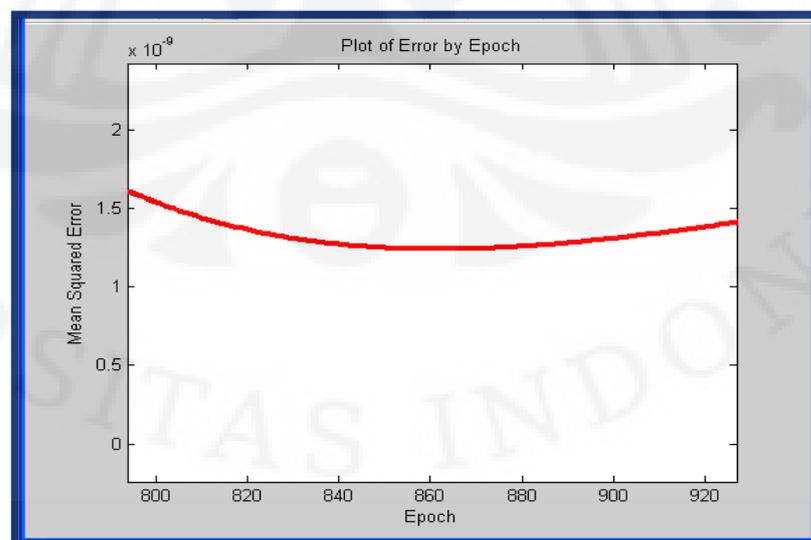
4.3.a



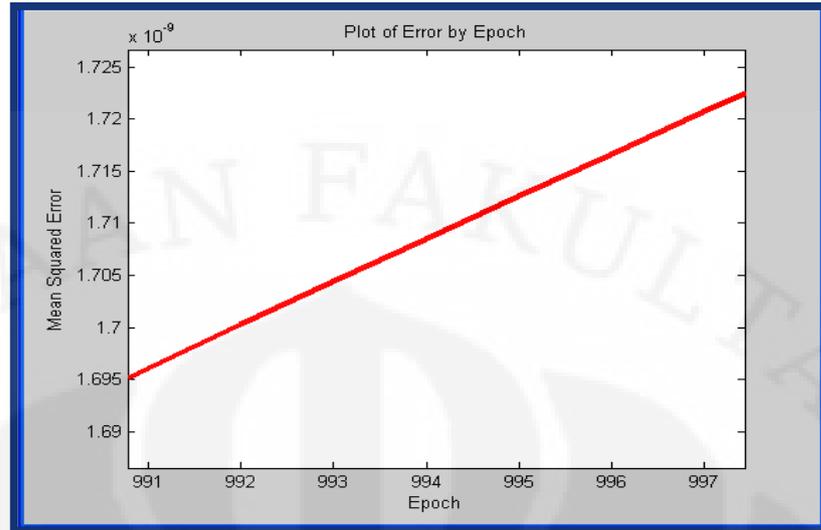
4.3.b



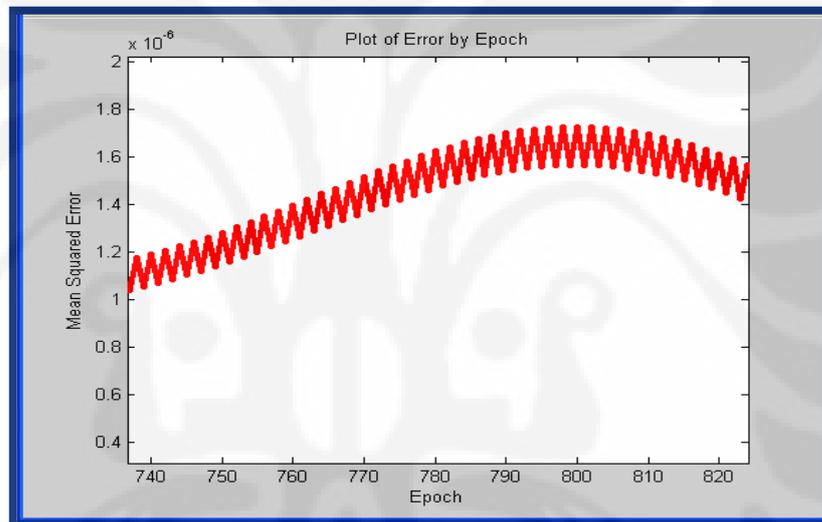
4.3.c



4.3.d



4.3.e



4.3.f

**Gambar 4.3.** Grafik Error terhadap Epoch

Gambar 4.3 menunjukkan grafik error terhadap epoch yang tercatat selama pelatihan berlangsung. Fungsi error yang digunakan di sini adalah fungsi kuadratik, di mana:  $E = \frac{1}{2} \sum_k (\mathbf{C}_k - Y_k)^2$ . Gambar 4.3a menunjukkan grafik secara keseluruhan, mulai dari iterasi pertama sampai iterasi ke-1000. Dari grafik ini terlihat pada iterasi-iterasi awal nilai error cukup besar dan pada suatu iterasi tertentu langsung mengecil hingga mendekati nol.

Gambar 4.3b hingga 4.3f menunjukkan grafik 4.3a setelah diperbesar pada iterasi-iterasi tertentu. Gambar 4.3b menunjukkan pada epoch yang

masih sedikit, nilai error berosilasi dengan kecenderungan semakin mengecil. Gambar 4.3c menunjukkan pada sekitar epoch ke-150, osilasi nilai error sudah tidak signifikan lagi (skala  $10^{-6}$ ) dan kecenderungannya terus menurun. Hal ini memperlihatkan bahwa pada akhirnya pelatihan ini menghasilkan penurunan nilai error yang konvergen meskipun pada awalnya mengalami fluktuasi. Fluktuasi di awal iterasi disebabkan karena pada awalnya inisialisasi bobot ditentukan secara acak (Nguyen-Widrow Initialization) dan banyaknya bobot yang harus dimodifikasi nilainya. Pada pelatihan standar ini, terdapat  $100 \times 100 + 100 = 10100$  bobot  $v_{ij}$  dan  $10 \times 100 + 10 = 1010$  bobot  $w_{jk}$  yang harus diperbarui terus-menerus.

Gambar 4.3d menunjukkan suatu kemungkinan, di mana pada sekitar epoch ke-860 nilai error menjadi tidak konvergen lagi, dan grafik error berbalik arah menjadi lebih besar. Hal ini mungkin terjadi karena pengaruh laju pembelajaran ( $\alpha$ ). Semakin besar harga  $\alpha$ , semakin sedikit iterasi yang diperlukan. Namun apabila  $\alpha$  terlalu besar, maka akan merusak pola yang sudah benar sehingga pemahaman menjadi lambat. Dengan demikian, untuk  $\alpha$  yang besarnya 0.5 ini, setelah iterasi sekitar 860 kali, pembelajaran telah mencapai pemahaman tertinggi. Hal ini dinyatakan oleh nilai errornya yang paling minimum.

Namun kenaikan error yang terjadi tidaklah signifikan. Gambar 4.3e menunjukkan kenaikan error tersebut hanya sekitar  $5 \times 10^{-12}$  per epoch, sehingga ketika pelatihan berhenti (epoch ke-1000), nilai error pun masih sangat kecil ( $1,74 \times 10^{-9}$ ), jauh di bawah toleransi umum ( $1 \times 10^{-5}$ ), dan sistem masih dapat mengenali gambar dengan baik.

Gambar 4.3a sampai 4.3e merupakan grafik error yang umum terjadi pada pelatihan. Gambar 4.3f menunjukkan kemungkinan yang lain ketika dilakukan pelatihan lain dengan parameter yang sama. Grafik error terus berfluktuasi meskipun pembelajaran telah mencapai epoch yang tinggi. Hal ini juga disebabkan oleh pengaruh laju pembelajaran yang mungkin saja tidak dapat mencapai konvergensi dengan inisialisasi awal bobot. Akan tetapi, fluktuasi ini hanya menyebabkan perubahan yang sangat kecil sehingga

perubahan bobot pun tidak terlalu signifikan dan gambar wajah masih dapat dikenali dengan baik.

#### 4.2. Principal Component Analysis (PCA) untuk Mereduksi Dimensi Data

Seperti yang telah diketahui, setiap data gambar yang digunakan dalam pelatihan memiliki 1200 nilai yang dapat dijadikan informasi untuk input. Jika kita menggunakan parameter-parameter standar yang ditetapkan di atas, maka banyaknya bobot antara lapisan masukan dan lapisan tersembunyi yang harus diinisialisasi dan dimodifikasi pada proses pembelajaran mencapai  $100 \times 1200 + 100 = 120100$ . Tentunya hal ini akan membutuhkan waktu pembelajaran yang lama, iterasi yang banyak, dan bisa jadi tidak konvergen.

Oleh karena itu, pada penelitian ini digunakanlah PCA sebagai sarana untuk mereduksi banyaknya dimensi data tersebut tanpa mengubah ataupun mengurangi informasi yang signifikan di dalamnya. PCA menggunakan prinsip di mana setiap set data memiliki nilai dan vektor eigen yang tidak semuanya signifikan. PCA hanya mengambil sebagian vektor eigen yang paling signifikan dengan memilih sejumlah nilai-nilai eigen terbesar saja. Kumpulan vektor-vektor eigen terpilih ini akan membentuk suatu matriks transformasi yang disebut dengan Principal Component atau komponen utama, yang nantinya akan menghasilkan data set baru yang memiliki dimensi baru sesuai banyaknya vektor eigen yang dipilih.

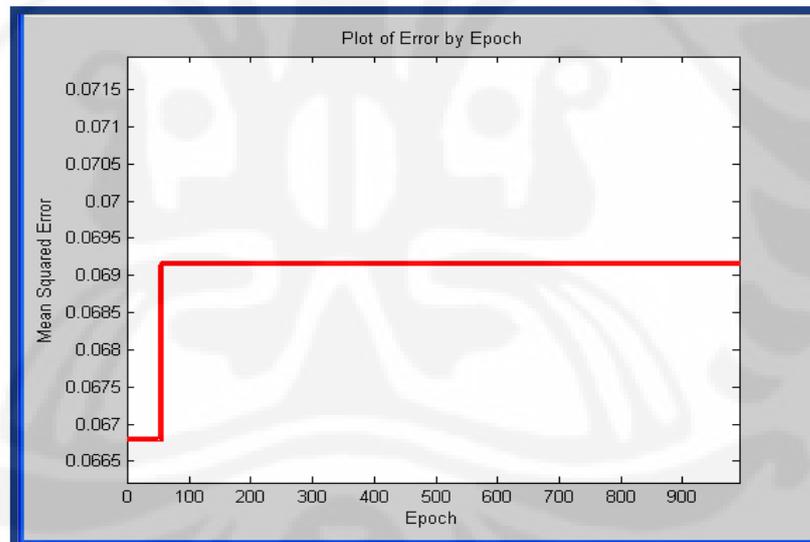
Pada penelitian ini dipilih nilai eigen terbesar sebanyak 20, 50, 100, dan 1200. Skema default ditandakan dengan latar berwarna kemerahan. Dari percobaan sebanyak sepuluh kali, berikut adalah hasilnya:

**Tabel 4.3.** Recognition Rate dan Recognition Time dengan Variasi Dimensi yang Direduksi PCA

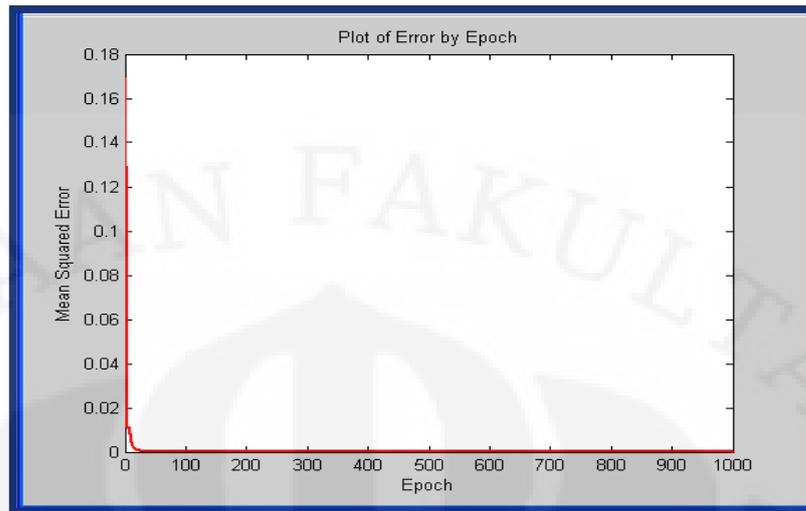
PCA	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>Tanpa PCA</b>	-	-	511.30	-
<b>20 dimensi</b>	100%	94.2%	13.95	0.0097
<b>50 dimensi</b>	100%	95.6%	22.35	0.0100
<b>100 dimensi</b>	100%	96.4%	36.97	0.0107
<b>1200 dimensi</b>	100%	96.6%	556.92	0.0246

Pada percobaan tanpa PCA, jaringan tidak mampu melakukan pelatihan dengan baik. Bobot-bobot akhir pelatihan hanya mengacu kepada target terakhir yaitu target ke-10, sehingga ketika sistem diminta untuk mengenali gambar wajah, semua gambar menghasilkan jawaban target ke-10. Hal ini terjadi karena besarnya dimensi sehingga komputasi yang dilakukan oleh bahasa pemrograman menjadi kacau.

Gambar 4.4a menunjukkan grafik error terhadap epoch yang dihasilkan pada saat pelatihan. Dari grafik tersebut dapat dilihat bahwa error tidak mencapai konvergensi dan nilainya pun konstan dan besar. Hal ini berbeda bila dibandingkan dengan gambar 4.4b yang menggunakan PCA dengan dimensi baru sebesar 20. Pada grafik 4.4b, error tampak semakin kecil dengan besarnya itersai dan menunjukkan adanya konvergensi. Secara garis besar grafik untuk PCA dengan dimensi baru sebesar 50, 100, dan 1200 juga menyerupai grafik pada gambar 4.4b.



4.4.a

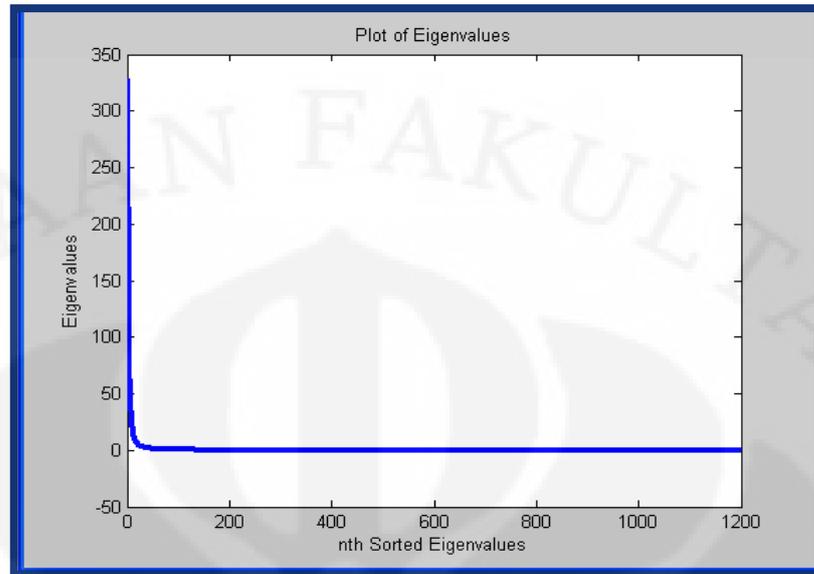


4.4. b

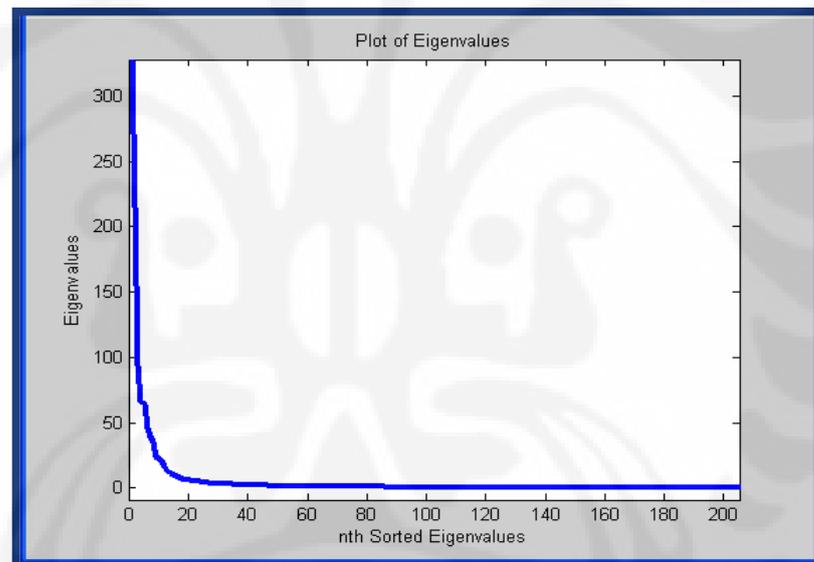
**Gambar 4.4.** Grafik Error terhadap Epoch a) Tanpa PCA dan b) Dengan PCA (100 dimensi)

Dari tabel 4.2, dapat dilihat bahwa semakin besar dimensi baru yang ditentukan, semakin besar recognition rate dan semakin lama waktu pembelajaran (training) dan waktu pengenalan (testing 1 gambar) yang dibutuhkan. Namun, peningkatan recognition ini tidaklah terlalu tajam. Jadi untuk hasil yang cukup optimal, dengan mempertimbangkan recognition rate dan waktu pelatihan maka PCA dengan dimensi 50 dianggap sudah cukup baik untuk sistem pengenalan wajah ini.

Karena pelatihan tanpa PCA tidak dapat dilakukan, peneliti mencoba melakukan PCA dengan dimensi baru 1200, yang artinya sama dengan dimensi awal. Jadi tidak ada bagian dari data set yang diabaikan atau dibuang, hanya saja susunan data set dinormalisasi dan ditata sedemikian hingga memiliki vektor-vektor eigen yang nilai eigennya berurutan, mulai dari yang terbesar sampai yang terkecil. Ternyata dengan cara ini bahasa pemrograman mampu melakukan komputasi dan menghasilkan recognition rate yang baik serta grafik error yang konvergen. Namun waktu pelatihan yang dibutuhkan terlalu lama jika dibandingkan dengan variasi lainnya. Selain itu kenaikan recognition rate terhadap variasi lain juga tidak terlalu signifikan. Hal ini tidak sebanding dengan waktu pelatihannya sehingga peneliti merekomendasikan untuk tidak memakai pilihan ini dalam sistem pengenalan wajah.



4.5.a



4.5.b

**Gambar 4.5.** Grafik Nilai Eigen Terbesar Sampai Terkecil

Gambar 4.5 menunjukkan grafik besarnya nilai eigen data set jika diurutkan dari yang terbesar sampai yang terkecil. Gambar 4.5b merupakan perbesaran dari gambar 4.5a. Data 250 nilai eigen yang telah diurutkan terdapat di lampiran C, di mana nilai eigen terbesar adalah 328.16 sedangkan nilai eigen terkecil mendekati nol ( $10^{-15}$ ). Jika dibandingkan dengan nilai eigen yang terbesar, dari data dan grafik tersebut dapat dilihat bahwa nilai

eigen menjadi sangat kecil pada nilai ke-30 (perbandingannya 0.01) dan menjadi ekstrim kecil pada nilai ke-200 (perbandingannya  $7.6 \times 10^{-16}$ ). Semakin kecil nilai perbandingan tersebut, semakin kecil pula signifikansi dan kontribusi vektor eigen yang berpasangan dengan nilai eigen tersebut terhadap pembentukan matriks transformasi komponen utama.

Dari data tersebut, dapat diindikasikan bahwa hanya sekitar 199 vektor eigen yang nilainya cukup signifikan untuk mereduksi dimensi data set tanpa mengurangi informasi di dalamnya. Namun jika perbandingan 0.01 sudah dianggap cukup kecil dan kurang dari itu tidak signifikan lagi, maka dengan hanya mengambil 30 vektor eigen saja sudah cukup baik untuk mereduksi dimensi data set. Hal inilah yang menyebabkan recognition rate pada percobaan dengan dimensi baru 50, 100 dan 1200 tidak terlalu signifikan perbaikannya, karena kecilnya perbandingan nilai eigen setelah 30 nilai terbesar.

### **4.3. Pengaruh Parameter-Parameter pada Algoritma Backpropagation**

Seperti yang dipaparkan pada metodologi penelitian, peneliti melakukan modifikasi beberapa parameter untuk mengamati dan menganalisis pengaruh-pengaruhnya terhadap performa sistem pengenalan wajah. Parameter-parameter yang dimodifikasi itu antara lain adalah metode inisialisasi awal bobot, jumlah unit pada lapisan tersembunyi, besarnya laju pembelajaran ( $\alpha$ ), banyaknya iterasi/epoch, dan digunakannya momentum. Ketika melakukan eksperimen terhadap suatu parameter, maka parameter-parameter lainnya ditetapkan sesuai standar *default* seperti yang telah disebutkan pada bagian 4.1. Pada tabel hasil eksperimen, terdapat sebuah skema default, yang ditandakan dengan latar berwarna kemerahan.

#### **4.3.1. Inisialisasi bobot awal**

Terdapat dua metode yang digunakan dalam eksperimen, yaitu metode acak dan metode Nguyen-Widrow. Pada metode acak, setiap bobot antarlapisan ( $v_{ij}$  dan  $w_{jk}$ ) diberi nilai awal yang berkisar -0.5 sampai 0.5 dengan distribusi seragam. Sedangkan pada metode

Nguyen-Widrow, bobot-bobot antara lapisan masukan dengan lapisan tersembunyi ( $v_{ij}$ ) ditentukan secara acak dan dengan perhitungan tertentu, seperti yang dijelaskan pada bab II. Selain itu bobot-bobot antara lapisan tersembunyi dengan lapisan keluarannya ( $w_{jk}$ ) ditentukan secara acak, sama seperti pada metode bias.

**Tabel 4.4.** Recognition Rate dan Recognition Time dengan Variasi Metode Inisialisasi Awal Bobot

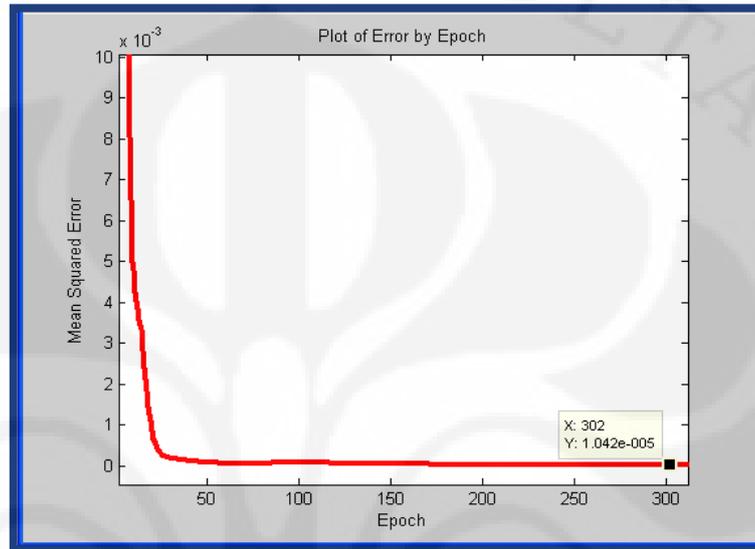
Metode Inisialisasi Bobot	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>Acak</b>	100%	87.8%	37.87	0.0106
<b>Nguyen-Widrow</b>	100%	96.4%	36.97	0.0107

Dari tabel 4.4, dapat dilihat bahwa kedua metode memiliki waktu pelatihan dan recognition yang relatif sama. Keduanya juga sudah baik untuk mengenali gambar wajah dari database dengan tercapainya recognition rate 100%. Namun inisialisasi bobot dengan metode Nguyen-Widrow akan menghasilkan recognition rate yang lebih baik pada pengenalan gambar wajah di luar database. Hal ini berarti dengan metode Nguyen-Widrow, bobot akhir yang dihasilkan sesuai pelatihan akan lebih peka terhadap noise karena gambar di luar database berbeda dengan gambar di dalam database yang telah dikenali sistem.

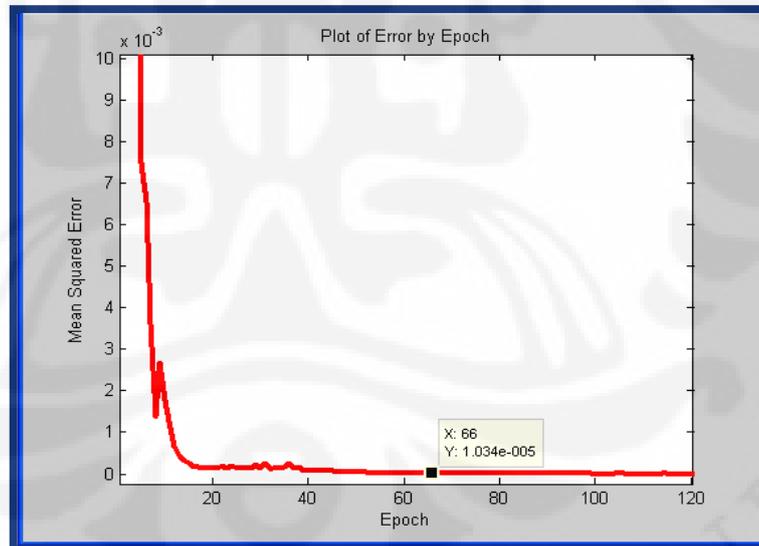
Hal ini dapat dijelaskan sebagai berikut. Berdasarkan Jong, Jek Siang (2005), bobot awal akan mempengaruhi apakah jaringan mencapai titik minimum lokal atau global, dan seberapa cepat konvergensinya. Nilai bobot awal tidak boleh terlalu besar karena nilai turunan fungsi aktivasinya akan menjadi sangat kecil sehingga perubahan bobotnya menjadi sangat kecil. Oleh karena itu, bobot dan bias sebaiknya diisi dengan bilangan acak kecil, seperti yang dilakukan pada metode Nguyen-Widrow.

Dari bobot-bobot yang diperoleh dari kedua metode, ternyata nilai  $v_{ij}$  dan  $v_{0j}$  pada metode Nguyen-Widrow lebih kecil dibandingkan

pada metode acak. Hal ini akan menyebabkan nilai turunan fungsi aktivasinya,  $f'(z_{net_j})$  dan  $f'(y_{net_k})$ , tidak terlalu kecil sehingga perubahan bobotnya,  $\Delta w$  dan  $\Delta v$ , juga tidak terlalu kecil. Akibatnya, iterasi yang diperlukan pun lebih singkat.



