



UNIVERSITAS INDONESIA

**STUDI KARAKTERISTIK NEURAL NETWORK ENSEMBLE
BERBASIS FUNGSI EROR
CROSS ENTROPY DAN KUADRATIS**

SKRIPSI

**ALIFIA FITHRITAMA
0706163590**

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
JUNI 2011**



UNIVERSITAS INDONESIA

**STUDI KARAKTERISTIK NEURAL NETWORK ENSEMBLE
BERBASIS FUNGSI EROR
CROSS ENTROPY DAN KUADRATIS**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

**ALIFIA FITHRITAMA
0706163590**

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
JUNI 2011**

HALAMAN PERNYATAAN ORISINALITAS

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

Nama : Alifia Fithritama
NPM : 0706163590
Tanda Tangan : *Alifia Fithritama*
Tanggal : 13 Juni 2011


HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Alifia Fithritama
NPM : 0706163590
Program Studi : Teknik Elektro
Judul Skripsi : Studi Karakteristik Neural Network Ensemble
Berdasarkan Fungsi Error Cross-Entropy dan Kuadratis

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 :
Ir. Aries Subianto, M.SEE. ()

Penguji 2 :
Dr. Abdul Halim, M.Eng. ()

Ditetapkan di : Depok

Tanggal : 4 Juli 2011

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) Stephen Roy Imantaka S.T., yang telah membantu penulis memahami sistem yang beliau bangun dalam skripsi beliau;
- 4) Dimas Adityamurthi, Harry Bian Pramudia, dan Stephen, teman satu kelompok bimbingan Prof. Dr.Eng. Drs. Benyamin Kusumoputro M.Eng. yang telah banyak memberikan bantuan pada saat penulis mengalami kesulitan.
- 5) Asisten Laboratorium Kendali, Digital, Elektronika, dan Jaringan Departemen Teknik Elektro UI yang telah mengizinkan penulis menggunakan komputer lab untuk menjalankan program dalam rangka proses pengambilan data.
- 6) teman-teman di Departemen Teknik Elektro UI, terutama teman-teman peminatan Kendali 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 pengetahuan dan teknologi.

Depok, Juni 2011

Penulis

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

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

Nama : Alifia Fithritama
NPM : 0706163590
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:

**STUDI KARAKTERISTIK NEURAL NETWORK ENSEMBLE
BERBASIS FUNGSI EROR
CROSS ENTROPY DAN KUADRATIS**

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 : 13 Juni 2011

Yang menyatakan



(Alifia Fithritama)

ABSTRAK

Nama : Alifia Fithritama
Program Studi : Teknik Elektro
Judul : Studi Karakteristik Neural Network Ensemble Berbasis Fungsi
Error Cross Entropy dan Kuadratis

Dalam beberapa tahun ini, telah banyak penelitian yang berhubungan dengan pengenalan pola dilakukan dengan menggunakan jaringan syaraf tiruan (*artificial neural network*). Skripsi ini membahas tentang sistem pengenal pola berbasis *neural network ensemble* (NNE), yang merupakan kumpulan dari beberapa *neural network* tunggal. Penelitian ini membandingkan antara NNE yang dilatih dengan fungsi error kuadratis dan *cross-entropy*. Terdapat 12 dataset pola yang digunakan pada penelitian ini yaitu 9 dataset pola yang didapatkan dari “*UCI Repository of Machine Learning Database*”, 2 dataset citra wajah dari kamera infra merah dan kamera cahaya tampak, dan 1 dataset campuran aroma. Prosedur kerja sistem terdiri dari tahap pra-pemrosesan, pelatihan, dan pengujian. Pada tahap pelatihan, diterapkan algoritma *Negative Correlated Learning* (NCL) yang merupakan pengembangan dari algoritma standar *backpropagation*. Hasil pengujian yang ditinjau dari *recognition rate* menunjukkan NNE yang dilatih dengan fungsi error *cross-entropy* memberikan performa yang lebih baik dibandingkan dengan NNE yang dilatih dengan fungsi error kuadratis.

Kata kunci:

Neural network ensemble (NNE), *Negative Correlated Learning* (NCL), fungsi error kuadratis, fungsi error *cross-entropy*.

ABSTRACT

Name : Alifia Fithritama

Study Program: Electrical Engineering

Title : Development of Neural Network Ensemble Based on Cross Entropy and Mean Square Error Function

In recent years, many people have been working on pattern recognition using artificial neural network. This bachelor thesis discuss about pattern recognition system based on neural network ensemble (NNE), which is a group of some individual neural networks. This research compares between NNE which is trained using mean-of-square and cross-entropy error function. There are 12 datasets used in this experiment, which are 9 pattern datasets obtained from “UCI Repository of Machine Learning Database” and 2 dataset of frontal face images from infra red and visible-light camera, and 1 dataset of odor mixtures. The working procedures of the system consist of pre-processing, training and testing stages. In the training stage, Negative Correlated Learning (NCL) algorithm, a developed standard back propagation method, is applied and some parameters are varied to obtain the optimum performance. The testing result which is measured from recognition rate shows that NNE which is trained using cross-entropy error function has a better performance than the one with mean-of-square error function.

Keywords:

Neural network ensemble (NNE), Negative Correlated Learning (NCL), sum-of square error function, cross-entropy error function.