4.6.a



4.6.b

**Gambar 4.6.** Grafik Error terhadap Epoch (diperbesar) (a) Metode Acak  
(b) Metode Nguyen-Widrow

Dari data eksperimen dan grafik error terhadap epoch pada gambar 4.6, dapat dilihat bahwa kedua metode menghasilkan error

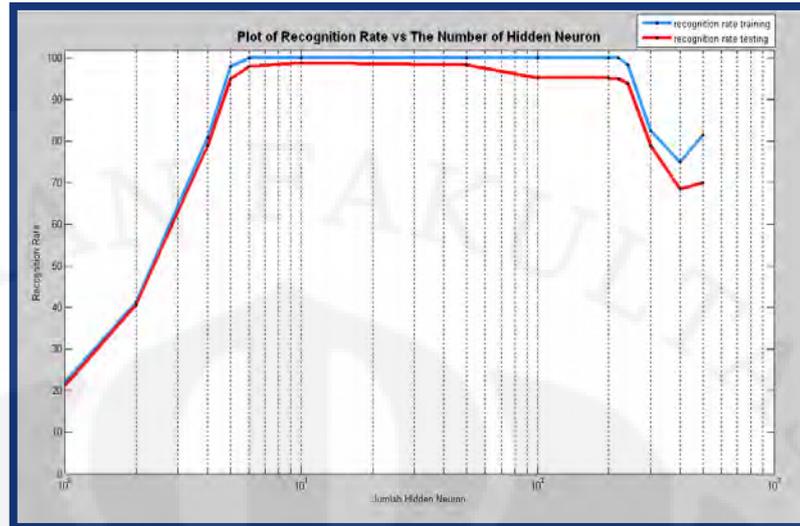
yang konvergen. Pada metode acak, error akan mencapai batas toleransi umum ( $1 \times 10^{-5}$ ) ketika iterasi mencapai sekitar 300 epoch sedangkan pada metode Nguyen-Widrow, batas toleransi error ini telah dicapai di sekitar epoch ke-60. Hal ini membuktikan bahwa inisialisasi dengan metode Nguyen-Widrow memerlukan iterasi yang lebih singkat dibandingkan metode acak.

#### 4.3.2. Jumlah Unit Lapisan Tersembunyi

Pada eksperimen, jumlah unit pada lapisan tersembunyi divariasikan antara 1, 10, 50, 100, 200, 300, 400, dan 500. Namun untuk mencari jumlah unit yang optimal, divariasikan pula untuk jumlah unit 2, 4, 5, 6, 220, dan 240. Perlu diingat eksperimen ini hanya berlaku untuk skema default. Berikut ini adalah hasilnya:

**Tabel 4.5.** Recognition Rate dan Recognition Time dengan Variasi Jumlah Unit Lapisan Tersembunyi

Jumlah Hidden Neuron	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
1	22.2%	21.3%	6.79	0.0106
2	41.3%	40.8%	5.43	0.0106
4	81.0%	78.9%	5.86	0.0106
5	98.0%	95.1%	6.11	0.0106
6	100%	98.1%	6.41	0.0106
10	100%	99.0%	7.11	0.0107
50	100%	98.4%	19.89	0.0108
100	100%	96.4%	36.97	0.0107
200	100%	95.3%	68.54	0.0109
220	100%	95.1%	75.21	0.0108
240	98.4	93.9%	57.40	0.0105
300	82.5%	78.9%	94.98	0.0108
400	75.0%	68.5%	128.53	0.0108
500	81.5%	70.1%	218.20	0.0107



**Gambar 4.7.** Grafik Recognition Rate terhadap Jumlah Unit Lapisan Tersembunyi

Dari tabel 4.5, secara umum dapat dinyatakan bahwa semakin banyak unit pada lapisan tersembunyi, semakin lama waktu pembelajaran (training), tetapi waktu pengenalannya (testing) ternyata sama. Hal ini disebabkan karena semakin banyak jumlah neuron pada lapisan tersembunyi, jumlah bobot  $w_{jk}$  dan  $v_{ij}$  akan menjadi lebih banyak sehingga modifikasi bobot dalam satu epoch juga lebih banyak dan pelatihan berlangsung lebih lama.

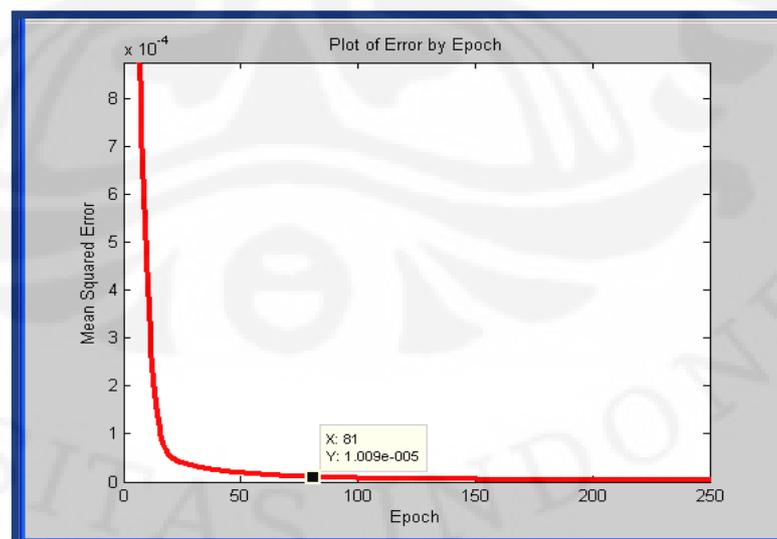
Awalnya peneliti melakukan eksperimen dengan 1, 10, 50, 100, 200, 300, 400, dan 500 neuron saja. Dari situ diperoleh bahwa recognition rate untuk data dalam database (training) mencapai 100% pada 10, 50, 100, dan 200 neuron saja, sedangkan di luar itu tidak. Dengan demikian, untuk mencapai hasil yang optimal ada batas-batas minimum dan maksimum jumlah neuron pada lapisan tersembunyi. Untuk mencari batas-batas tersebut, peneliti mencoba eksperimen tambahan untuk 2, 4, 5, 6, 220, dan 240 neuron.

Ternyata dari eksperimen diperoleh bahwa recognition rate untuk data dalam database mencapai 100% saat jumlah neuron sebanyak 6. Hal ini berarti untuk skema default ini, jumlah minimal unit pada lapisan tersembunyi untuk mencapai hasil yang optimal adalah enam unit. Semakin kecil di bawah enam unit, recognition rate

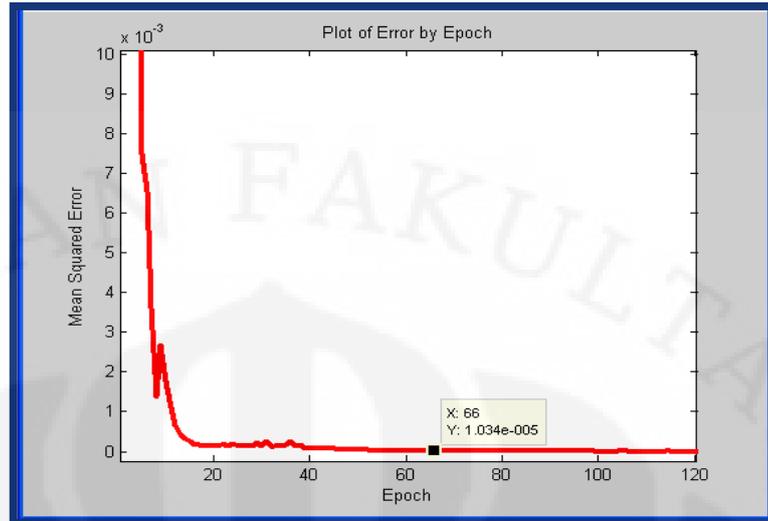
baik untuk data di dalam ataupun luar database akan semakin kecil, seperti yang ditunjukkan pada grafik.

Sementara itu, jumlah neuron maksimum untuk skema default ini adalah antara 220 dan 240. Dari tabel 4.5 dan gambar 4.7, dapat dilihat bahwa recognition rate untuk data di dalam database masih mencapai 100% ketika jumlah neuron pada lapisan tersembunyi berjumlah 220. Lebih dari itu, tampak kecenderungan di mana recognition rate semakin rendah seiring bertambahnya jumlah neuron. Dari eksperimen yang telah dilakukan, dapat dilihat bahwa skema yang paling optimal adalah ketika jumlah neuron pada lapisan tersembunyi sebanyak 10 unit, karena selain recognition rate yang tinggi, waktu yang diperlukan untuk melatih jaringan juga relatif singkat.

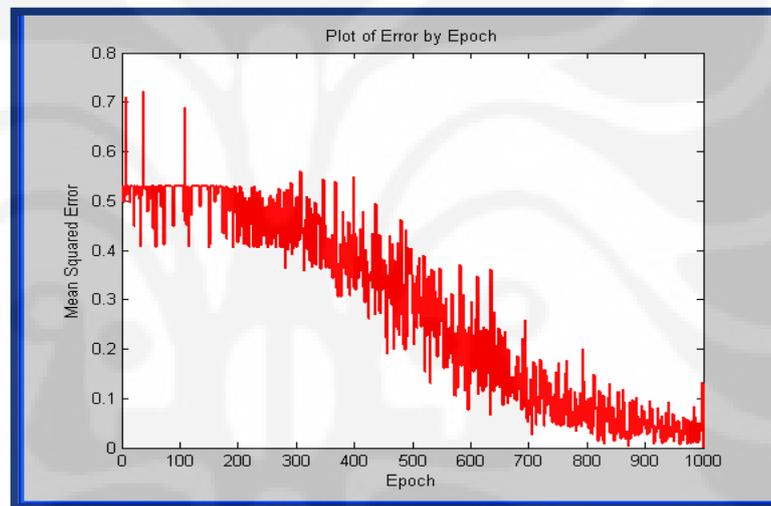
Gambar 4.8 menunjukkan perbandingan grafik untuk jumlah neuron pada lapisan tersembunyi sebanyak 10, 100, dan 500 untuk mewakili jumlah yang kecil, sedang, dan besar. Secara keseluruhan, untuk mencapai batas toleransi error ( $1 \times 10^{-5}$ ) jaringan dengan 10 neuron pada lapisan tersembunyi memerlukan iterasi sebanyak 80-140 kali, sedangkan jaringan dengan 100 neuron memerlukan hanya 50-70 iterasi.



4.8.a



4.8.b



4.8.c

**Gambar 4.8.** Grafik Error terhadap Epoch dengan Jumlah Unit Lapisan Tersembunyi (a)10 neuron (diperbesar) (b)100 neuron (diperbesar) (c)500 neuron

Namun secara umum jumlah unit lapisan tersembunyi sebanyak 10 neuron memiliki konvergensi yang lebih baik dan tidak sefluktuatif seperti pada 100 neuron. Sementara itu, jaringan dengan 500 neuron pada lapisan tersembunyi tampak tinggi dan sangat fluktuatif meskipun pada akhirnya error semakin kecil seiring bertambahnya iterasi. Dengan 1000 iterasi pada percobaan, jaringan belum mencapai nilai toleransi error. Hal ini menjelaskan mengapa recognition rate cenderung buruk pada jumlah neuron yang terlalu besar.

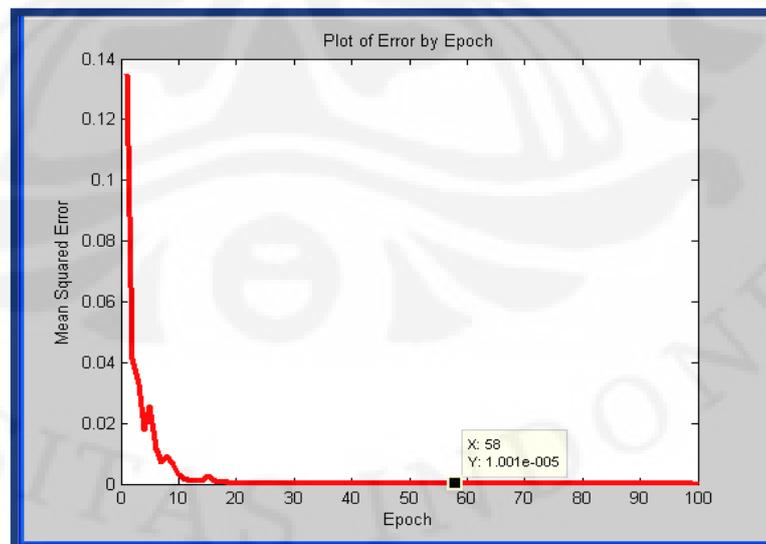
### 4.3.3. Banyaknya Iterasi yang Dilakukan

Pada parameter ini, dilakukan eksperimen dengan tiga variasi, yaitu 100, 1000, dan 10000 epoch. Berikut ini adalah hasilnya:

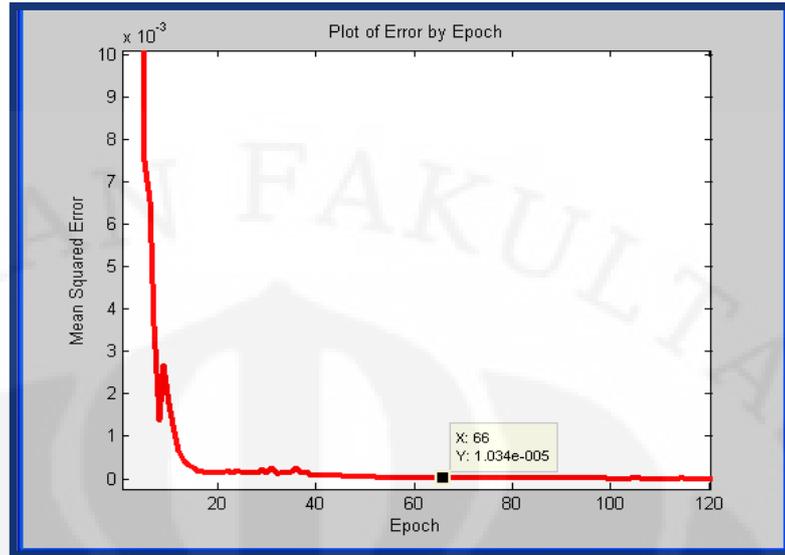
**Tabel 4.6.** Recognition Rate dan Recognition Time dengan Variasi Banyaknya Iterasi yang Dilakukan

Banyaknya Iterasi yang Dilakukan	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>100</b>	100%	96.3%	4.65	0.0106
<b>1000</b>	100%	96.4%	36.97	0.0107
<b>10000</b>	100%	96.0%	386.10	0.0106

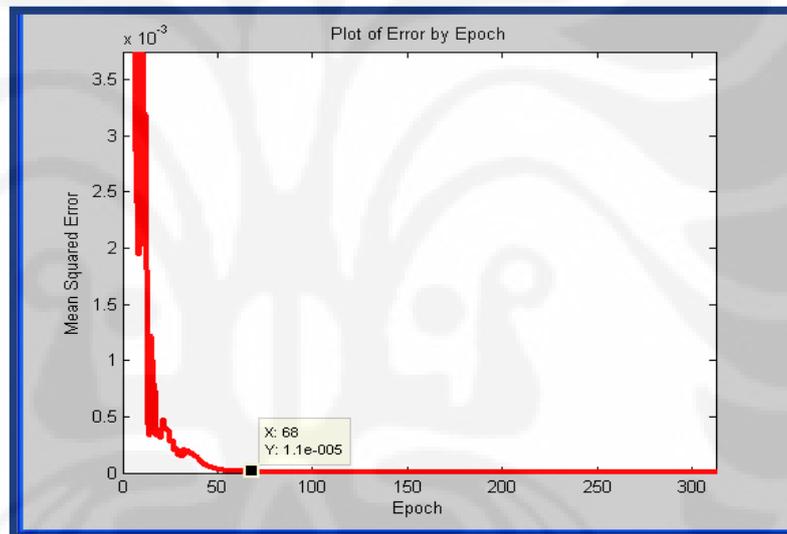
Dari tabel 4.6, dapat diamati bahwa ketiga variasi memiliki recognition rate yang besarnya tidak jauh berbeda. Untuk data dari dalam database, recognition rate mencapai 100% sedangkan untuk data di luar database, recognition rate berkisar di antara 96.0-96.4%. Hal ini disebabkan karena batas toleransi error ( $1 \times 10^{-5}$ ) yang dicapai ketiga variasi kurang lebih sama, yaitu di antara epoch ke-50 sampai 80. Lebih dari epoch tersebut, error akan berkurang seiring iterasi tetapi tidak terlalu signifikan, meskipun iterasi dilakukan sampai 10000 kali. Berikut adalah grafik error terhadap epoch yang telah diperbesar.



4.9.a



4.9.b



4.9.c

**Gambar 4.9.** Grafik Error terhadap Epoch

(a)100 epoch (b)1000 epoch (c)10000 epoch

Pada variasi ini, yang menjadi perbedaan hanyalah waktu pembelajarannya. Semakin banyak iterasi yang dilakukan, semakin lama waktu yang dibutuhkan jaringan untuk melatih dirinya. Atas dasar recognition rate yang besarnya tidak jauh berbeda, maka dengan iterasi sebanyak 100 epoch sebenarnya sudah cukup baik untuk diterapkan pada sistem. Atau dapat disimpulkan, jika banyaknya iterasi yang ditetapkan sebagai syarat henti telah melampaui epoch dengan

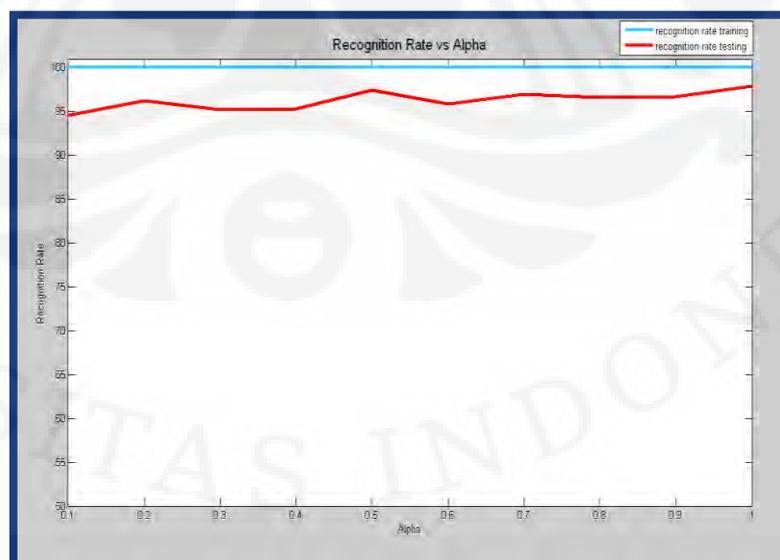
batas toleransi error, sistem dapat mempelajari pola dan mengenali wajah dengan baik.

#### 4.3.4. Besarnya Laju Pembelajaran ( $\alpha$ )

Besarnya laju pembelajaran mempengaruhi seberapa cepat jaringan akan mencapai konvergensi error. Pada eksperimen, terdapat sepuluh variasi nilai  $\alpha$ , yaitu 0.1, 0.2, 0.3, hingga 1.0. Berikut ini adalah hasil percobaannya:

**Tabel 4.7.** Recognition Rate dan Recognition Time dengan Variasi Besarnya Laju Pembelajaran

Laju Pembelajaran ( $\alpha$ )	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
0.1	100%	94.6%	37.72	0.0105
0.2	100%	96.2%	37.03	0.0105
0.3	100%	95.2%	37.16	0.0106
0.4	100%	95.3%	37.22	0.0105
0.5	100%	96.4%	36.97	0.0107
0.6	100%	95.9%	37.33	0.0106
0.7	100%	97.0%	36.91	0.0106
0.8	100%	96.6%	37.33	0.0106
0.9	100%	96.7%	36.84	0.0107
1.0	100%	97.9%	37.28	0.0106

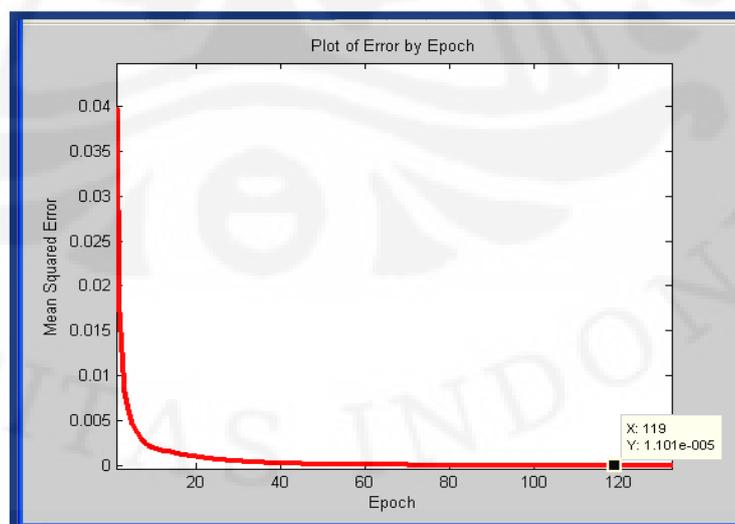


**Gambar 4.10.** Grafik Recognition Rate vs. Alpha

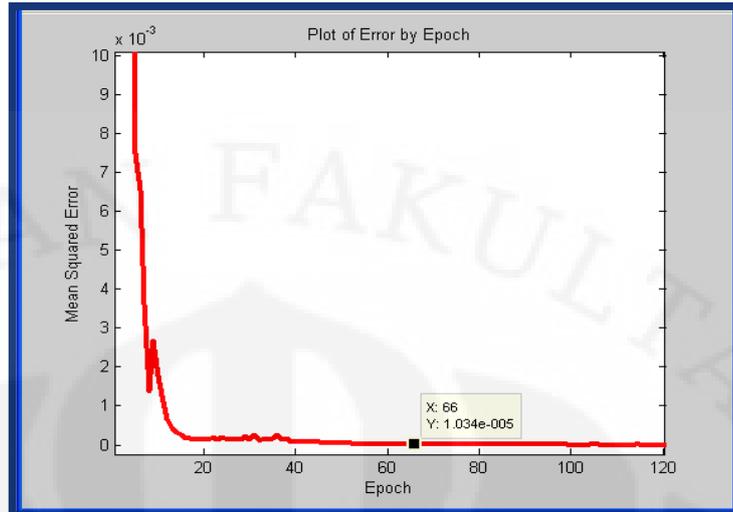
Dari tabel 4.7 dan grafik 4.10, dapat diamati recognition rate untuk data di dalam database (training) mencapai 100% untuk semua nilai  $\alpha$ , namun untuk data di luar database (testing), nilainya bervariasi dan berfluktuasi, antara 95.2% - 97.9%. Dengan demikian, besarnya nilai  $\alpha$  tidak akan mempengaruhi tingkat akurasi pengenalan wajah karena pada intinya sebesar apapun nilai  $\alpha$ , konvergensi berkurangnya error dapat tercapai dan sistem dapat mengenali wajah dengan baik apabila errornya kecil.

Sedangkan waktu pembelajaran semua variasi  $\alpha$  juga tidak berbeda jauh. Hal ini disebabkan karena syarat henti pada percobaan ini adalah jumlah iterasi/epoch, sehingga sebesar apapun nilai  $\alpha$ , jaringan tetap akan melakukan modifikasi bobot sebanyak iterasi yang ditentukan. Dengan demikian nilai  $\alpha$  tidak akan mempengaruhi lamanya waktu pembelajaran. Kecuali apabila syarat henti yang digunakan adalah besarnya nilai error. Jika demikian, maka semakin besar nilai  $\alpha$ , jaringan akan semakin cepat mencapai konvergensi dan waktu pembelajaran pun menjadi lebih singkat.

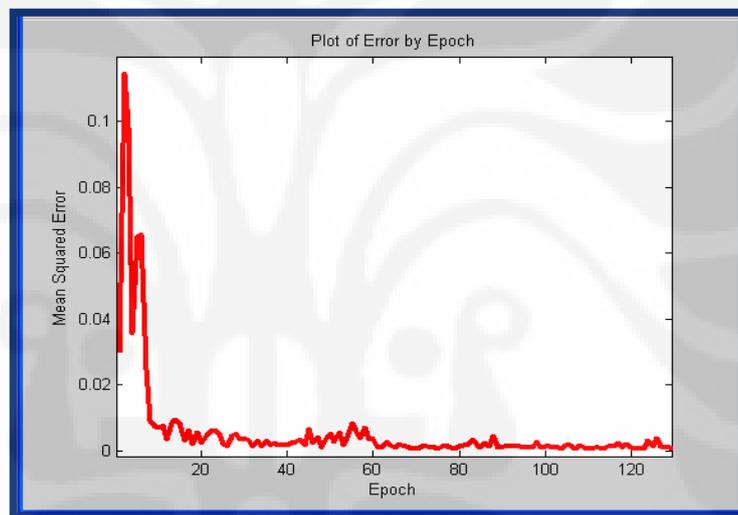
Perbedaan yang paling mencolok pada percobaan dengan variasi nilai  $\alpha$  ini adalah grafik error terhadap epoch, seperti yang ditunjukkan pada gambar 4.11.



4.11.a



4.11.b



4.11.c

**Gambar 4.11.** Grafik Error terhadap Epoch (a)  $\alpha=0.1$  (b)  $\alpha=0.5$  (c)  $\alpha=0.9$

Gambar 4.11 menunjukkan grafik error terhadap epoch untuk  $\alpha = 0.1, 0.5,$  dan  $0.9$  dengan tujuan mewakili nilai  $\alpha$  yang kecil, sedang, dan besar. Untuk  $\alpha = 0.1$ , grafik error tampak sangat halus (tidak berfluktuasi), karena nilai  $\alpha$  kecil sehingga modifikasi bobot dan penurunan error terjadi sedikit demi sedikit. Secara keseluruhan, error akan mencapai batas toleransi pada epoch ke-100 sampai 140.

Untuk  $\alpha = 0.5$ , grafik error agak berfluktuasi di awal karena modifikasi bobot lebih besar dan penurunan error terjadi lebih cepat, sehingga penurunan error ini melewati nilai minima lokalnya.

Akibatnya, dengan  $\alpha$  yang sama, perhitungan akan berbalik arah dan error menjadi besar kembali. Hal ini berlangsung sampai minima global ditemukan. Namun kadangkala nilai  $\alpha$  ini masih terlalu besar sehingga meskipun telah mendekati error minimum, grafik akan terus berosilasi, naik-turun mencari minima globalnya.

Untuk  $\alpha = 0.9$ , grafik error sangat berfluktuasi dari awal hingga akhir iterasi. Hal ini disebabkan karena nilai  $\alpha$  sangat besar sehingga sistem sering melampaui minima lokal, dan grafik akan berosilasi terus-menerus. Akibatnya minima global tidak dapat ditemukan. Dengan demikian error minimum tidak akan pernah dicapai.

#### 4.3.5. Penggunaan Momentum

Pada algoritma backpropagation, perubahan bobot didasarkan atas gradien yang terjadi untuk pola yang dimasukkan saat itu. Modifikasi bobot yang didasarkan atas arah gradien pola terakhir dan pola sebelumnya disebut dengan momentum. Dengan demikian, tidak hanya pola masukan terakhir saja yang diperhitungkan.

Penambahan momentum (persamaan 2.14 dan 2.15) dimaksudkan untuk menghindari perubahan bobot yang mencolok akibat adanya data yang sangat berbeda dengan data yang lain (outlier). Apabila beberapa data terakhir yang diberikan ke jaringan memiliki pola serupa, berarti arah gradien sudah benar, maka perubahan bobot dilakukan secara cepat. Namun apabila data terakhir yang dimasukkan memiliki pola yang berbeda dengan pola sebelumnya, maka perubahan dilakukan secara lambat.<sup>20</sup>

Pada eksperimen ini, akan dibandingkan performa sistem jika tidak menggunakan momentum, menggunakan momentum dengan  $\mu = 0.1, 0.2$ , dan seterusnya hingga 1.0. Berikut ini adalah hasilnya jika menggunakan parameter  $\alpha = 0.5$ :

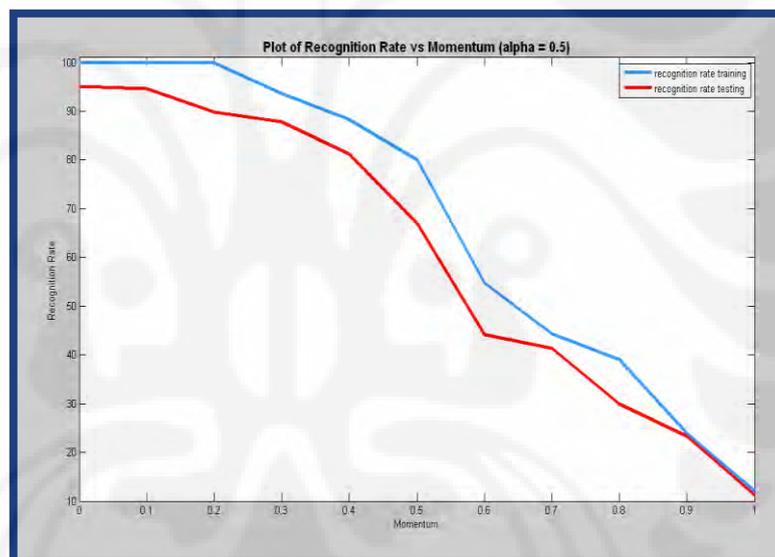
---

<sup>20</sup> Ibid, hal.113.

\*tidak menggunakan momentum

**Tabel 4.8.** Recognition Rate dan Recognition Time dengan Variasi Momentum ( $\alpha = 0.5$ )

$\mu$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0*</b>	100%	96.4%	36.97	0.0107
<b>0.1</b>	100%	93.9%	42.42	0.0101
<b>0.2</b>	100%	89.8%	42.45	0.0103
<b>0.3</b>	93.6%	87.8%	42.26	0.0103
<b>0.4</b>	88.4%	81.2%	42.49	0.0104
<b>0.5</b>	80.0%	67.0%	42.24	0.0103
<b>0.6</b>	54.8%	44.2%	42.32	0.0102
<b>0.7</b>	44.4%	41.4%	42.10	0.0102
<b>0.8</b>	39.0%	30.0%	42.09	0.0103
<b>0.9</b>	24.0%	23.4%	42.33	0.0103
<b>1.0</b>	12.0%	11.2%	44.09	0.0103



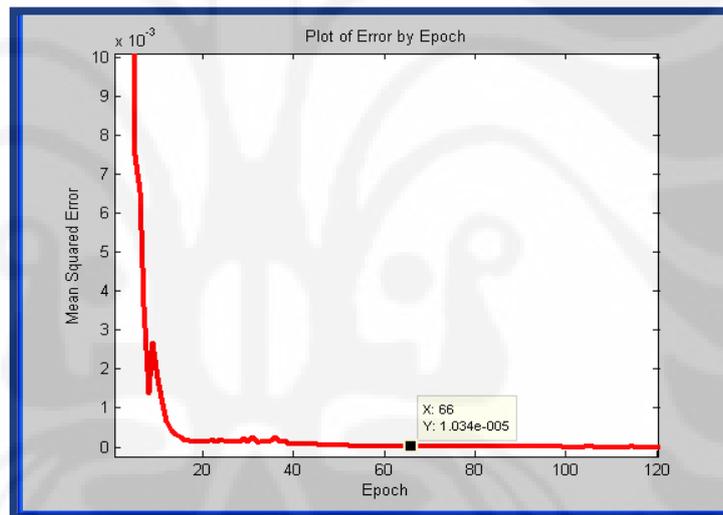
**Gambar 4.12.** Grafik Recognition Rate vs Momentum ( $\alpha = 0.5$ )

Dari tabel 4.8 dan gambar 4.12, dapat dilihat bahwa penambahan momentum untuk jaringan dengan  $\alpha = 0.5$  tidak memberi efek yang lebih baik ditinjau dari recognition rate. Untuk nilai  $\mu$  yang kecil (0.1), sistem masih dapat mempelajari pola dan mengenali gambar wajah dengan baik, meskipun ada sedikit penurunan recognition rate jika dibandingkan dengan tanpa memakai momentum.

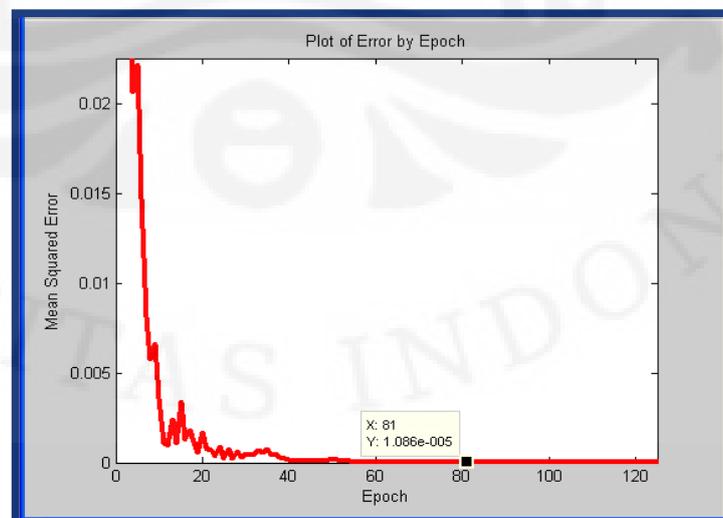
Waktu pembelajaran untuk jaringan yang menggunakan momentum sedikit lebih lama dibandingkan dengan yang tidak

menggunakan momentum karena adanya penambahan kode program untuk penggunaan momentum. Untuk semua variasi nilai  $\mu$ , waktu pembelajaran dan waktu pengenalan tidak jauh berbeda.

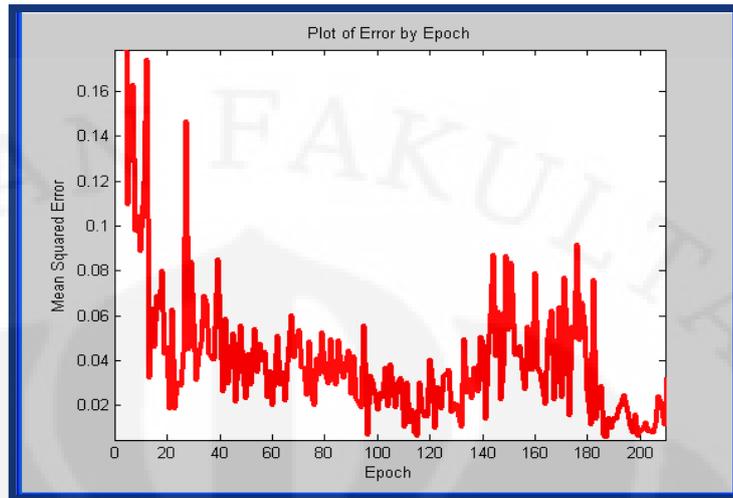
Semakin besar nilai  $\mu$ , recognition rate menjadi lebih rendah. Hal ini disebabkan karena semakin besar nilai  $\mu$ , semakin besar perubahan bobot yang diberikan, sehingga error tidak dapat menemukan minima global dan mencapai konvergensi. Grafik error terhadap epoch ditunjukkan pada gambar 4.13, dengan mengambil nilai  $\mu = 0$ ,  $\mu = 0.1$ ,  $\mu = 0.5$ , dan  $\mu = 0.9$ . Semakin besar nilai  $\mu$ , semakin besar fluktuasi error, dan bahkan untuk  $\mu = 0.9$  error bernilai sangat besar dan tidak konvergen.



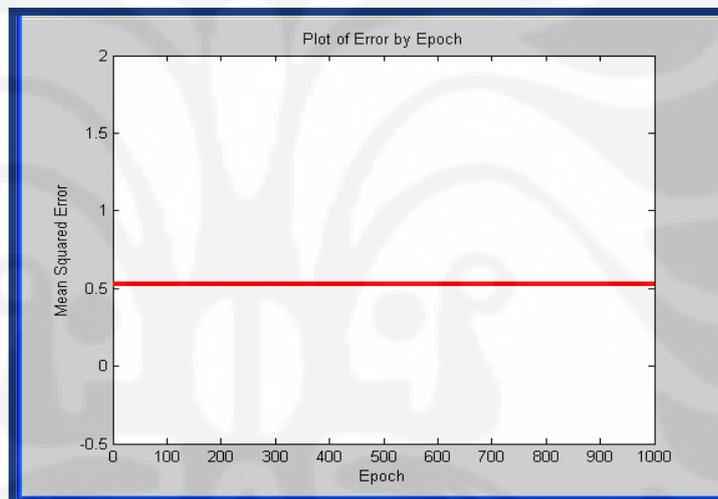
4.13.a



4.13.b



4.13.c

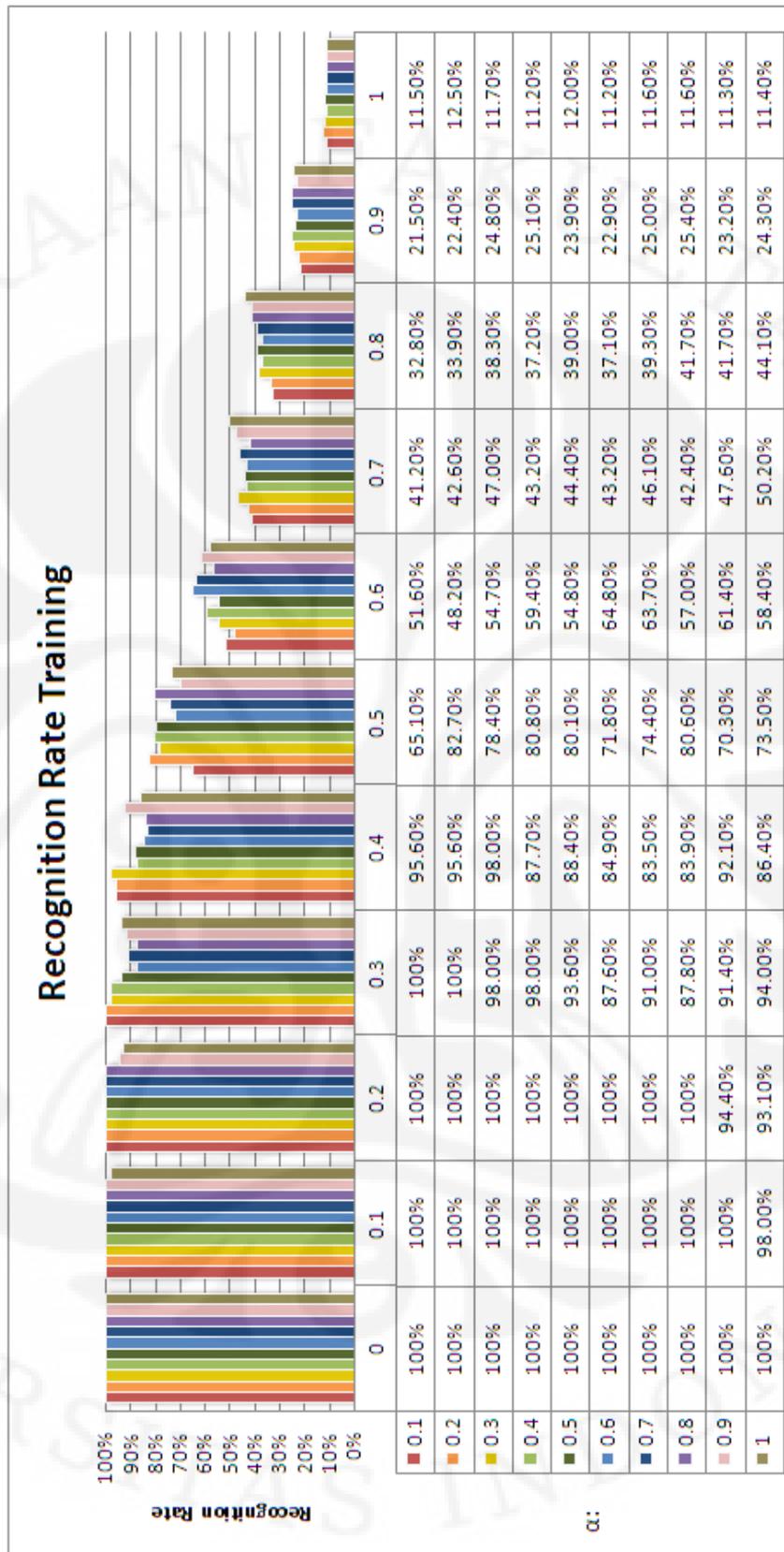


4.13.d

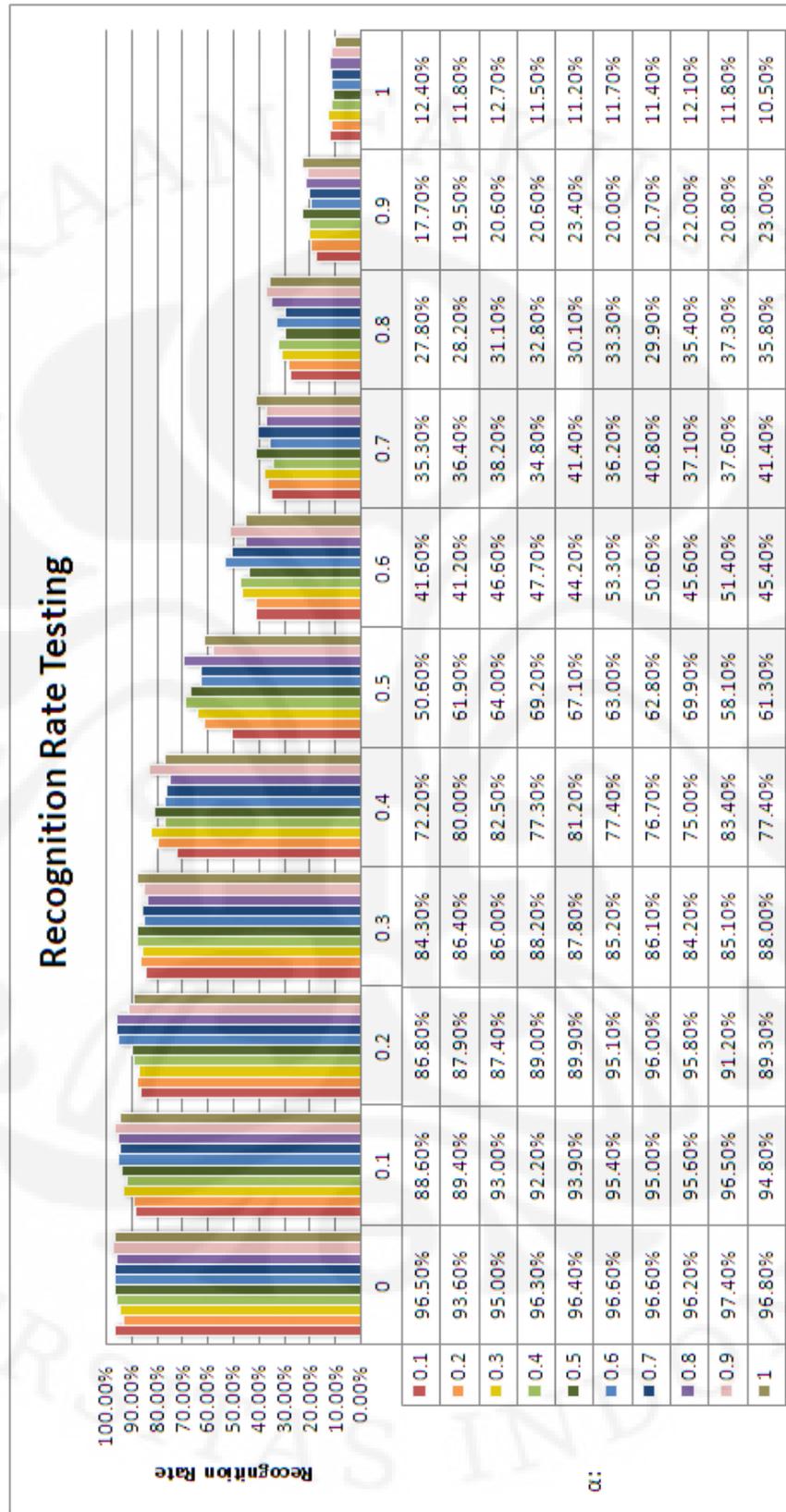
**Gambar 4.13.** Grafik Error terhadap Epoch untuk  $\alpha=0.5$ 

(a) Tanpa Momentum  $\mu = 0$ , (b)  $\mu = 0.1$  (c)  $\mu = 0.5$  (d)  $\mu = 0.9$

Karena nilai  $\mu$  dan  $\alpha$  sangat berhubungan, maka peneliti mencoba melakukan eksperimen tambahan untuk  $\alpha = 0.1, 0.2, 0.3, 0.4, 0.6, 0.7, 0.8, 0.9$ , dan  $1.0$ . Berikut ini adalah grafik yang menunjukkan recognition rate training dan testing eksperimen tambahan.



Gambar 4-14. Grafik Recognition Rate Training dengan Variasi  $\alpha$  dan  $\mu$



Gambar 4-15. Grafik Recognition Rate Testing dengan Variasi  $\alpha$  dan  $\mu$

Dari grafik recognition rate training pada gambar 4.14, dapat dilihat jika momentum tidak digunakan ( $\mu = 0$ ), jaringan dapat mengenali seluruh gambar uji dalam database (recognition rate = 100%) untuk nilai  $\alpha$  apapun. Namun, ketika momentum digunakan, semakin besar nilai  $\mu$ , semakin rendah recognition rate. Untuk nilai  $\mu$  yang relatif kecil (0.1 sampai dengan 0.4), semakin besar nilai  $\alpha$ , recognition rate cenderung menurun. Sedangkan untuk nilai  $\mu$  yang relatif besar (0.5 sampai dengan 1.0), recognition rate tampaknya tidak dipengaruhi oleh  $\alpha$ .

Dari tabel di bawah grafik, dapat dilihat bahwa nilai  $\alpha = 0.1$ , 0.2, dan 0.3 memberikan hasil yang lebih baik untuk penggunaan momentum. Ketika  $\mu = 0.3$  dan 0.4, recognition rate masih dapat mengenali gambar dengan baik. Hal ini tampak berbeda bila dibandingkan dengan recognition rate pada  $\alpha$  lainnya, di mana pada nilai  $\mu$  tersebut, jaringan sudah tidak dapat mengenali gambar uji dengan baik.

Dari grafik recognition rate testing pada gambar 4.15, dapat dilihat bahwa sama halnya dengan recognition rate training, penggunaan momentum juga berdampak pada berkurangnya recognition rate testing. Semakin besar nilai  $\mu$ , semakin rendah recognition rate untuk nilai  $\alpha$  apapun.

Untuk nilai  $\mu$  yang relatif kecil (0.1 sampai dengan 0.3), ada kecenderungan recognition rate meningkat seiring semakin besar nilai  $\alpha$  yang digunakan. Peningkatan ini terjadi hingga suatu batas nilai  $\alpha$  tertentu, dan setelah itu recognition rate akan turun atau berfluktuasi. Misalnya ketika  $\mu = 0.2$ , recognition rate meningkat dari  $\alpha = 0.1$  sampai dengan 0.7, lalu kemudian berkurang sampai dengan 1.0. Sementara itu, untuk nilai  $\mu$  yang cukup besar (0.4 sampai 1.0), recognition rate tampaknya juga tidak dipengaruhi oleh  $\alpha$ .

Dari tabel di bawah grafik, dapat dilihat bahwa nilai  $\alpha = 0.6$ , 0.7, 0.8, dan 0.9 tampak lebih baik untuk mengenali data dari luar database, jika momentum tetap ingin digunakan. Namun bila

dibandingkan dengan recognition rate training, nilai-nilai  $\alpha$  tersebut tidak memberikan hasil yang baik (tidak mencapai 100%). Jadi, penggunaan momentum kurang tepat untuk nilai  $\alpha$  yang relatif besar. Momentum lebih tepat digunakan untuk nilai  $\alpha$  yang relatif kecil (0.2 dan 0.3), kecuali  $\alpha = 0.1$  karena recognition rate testing pada nilai tersebut tidak terlalu baik.

Hal ini dapat dijelaskan sebagai berikut. Nilai  $\alpha$  yang relatif kecil, menyebabkan penambahan momentum ( $\mu$ ) seakan-akan memberikan suatu nilai  $\alpha$  tambahan dan membuat “ $\alpha$  baru” ini lebih besar. Akibatnya, error akan lebih cepat menemukan global minima dan mencapai konvergensi. Namun apabila nilai  $\mu$  yang diberikan terlalu besar, maka hal sebaliknya akan terjadi, yaitu error sulit menemukan global minima, sehingga grafik error akan terus berosilasi dan bahkan tidak mencapai konvergensi.

Hasil eksperimen menunjukkan penambahan momentum malah memperkecil recognition rate. Hal ini berbeda dengan tujuan ditamahnya momentum, yang seharusnya memperbaiki perubahan bobot sehingga pengenalan gambar wajah seharusnya lebih akurat. Hal ini mungkin disebabkan oleh beberapa hal, misalnya penambahan momentum tidak sesuai untuk komputasi pelatihan pola gambar wajah.

Selain itu, tujuan lain digunakannya momentum adalah untuk mempercepat waktu pelatihan dengan mencapai konvergensi error. Namun hasil eksperimen ini tidak menunjukkan waktu pelatihan yang lebih cepat bila menggunakan momentum. Hal ini disebabkan karena syarat henti yang digunakan pada pelatihan ini adalah jumlah epoch, bukan batas minimum toleransi error. Pelatihan berlangsung selama epoch maksimum belum terpenuhi, dan akibatnya waktu pelatihan tidak terpengaruh oleh konvergensi error.

Jadi, momentum sebaiknya diberikan jika nilai  $\alpha$  relatif kecil (0.2 dan 0.3) dan jaringan menggunakan batas minimal error sebagai syarat henti. Hal ini dapat mempercepat proses pembelajaran, namun belum tentu memberikan performa (recognition rate) yang lebih baik.

#### 4.4. Optimalisasi Jaringan Syaraf Tunggal

Dari seluruh skema eksperimen yang dilakukan, dapat disimpulkan bahwa (khusus untuk penelitian ini) sistem dapat menghasilkan recognition rate yang semakin baik jika:

- Dimensi yang direduksi PCA semakin besar (tetapi tidak terlalu signifikan)
- Menggunakan inisialisasi bobot awal Nguyen-Widrow
- Menggunakan jumlah neuron lapisan tersembunyi yang relatif sedikit
- Tidak menggunakan momentum

Sedangkan nilai laju pembelajaran  $\alpha$  tidak berpengaruh terhadap recognition rate, dan besarnya epoch juga tidak mempengaruhi recognition rate jika telah melampaui epoch dengan batas toleransi errornya.

Untuk mencapai hasil yang optimal, tidak hanya recognition rate yang menjadi acuan, tetapi waktu pelatihan dan pengenalan gambar juga menjadi pertimbangan. Dengan demikian, berdasarkan hasil yang diperoleh pada eksperimen, peneliti menentukan parameter-parameter baru yang memberikan hasil yang lebih optimal dibandingkan parameter default. Perbandingan parameter optimalisasi dengan parameter default dapat dilihat pada tabel 4.9.

**Tabel 4.9.** Perbandingan Parameter Default dan Parameter Optimalisasi

No.	Parameter	Parameter Default	Parameter Optimalisasi
1	Dimensi yang direduksi PCA	100	50
2	Metode inisialisasi bobot awal	Nguyen-Widrow	Nguyen-Widrow
3	Banyaknya unit lapisan tersembunyi	100	10
4	Besarnya epoch	1000	100
5	Besarnya laju pembelajaran ( $\alpha$ )	0.5	0.5
6	Penggunaan momentum ( $\mu$ )	0.0	0.0

Berikut ini adalah perbandingan hasil yang diperoleh parameter optimalisasi dan parameter default, ditinjau dari recognition rate dan recognition time.

**Tabel 4.10.** Perbandingan Recognition Rate dan Recognition Time Parameter Default dan Parameter Optimalisasi

Parameter	Recognition Rate		Waktu (detik)			
	Training	Testing	Pre-Processing	Training	Testing (1 gambar)	Total
<b>Default</b>	100%	96.4%	32.06	36.97	0.0107	69.04
<b>Optimalisasi</b>	100%	99.2%	34.11	0.77	0.0102	34.89

Dari tabel 4.10, dapat dilihat dengan parameter-parameter yang diubah seperti pada tabel 4.9, recognition rate dapat meningkat dari 96.4% ke 99.2%. Sementara itu, waktu pelatihan juga dipersingkat cukup banyak, yaitu dari 36.97 detik menjadi hanya 0.77 detik. Dalam aplikasi, hasil yang menunjukkan recognition rate yang tinggi dan waktu yang singkat seperti ini amat diutamakan. Namun hasil yang ditunjukkan ini belum tentu merupakan hasil yang paling optimal, karena masih ada kemungkinan-kemungkinan lain yang belum dilakukan pada eksperimen.

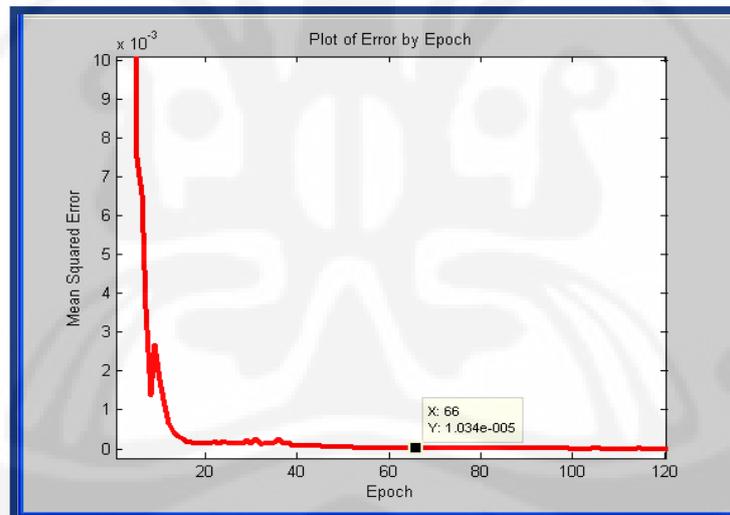
**Tabel 4.11.** Recognition Rate per Kelas pada Parameter Optimalisasi

No	Kelas	Recognition Rate (Testing)	Gambar Salah Uji
1	ann 	100%	-
2	dau 	100%	-
3	cin 	100%	-
4	roy 	100%	-
5	dev 	100%	-
6	chr 	100%	-
7	pel 	91%	69,70
8	rud 	100%	-
9	eni 	98%	87
10	pep 	100%	-

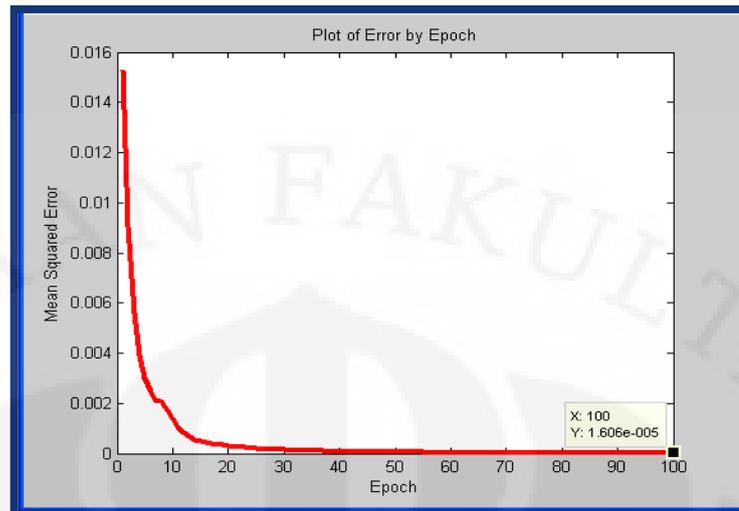
Tabel 4.11 menunjukkan recognition rate tiap kelas untuk parameter optimalisasi. Dari tabel tersebut, dapat dilihat jumlah kelas yang mencapai 100% bertambah banyak. Kelas ke-1, 2, dan 4 yang sebelumnya (pada

parameter default) tidak mencapai 100%, kini dapat mencapai 100%. Selain itu recognition rate pada kelas ke-7 dan 9 juga meningkat. Dari percobaan yang dilakukan, beberapa gambar yang salah teridentifikasi pada parameter default telah dapat diidentifikasi dengan benar, seperti gambar 2, 7, 13, dan lainnya. Namun ada beberapa gambar yang masih salah teridentifikasi, yaitu gambar 69, 70, dan 87, di mana ketiga gambar ini termasuk gambar yang paling sering salah teridentifikasi pada parameter default.

Faktor yang menyebabkan masih salahnya pengenalan wajah pada gambar-gambar tersebut yaitu perbedaan yang cukup mencolok antara gambar tersebut dengan gambar-gambar lainnya pada pelatihan. Dengan demikian, dapat disimpulkan perubahan parameter ini memang dapat meminimalisir kesalahan pada pengenalan wajah, tetapi jaringan masih belum dapat mengidentifikasi 100% dengan benar wajah-wajah yang terlalu berbeda dengan gambar yang dipelajari.



4.16.a



4.16.b

**Gambar 4.16.** Grafik Error terhadap Epoch  
(a) Parameter Default, (b) Parameter Optimalisasi

Gambar 4.16 menunjukkan perbandingan grafik error terhadap epoch untuk parameter default dan optimalisasi. Tampak bahwa grafik error kedua parameter dapat mencapai konvergensi dan nilai yang cukup kecil dalam 100 epoch. Selibuhnya, tidak ada perbedaan yang mencolok pada grafik error terhadap epoch antara kedua parameter.

## BAB V

### ANALISIS PERFORMA JARINGAN SYARAF ENSEMBLE

Dalam bab sebelumnya, penulis telah memaparkan analisis mengenai jaringan syaraf tiruan tunggal (*single neural network*) dengan algoritma backpropagation untuk mempelajari dan mengenali pola gambar wajah. Dari hasil eksperimen tersebut dapat disimpulkan bahwa jaringan syaraf tunggal sudah dapat melakukan pembelajaran dan pengenalan pola dengan baik. Performa jaringan dipengaruhi oleh parameter-parameter seperti dimensi baru yang direduksi PCA, inisialisasi bobot awal, jumlah unit neuron pada lapisan tersembunyi, dan penggunaan momentum.

Namun, perkembangan teknologi memacu manusia untuk terus menciptakan sistem dengan performa yang lebih baik. Demikian pula dengan sistem pengenalan wajah, di mana pengembangan-pengembangan dilakukan untuk memperbaiki performanya. Salah satu pengembangan tersebut adalah dengan mengaplikasikan jaringan syaraf ensemble (*ensemble neural network*) pada sistem yang telah dibangun sebelumnya. Pada bab ini, penulis akan memaparkan hasil eksperimen dan analisis performa jaringan syaraf ensemble dengan empat skema, yaitu skema algoritma backpropagation, skema parsialisasi data dengan algoritma backpropagation, skema algoritma *Negative Correlation Learning* (NCL), dan skema NCL dengan parsialisasi data. Setelah itu, keempat skema akan dibandingkan.

#### 5.1. Skema 1: Algoritma Backpropagation

Sistem jaringan syaraf ensemble dengan algoritma backpropagation pada umumnya sama seperti jaringan syaraf tunggal. Yang menjadi perbedaan adalah jumlah jaringan yang digunakan tidak hanya satu, melainkan beberapa jaringan. Selain itu, proses pengambilan keputusan juga memperhatikan hasil atau keluaran dari setiap jaringan.

Proses pembelajaran jaringan syaraf ensemble dengan algoritma backpropagation dapat dijelaskan sebagai berikut. Awalnya, jumlah jaringan ditentukan. Setelah itu, setiap jaringan diberi bobot awal yang berbeda-beda

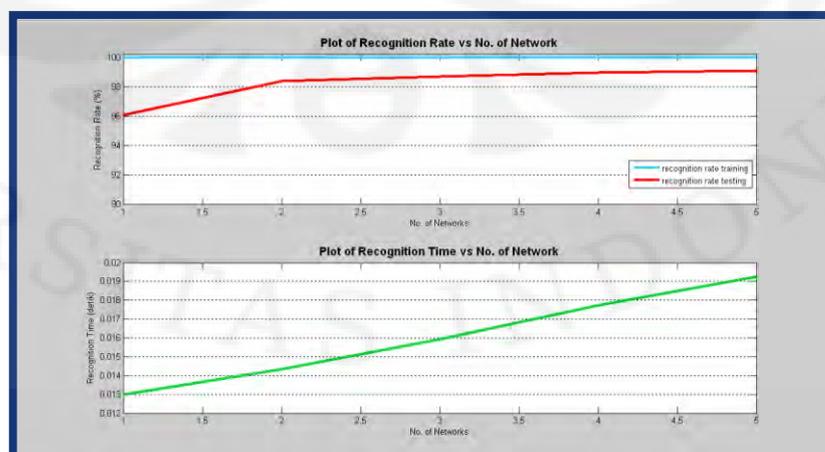
namun menggunakan metode yang sama (dalam hal ini metode Nguyen-Widrow). Setiap jaringan melakukan pembelajaran secara bersama-sama dengan parameter-parameter yang sama pula hingga akhirnya diperoleh bobot akhir setiap jaringan yang berbeda-beda. Proses pembelajaran ini menggunakan algoritma yang sama dengan backpropagation, yang terdiri dari propagasi maju dan propagasi balik.

Setelah masing-masing jaringan memiliki bobot akhir, tahap selanjutnya adalah melakukan pengujian gambar. Masing-masing gambar uji dijadikan masukan setiap jaringan, dan pada akhirnya setiap jaringan akan menghasilkan suatu vektor keluaran. Untuk mengambil keputusan akhir, keluaran masing-masing jaringan dikombinasikan dengan metode rata-rata sederhana (*simple averaging*).

Peneliti menentukan banyaknya jaringan yang akan diujicoba, yaitu 1, 2, 3, 4, dan 5 jaringan. Seperti yang dijelaskan pada bab III, skema penelitian menggunakan parameter default seperti yang dilakukan pada eksperimen jaringan syaraf tunggal. Berikut ini adalah hasilnya:

**Tabel 5.1.** Recognition Rate dan Recognition Time Skema 1 Algoritma Backpropagation

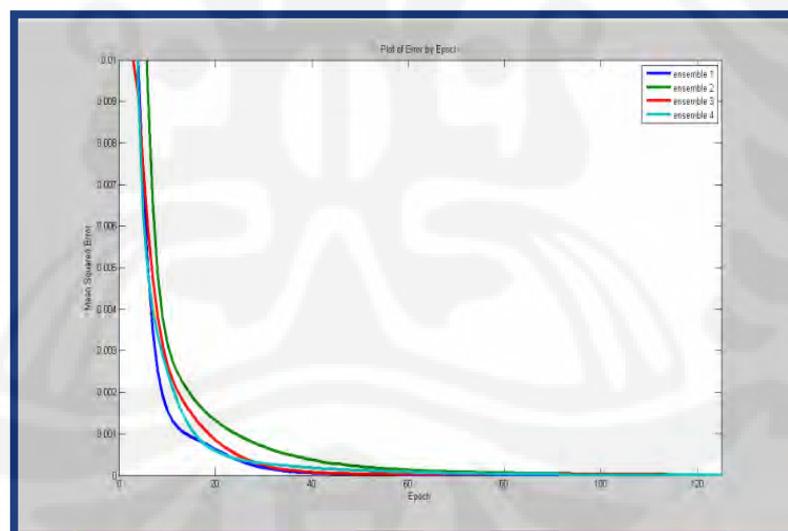
Jumlah Jaringan	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
1	100%	96.1%	25.9373	0.0130
2	100%	98.4%	51.8802	0.0143
3	100%	98.7%	77.8318	0.0159
4	100%	99.0%	104.4128	0.0177
5	100%	99.1%	130.0167	0.0193



**Gambar 5.1.** Grafik Recognition Rate dan Recognition Time terhadap Jumlah Network

Dari tabel 5.1 dan gambar 5.1, dapat dilihat bahwa semakin banyak jumlah jaringan yang digunakan, semakin tinggi recognition rate, terutama untuk data uji di luar database. Untuk data uji dari dalam database, recognition rate mencapai 100%. Peningkatan paling signifikan tampak dari satu jaringan ke dua jaringan. Semakin banyak jaringan, maka semakin banyak pula vektor keluaran yang dijadikan sebagai referensi keluaran akhir. Dengan menggunakan metode rata-rata untuk menghasilkan keluaran akhir, semakin banyak referensi maka kemungkinan jawaban untuk benar juga semakin besar.

Namun, semakin banyak jaringan yang digunakan, semakin lama waktu yang dibutuhkan untuk mempelajari pola gambar dan semakin lama waktu yang dibutuhkan untuk mengenali pola pada satu gambar. Hal ini dikarenakan banyaknya jaringan akan membuat komputasi menjadi lebih banyak sehingga waktu yang dibutuhkan pun cukup lama. Namun kenaikan pada recognition time tidak terlalu besar, di mana sampai dengan lima jaringan waktu yang dibutuhkan pun masih relatif singkat.



**Gambar 5.2.** Grafik Error terhadap Epoch Skema 1 untuk Jumlah Jaringan = 4

Gambar 5.2 menunjukkan grafik error terhadap epoch, dengan mengambil contoh jaringan ensemble dengan empat jaringan. Fungsi error yang digunakan adalah *Root Mean Squared (RMS) Error*. Dari grafik tersebut dapat dilihat bahwa setiap jaringan memiliki nilai error sendiri yang berbeda-

beda tetapi menunjukkan konvergensi, di mana nilai errornya semakin mendekati nol seiring dilakukannya iterasi dalam pelatihan.

## 5.2. Skema 2: Parsialisasi Data dengan Algoritma Backpropagation

Seperti yang telah dijelaskan pada bab II mengenai teori *Wisdom of Crowds*, untuk menghasilkan keputusan yang baik, diperlukan diversitas opini. Dalam sistem pengenalan wajah ini, data masukan untuk pelatihan dibuat bervariasi dan acak dengan persentase tertentu. Misalnya dengan persentase 70%, ini berarti dari database gambar yang terdiri dari 200 gambar hanya diambil 140 gambar secara acak untuk dijadikan masukan pada suatu jaringan. Dengan jumlah yang sama, gambar pun diambil secara acak dari database untuk dijadikan masukan pada jaringan berikutnya.

Peneliti menggunakan persentase sebanyak 50%, 60%, 70%, 80%, 90%, dan 100% untuk melakukan eksperimen. Selain itu jumlah jaringan pun divariasikan dari 1 hingga 5 jaringan. Sama seperti pada skema sebelumnya, pada awalnya setiap jaringan diberi bobot awal dengan metode inisialisasi Nguyen-Widrow. Setiap jaringan dilatih bersamaan dengan menggunakan algoritma backpropagation sampai diperoleh suatu bobot akhir. Setelah itu, pengujian dilakukan, baik untuk data dari dalam database maupun dari luar database.

Parameter-parameter yang digunakan adalah parameter default. Berikut ini adalah hasil eksperimen.

**Tabel 5.2.** Recognition Rate dan Recognition Time Skema 2 untuk 1 Jaringan

Persentase	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>50%</b>	98.8%	91.8%	13.0306	0.0129
<b>60%</b>	99.0%	91.1%	15.6546	0.0133
<b>70%</b>	99.8%	93.4%	18.2553	0.0138
<b>80%</b>	99.6%	93.7%	20.8509	0.0134
<b>90%</b>	100%	95.2%	23.4267	0.0129
<b>100%</b>	100%	96.7%	25.9685	0.0131

**Tabel 5.3.** Recognition Rate dan Recognition Time Skema 2 untuk 2 Jaringan

Persentase	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>50%</b>	99.6%	95.7%	37.5801	0.0154
<b>60%</b>	99.9%	96.1%	47.1084	0.0151
<b>70%</b>	100%	97.0%	53.7065	0.0156
<b>80%</b>	100%	97.4%	57.0183	0.0154
<b>90%</b>	100%	96.1%	72.5118	0.0155
<b>100%</b>	100%	97.6%	76.8299	0.0147

**Tabel 5.4.** Recognition Rate dan Recognition Time Skema 2 untuk 3 Jaringan

Persentase	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>50%</b>	99.9%	96.4%	55.0387	0.0164
<b>60%</b>	100%	97.6%	61.1023	0.0163
<b>70%</b>	100%	97.1%	72.9564	0.0164
<b>80%</b>	100%	98.5%	88.2864	0.0163
<b>90%</b>	100%	98.8%	99.4083	0.0166
<b>100%</b>	100%	98.5%	95.6300	0.0164

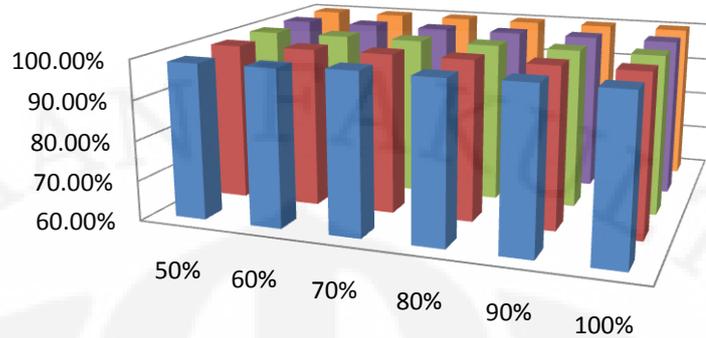
**Tabel 5.5.** Recognition Rate dan Recognition Time Skema 2 untuk 4 Jaringan

Persentase	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>50%</b>	100%	97.9%	72.8168	0.0182
<b>60%</b>	100%	97.5%	86.7051	0.0183
<b>70%</b>	100%	98.1%	107.7397	0.0184
<b>80%</b>	100%	98.7%	122.1786	0.0185
<b>90%</b>	100%	99.3%	134.4116	0.0182
<b>100%</b>	100%	99.2%	163.1739	0.0182

**Tabel 5.6.** Recognition Rate dan Recognition Time Skema 2 untuk 5 Jaringan

Persentase	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>50%</b>	100%	97.9%	88.7148	0.0199
<b>60%</b>	100%	98.1%	101.2421	0.0198
<b>70%</b>	100%	99.1%	131.5283	0.0198
<b>80%</b>	100%	99.4%	155.0767	0.0196
<b>90%</b>	100%	99.7%	172.7592	0.0196
<b>100%</b>	100%	99.3%	175.3710	0.0197

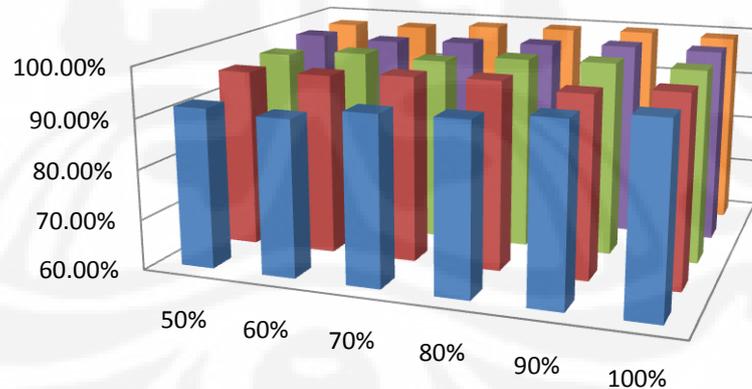
### Recognition Rate - Training



	50%	60%	70%	80%	90%	100%
1 network	98.80%	99.00%	99.80%	99.60%	100%	100%
2 networks	99.60%	99.90%	100%	100%	100%	100%
3 networks	99.90%	100%	100%	100%	100%	100%
4 networks	100%	100%	100%	100%	100%	100%
5 networks	100%	100%	100%	100%	100%	100%

Gambar 5.3. Grafik Recognition Rate Training untuk Skema 2

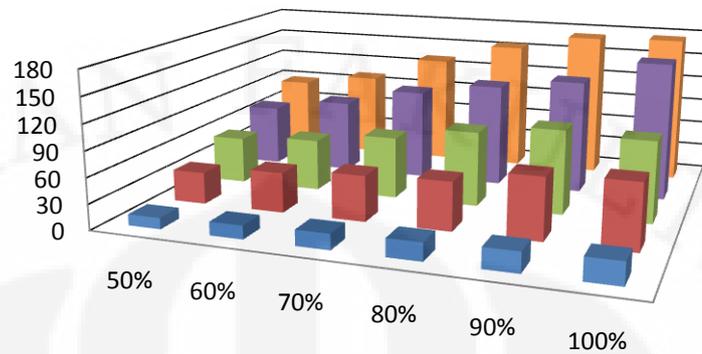
### Recognition Rate - Testing



	50%	60%	70%	80%	90%	100%
1 network	91.80%	91.10%	93.40%	93.70%	95.20%	96.70%
2 networks	95.70%	96.10%	97.00%	97.40%	96.10%	97.60%
3 networks	96.40%	97.60%	97.10%	98.50%	98.80%	98.50%
4 networks	97.90%	97.50%	98.10%	98.70%	99.30%	99.20%
5 networks	97.90%	98.10%	99.10%	99.40%	99.70%	99.30%

Gambar 5.4. Grafik Recognition Rate Testing untuk Skema 2

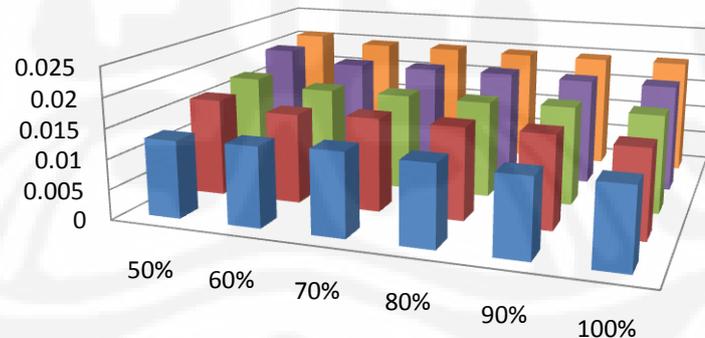
## Training Time (detik)



	50%	60%	70%	80%	90%	100%
1 network	13.0306	15.6546	18.2553	20.8509	23.4267	25.9685
2 networks	37.5801	47.1084	53.7065	57.0183	72.5118	76.8299
3 networks	55.0387	61.1023	72.9564	88.2864	99.4083	95.63
4 networks	72.8168	86.7051	107.7397	122.1786	134.4116	163.1739
5 networks	88.7148	101.2421	131.5283	155.0767	172.7592	175.371

**Gambar 5.5.** Grafik Waktu Pembelajaran (Training Time) untuk Skema 2

## Recognition Time (detik)



	50%	60%	70%	80%	90%	100%
1 network	0.0129	0.0133	0.0138	0.0134	0.0129	0.0131
2 networks	0.0164	0.0151	0.0156	0.0154	0.0155	0.0147
3 networks	0.0174	0.0163	0.0164	0.0163	0.0166	0.0164
4 networks	0.0202	0.0183	0.0184	0.0185	0.0182	0.0182
5 networks	0.0209	0.0198	0.0198	0.0196	0.0196	0.0197

**Gambar 5.6.** Grafik Waktu Pengenalan (Recognition Time) untuk Skema 2

Tabel 5.2 sampai dengan 5.6 menunjukkan recognition rate, training time, dan recognition time untuk setiap persentase pada setiap jumlah jaringan. Grafik- grafik tiga dimensi tersebut secara terpisah menampilkan hasil tersebut dengan menambahkan jumlah jaringan sebagai sumbu ketiga sehingga perbandingan antara setiap jumlah jaringan dapat diamati.

Dari gambar 5.2, dapat dilihat recognition rate untuk data dari dalam database (training) untuk persentase 50% - 80% pada 1 jaringan tidak mencapai 100%. Hal ini disebabkan karena hanya sebagian dari database yang digunakan sebagai proses pembelajaran. Akibatnya ketika sistem diminta untuk mengenali gambar dalam database secara keseluruhan, sistem tidak dapat mengidentifikasi semuanya dengan benar. Sedangkan itu, untuk persentase 90% dan 100% pada 1 jaringan, recognition rate dapat mencapai 100%. Hal ini disebabkan karena data yang digunakan untuk pelatihan sudah cukup banyak sehingga sistem dapat mengenali gambar dalam database dengan baik.

Namun, ketika jumlah jaringan bertambah menjadi 2, persentase 70% dan 80% mencapai recognition rate 100%, sementara itu persentase 50% dan 60% juga mengalami kenaikan recognition rate. Ketika jumlah jaringan bertambah menjadi 3, hanya persentase 50% yang belum mencapai recognition rate 100%, tetapi mengalami kenaikan. Pada akhirnya, ketika jumlah jaringan bertambah menjadi 4 dan 5, setiap persentase berhasil mencapai recognition rate 100%.

Sementara itu, dari gambar 5.3 dapat dilihat recognition rate untuk data dari luar database (testing). Untuk setiap jaringan, secara umum semakin besar persentase parsialisasi data, semakin tinggi recognition rate. Sedangkan untuk setiap persentase, semakin banyak jumlah jaringan yang digunakan, semakin tinggi recognition rate. Bila diperhatikan, pada jumlah jaringan yang banyak, kenaikan recognition rate akan menunjukkan pergeseran ke arah persentase kurang dari 100%. Pada eksperimen ini, tampak persentase 90% memberikan recognition rate yang sedikit lebih baik dibandingkan 100%.

Fenomena pada gambar 5.2 dan 5.3 dapat dijelaskan sebagai berikut. Ketika persentase parsialisasi data kecil, maka jumlah pola pada set data yang

digunakan untuk pelatihan juga sedikit. Sama halnya ketika kita mempelajari suatu materi dengan referensi yang sedikit, maka ilmu atau pemahaman terhadap materi tersebut pun kurang. Ketika persentase parsialisasi data dinaikkan, maka referensi untuk pembelajaran sistem menjadi lebih banyak. Dengan demikian, pemahaman sistem terhadap pola-pola wajah menjadi lebih baik dan recognition rate akan meningkat.

Ketika jumlah jaringan hanya satu, maka keputusan akhir diambil hanya berdasarkan jaringan tersebut saja. Jika keputusan dari jaringan tersebut salah, maka keputusan akhir pun salah. Namun ketika jumlah jaringan ditambah, keputusan akhir akan diambil berdasarkan jaringan-jaringan yang ada. Dengan menggunakan metode rata-rata, jaringan yang salah akan digabungkan dengan jaringan lain sehingga kemungkinan keputusan akhir yang salah menjadi kecil. Keputusan akhir akan menjadi salah bila kebanyakan jaringan memberi keputusan yang salah.

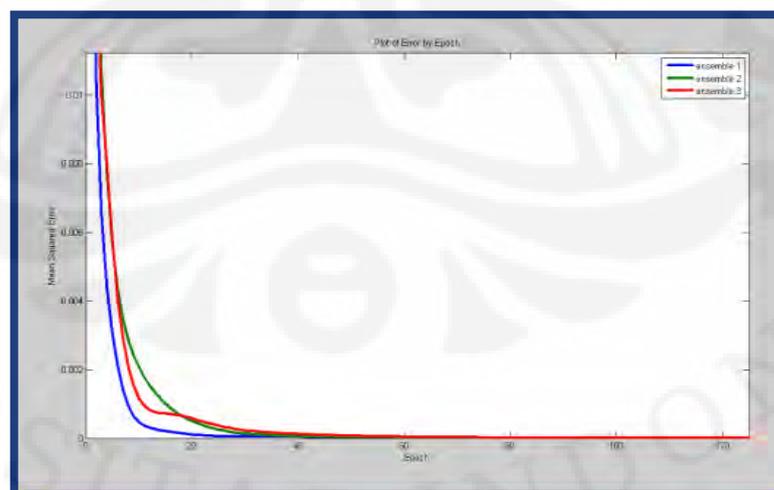
Namun penambahan jumlah jaringan yang memberi perubahan signifikan hanyalah dari 1 ke 2 jaringan dan 2 ke 3 jaringan. Sementara itu penambahan jaringan dari 3 ke 4 dan 4 ke 5 juga memberikan peningkatan recognition rate, tetapi tidak terlalu signifikan. Artinya, pengambilan keputusan dengan metode rata-rata tidak memerlukan banyak variabel untuk mengambil keputusan. Faktor lain yang mempengaruhi adalah besarnya nilai vektor target yang ditentukan dan metode pengambilan keputusan benar atau salah yang berdasarkan *Euclidean Distance*.

Dengan demikian, untuk menghasilkan keputusan yang lebih tepat, diperlukan persentase parsialisasi yang relatif besar dan jumlah jaringan yang banyak. Namun, recognition yang paling baik ditunjukkan saat jumlah jaringan banyak dan parsialisasi data 90% (bukan 100%). Hal ini disebabkan karena pada parsialisasi 100%, seluruh data digunakan pada tiap jaringan untuk pelatihan. Sedangkan pada parsialisasi 90%, terdapat 10% data yang tidak digunakan pada tiap jaringan. Dengan kata lain, jaringan yang satu dengan yang lainnya bersifat *mutually exclusive* yang memiliki elemen atau anggota yang sama dan tidak sama. Oleh karena itu, perbedaan antarjaringan

pada parsialisasi 90% lebih besar dibandingkan 100%, yang menyebabkan diversitas yang lebih baik.

Gambar 5.4 menunjukkan waktu yang dibutuhkan untuk melatih pola-pola, di luar waktu pra-pemrosesan dan waktu pengenalan. Tampak bahwa untuk setiap jumlah jaringan yang digunakan, semakin besar persentase parsialisasi data, semakin lama waktu yang dibutuhkan sistem untuk mempelajari pola-pola yang ada. Hal ini jelas terjadi karena semakin tinggi persentase, semakin banyak pola yang harus dipelajari, sehingga waktu pembelajaran juga akan lebih lama. Sedangkan untuk setiap persentase, semakin besar jumlah jaringan semakin lama waktu pembelajaran, sama seperti pada skema pertama (algoritma backpropagation).

Gambar 5.5 menunjukkan waktu yang dibutuhkan sistem untuk mengenali atau mengidentifikasi sebuah foto. Untuk setiap jumlah jaringan, besarnya persentase parsialisasi data tidak mempengaruhi waktu pengenalan. Fluktuasi yang terjadi disebabkan oleh pengaruh prosesor komputer yang digunakan, dan nilainya pun sangat kecil. Sementara itu, waktu pengenalan meningkat seiring bertambahnya jumlah jaringan yang digunakan. Namun sampai dengan lima jaringan, waktu pengenalan masih sekitar 0.02 detik sehingga tidak menjadi masalah untuk mengidentifikasi suatu gambar wajah.



**Gambar 5.7.** Grafik Error terhadap Epoch Skema 2 untuk Jumlah Jaringan = 3, Persentase 100%

Gambar 5.7 menunjukkan grafik error terhadap epoch, di mana jumlah jaringan yang digunakan adalah tiga, dengan persentase parsialisasi data 100%. Dari grafik tersebut dapat dilihat bahwa setiap jaringan memiliki grafik error sendiri dan berbeda-beda namun mencapai konvergensi.

### 5.3. Skema 3: Algoritma Negative Correlation Learning (NCL)

Negative Correlation Learning (NCL) adalah metode lain untuk menghasilkan *diversity of opinion*. Seperti yang telah dijelaskan pada bab II, algoritma NCL merupakan ekstensi dari backpropagation dengan ditambahkan faktor penalty term pada fungsi errornya. Akibatnya, perubahan bobot antarlapisan pada jaringan juga dipengaruhi penalty term ini. Penurunan rumus perubahan bobot ini terdapat di lampiran.

Perbedaan yang paling mencolok pada pelatihan dengan algoritma NCL adalah digunakannya keluaran tiap jaringan  $F_i(n)$  dan keluaran akhir yang merupakan rata-rata keluaran tiap jaringan  $F(n)$  dalam komputasi propagasi balik (backward) dalam proses pelatihan. Pada skema 1 dan 2 (algoritma backpropagation dan parsialisasi data), hanya keluaran tiap jaringan yang diperhitungkan dalam pelatihan dalam jaringan itu, tanpa mencari keluaran akhir yang merupakan gabungan semua keluaran jaringan.

Pada eksperimen, peneliti menggunakan nilai parameter penalty ( $\lambda$ ) sebesar 0, 0.2, 0.4, 0.6, 0.8, dan 1.0. Berikut ini adalah hasil eksperimen.

**Tabel 5.7.** Recognition Rate dan Recognition Time Skema 3 untuk 1 Jaringan

$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	100%	93.9%	47.8104	0.1016
<b>0.2</b>	100%	94.5%	48.3463	0.1012
<b>0.4</b>	100%	95.3%	48.4362	0.1010
<b>0.6</b>	100%	95.0%	49.0750	0.1009
<b>0.8</b>	100%	96.5%	48.9371	0.1009
<b>1.0</b>	100%	95.9%	48.9342	0.1004

**Tabel 5.8.** Recognition Rate dan Recognition Time Skema 3 untuk 2 Jaringan

$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	100%	97.6%	88.2273	0.1881
<b>0.2</b>	100%	97.1%	90.0988	0.1878
<b>0.4</b>	100%	97.4%	94.9043	0.1874
<b>0.6</b>	100%	98.4%	95.1199	0.1871
<b>0.8</b>	100%	98.7%	92.8513	0.1870
<b>1.0</b>	100%	98.5%	90.7994	0.1872

**Tabel 5.9.** Recognition Rate dan Recognition Time Skema 3 untuk 3 Jaringan

$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	100%	99.0%	141.3045	0.2746
<b>0.2</b>	100%	98.4%	141.2445	0.2748
<b>0.4</b>	100%	98.2%	144.2979	0.2740
<b>0.6</b>	100%	98.1%	143.8044	0.2731
<b>0.8</b>	100%	98.9%	147.2998	0.2731
<b>1.0</b>	100%	99.2%	143.4021	0.2730

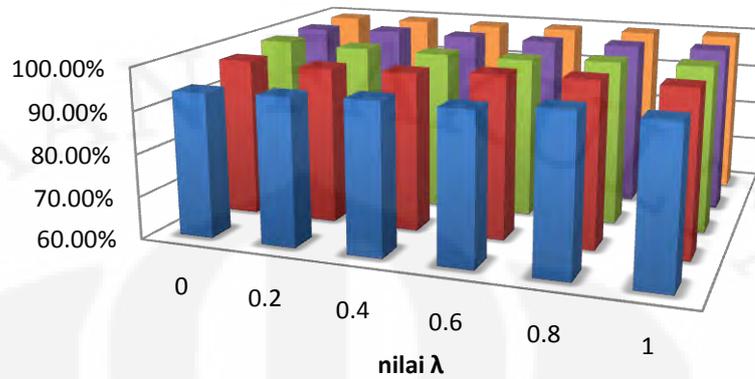
**Tabel 5.10.** Recognition Rate dan Recognition Time Skema 3 untuk 4 Jaringan

$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	100%	99.1%	187.7875	0.3670
<b>0.2</b>	100%	99.6%	185.2650	0.3663
<b>0.4</b>	100%	99.3%	187.9978	0.3653
<b>0.6</b>	100%	99.2%	184.8414	0.3660
<b>0.8</b>	100%	99.3%	178.8349	0.3673
<b>1.0</b>	100%	99.5%	186.6382	0.3672

**Tabel 5.11.** Recognition Rate dan Recognition Time Skema 3 untuk 5 Jaringan

$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	100%	99.3%	225.3650	0.4506
<b>0.2</b>	100%	99.4%	228.6918	0.4605
<b>0.4</b>	100%	99.1%	232.6813	0.4610
<b>0.6</b>	100%	99.3%	219.1794	0.4684
<b>0.8</b>	100%	99.5%	225.4366	0.4459
<b>1.0</b>	100%	99.6%	229.9585	0.4538

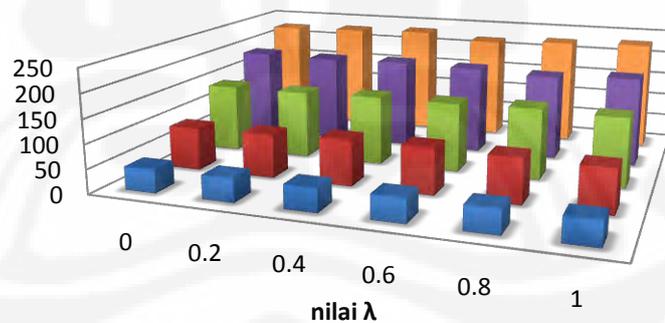
## Recognition Rate - Testing



	0	0.2	0.4	0.6	0.8	1
1 network	93.90%	94.50%	95.30%	95.00%	96.50%	95.90%
2 networks	97.60%	97.10%	97.40%	98.40%	98.70%	98.50%
3 networks	99.00%	98.40%	98.20%	98.10%	98.90%	99.20%
4 networks	99.10%	99.60%	99.30%	99.20%	99.30%	99.50%
5 networks	99.30%	99.40%	99.10%	99.30%	99.50%	99.60%

Gambar 5.8. Grafik Recognition Rate Testing untuk Skema 3

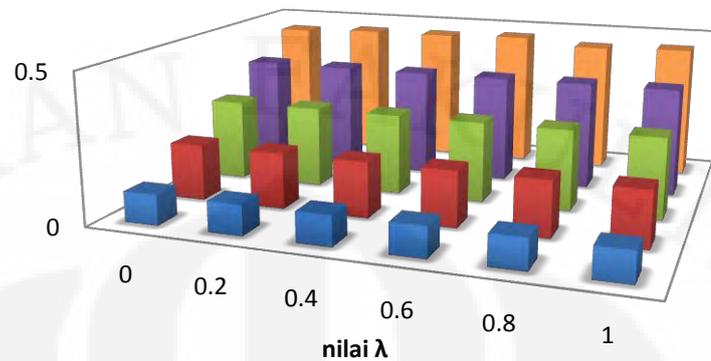
## Training Time (detik)



	0	0.2	0.4	0.6	0.8	1
1 network	47.8104	48.3463	48.4362	49.075	48.9371	48.9342
2 networks	88.2273	90.0988	94.9043	95.1199	92.8513	90.7994
3 networks	141.3045	141.2445	144.2979	143.8044	147.2998	143.4021
4 networks	187.7875	185.265	187.9978	184.8414	178.8349	186.6382
5 networks	225.365	228.6918	232.6813	219.1794	225.4366	229.9585

Gambar 5.9. Grafik Training Time untuk Skema 3

## Recognition Time (detik)



	0	0.2	0.4	0.6	0.8	1
1 network	0.1016	0.1012	0.101	0.1009	0.1009	0.1004
2 networks	0.1881	0.1878	0.1874	0.1871	0.187	0.1872
3 networks	0.2746	0.2748	0.274	0.2731	0.2731	0.273
4 networks	0.367	0.3663	0.3653	0.366	0.3673	0.3672
5 networks	0.4506	0.4605	0.461	0.4684	0.4459	0.4538

**Gambar 5.10.** Grafik Recognition Time untuk Skema 3

Dari tabel 5.7 sampai dengan 5.11, dapat dilihat bahwa recognition rate untuk data dari dalam database mencapai 100% untuk segala nilai  $\lambda$  dan jumlah jaringan yang digunakan. Sedangkan dari gambar 5.6 dapat dilihat secara tiga dimensi pengaruh dari parameter  $\lambda$  dan jumlah jaringan pada recognition rate untuk data di luar database. Ternyata besarnya nilai  $\lambda$  tidak menunjukkan keterkaitan dengan semakin tinggi atau semakin rendahnya recognition rate. Hanya saja pada nilai  $\lambda$  yang relatif besar (0.8 dan 1.0) recognition rate tampak sedikit lebih baik untuk segala jumlah jaringan yang digunakan.

Hal ini dapat dijelaskan sebagai berikut. Nilai  $\lambda$  hanyalah sebuah parameter pada penalty term. Semakin besar nilai  $\lambda$ , maka penalty term akan bertambah besar, dan hal ini akan mengakibatkan diversitas pada tiap-tiap jaringan yang melakukan pelatihan. Selama proses pelatihan, setiap jaringan berinteraksi satu sama lain melalui penalty term tersebut. Masing-masing jaringan tidak hanya meminimalisir perbedaan antara keluarannya  $F_i(n)$  dengan target  $d(n)$ , tetapi juga keluaran bersama  $F(n)$  dengan target  $d(n)$ .

Namun setelah proses pelatihan selesai, bobot-bobot antarlapisan pada setiap jaringan akan termodifikasi dan menjadi sesuai untuk melakukan proses pengujian gambar.

Seperti yang telah dijelaskan pada bab II, turunan parsial fungsi error terhadap keluaran jaringan ke- $i$  dapat dinyatakan sebagai berikut:

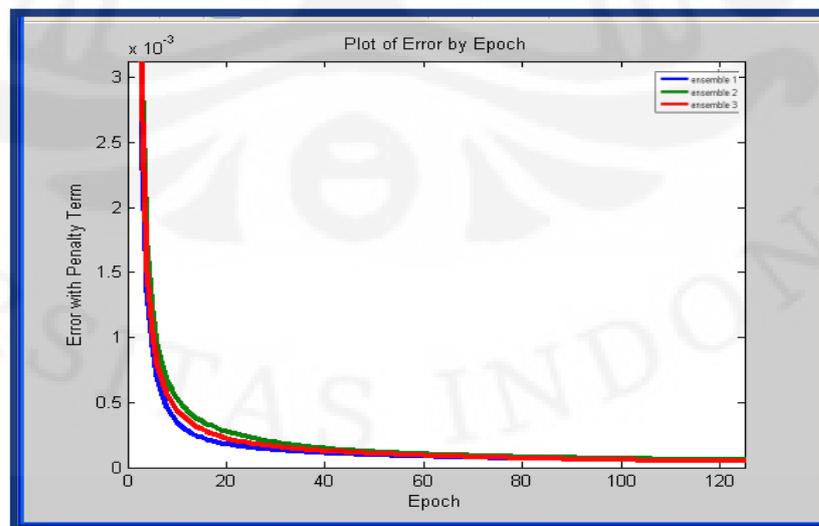
$$\frac{\partial E_i(n)}{\partial F_i(n)} = F_i(n) - d(n) + \lambda \frac{\partial p_i(n)}{\partial F_i(n)} \quad (2-24a)$$

$$= \lambda (F_i(n) - d(n)) + \lambda (F(n) - d(n)) \quad (2-24b)$$

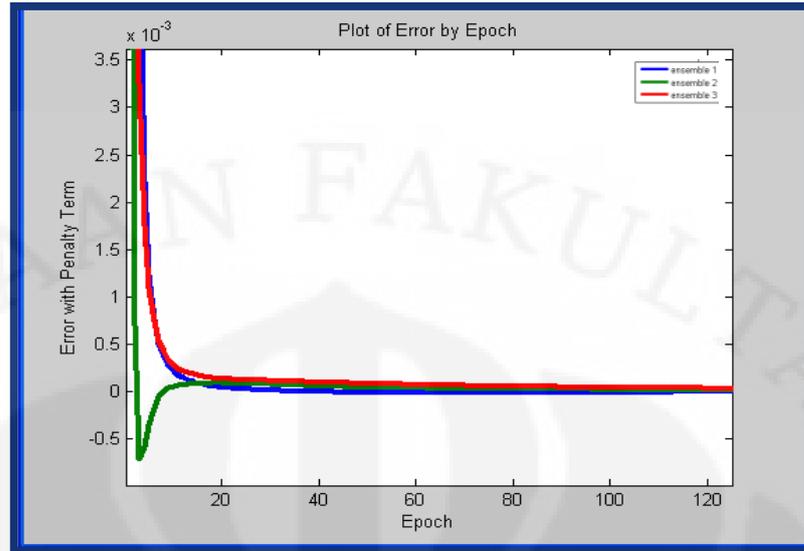
Ketika  $\lambda = 0$ , maka tidak ada penalty term sehingga dapat dikatakan pelatihan berlangsung secara independen. Hal ini tidak ada bedanya dengan skema algoritma backpropagation. Ketika  $\lambda = 1$ , fungsi error hanya memiliki komponen  $F(n) - d(n)$ .

Sama halnya dengan kedua skema sebelumnya, semakin banyak jumlah jaringan yang digunakan, semakin baik (tinggi) recognition rate. Kenaikan recognition rate ini terlihat paling signifikan pada saat 1 jaringan ke 2 jaringan. Selanjutnya, kenaikan tetap terjadi tetapi tidak signifikan.

Pada gambar 5.7 dan 5.8, dapat dilihat bahwa waktu pembelajaran dan waktu pelatihan tidak dipengaruhi oleh besarnya nilai  $\lambda$ , tetapi oleh banyaknya jaringan yang digunakan. Sama seperti skema-skema sebelumnya, semakin banyak jumlah jaringan yang digunakan, waktu pembelajaran dan waktu pengenalan gambar akan menjadi lebih lama.



5-11 a.



5-11 b.

**Gambar 5.11.** Grafik Error terhadap Epoch Skema 3 untuk Jumlah Jaringan = 3  
(a)  $\lambda = 0.2$  (b)  $\lambda = 0.8$

Gambar 5.11 menunjukkan grafik error terhadap epoch dengan mengambil contoh jumlah jaringan sebanyak tiga. Setiap jaringan memiliki grafik error yang berbeda-beda (ditandai dengan warna yang berbeda) dan tampak semua grafik mencapai konvergensi.

Namun bila diperhatikan, ketika nilai  $\lambda$  besar (0.8), salah satu grafik ada yang mencapai nilai negatif pada awal iterasi. Hal ini mungkin terjadi akibat nilai  $\lambda$ . Semakin besar nilai  $\lambda$ , semakin besar penalty term pada fungsi error, sehingga pada awal iterasi nilai fungsi error yang melewati batas nol (negatif) mungkin terjadi. Akan tetapi, seiring bertambahnya iterasi, penalty term akan semakin kecil dan error akan kembali menuju konvergensi.

#### 5.4. Skema 4: Parsialisasi Data dengan Algoritma NCL

Skema ini menggabungkan skema parsialisasi data dan skema algoritma NCL. Database yang berisi 200 data gambar wajah hanya diambil sebagian secara acak dan tertentu melalui persentase. Sama seperti pada skema parsialisasi data, persentase yang diuji adalah 50%, 60%, 70%, 80%, 90%, dan 100%. Pelatihan pola menggunakan algoritma NCL yang memperhatikan keluaran setiap jaringan. Sama halnya dengan skema NCL, nilai kekuatan penalty  $\lambda$  yang digunakan adalah 0, 0.2, 0.4, 0.6, 0.8, dan 1.0. Untuk setiap

nilai persentase dan  $\lambda$ , peneliti melakukan eksperimen pada jaringan berjumlah 1, 2, 3, 4, dan 5. Untuk setiap nilai  $\lambda$ , jumlah jaringan, dan besarnya persentase, dilakukan pengambilan data sebanyak sepuluh kali.

Karena banyaknya data yang diperoleh, peneliti mengelompokkannya berdasarkan jumlah jaringan dan persentase. Dengan demikian, hasil yang diperoleh hanya ditinjau dari besarnya nilai  $\lambda$ . Berikut ini hasil eksperimennya.

**Tabel 5.12.** Recognition Rate dan Recognition Time Skema 4 untuk 1 Jaringan, 50% Data

$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	99.3%	92.2%	24.4211	0.0994
<b>0.2</b>	98.7%	90.2%	24.3688	0.0994
<b>0.4</b>	98.0%	90.9%	24.2465	0.0996
<b>0.6</b>	98.1%	90.7%	24.6554	0.0998
<b>0.8</b>	98.6%	90.3%	24.2745	0.0999
<b>1.0</b>	98.5%	91.5%	24.3112	0.0994

**Tabel 5.13.** Recognition Rate dan Recognition Time Skema 4 untuk 1 Jaringan, 60% Data

$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	99.5%	94.0%	28.8550	0.0982
<b>0.2</b>	99.5%	92.7%	29.1219	0.0976
<b>0.4</b>	99.4%	92.6%	28.9157	0.0979
<b>0.6</b>	99.0%	92.3%	28.8623	0.0985
<b>0.8</b>	99.4%	94.0%	28.9053	0.0984
<b>1.0</b>	99.6%	91.7%	29.1020	0.0989

**Tabel 5.14.** Recognition Rate dan Recognition Time Skema 4 untuk 1 Jaringan, 70% Data

$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	99.4%	94.1%	34.4745	0.0993
<b>0.2</b>	99.7%	93.6%	34.0784	0.0992
<b>0.4</b>	99.6%	92.4%	34.4128	0.0996
<b>0.6</b>	99.5%	92.1%	34.3122	0.0999
<b>0.8</b>	99.7%	93.3%	34.6713	0.0995
<b>1.0</b>	99.6%	93.4%	34.5341	0.0994

**Tabel 5.15.** Recognition Rate dan Recognition Time Skema 4 untuk 1 Jaringan, 80% Data

$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	99.8%	95.7%	39.0100	0.1001
<b>0.2</b>	99.8%	95.7%	38.3588	0.0996
<b>0.4</b>	99.7%	95.7%	38.6132	0.0995
<b>0.6</b>	99.8%	94.6%	38.3352	0.0995
<b>0.8</b>	99.6%	94.6%	38.3556	0.0993
<b>1.0</b>	99.8%	94.5%	38.9483	0.0995

**Tabel 5.16.** Recognition Rate dan Recognition Time Skema 4 untuk 1 Jaringan, 90% Data

$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	100%	96.0%	42.4136	0.0990
<b>0.2</b>	99.9%	93.9%	42.1582	0.0986
<b>0.4</b>	99.9%	94.2%	42.2111	0.0984
<b>0.6</b>	99.7%	94.3%	42.2299	0.0988
<b>0.8</b>	100%	95.0%	42.1878	0.0987
<b>1.0</b>	99.8%	94.2%	42.0485	0.0988

**Tabel 5.17.** Recognition Rate dan Recognition Time Skema 4 untuk 1 Jaringan, 100% Data

$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	100%	96.8%	48.0254	0.0980
<b>0.2</b>	100%	94.7%	47.8557	0.0972
<b>0.4</b>	100%	94.4%	47.9627	0.0974
<b>0.6</b>	100%	95.0%	47.6273	0.0975
<b>0.8</b>	100%	94.7%	47.8055	0.0974
<b>1.0</b>	100%	94.9%	47.4210	0.0973

**Tabel 5.18.** Recognition Rate dan Recognition Time Skema 4 untuk 2 Jaringan, 50% Data

$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	99.7%	95.2%	46.4360	0.1878
<b>0.2</b>	99.7%	93.8%	45.9675	0.1871
<b>0.4</b>	99.5%	94.8%	45.7889	0.1873
<b>0.6</b>	99.9%	94.2%	45.5914	0.1874
<b>0.8</b>	99.8%	95.9%	45.6236	0.1873
<b>1.0</b>	99.8%	97.4%	45.5102	0.1875

**Tabel 5.19.** Recognition Rate dan Recognition Time Skema 4 untuk 2 Jaringan, 60% Data

$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	100%	94.7%	55.4762	0.1874
<b>0.2</b>	99.9%	95.4%	55.7344	0.1871
<b>0.4</b>	99.9%	95.5%	55.6838	0.1872
<b>0.6</b>	99.9%	95.4%	55.2757	0.1868
<b>0.8</b>	99.7%	96.0%	55.5249	0.1868
<b>1.0</b>	99.8%	98.0%	55.8625	0.1869

**Tabel 5.20.** Recognition Rate dan Recognition Time Skema 4 untuk 2 Jaringan, 70% Data

$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	99.9%	96.2%	64.5214	0.1877
<b>0.2</b>	99.9%	95.8%	64.5526	0.1874
<b>0.4</b>	100%	95.2%	64.2187	0.1874
<b>0.6</b>	100%	96.2%	64.9198	0.1875
<b>0.8</b>	99.9%	96.3%	64.0305	0.1873
<b>1.0</b>	100%	98.3%	63.9956	0.1873

**Tabel 5.21.** Recognition Rate dan Recognition Time Skema 4 untuk 2 Jaringan, 80% Data

$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	100%	96.7%	74.4339	0.1875
<b>0.2</b>	100%	96.8%	74.6913	0.1877
<b>0.4</b>	100%	97.5%	74.4287	0.1875
<b>0.6</b>	99.9%	97.1%	74.4140	0.1874
<b>0.8</b>	100%	96.5%	74.4222	0.1875
<b>1.0</b>	100%	98.3%	74.2936	0.1873

**Tabel 5.22.** Recognition Rate dan Recognition Time Skema 4 untuk 2 Jaringan, 90% Data

$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	100%	96.8%	83.8668	0.1908
<b>0.2</b>	100%	97.0%	84.3467	0.1897
<b>0.4</b>	100%	97.1%	83.1990	0.1895
<b>0.6</b>	100%	97.4%	83.2643	0.1896
<b>0.8</b>	100%	97.3%	82.9631	0.1896
<b>1.0</b>	100%	99.0%	82.6301	0.1897

**Tabel 5.23.** Recognition Rate dan Recognition Time Skema 4 untuk 2 Jaringan, 100% Data

$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	100%	97.5%	91.8324	0.1875
<b>0.2</b>	100%	97.1%	91.7381	0.1876
<b>0.4</b>	100%	97.1%	91.3451	0.1874
<b>0.6</b>	100%	97.5%	91.3471	0.1870
<b>0.8</b>	100%	97.4%	91.5652	0.1869
<b>1.0</b>	100%	99.0%	91.0862	0.1865

**Tabel 5.24.** Recognition Rate dan Recognition Time Skema 4 untuk 3 Jaringan, 50% Data

$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	100%	97.3%	66.2592	0.2678
<b>0.2</b>	100%	97.1%	66.6959	0.2687
<b>0.4</b>	99.8%	96.6%	65.8624	0.2668
<b>0.6</b>	100%	96.2%	65.9304	0.2678
<b>0.8</b>	99.9%	96.3%	65.9927	0.2689
<b>1.0</b>	99.8%	97.2%	66.1134	0.2687

**Tabel 5.25.** Recognition Rate dan Recognition Time Skema 4 untuk 3 Jaringan, 60% Data

$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	100%	98.1%	78.8603	0.2696
<b>0.2</b>	99.9%	97.3%	78.8731	0.2694
<b>0.4</b>	99.9%	97.9%	78.5850	0.2686
<b>0.6</b>	100%	96.3%	78.8981	0.2688
<b>0.8</b>	99.9%	95.5%	79.1171	0.2690
<b>1.0</b>	99.9%	97.4%	79.1607	0.2688

**Tabel 5.26.** Recognition Rate dan Recognition Time Skema 4 untuk 3 Jaringan, 70% Data

$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	100%	98.1%	93.0337	0.2693
<b>0.2</b>	100%	97.2%	93.0700	0.2687
<b>0.4</b>	100%	97.5%	93.2521	0.2692
<b>0.6</b>	100%	97.9%	93.1761	0.2694
<b>0.8</b>	100%	97.5%	93.1275	0.2689
<b>1.0</b>	100%	97.5%	93.1432	0.2688

**Tabel 5.27.** Recognition Rate dan Recognition Time Skema 4 untuk 3 Jaringan, 80% Data

$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	100%	97.9%	105.5493	0.2693
<b>0.2</b>	100%	98.2%	105.6164	0.2690
<b>0.4</b>	100%	98.8%	105.6555	0.2688
<b>0.6</b>	100%	97.8%	106.0980	0.2690
<b>0.8</b>	100%	98.3%	105.1423	0.2691
<b>1.0</b>	99.9%	98.3%	105.4538	0.2693

**Tabel 5.28.** Recognition Rate dan Recognition Time Skema 4 untuk 3 Jaringan, 90% Data

$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	100%	98.2%	120.9474	0.2684
<b>0.2</b>	100%	98.2%	120.8875	0.2688
<b>0.4</b>	100%	98.6%	120.3568	0.2685
<b>0.6</b>	100%	98.8%	119.7980	0.2685
<b>0.8</b>	100%	97.6%	120.2342	0.2690
<b>1.0</b>	100%	98.9%	120.2033	0.2685

**Tabel 5.29.** Recognition Rate dan Recognition Time Skema 4 untuk 3 Jaringan, 100% Data

$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	100%	98.6%	135.0317	0.2700
<b>0.2</b>	100%	98.5%	135.0747	0.2698
<b>0.4</b>	100%	99.0%	134.9241	0.2695
<b>0.6</b>	100%	98.5%	134.4382	0.2699
<b>0.8</b>	100%	98.5%	134.9581	0.2695
<b>1.0</b>	100%	98.8%	134.4096	0.2697

**Tabel 5.30.** Recognition Rate dan Recognition Time Skema 4 untuk 4 Jaringan, 50% Data

$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	100%	98.8%	90.0442	0.3598
<b>0.2</b>	99.9%	97.8%	89.3784	0.3594
<b>0.4</b>	100%	98.5%	89.2298	0.3590
<b>0.6</b>	100%	96.8%	89.4355	0.3594
<b>0.8</b>	100%	96.8%	88.9655	0.3592
<b>1.0</b>	100%	97.8%	88.8550	0.3593

**Tabel 5.31.** Recognition Rate dan Recognition Time Skema 4 untuk 4 Jaringan, 60% Data

$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	100%	97.4%	107.4297	0.3599
<b>0.2</b>	100%	97.5%	107.3279	0.3595
<b>0.4</b>	100%	98.0%	107.4928	0.3590
<b>0.6</b>	99.9%	97.1%	107.5220	0.3592
<b>0.8</b>	100%	98.0%	107.9283	0.3590
<b>1.0</b>	99.9%	97.8%	107.5307	0.3592

**Tabel 5.32.** Recognition Rate dan Recognition Time Skema 4 untuk 4 Jaringan, 70% Data

$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	100%	98.0%	127.0677	0.3593
<b>0.2</b>	100%	97.7%	126.7302	0.3596
<b>0.4</b>	100%	98.0%	126.1415	0.3593
<b>0.6</b>	100%	98.1%	126.1987	0.3595
<b>0.8</b>	100%	97.7%	126.4738	0.3598
<b>1.0</b>	100%	98.3%	126.2307	0.3596

**Tabel 5.33.** Recognition Rate dan Recognition Time Skema 4 untuk 4 Jaringan, 80% Data

$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	100%	98.8%	144.4937	0.3598
<b>0.2</b>	100%	99.2%	144.2866	0.3595
<b>0.4</b>	100%	98.4%	144.2471	0.3589
<b>0.6</b>	100%	98.8%	144.6342	0.3591
<b>0.8</b>	100%	98.7%	144.8818	0.3590
<b>1.0</b>	100%	98.0%	144.2798	0.3590

**Tabel 5.34.** Recognition Rate dan Recognition Time Skema 4 untuk 4 Jaringan, 90% Data

$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	100%	99.0%	161.3399	0.3592
<b>0.2</b>	100%	98.9%	161.2546	0.3588
<b>0.4</b>	100%	98.9%	161.6145	0.3590
<b>0.6</b>	100%	98.4%	161.4702	0.3590
<b>0.8</b>	100%	98.5%	161.7136	0.3592
<b>1.0</b>	100%	99.1%	161.4090	0.3589

**Tabel 5.35.** Recognition Rate dan Recognition Time Skema 4 untuk 4 Jaringan, 100% Data

$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	100%	99.2%	181.1282	0.3599
<b>0.2</b>	100%	99.0%	181.1328	0.3598
<b>0.4</b>	100%	99.2%	181.6679	0.3601
<b>0.6</b>	100%	99.4%	181.4823	0.3598
<b>0.8</b>	100%	98.9%	181.7716	0.3599
<b>1.0</b>	100%	98.6%	181.8689	0.3598

**Tabel 5.36.** Recognition Rate dan Recognition Time Skema 4 untuk 5 Jaringan, 50% Data

$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	100%	97.6%	130.8071	0.4498
<b>0.2</b>	100%	98.5%	130.5575	0.4497
<b>0.4</b>	100%	98.5%	130.6163	0.4498
<b>0.6</b>	100%	98.2%	130.6116	0.4501
<b>0.8</b>	99.9%	97.4%	130.6811	0.4499
<b>1.0</b>	99.8%	97.3%	130.4138	0.4498

**Tabel 5.37.** Recognition Rate dan Recognition Time Skema 4 untuk 5 Jaringan, 60% Data

$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	100%	98.0%	134.3341	0.4502
<b>0.2</b>	100%	98.9%	135.0074	0.4499
<b>0.4</b>	100%	98.3%	134.7742	0.4496
<b>0.6</b>	100%	98.5%	134.8528	0.4499
<b>0.8</b>	100%	97.9%	134.7198	0.4498
<b>1.0</b>	99.9%	98.1%	134.6350	0.4999

**Tabel 5.38.** Recognition Rate dan Recognition Time Skema 4 untuk 5 Jaringan, 70% Data

$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	100%	98.8%	161.4367	0.4506
<b>0.2</b>	100%	99.4%	160.7189	0.4502
<b>0.4</b>	100%	99.1%	161.6501	0.4503
<b>0.6</b>	100%	98.4%	161.5434	0.4499
<b>0.8</b>	100%	98.7%	161.0491	0.4498
<b>1.0</b>	100%	98.0%	160.9873	0.4499

**Tabel 5.39.** Recognition Rate dan Recognition Time Skema 4 untuk 5 Jaringan, 80% Data

$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	100%	99.2%	184.4458	0.4496
<b>0.2</b>	100%	99.1%	183.8443	0.4495
<b>0.4</b>	100%	99.0%	184.2296	0.4502
<b>0.6</b>	100%	98.5%	184.4776	0.4499
<b>0.8</b>	100%	99.0%	183.8998	0.4501
<b>1.0</b>	100%	98.9%	183.6091	0.4502

**Tabel 5.40.** Recognition Rate dan Recognition Time Skema 4 untuk 5 Jaringan, 90% Data

$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	100%	99.3%	200.5539	0.4503
<b>0.2</b>	100%	98.9%	199.9162	0.4501
<b>0.4</b>	100%	98.6%	200.8747	0.4496
<b>0.6</b>	100%	98.7%	199.8001	0.4496
<b>0.8</b>	100%	98.9%	199.8804	0.4501
<b>1.0</b>	100%	98.7%	200.0535	0.4499

**Tabel 5.41.** Recognition Rate dan Recognition Time Skema 4 untuk 5 Jaringan, 100% Data

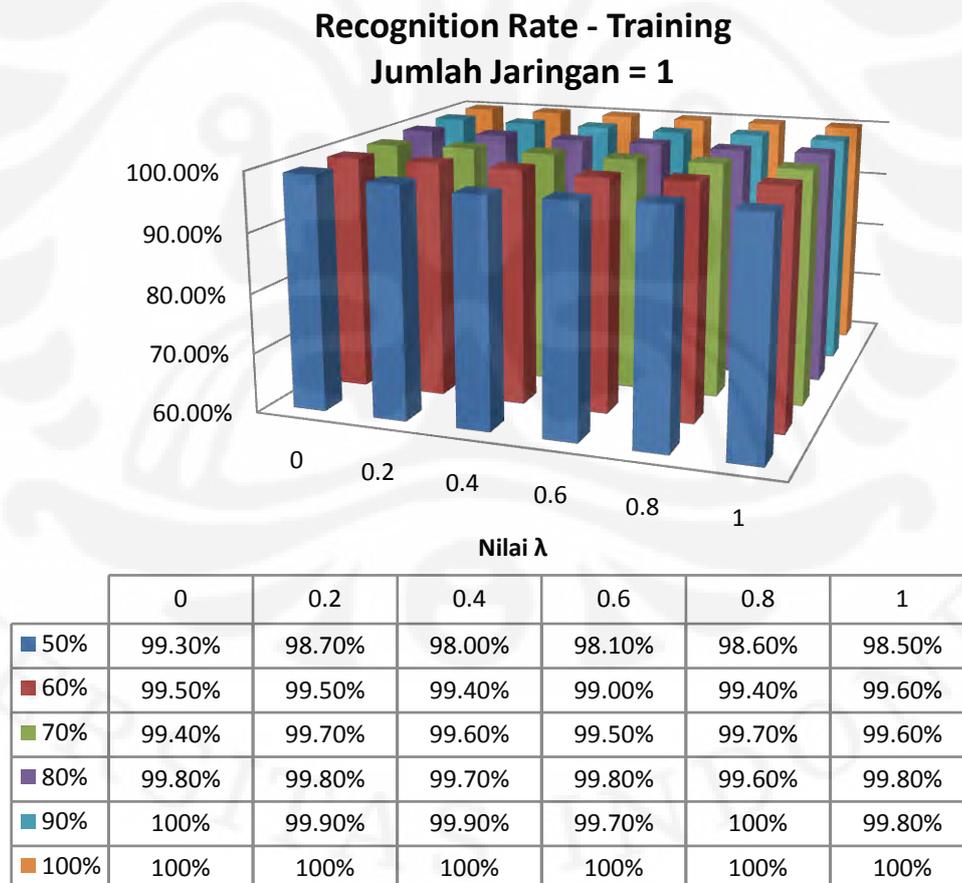
$\lambda$	Recognition Rate		Waktu (detik)	
	Training	Testing	Training	Testing (1 gambar)
<b>0.0</b>	100%	99.1%	219.2931	0.4489
<b>0.2</b>	100%	99.3%	219.7223	0.4490
<b>0.4</b>	100%	99.6%	220.2620	0.4486
<b>0.6</b>	100%	99.0%	219.8574	0.4493
<b>0.8</b>	100%	99.3%	219.5824	0.4489
<b>1.0</b>	100%	99.3%	220.0212	0.4493

Data yang diperoleh menunjukkan sama halnya pada skema-skema sebelumnya, semakin banyak jumlah jaringan yang digunakan, semakin besar recognition rate dan waktu pelatihan. Untuk mempermudah analisis, maka dibuat grafik tiga dimensi yang menunjukkan recognition rate untuk setiap jumlah jaringan, di mana variabel yang menjadi patokan adalah persentase parsialisasi data dan nilai  $\lambda$ . Pada jumlah jaringan tertentu dan setiap nilai  $\lambda$

apapun, waktu pelatihan bertambah seiring besarnya persentase, tetapi waktu pelatihannya sama.

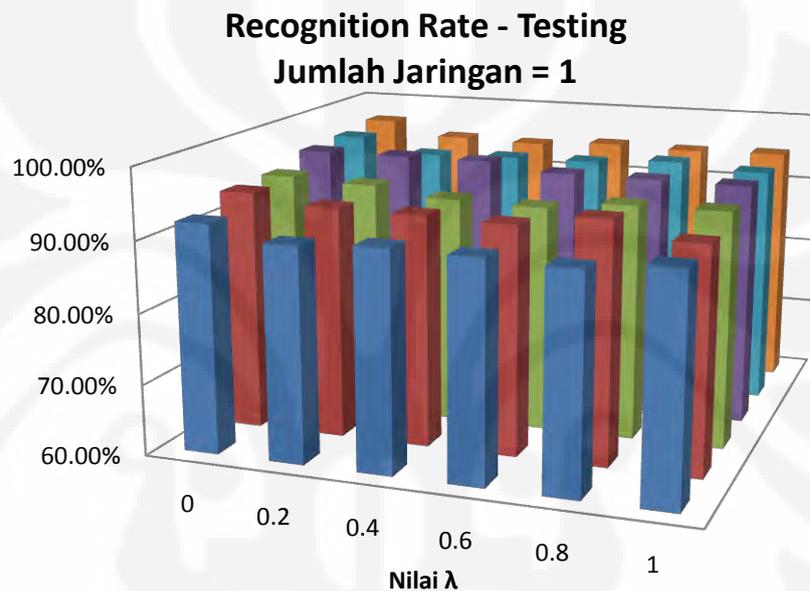
Jika jumlah jaringan semakin banyak atau persentase parsialisasi semakin besar, waktu yang dibutuhkan jaringan untuk melakukan pelatihan semakin lama. Sementara itu, semakin banyak jumlah jaringan yang digunakan, semakin lama juga waktu pengenalan gambar. Namun besarnya persentase parsialisasi data tidak memengaruhi waktu pengenalan gambar. Selama jumlah jaringan sama, berapapun persentasenya, waktu pengenalan gambar adalah sama lamanya. Karena banyaknya data pada skema ini, maka grafik waktu pelatihan dan pengenalan tidak ditampilkan.

Berikut ini adalah grafik tiga dimensi yang menampilkan hubungan recognition rate terhadap persentase parsialisasi data dan nilai  $\lambda$  yang digunakan.



**Gambar 5.12.** Grafik Recognition Rate (Training) Skema 4 untuk Jumlah Jaringan = 1

Pada satu jaringan, recognition rate untuk gambar uji dari dalam database tidak mencapai 100% pada persentase parsialisasi yang relatif kecil. Hal ini disebabkan pada persentase yang kecil, jaringan tidak mempelajari keseluruhan pola. Semakin besar persentase, semakin banyak pola yang dipelajari sehingga semakin besar recognition rate. Bahkan untuk persentase 90%, recognition rate telah dapat mencapai 100%.



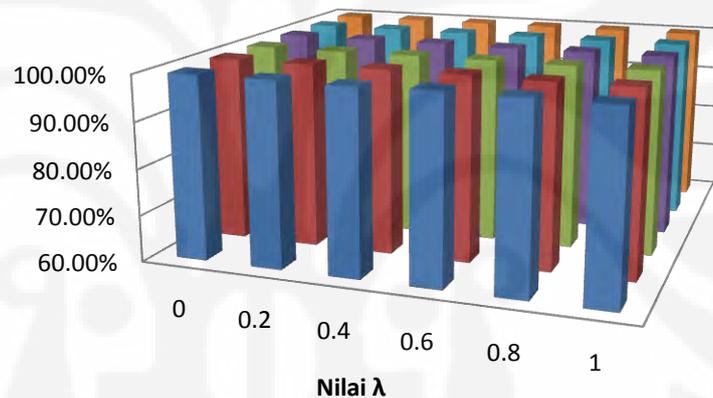
	0	0.2	0.4	0.6	0.8	1
50%	92.20%	90.20%	90.90%	90.70%	90.30%	91.50%
60%	94.00%	92.70%	92.60%	92.30%	94.00%	91.70%
70%	94.10%	93.60%	92.40%	92.10%	93.30%	93.40%
80%	95.70%	95.70%	95.70%	94.60%	94.60%	94.50%
90%	96.00%	93.90%	94.20%	94.30%	95.00%	94.20%
100%	96.80%	94.70%	94.40%	95.00%	94.70%	94.90%

**Gambar 5.13.** Grafik Recognition Rate (Testing) Skema 4 untuk Jumlah Jaringan = 1

Gambar 5.13 menunjukkan recognition rate untuk gambar uji di luar database. Sama halnya dengan gambar 5.12, semakin besar persentase parsialisasi data, semakin tinggi recognition rate. Namun gambar 5.12 dan 5.13 tidak menunjukkan relasi antara recognition rate dengan nilai  $\lambda$ , kecuali

pada  $\lambda = 0$  yang sedikit lebih baik. Selain itu recognition rate antara  $\lambda$  yang satu dengan yang lainnya tidak berbeda jauh. Hal ini disebabkan karena pada satu jaringan, keluaran jaringan individu ( $F_i(n)$ ) dan keluaran ensemble ( $F(n)$ ) sama. Dengan demikian, seberapa besar nilai  $\lambda$  seharusnya tidak akan mempengaruhi hasil pelatihan. Untuk lebih jelasnya, penurunan rumus perubahan bobot dapat dilihat pada lampiran. Hasil yang menunjukkan recognition rate lebih baik pada  $\lambda = 0$  pada gambar tersebut mungkin disebabkan oleh faktor kurangnya pengambilan data.

**Recognition Rate - Training**  
**Jumlah Jaringan = 2**

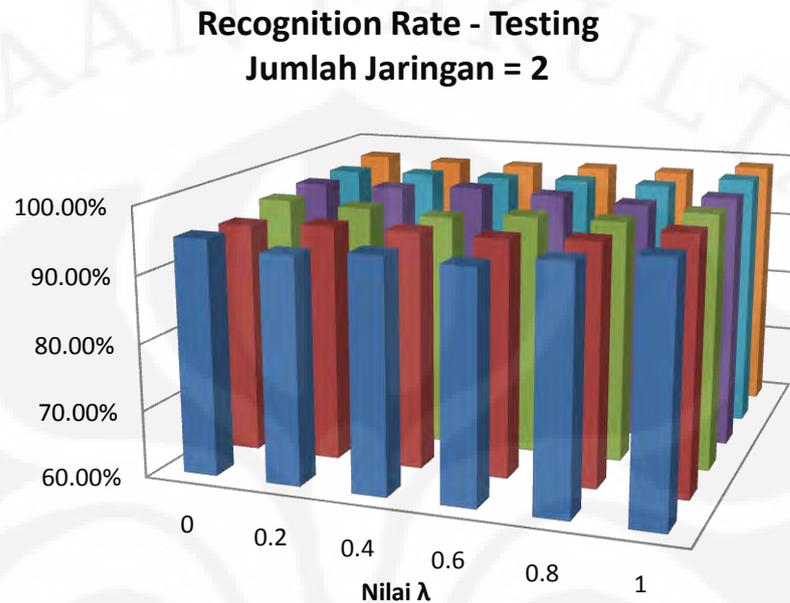


	0	0.2	0.4	0.6	0.8	1
50%	99.70%	99.70%	99.50%	99.90%	99.80%	99.80%
60%	100%	99.90%	99.90%	99.90%	99.70%	99.80%
70%	99.90%	99.90%	100%	100%	99.90%	100%
80%	100%	100%	100%	99.90%	100%	100%
90%	100%	100%	100%	100%	100%	100%
100%	100%	100%	100%	100%	100%	100%

**Gambar 5.14.** Grafik Recognition Rate (Training) Skema 4 untuk Jumlah Jaringan = 2

Gambar 5.14 menunjukkan grafik recognition rate untuk gambar uji dari dalam database. Sama seperti pada satu jaringan, semakin besar persentase parsialisasi data, semakin tinggi recognition rate. Namun untuk persentase yang kecil, recognition rate untuk dua jaringan mengalami peningkatan dibanding satu jaringan. Akan tetapi, pengaruh nilai  $\lambda$  tidak dapat

terlihat pada grafik tersebut, karena besarnya recognition rate tidak berbeda terlalu jauh terhadap nilai  $\lambda$  berapapun.



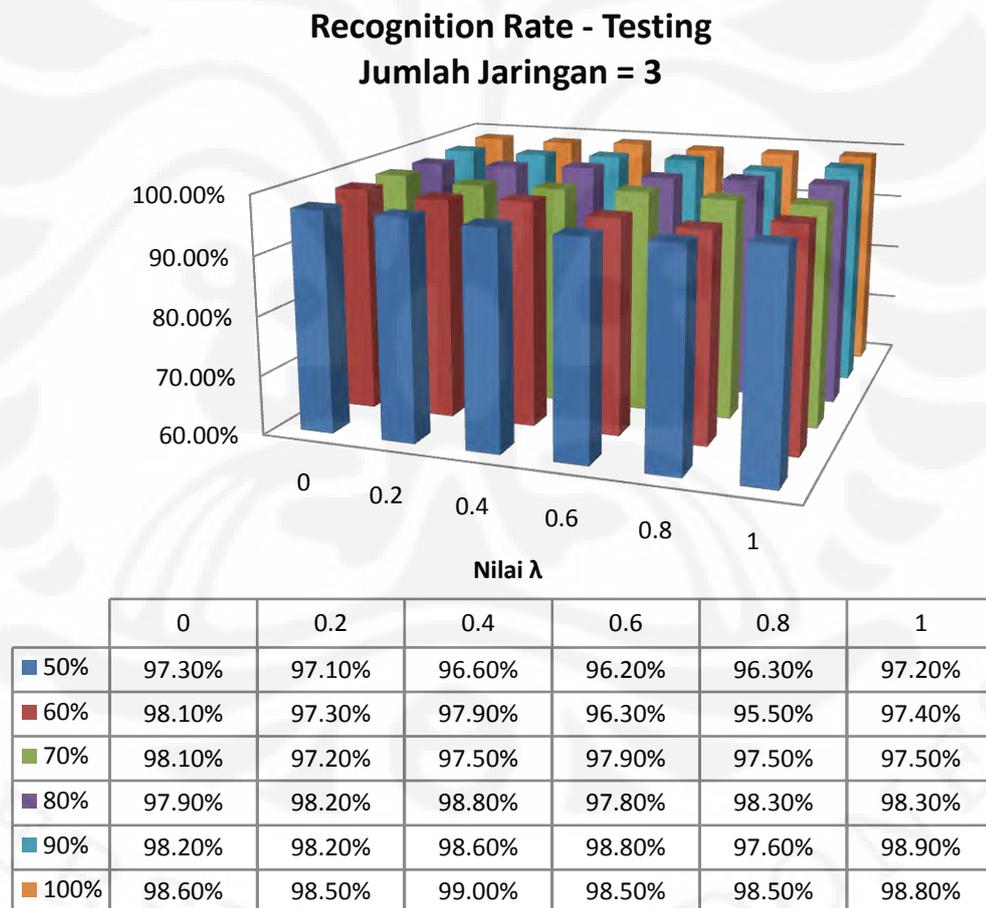
	0	0.2	0.4	0.6	0.8	1
■ 50%	95.20%	93.80%	94.80%	94.20%	95.90%	97.40%
■ 60%	94.70%	95.40%	95.50%	95.40%	96.00%	98.00%
■ 70%	96.20%	95.80%	95.20%	96.20%	96.30%	98.30%
■ 80%	96.70%	96.80%	97.50%	97.10%	96.50%	98.30%
■ 90%	96.80%	97.00%	97.10%	97.40%	97.30%	99.00%
■ 100%	97.50%	97.10%	97.10%	97.50%	97.40%	99.00%

**Gambar 5.15.** Grafik Recognition Rate (Testing) untuk Jumlah Jaringan = 2

Gambar 5.15 menunjukkan recognition rate untuk gambar uji di luar database. Ketika jumlah jaringan lebih dari satu, pelatihan dengan algoritma NCL akan memperbarui bobot dengan memperhatikan keluaran jaringan ke- $i$  ( $F_i(n)$ ) dan keluaran ensemble ( $F(n)$ ). Dari grafik tersebut, dapat dilihat bahwa recognition rate pada persentase berapapun menunjukkan hasil yang sedikit lebih baik pada nilai  $\lambda = 1.0$ , selain itu tampaknya tidak ada pengaruh nilai  $\lambda$  terhadap recognition rate.

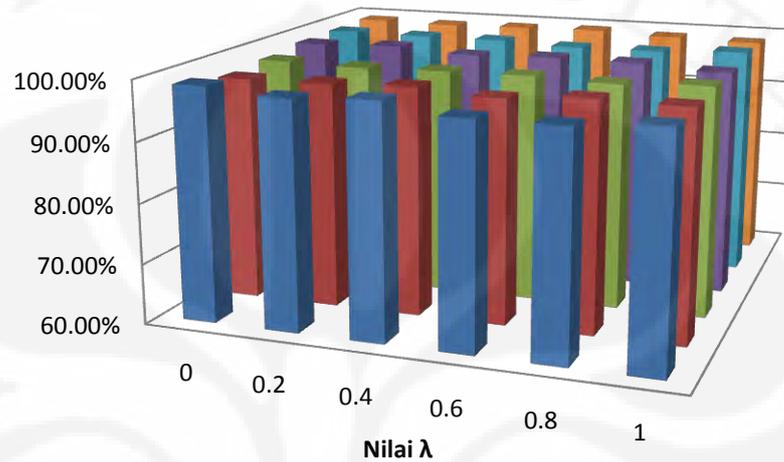
Ketika  $\lambda = 1.0$ , perubahan bobot pada tiap jaringan hanya dipengaruhi oleh faktor keluaran ensemble ( $F(n)$ ). Dengan demikian, besarnya perubahan bobot pada setiap jaringan adalah sama. Berbeda dengan nilai  $\lambda$  antara 0 dan 1, di mana perubahan bobot tergantung dari keluaran individual dan keluaran ensemble.

Pada jumlah jaringan sebanyak 3 dan 4, recognition rate untuk gambar uji dari dalam database (training) hampir mendekati 100% untuk persentase berapapun. Pada lima jaringan, recognition rate training telah mencapai 100% untuk semua persentase. Oleh karena itu, grafiknya tidak ditampilkan. Untuk gambar uji dari luar database (testing), berikut ini adalah hasilnya:



**Gambar 5.16.** Grafik Recognition Rate (Testing) Skema 4 untuk Jumlah Jaringan = 3

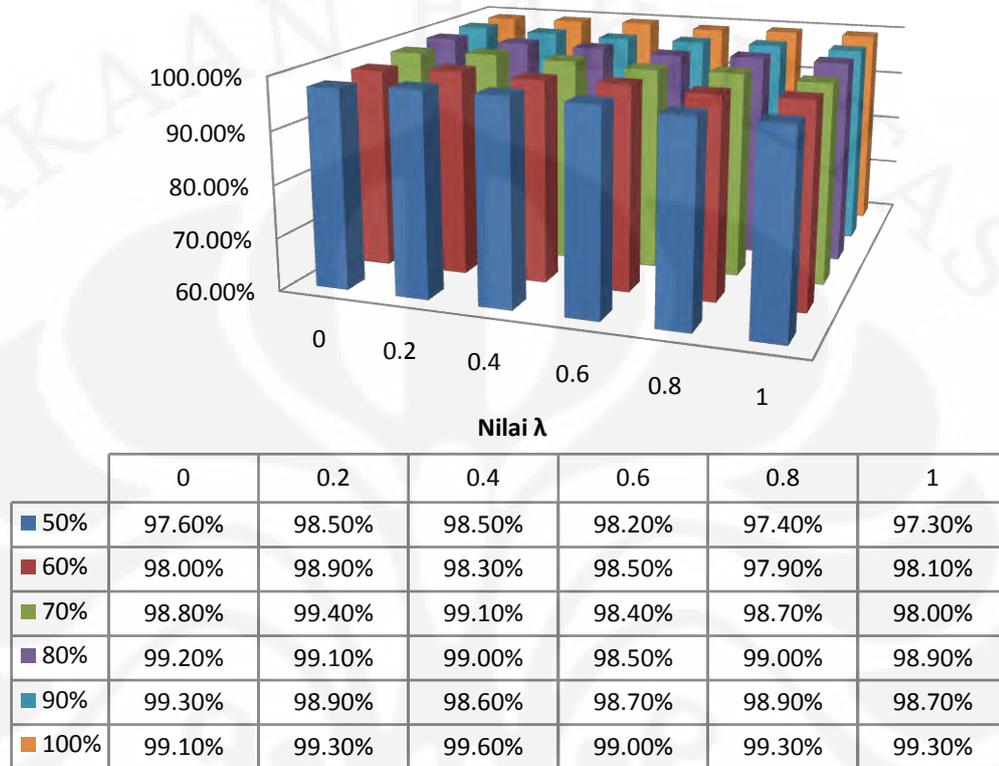
**Recognition Rate - Testing**  
**Jumlah Jaringan = 4**



	0	0.2	0.4	0.6	0.8	1
■ 50%	98.80%	97.80%	98.50%	96.80%	96.80%	97.80%
■ 60%	97.40%	97.50%	98.00%	97.10%	98.00%	97.80%
■ 70%	98.00%	97.70%	98.00%	98.10%	97.70%	98.30%
■ 80%	98.80%	99.20%	98.40%	98.80%	98.70%	98.00%
■ 90%	99.00%	98.90%	98.90%	98.40%	98.50%	99.10%
■ 100%	99.20%	99.00%	99.20%	99.40%	98.90%	98.60%

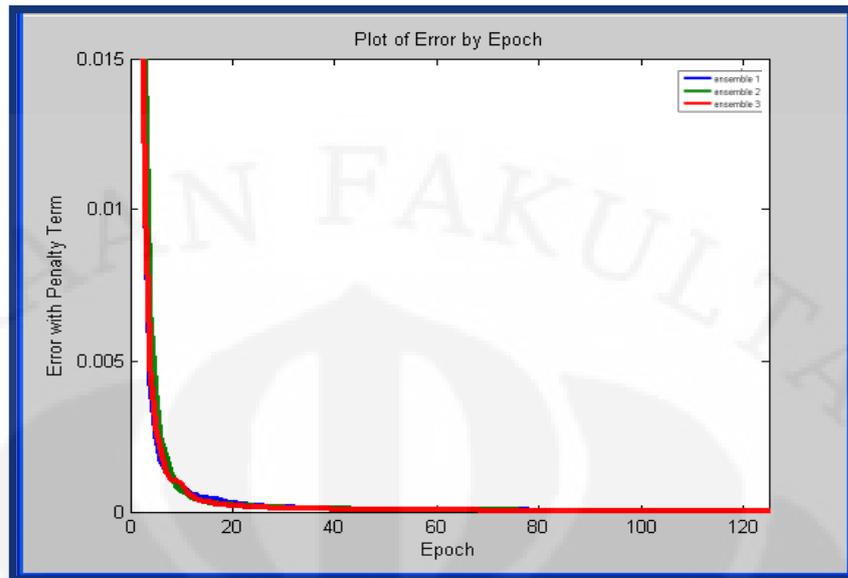
**Gambar 5.17.** Grafik Recognition Rate (Testing) Skema 4 untuk Jumlah Jaringan = 4

### Recognition Rate - Testing Jumlah Jaringan = 5

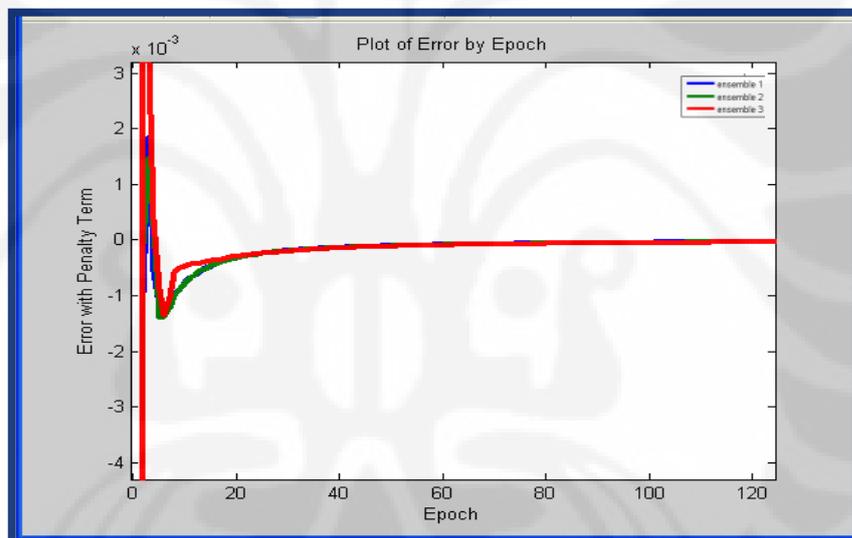


**Gambar 5.18.** Grafik Recognition Rate (Testing) Skema 4 untuk Jumlah Jaringan = 5

Untuk jaringan sebanyak 3, 4, dan 5, tidak terlihat adanya pengaruh nilai  $\lambda$  terhadap recognition rate. Untuk jumlah jaringan yang relatif sedikit, seperti dua jaringan, recognition rate tampak lebih baik pada  $\lambda = 1$ . Hal ini serupa dengan skema NCL di mana pada umumnya tidak ada keterkaitan antara tinggi-rendahnya recognition rate dengan nilai  $\lambda$ . Dengan demikian, dapat dikatakan skema parsialisasi data dengan algoritma NCL tidak juga menghasilkan performa yang lebih baik untuk penelitian ini.



5.19 a.



5.19 b.

**Gambar 5.19.** Grafik Error terhadap Epoch Skema 4 untuk Jumlah Jaringan = 3, Persentase 100% (a)  $\lambda = 0.2$  (b)  $\lambda = 0.8$

Gambar 5.19 menunjukkan grafik error terhadap epoch dengan mengambil contoh jumlah jaringan sebanyak tiga dan persentase parsialisasi 100%. Karna setiap jaringan memiliki fungsi error sendiri, maka grafik error setiap jaringan berbeda-beda (ditandai dengan warna yang berbeda). Pada grafik, tampak bahwa grafik error semua jaringan mencapai kovertensi.

Namun bila diperhatikan, ketika nilai  $\lambda$  besar (0.8), grafik dapat mencapai nilai negatif pada awal iterasi. Sama seperti pada skema 3 (NCL),

hal ini mungkin terjadi akibat nilai  $\lambda$ . Semakin besar nilai  $\lambda$ , semakin besar penalty term pada fungsi error, sehingga pada awal iterasi nilai fungsi error yang melewati batas nol (negatif) mungkin terjadi. Akan tetapi, seiring bertambahnya iterasi, penalty term akan semakin kecil dan error akan kembali menuju konvergensi. Dengan demikian, dari skema 3 dan 4, dapat disimpulkan bahwa nilai  $\lambda$  tidak berpengaruh pada recognition rate, tetapi berpengaruh pada grafik error yang dihasilkan.

### 5.5. Perbandingan Keempat Skema yang Digunakan

Setelah melakukan eksperimen dengan keempat skema, peneliti membuat perbandingan dengan memperhatikan recognition rate untuk gambar uji dari luar database (testing) dan recognition time dari keempat skema.

Pada skema 2 (parsialisasi data), persentase 90% dipilih untuk dibandingkan dengan skema 1 (algoritma backpropagation) karena menunjukkan hasil terbaik. Pada skema 1, setiap jaringan individu dilatih dengan urutan 200 gambar (pola) yang sama, sedangkan pada skema 2 – 90%, hanya 180 gambar (pola) yang digunakan untuk pelatihan dan diacak sedemikian hingga tiap jaringan individu dilatih dengan data dan urutan pola yang berbeda-beda.

Pada skema 3 (NCL), nilai  $\lambda = 0.8$  dipilih untuk menunjukkan pengaruh penalty term, meskipun besarnya nilai  $\lambda$  tidak mempengaruhi recognition rate. Perlu diketahui bahwa pada algoritma NCL, data yang digunakan untuk pelatihan tidak diacak sehingga setiap jaringan individu akan dilatih dengan urutan pola yang sama. Dengan demikian, jika nilai  $\lambda = 0$ , jaringan akan dilatih seperti halnya skema 1, yaitu backpropagation murni. Jika nilai  $\lambda = 1$ , maka hanya selisih target dengan keluaran ensemble yang akan berpengaruh terhadap perubahan bobot dalam pelatihan. Keluaran tiap jaringan individu tidak akan diperhitungkan.

Pada skema 4 (NCL dengan parsialisasi data), persentase 100% dipilih karena memberikan hasil yang paling baik, dan nilai  $\lambda = 0.8$  dipilih agar dapat dibandingkan dengan skema 3. Yang menjadi perbedaan antara skema 4 dan skema 3 di sini hanyalah urutan pola pelatihan yang diacak pada skema 4,

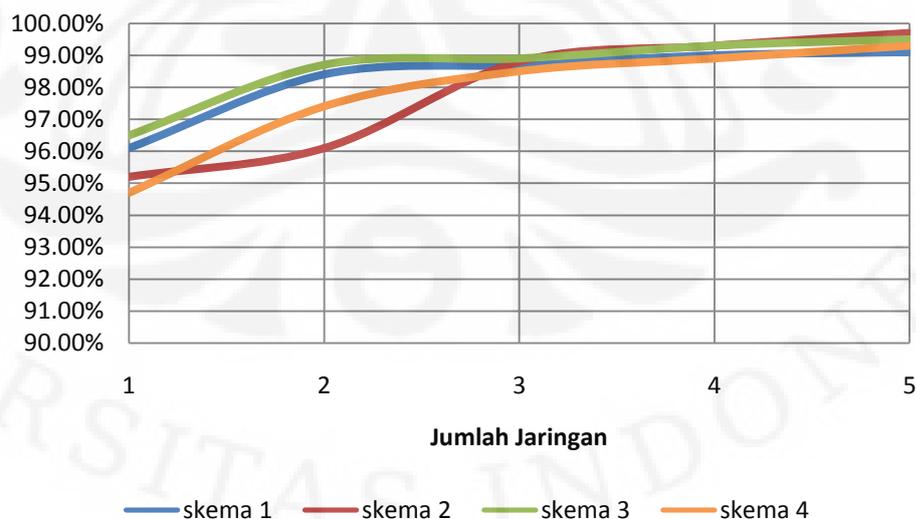
sedangkan pada skema 3 tidak diacak. Selain itu, yang membedakan skema 4 dan skema 2 di sini adalah besarnya persentase dan algoritma yang digunakan. Pada skema 2, persentase 90% dipilih dan algoritma backpropagation murni digunakan, sedangkan pada skema 4, persentase 100% dipilih dan algoritma NCL digunakan. Jika diperhatikan, nilai  $\lambda = 0$  pada skema 4 sama halnya melakukan kembali skema 2 karena penalty term tidak diperhitungkan dalam pelatihan, dan oleh karena itu dapat disebut backpropagation murni.

Berikut ini adalah tabel dan grafik recognition rate untuk gambar uji dari luar database (testing). Recognition rate untuk gambar uji dari dalam database tidak ditunjukkan karena semuanya mencapai 100%.

**Tabel 5.42.** Tabel Perbandingan Recognition Rate (Testing) untuk Keempat Skema

Jumlah Jaringan	Skema 1 Algoritma Backpropagation	Skema 2 Parsialisasi data (90%)	Skema 3 NCL ( $\lambda = 0.8$ )	Skema 4 NCL & Parsialisasi ( $\lambda = 0.8, 100\%$ )
1	96.1%	95.2%	96.5%	94.7%
2	98.4%	96.1%	98.7%	97.4%
3	98.7%	98.8%	98.9%	98.5%
4	99.0%	99.3%	99.3%	98.9%
5	99.1%	99.7%	99.5%	99.3%

### Recognition Rate (Testing)



**Gambar 5.20.** Grafik Recognition Rate (Testing) untuk Keempat Skema

Jika ditinjau dari recognition rate, semua skema menunjukkan adanya peningkatan recognition rate (testing) terhadap penambahan jumlah jaringan yang digunakan. Semakin banyak jaringan yang digunakan, semakin tinggi recognition rate. Namun peningkatan paling signifikan terlihat pada saat 1 ke 2 jaringan dan saat 2 ke 3 jaringan. Pada saat 3 ke 4 dan 4 ke 5 jaringan, peningkatan recognition rate masih terjadi, tetapi tampak semakin sedikit (tidak signifikan). Dari keempat skema yang dilakukan, hasil yang ditunjukkan tidak menunjukkan adanya perbedaan yang mencolok. Dengan demikian, tidak dapat disimpulkan adanya skema yang lebih baik performanya terhadap skema yang lain.

Seharusnya skema 2 (parsialisasi data) 100% dapat memiliki performa yang lebih baik dibandingkan skema 1 karena jaringan ensemble dilatih dengan pola yang urutannya berbeda pada tiap jaringannya. Hal ini berbeda dengan skema 1 di mana setiap jaringan mempelajari pola dengan urutan yang sama. Meskipun pada kedua skema setiap jaringan diberi bobot awal yang berbeda-beda, skema parsialisasi data seharusnya menghasilkan diversifikasi data yang lebih baik sehingga bobot akhir yang terbentuk dari pelatihan memiliki kemampuan yang lebih baik dalam mengelompokkan atau mengidentifikasi suatu gambar baru.

Skema 3 (NCL) yang memiliki penalty term seharusnya juga memiliki performa yang lebih baik dibanding skema algoritma backpropagation karena setiap jaringan tidak dilatih secara independen (terpisah), melainkan ada interaksi di antara jaringan. Pada proses pelatihan, keluaran setiap jaringan digabung terlebih dahulu dan kemudian diperhitungkan dalam modifikasi bobot pada langkah selanjutnya. Perhitungan ini dimasukkan ke dalam penalty term pada fungsi error. Jadi, tidak hanya selisih antara target dengan keluaran jaringan ke- $i$  yang diperhitungkan, tetapi juga keluaran rata-rata setiap jaringan dengan target. Hal inilah yang membuat skema NCL memiliki diversifikasi data yang lebih baik. Oleh karena itu, bobot akhir yang terbentuk dari pelatihan ini seharusnya memiliki kemampuan yang lebih baik dalam mengelompokkan dan mengidentifikasi gambar baru.

Skema 4 (parsialisasi data dengan NCL) seharusnya memiliki performa yang paling baik di antara skema-skema lainnya. Hal ini disebabkan selain menggunakan algoritma NCL yang memperhitungkan penalty term, data pelatihan pada skema ini diacak sedemikian rupa setiap jaringan individu menerima masukan pola yang urutannya berbeda satu sama lain. Seharusnya, skema ini menghasilkan diversifikasi yang paling baik, sehingga pada akhirnya dapat mengklasifikasikan dan mengenali gambar baru dengan lebih baik.

Namun hipotesis tersebut tidak tampak pada eksperimen yang dilakukan. Hal ini dapat disebabkan karena recognition rate jaringan individu pada awalnya sudah cukup tinggi ( $\pm 96\%$ ), sehingga efek skema-skema yang dilakukan pada jaringan ensemble tidak dapat teramati. Recognition rate awal ini berhubungan dengan parameter-parameter pada jaringan individu yang hanya menggunakan parameter default (halaman 32). Jika recognition rate jaringan individu masih rendah, maka efek dari masing-masing skema kemungkinan dapat teramati dengan lebih baik.

**Tabel 5.43.** Tabel Perbandingan Training Time untuk Keempat Skema (dalam satuan detik)

Jumlah Jaringan	Skema 1 Algoritma Backpropagation	Skema 2 Parsialisasi data (90%)	Skema 3 NCL ( $\lambda = 0.8$ )	Skema 4 NCL & Parsialisasi ( $\lambda = 0.8, 100\%$ )
1	25.9373	23.4267	48.9371	47.4210
2	51.8802	72.5118	92.8513	91.5652
3	77.8318	99.4083	147.2998	134.9581
4	104.4128	134.4116	178.8349	181.7716
5	130.0167	172.3710	225.4366	219.5824

**Tabel 5.44.** Tabel Perbandingan Recognition Time untuk Keempat Skema (dalam satuan detik)

Jumlah Jaringan	Skema 1 Algoritma Backpropagation	Skema 2 Parsialisasi data (90%)	Skema 3 NCL ( $\lambda = 0.8$ )	Skema 4 NCL & Parsialisasi ( $\lambda = 0.8, 100\%$ )
1	0.0130	0.0129	0.1009	0.0973
2	0.0143	0.0155	0.1870	0.1869
3	0.0159	0.0166	0.2731	0.2695
4	0.0177	0.0182	0.3673	0.3599
5	0.0193	0.0196	0.4459	0.4489

Jika ditinjau dari lama waktu pelatihan dan waktu pengenalan gambar, semua skema menunjukkan semakin banyak jumlah jaringan yang digunakan, semakin lama waktu pelatihan dan pengenalan gambarnya. Namun bila dibandingkan satu persatu, skema 2 memiliki waktu yang sedikit lebih lama daripada skema 1, karena adanya tahap pengacakan urutan pola pelatihan. Tahap ini tidak dapat dipisahkan dari pelatihan.

Sementara itu, skema 3 dan 4 memiliki waktu yang paling lama dibanding skema 1 dan 2. Hal ini disebabkan karena skema 3 dan 4 menggunakan algoritma NCL. Algoritma NCL membutuhkan perhitungan yang lebih banyak dan kompleks sehingga waktu yang dibutuhkan untuk mengeksekusi program pun menjadi lebih lama. Skema 3 dan 4 tidak menunjukkan perbedaan yang signifikan, meskipun seharusnya skema 4 membutuhkan waktu yang lebih lama karena adanya tahap pengacakan urutan pola pelatihan.

Perlu diperhatikan pula pengukuran waktu pelatihan dan pengenalan gambar pada eksperimen ini sangat tergantung pada kecepatan prosesor komputer. Pada penelitian digunakan prosesor *Core 2 Duo* 1.66 GHz dengan RAM 502 MB. Waktu pelatihan dan pengenalan gambar dapat berubah-ubah pada setiap pengambilan data. Mengacu pada hasil eksperimen ini, performa keempat skema tidak jauh berbeda. Dengan demikian, skema 1 dan 2 lebih direkomendasikan untuk diterapkan karena waktu pelatihan dan pengenalan gambar yang lebih singkat.

Seperti yang telah dikemukakan pada bab IV, performa suatu jaringan yang optimal tidak hanya ditinjau dari tingginya recognition rate, tetapi juga singkatnya waktu pelatihan dan pengenalan gambar. Berdasarkan pertimbangan-pertimbangan tersebut, maka jumlah jaringan yang cukup optimal pada eksperimen ini adalah tiga jaringan (untuk semua skema). Jaringan ensemble dengan empat atau lima jaringan memang memiliki recognition rate yang lebih tinggi, tetapi jika dibandingkan terhadap tiga jaringan, perbedaannya sangat kecil, tidak sebanding dengan penambahan waktu pelatihannya yang cukup besar perbedaannya.

## 5.6. Optimalisasi Jaringan Syaraf Ensemble

Optimalisasi performa dengan mengubah parameter juga dapat dilakukan untuk jaringan ensemble, agar recognition rate-nya mencapai 100% dan waktu pelatihan yang lebih singkat. Sama seperti yang dilakukan pada bab IV, peneliti mencoba untuk melakukan optimalisasi jaringan dengan parameter-parameter yang sama seperti pada tabel 4.9 (halaman 68). Parameter optimalisasi diterapkan pada keempat skema yang memiliki hasil terbaik pada percobaan (mengacu pada pembahasan 5.5) dengan jumlah jaringan sebanyak lima.

**Tabel 5.45.** Recognition Rate dan Recognition Time pada Parameter Default

Skema		Recognition Rate		Waktu (detik)	
		Training	Testing	Training	Testing (1 gambar)
<b>Jaringan Syaraf Tunggal</b>		100%	96.4%	36.97	0.0107
1	<b>Algoritma Backpropagation</b>	100%	99.1%	130.02	0.0193
2	<b>Parsialisasi data dengan Algoritma Backpropagation</b>	100%	99.7%	172.37	0.0196
3	<b>Algoritma NCL</b>	100%	99.5%	225.47	0.4459
4	<b>Parsialisasi Data dengan Algoritma NCL</b>	100%	99.3%	219.58	0.4489

**Tabel 5.46.** Recognition Rate dan Recognition Time pada Parameter Optimalisasi

Skema		Recognition Rate		Waktu (detik)	
		Training	Testing	Training	Testing (1 gambar)
<b>Jaringan Syaraf Tunggal</b>		100%	99.2%	0.77	0.0102
1	<b>Algoritma Backpropagation</b>	100%	99.6%	3.44	0.0192
2	<b>Parsialisasi data dengan Algoritma Backpropagation</b>	100%	99.8%	3.87	0.0195
3	<b>Algoritma NCL</b>	100%	99.9%	10.96	0.0215
4	<b>Parsialisasi Data dengan Algoritma NCL</b>	100%	99.8%	12.40	0.0208

Tabel 5.45 menunjukkan performa sistem pada parameter default yang terdiri dari recognition rate, waktu pelatihan, dan waktu pengenalan gambar, dengan merangkum hasil-hasil yang ditunjukkan pada tabel 4.10 (default), 5.42, 5.43, dan 5.44 (jumlah jaringan = 5). Sebagai pembandingan, tabel 5.46 menunjukkan performa pada parameter optimalisasi, di mana hasil untuk

jaringan syaraf tiruan mengacu pada tabel 4.10 dan hasil lainnya diperoleh melalui eksperimen baru (jumlah jaringan = 5).

Dari kedua tabel tersebut, dapat dilihat bahwa dengan jumlah jaringan sebanyak lima, parameter optimalisasi memberikan performa yang lebih baik dibandingkan parameter default, baik pada jaringan syaraf tunggal maupun ensemble. Selain recognition rate yang meningkat hingga mendekati 100%, waktu pelatihan berhasil dipersingkat cukup besar, hingga mencapai 95 - 98% dari parameter default. Selain itu, waktu pengenalan gambar juga berhasil dipersingkat, walaupun tidak terlalu signifikan karena nilainya yang kecil. Perlu diingat bahwa parameter optimalisasi yang dilakukan pada penelitian ini bukanlah parameter yang paling optimal, karena masih ada kemungkinan-kemungkinan lain yang dapat menghasilkan performa yang lebih optimal.

Jika recognition rate ditinjau secara per kelas, maka hasil yang diperoleh ditunjukkan sebagai berikut:

**Tabel 5.47.** Recognition Rate per Kelas pada Parameter Optimalisasi Jaringan Syaraf Ensemble

No	Kelas		Recognition Rate (Testing) Skema:				Gambar Salah Uji
			1	2	3	4	
1	ann		100%	100%	100%	100%	-
2	dau		100%	100%	100%	100%	-
3	cin		100%	100%	100%	100%	-
4	roy		100%	100%	100%	100%	-
5	dev		100%	100%	100%	100%	-
6	chr		100%	100%	100%	100%	-
7	pel		97%	98%	100%	99%	69
8	rud		100%	100%	100%	100%	-
9	eni		99%	100%	99%	99%	87
10	pep		100%	100%	100%	100%	-

Jika dibandingkan dengan tabel 4.2 pada halaman 39 dan tabel 4.11 pada halaman 69, recognition rate tiap kelas yang ditunjukkan pada tabel 5.47

mengalami peningkatan, khususnya pada kelas ke-7 dan 9. Gambar-gambar yang sering salah dikenali, seperti gambar 70 telah dapat dikenali dengan benar dengan jaringan syaraf ensemble. Selain itu, gambar 69 dan 87 juga semakin banyak dikenali dengan benar. Hanya pada beberapa percobaan, masih ada kesalahan identifikasi pada gambar-gambar tersebut. Hal ini disebabkan kedua gambar tersebut memang memiliki perbedaan yang cukup mencolok dibandingkan dengan gambar-gambar di kelas yang sama pada data pelatihan. Jika jaringan syaraf ini dapat dioptimalkan lebih baik lagi, bukan tidak mungkin gambar-gambar tersebut dapat selalu diidentifikasi dengan benar.

Dari keseluruhan eksperimen, dapat disimpulkan jaringan syaraf ensemble (dengan jumlah jaringan lebih dari satu) akan menghasilkan recognition rate yang lebih baik dibandingkan dengan jaringan syaraf tunggal jika menggunakan parameter-parameter yang sama. Namun sebenarnya jaringan syaraf tunggal pun juga dapat memiliki recognition rate yang cukup baik seperti jaringan syaraf ensemble. Misalnya, recognition rate pada jaringan syaraf tunggal dengan parameter optimalisasi dapat menghasilkan recognition rate 99.2%, tidak jauh berbeda dengan jaringan syaraf ensemble dengan jumlah jaringan sebanyak empat (lihat tabel 5.42).

Jika waktu pembelajaran dan waktu pengenalan gambar yang lebih lama menjadi masalah, maka jaringan syaraf tunggal sebenarnya sudah cukup baik digunakan untuk sistem pengenalan wajah, dibandingkan menggunakan jaringan syaraf ensemble. Namun bila optimalisasi telah dilakukan pada jaringan syaraf tunggal dan hasil yang ditunjukkan masih belum memuaskan, jaringan syaraf ensemble adalah solusi untuk meningkatkan performanya. Atau, jika jaringan syaraf tunggal telah optimal, jaringan syaraf ensemble dapat digunakan untuk memaksimalkan performanya.

Dengan demikian, dapat disimpulkan untuk menghasilkan performa jaringan syaraf tiruan yang optimal, dapat dilakukan dengan cara mengubah parameter-parameter di dalamnya atau menggunakan ensemble jaringan. Sejauh ini, cara-cara tersebut tergolong dalam optimalisasi tahap pelatihan.

Banyak cara lain yang dapat dilakukan pada tahap pelatihan untuk mengoptimalkan performa jaringan, misalnya dengan mencoba algoritma lain pada jaringan syaraf ensemble. Sebagai saran, optimalisasi tidak hanya dapat diterapkan pada tahap pelatihan saja seperti yang dilakukan pada penelitian ini. Optimalisasi juga dapat dilakukan pada tahap pengambilan data, misalnya dengan mengatur *ambient light*, mengambil data yang lebih banyak variasinya, dan memperbaiki kinerja sistem pendeteksi wajah.

Selain itu, optimalisasi pada tahap pra-pemrosesan dapat dilakukan, misalnya dengan mencoba menambahkan *frame* bundar pada gambar wajah agar mengurangi *background noise*. Optimalisasi pada tahap ekstraksi fitur, misalnya dengan mencoba menggunakan pereduksi dimensi selain PCA. Dan optimalisasi pada tahap pengujian, misalnya dengan mencoba menggunakan metode-metode pengambilan keputusan yang lain, seperti metode voting, nilai maksimum, atau perkalian produk.

Semua ini dapat dicapai dengan melakukan eksperimen-eksperimen di luar apa yang telah dikerjakan pada penelitian ini. Penulis berharap dari penelitian ini, muncul penelitian-penelitian lain yang dapat melanjutkan dan mengoptimalkan performa sistem pengenalan wajah untuk citra infra merah.

## BAB VI

### KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari keseluruhan penelitian skripsi ini dan saran untuk penelitian selanjutnya.

#### 6.1. Kesimpulan

Dari hasil penelitian, Jaringan Syaraf Tiruan (JST) dengan algoritma backpropagation dapat digunakan untuk melatih pola yang ada pada gambar wajah dan mengenali pola lain yang diberikan dengan baik. JST yang digunakan memiliki masing-masing sebuah lapisan masukan, lapisan tersembunyi, dan lapisan keluaran. Pelatihan dilakukan dengan algoritma backpropagation yang terdiri dari tahap propagasi maju, propagasi balik, dan perubahan bobot. Pengenalan gambar atau pengujian dilakukan hanya dengan tahap propagasi maju dan keputusan diambil dari pengukuran jarak Euclidean.

Principal Component Analysis (PCA) merupakan metode yang baik untuk mengurangi dimensi data yang sangat besar menjadi lebih sederhana sehingga proses komputasi dapat berjalan dengan lebih cepat. Semakin besar dimensi yang direduksi PCA tidak memberikan kenaikan tingkat akurasi pengenalan wajah (recognition rate) yang signifikan. Hal ini disebabkan karena banyaknya dimensi didasarkan urutan besarnya nilai eigen, semakin kecil nilai eigen, maka vektor eigen yang berpasangan dengannya tidak berkontribusi sebesar vektor eigen dengan nilai eigen yang besar.

Beberapa parameter dan variasi eksperimen pada jaringan syaraf tunggal menunjukkan bahwa:

1. Metode Nguyen-Widrow untuk memberikan recognition rate yang lebih baik dibandingkan dengan metode acak.
2. Banyaknya neuron pada lapisan tersembunyi memiliki jumlah minimal dan maksimal tertentu agar memberikan performa yang optimal untuk sistem.

3. Banyaknya iterasi yang dilakukan sebagai syarat henti pelatihan tidak mempengaruhi recognition rate apabila jumlah iterasi tersebut telah melewati batas toleransi error atau jaringan telah mencapai konvergensi.
4. Besarnya laju pembelajaran akan mempengaruhi pencapaian konvergensi error tetapi tidak terlalu berpengaruh terhadap recognition rate. Nilai  $\alpha$  yang semakin besar akan membuat grafik error terhadap epoch semakin berfluktuasi.
5. Penggunaan momentum lebih tepat diberikan untuk nilai  $\alpha$  yang relatif kecil (0.2 dan 0.3). Namun hal ini belum tentu memberikan performa (recognition rate) yang lebih baik.

Optimalisasi performa jaringan syaraf tiruan dapat dilakukan dengan pengubahan parameter-parameter pada jaringan syaraf tunggal dan menggunakan ensemble jaringan. Penelitian pada jaringan syaraf ensemble menunjukkan bahwa semakin banyak jumlah jaringan yang digunakan, semakin baik sistem dapat mengenali gambar wajah, namun waktu pelatihan dan pengenalan gambar menjadi lebih lama. Dari keempat skema yang digunakan, performa yang ditunjukkan dari recognition rate tidak jauh berbeda.

## 6.2. Saran

Peneliti berharap agar sistem pengenalan wajah ini dapat diaplikasikan ke dalam kehidupan sehari-hari, misalnya dalam sistem sekuriti. Namun perlu adanya pengembangan lebih lanjut mengenai sistem pendeteksi wajah (face detection system) agar dapat dijalankan bersama dengan sistem pengenalan wajah secara real-time.

Peneliti juga berharap hasil penelitian ini dapat dipakai sebagai referensi peneliti-peneliti lain yang ingin meneruskan riset dalam bidang sistem pengenalan wajah untuk mengembangkan sistem ini agar performanya lebih optimal lagi.

## LAMPIRAN

### A. Penurunan rumus perubahan bobot untuk algoritma NCL:

Fungsi error NCL dinyatakan oleh:

$$E_i = \frac{1}{N} \sum_{n=1}^N E_i(n) = \frac{1}{N} \sum_{n=1}^N \frac{1}{2} \left( F_i(n) - d(n) \right)^2 + \frac{1}{N} \sum_{n=1}^N \lambda p_i(n) \quad (\text{A.1})$$

Fungsi aktivasi yang digunakan pada lapisan keluaran adalah fungsi sigmoid, yang dinyatakan oleh:

$$F_i \left( f_{in} \right) = f(f_{in}) = \frac{1}{1 + e^{-f_{in}}} \quad (\text{A.2})$$

di mana  $f_{in}$  adalah fungsi keluaran, yang dinyatakan oleh:

$$f_{in} = w_{k0} + \sum_{j=1}^p z_j w_{kj} \quad (\text{A.3})$$

$z_j$  adalah keluaran lapisan tersembunyi yang telah diaktivasi, yang dinyatakan oleh:

$$z_j = f(z\_net_j) = \frac{1}{1 + e^{-z\_net_j}} \quad (\text{A.4})$$

di mana  $z\_net_j$  adalah fungsi keluaran, yang dinyatakan oleh:

$$z\_net_j = v_{j0} + \sum_{i=1}^n x_i v_{ji} \quad (\text{A.5})$$

Turunan parsial Error Function terhadap keluaran jaringan ke- $i$  disebut pula *local gradient*, yang dinyatakan oleh: ( $M$  = jumlah jaringan ensemble)

$$\begin{aligned} \frac{\partial E_i(n)}{\partial F_i(n)} &= F_i(n) - d(n) + \lambda \frac{\partial p_i(n)}{\partial F_i(n)} \\ &= F_i(n) - d(n) + \lambda \sum_{j \neq i} \left( F_j(n) - F(n) \right) \\ &= F_i(n) - d(n) + \lambda \left( M F(n) - F_i(n) - (M - 1) F(n) \right) \\ &= F_i(n) - d(n) + \lambda \left( F(n) - F_i(n) \right) \\ &= F_i(n) - d(n) - \lambda \left( F_i(n) - F(n) \right) \\ &= \left( -\lambda \right) \left( F_i(n) - d(n) \right) + \lambda \left( F(n) - d(n) \right) \\ &= \left[ -\lambda \left( d(n) - F_i(n) \right) + \lambda \left( d(n) - F(n) \right) \right] \end{aligned} \quad (\text{A.6})$$

1. Perubahan bobot antara lapisan tersembunyi dengan lapisan keluaran:

$$\Delta w_{kj(i)} = -\alpha \frac{\partial E_i}{\partial w_i} \quad \text{di mana} \quad \frac{\partial E_i}{\partial w_i} = \frac{\partial E_i}{\partial F_i} \times \frac{\partial F_i}{\partial \text{fin}_i} \times \frac{\partial \text{fin}_i}{\partial w_i} \quad (\text{A.7})$$

berikut ini adalah turunan parsial  $F_i$  terhadap  $\text{fin}_i$

$$\frac{\partial F_i}{\partial \text{fin}_i} = f'(\text{fin}_i) = F_i(n) \llcorner -F_i(n) \lrcorner \quad (\text{A.8})$$

dan berikut adalah turunan parsial  $\text{fin}_i$  terhadap  $w_i$

$$\frac{\partial \text{fin}_i}{\partial w_i} = z \quad (\text{A.9})$$

Substitusi (A.6), (A.8) dan (A.9) ke (A.7):

$$\begin{aligned} \Delta w_{kj(i)} &= -\alpha \llcorner \llcorner -\lambda (d(n) - F_i(n)) + \lambda (d(n) - F(n)) \llcorner F_i(n) \llcorner -F_i(n) \lrcorner \lrcorner z \\ &= \alpha \llcorner \llcorner -\lambda (d(n) - F_i(n)) + \lambda (d(n) - F(n)) \llcorner F_i(n) \llcorner -F_i(n) \lrcorner \lrcorner z \end{aligned} \quad (\text{A.10})$$

Suku  $\llcorner \llcorner -\lambda (d(n) - F_i(n)) + \lambda (d(n) - F(n)) \llcorner F_i(n) \llcorner -F_i(n) \lrcorner \lrcorner$  dapat dilambangkan dengan  $\delta_k$ , sehingga persamaan di atas dapat disingkat menjadi:

$$\Delta w_{kj(i)} = \alpha \delta_k z \quad (\text{A.11})$$

2. Perubahan bobot antara lapisan tersembunyi dengan lapisan keluaran:

$$\Delta v_{ji(i)} = -\alpha \frac{\partial E_i}{\partial v_i} \quad \text{di mana} \quad \frac{\partial E_i}{\partial v_i} = \frac{\partial E_i}{\partial F_i} \times \frac{\partial F_i}{\partial \text{fin}_i} \times \frac{\partial \text{fin}_i}{\partial z_i} \times \frac{\partial z_i}{\partial \text{zin}_i} \times \frac{\partial \text{zin}_i}{\partial v_i} \quad (\text{A.12})$$

berikut ini adalah turunan parsial  $F_i$  terhadap  $\text{fin}_i$  (sama seperti A.8)

$$\frac{\partial F_i}{\partial \text{fin}_i} = f'(\text{fin}_i) = F_i(n) \llcorner -F_i(n) \lrcorner \quad (\text{A.13})$$

dan berikut adalah turunan parsial  $\text{fin}_i$  terhadap  $w_i$

$$\frac{\partial \text{fin}_i}{\partial z_i} = w_{kj(i)} \quad (\text{A.14})$$

Sedangkan turunan parsial  $z_i$  terhadap  $z_{in_i}$  dinyatakan oleh:

$$\frac{\partial z_i}{\partial z_{in_i}} = f'(z_{in_i}) = z_i(n) \left( -z_i(n) \right) \quad (\text{A.15})$$

dan turunan parsial  $z_{in_i}$  terhadap  $v_i$  dinyatakan oleh:

$$\frac{\partial z_{in_i}}{\partial v_i} = x \quad (\text{A.16})$$

Substitusi (A.6), (A.13), (A.14), (A.15), dan (A.16) ke (A.12):

$$\begin{aligned} \Delta v_{ji(i)} &= -\alpha \left[ \left( -\lambda \left( d(n) - F_i(n) \right) + \lambda \left( d(n) - F(n) \right) \right) F_i(n) \left( -F_i(n) \right) w \left( -z_i(n) \right) x \right. \\ &= \alpha \left[ -\lambda \left( d(n) - F_i(n) \right) + \lambda \left( d(n) - F(n) \right) \right] F_i(n) \left( -F_i(n) \right) w \left( -z_i(n) \right) x \\ &= \alpha \delta_k w \left( -z_i(n) \right) x \end{aligned} \quad (\text{A.17})$$

Suku  $\delta_k w \left( -z_i(n) \right)$  dapat dilambangkan dengan  $\delta_j$  sehingga persamaan di atas dapat disingkat menjadi:

$$\Delta v_{ji(i)} = \alpha \delta_j x \quad (\text{A.18})$$

## B. Database Pelatihan dan Pengujian

Database pelatihan (training) terdiri dari 200 gambar, berikut ini gambar dengan ukuran aslinya:

- Target 1 (ann)



- Target 2 (dau)



- Target 3 (cin)



- Target 4 (roy)



- Target 5 (dev)



- Target 6 (chr)



- Target 7 (pel)



- Target 8 (rud)



- Target 9 (eni)



- Target 10 (pep)



Data pengujian terdiri dari 100 foto yang diambil dari dalam database di atas (training) dan 100 foto yang diambil dari luar database (testing).

- Gambar-gambar uji dari dalam database (training):





- Gambar-gambar uji dari luar database (testing):



### C. Nilai Eigen Matriks Kovarians pada Proses PCA

Urutan	Nilai Eigen	Perbandingan terhadap Nilai Eigen Terbesar
1	328.1592	1
2	215.6599	0.657181
3	93.61264	0.285266
4	66.18376	0.201682
5	64.12812	0.195418
6	45.27672	0.137972
7	39.79504	0.121267
8	34.27214	0.104438
9	24.15623	0.073611
10	22.96424	0.069979
11	21.13649	0.064409
12	15.91667	0.048503
13	12.92856	0.039397
14	10.89781	0.033209
15	10.39181	0.031667
16	9.42403	0.028718
17	7.887636	0.024036
18	7.346981	0.022388
19	6.548402	0.019955
20	6.17447	0.018815
21	6.116899	0.01864
22	5.529439	0.01685
23	5.396109	0.016444
24	4.846025	0.014767
25	4.466251	0.01361
26	4.383659	0.013358
27	4.013559	0.012231
28	3.590638	0.010942
29	3.393257	0.01034
30	3.283065	0.010004
31	3.136108	0.009557
32	2.922943	0.008907
33	2.868847	0.008742
34	2.746669	0.00837
35	2.656784	0.008096
36	2.583059	0.007871
37	2.516091	0.007667
38	2.408402	0.007339
39	2.261673	0.006892
40	2.229908	0.006795
41	2.127489	0.006483
42	2.063376	0.006288
43	1.98001	0.006034
44	1.96263	0.005981
45	1.861411	0.005672
46	1.796098	0.005473
47	1.717207	0.005233
48	1.614329	0.004919
49	1.580144	0.004815
50	1.51824	0.004627
51	1.454059	0.004431
52	1.438226	0.004383
53	1.394939	0.004251
54	1.379371	0.004203
55	1.345301	0.0041
56	1.297426	0.003954
57	1.268767	0.003866
58	1.199232	0.003654
59	1.187789	0.00362
60	1.168606	0.003561
61	1.156949	0.003526
62	1.122696	0.003421
63	1.091887	0.003327
64	1.069803	0.00326
65	1.044828	0.003184
66	1.004129	0.00306
67	0.997867	0.003041
68	0.953919	0.002907
69	0.931518	0.002839
70	0.913739	0.002784
71	0.888325	0.002707
72	0.870615	0.002653
73	0.860518	0.002622

74	0.839209	0.002557
75	0.818278	0.002494
76	0.808269	0.002463
77	0.801241	0.002442
78	0.772953	0.002355
79	0.771012	0.00235
80	0.761598	0.002321
81	0.751765	0.002291
82	0.738895	0.002252
83	0.720121	0.002194
84	0.707456	0.002156
85	0.697046	0.002124
86	0.684887	0.002087
87	0.678733	0.002068
88	0.657037	0.002002
89	0.639456	0.001949
90	0.629006	0.001917
91	0.622629	0.001897
92	0.613867	0.001871
93	0.599843	0.001828
94	0.595227	0.001814
95	0.583096	0.001777
96	0.574264	0.00175
97	0.571729	0.001742
98	0.55497	0.001691
99	0.540514	0.001647
100	0.533992	0.001627
101	0.524061	0.001597
102	0.518245	0.001579
103	0.514019	0.001566
104	0.504579	0.001538
105	0.495911	0.001511
106	0.494101	0.001506
107	0.480579	0.001464
108	0.470835	0.001435
109	0.465935	0.00142
110	0.456153	0.00139
111	0.448807	0.001368
112	0.443004	0.00135
113	0.43401	0.001323
114	0.428264	0.001305

115	0.419717	0.001279
116	0.415934	0.001267
117	0.407949	0.001243
118	0.401585	0.001224
119	0.397564	0.001211
120	0.392676	0.001197
121	0.3881	0.001183
122	0.38068	0.00116
123	0.377555	0.001151
124	0.369064	0.001125
125	0.363501	0.001108
126	0.35285	0.001075
127	0.345667	0.001053
128	0.345233	0.001052
129	0.332662	0.001014
130	0.329358	0.001004
131	0.318283	0.00097
132	0.316664	0.000965
133	0.311648	0.00095
134	0.305083	0.00093
135	0.296888	0.000905
136	0.293777	0.000895
137	0.285313	0.000869
138	0.277859	0.000847
139	0.271435	0.000827
140	0.270301	0.000824
141	0.263946	0.000804
142	0.259422	0.000791
143	0.254562	0.000776
144	0.248535	0.000757
145	0.246817	0.000752
146	0.241824	0.000737
147	0.23851	0.000727
148	0.226927	0.000692
149	0.219168	0.000668
150	0.204625	0.000624
151	0.203838	0.000621
152	0.198647	0.000605
153	0.196118	0.000598
154	0.18849	0.000574
155	0.182115	0.000555

156	0.176853	0.000539
157	0.174027	0.00053
158	0.167835	0.000511
159	0.161473	0.000492
160	0.160245	0.000488
161	0.149108	0.000454
162	0.14664	0.000447
163	0.145932	0.000445
164	0.143278	0.000437
165	0.134933	0.000411
166	0.131528	0.000401
167	0.123355	0.000376
168	0.1212	0.000369
169	0.117191	0.000357
170	0.114147	0.000348
171	0.107635	0.000328
172	0.107294	0.000327
173	0.102675	0.000313
174	0.097546	0.000297
175	0.092762	0.000283
176	0.091185	0.000278
177	0.087149	0.000266
178	0.085016	0.000259
179	0.084127	0.000256
180	0.079681	0.000243
181	0.07913	0.000241
182	0.075875	0.000231
183	0.074781	0.000228
184	0.072125	0.00022
185	0.069875	0.000213
186	0.069112	0.000211
187	0.066543	0.000203
188	0.064625	0.000197
189	0.061543	0.000188
190	0.060482	0.000184
191	0.060244	0.000184
192	0.055218	0.000168
193	0.052889	0.000161
194	0.050654	0.000154
195	0.050284	0.000153
196	0.048925	0.000149

197	0.046448	0.000142
198	0.041948	0.000128
199	0.038361	0.000117
200	2.50E-13	7.63E-16
201	2.15E-13	6.56E-16
202	1.18E-13	3.59E-16
203	7.56E-14	2.3E-16
204	5.60E-14	1.71E-16
205	4.88E-14	1.49E-16
206	4.41E-14	1.34E-16
207	4.13E-14	1.26E-16
208	3.75E-14	1.14E-16
209	3.28E-14	1E-16
210	3.17E-14	9.67E-17
211	2.90E-14	8.84E-17
212	2.51E-14	7.64E-17
213	2.47E-14	7.53E-17
214	2.37E-14	7.23E-17
215	2.21E-14	6.75E-17
216	2.01E-14	6.12E-17
217	1.82E-14	5.54E-17
218	1.71E-14	5.22E-17
219	1.66E-14	5.05E-17
220	1.58E-14	4.82E-17
221	1.52E-14	4.65E-17
222	1.50E-14	4.57E-17
223	1.45E-14	4.41E-17
224	1.36E-14	4.15E-17
225	1.33E-14	4.06E-17
226	1.32E-14	4.03E-17
227	1.26E-14	3.85E-17
228	1.16E-14	3.53E-17
229	9.86E-15	3E-17
230	8.80E-15	2.68E-17
231	8.29E-15	2.53E-17
232	4.28E-15	1.3E-17
233	4.09E-15	1.25E-17
234	3.76E-15	1.14E-17
235	3.52E-15	1.07E-17
236	3.47E-15	1.06E-17
237	3.40E-15	1.04E-17

<b>238</b>	3.35E-15	1.02E-17
<b>239</b>	3.28E-15	9.98E-18
<b>240</b>	3.24E-15	9.88E-18
<b>241</b>	3.21E-15	9.78E-18
<b>242</b>	3.16E-15	9.62E-18
<b>243</b>	3.15E-15	9.59E-18
<b>244</b>	3.11E-15	9.48E-18
<b>245</b>	3.08E-15	9.39E-18
<b>246</b>	3.04E-15	9.28E-18

<b>247</b>	3.03E-15	9.24E-18
<b>248</b>	3.01E-15	9.17E-18
<b>249</b>	2.99E-15	9.12E-18
<b>250</b>	2.98E-15	9.08E-18
<b>251</b>	2.97E-15	9.06E-18
<b>252</b>	2.96E-15	9.01E-18
<b>253</b>	2.95E-15	8.99E-18
<b>254</b>	2.93E-15	8.93E-18
<b>255</b>	2.88E-15	8.79E-18

## DAFTAR REFERENSI

Buchala, Samarasena et al. Principal Component Analysis of Gender, Ethnicity, Age, and Identity of Face Images. <http://info.scopus.com>.

Chan Z. & Kasabov N. (2005). A Preliminary Study on Negative Correlation Learning via Correlation-Corrected Data (NCCD). *Neural Processing Letters*, 21:207-214.

Chen Xin, Flynn,P., & Bowyer K.W. (2005). IR and Visible Light Face Recognition. *Computer Vision and Image Understanding* 99, 332-358.

Dam, H.H et al. (2008). Neural-Based Learning Classifier Systems. *IEEE Transactions on Knowledge and Data Engineering*, Vol.20, No.1, 26-39.

Debnath, K.K, Ahdikary J.K, & Shahjahan, Md. (2009). A Currency Recognition System Using Negatively Correlated Neural Network Ensemble. *IEEE Proc. of 2009 12<sup>th</sup> International Conference on Computer and Information*, 367-372.

Fadlil, Abdul. (2007). Perbandingan Pengklasifikasi Fungsi Jarak dan Jaringan Syaraf Tiruan pada Sistem Pengenalan Wajah. *Seminar Nasional Aplikasi Teknologi Informasi (SNATI 2007)*, ISSN: 1907-5022, G-7 – G-10.

Fadlil, Abdul. (2008). Sistem Pengenalan Wajah Manusia Secara Realtime Menggunakan Algoritma Jaringan Syaraf Tiruan. *Seminar Nasional Aplikasi Teknologi Informasi (SNATI 2008)*, ISSN: 1907-5022, J-5 – J-8.

Howard, Anton. (2002). *Dasar-Dasar Aljabar Linear Jilid 2*. Batam: Interaksara.

Islam, M. et.al. (2003). A Constructive Algorithm for Training Cooperative Neural Network Nsembles. *IEEE Transactions on Neural Networks*, Vol.14, No. 4, 820-834.

Jiang Qing. Principal Component Analysis and Neural Network Based Face Recognition. <http://info.scopus.com>.

Jolliffe, I.T. (2002). *Principal Component Analysis* (second edition). Springer Series in Statistics.

Jong Jek Siang. (2005). *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan MATLAB*. Yogyakarta: Penerbit Andi.

Kai Hansen, L & Salamon, P. (1990). Neural Network Ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.12, No.10, 993-1001.

Kang Jinwoo et.al. (2006). Short Wavelength Infrared Face Recognition For Personalization. *IEEE*, 1-4244-0481-9, 2757-2760.

Kusumoputro, Benyamin. (2001). *Jaringan Neural Buatan*. Fakultas Ilmu Komputer Universitas Indonesia.

Lee H., Hong S., & Kim E. (2009). Neural Network Ensemble with Probabilistic Fusion and Its Application to Gait Recognition. *Elsevier Neurocomputing*, 72, 1557-1564.

Li Minku, F., Inoue, H., & Yao X. (2007). Negative Correlation in Incremental Learning. *Springer Science+Business Media B.V. Nat Comput*, 8:289-320.

Liu J et al. (2009). Weights Updated Voting for Ensemble of Neural Networks Based Incremental Learning. *Springer-Verlag ISSN 2009*, Part I, LNCS 5551, 661-669.

Liu Y. & Yao X. (1999). Ensemble Learning via Negative Correlation. *Pergamon Neural Networks 12*, 1299-1404.

O'Toole, A. & Tistarelli, M. (2009). Chapter 5: Face Recognition in Humans and Machines. *Handbook of Remote Biometrics, Advances in Pattern Recognition*, DOI 10.1007/978-1-84882-385-3 5

Rokach, Lior (2009). Ensemble-based Classifiers. *Spinger Science+Business Media B.V. Artif.Intell.Rev 2010*, 33:1-39.

Shlens, Jonathan. (2009). *A Tutorial on Principal Component Analysis*. [shlens@salk.edu](mailto:shlens@salk.edu).

Smith, Lindsay I. (2002). *A Tutorial on Principal Component Analysis*. <http://info.scopus.com>.

Socolinsky, D.A., Wolff, L.B., & Lundberg.A.J. (2006). Face Recognition with Intensified NIR Imagery. *Biometric Technology for Human Identification III*, Proc. of SPIE Vol.6202, 620203.

Wang Jian et.al. (2007). Number Image Recognition Based on Neural Network Ensemble. *IEEE Third International Conference on Natural Computation 2007*, 0-7695-2975-9/07.

Yong L. & Zou X.F. (2003). Analysis of Negative Correlation Learning. *Wuhan University Journal of Natural Sciences*, Vol. 8, No. 18, 165-175.

Zhang Xingming et.al. (2008). A Biological Intelligent Access Control System Based on DSP and NIR Technology. *Springer-Verlag, ICIC2008, LNCS5226*, 55-62.

Zhao Shuyan & Grigat, Rolf-Rainer. (2005). An Automatic Face Recognition System in the Near Infrared Spectrum. *MLDM 2005*, 437-444.

Zhao W et.al. (2003). Face Recognition : A Literature Survey. *ACM Computing Surveys*, Vol 35, No.4, 399-458.