DAFTAR ISI

| | |
|--|-----------|
| HALAMAN JUDUL | i |
| HALAMAN PERNYATAAN ORISINALITAS | ii |
| HALAMAN PENGESAHAN | iii |
| KATA PENGANTAR | iv |
| HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI | v |
| ABSTRAK | vi |
| ABSTRACT | vii |
| DAFTAR ISI | viii |
| DAFTAR TABEL | xi |
| DAFTAR GAMBAR | xiv |
| BAB I : PENDAHULUAN..... | 1 |
| 1.1.Latar Belakang | 1 |
| 1.2.Pernyataan Masalah..... | 3 |
| 1.3.Tujuan Penulisan | 3 |
| 1.4.Batasan Masalah..... | 4 |
| 1.5.Manfaat Penelitian..... | 5 |
| 1.6.Tahapan Penelitian | 6 |
| 1.7.Sistematika Penulisan Laporan | 12 |
| BAB II : NEURAL NETWORK ENSEMBLE, FUNGSI EROR CROSS ENTROPY DAN KUADRATIS | 14 |
| 2.1. Neural Network dengan Algoritma Backpropagation | 14 |
| 2.2. Neural Network Ensemble dengan Algoritma NCL | 22 |
| 2.3. Fungsi Error Cross Entropy dan Kuadratis | 31 |
| 2.4. Neural Network untuk Pengenalan Pola..... | 34 |
| BAB III : SISTEM PENGENAL POLA BERBASIS NEURAL NETWORK ENSEMBLE..... | 38 |
| 3.1. Pendahuluan | 38 |
| 3.1.1. Latar Belakang Penelitian..... | 39 |
| 3.1.2. Skema Penelitian..... | 41 |
| 3.2. Hasil Penelitian dan Pembahasan..... | 44 |
| 3.3. Penutup | 51 |
| 3.3.1. Kesimpulan Penelitian..... | 51 |
| 3.3.2. Saran untuk Penelitian Lanjutan | 52 |

BAB IV : SISTEM PENGENAL WAJAH BERBASIS NEURAL NETWORK ENSEMBLE..... 53

| | |
|---|----|
| 4.1. Pendahuluan | 53 |
| 4.1.1. Latar Belakang Penelitian..... | 53 |
| 4.1.2. Skema Penelitian..... | 56 |
| 4.1.2.1. Skema Penelitian Citra Wajah Inframerah..... | 57 |
| 4.1.2.2. Skema Penelitian Citra Wajah Cahaya Tampak..... | 59 |
| 4.2. Hasil Penelitian dan Pembahasan..... | 61 |
| 4.2.1. Hasil Penelitian dengan Citra Wajah Inframerah | 62 |
| 4.2.2. Hasil Penelitian dengan Citra Wajah Cahaya Tampak | 71 |
| 4.2.3. Ringkasan Hasil Penelitian Citra Inframerah dan Cahaya Tampak | 77 |
| 4.3. Penutup | 80 |
| 4.3.1. Kesimpulan..... | 80 |
| 4.3.2. Saran untuk Penelitian Lanjutan..... | 81 |

BAB V : SISTEM PENGENAL AROMA BERBASIS NEURAL NETWORK ENSEMBLE..... 83

| | |
|---|----|
| 5.1. Pendahuluan | 83 |
| 5.1.1. Latar Belakang Penelitian..... | 83 |
| 5.1.2. Skema Penelitian..... | 85 |
| 5.2. Hasil Penelitian dan Pembahasan..... | 88 |
| 5.3. Penutup | 97 |
| 5.3.1. Kesimpulan..... | 97 |
| 5.3.2. Saran untuk Penelitian Lanjutan..... | 98 |

BAB VI KARAKTERISTIK NEURAL NETWORK ENSEMBLE BERBASIS FUNGSI EROR CROSS ENTROPY DAN KUADRATIS..... 99

| | |
|---|-----|
| 6.1. Nilai Error Pelatihan | 99 |
| 6.2. Jumlah Epoch Pelatihan..... | 101 |
| 6.3. Waktu Pelatihan | 103 |
| 6.4. Derajat Pengenalan..... | 105 |
| 6.4.1. Pengaruh Ukuran Dataset terhadap Derajat Pengenalan | 111 |
| 6.4.2. Pengaruh Nilai Parameter Penalti NCL (γ) terhadap Derajat Pengenalan..... | 112 |
| 6.4.3. Pengaruh Jumlah Jaringan terhadap Derajat Pengenalan | 113 |
| 6.5. Kesimpulan | 115 |

BAB VII : PENUTUP 117

| | |
|-----------------------|-----|
| 7.1. Kesimpulan | 117 |
| 7.2. Saran | 119 |

| | |
|---|------------|
| LAMPIRAN | 120 |
| A. Penurunan Rumus Perubahan Bobot untuk Fungsi Cross Entropy | 120 |
| A.1.Perubahan Bobot antara Lapisan Tersembunyi dan Lapisan Keluaran | 120 |
| A.2.Perubahan Bobot antara Lapisan Masukan dan Lapisan Tersembunyi | 121 |
| B. Penurunan Rumus Perubahan Bobot untuk Fungsi Kuadratis | 123 |
| B.1.Perubahan Bobot antara Lapisan Tersembunyi dan Lapisan Keluaran. | 123 |
| B.2.Perubahan Bobot antara Lapisan Masukan dan Lapisan Tersembunyi. | 124 |
| C. Penurunan Rumus Perubahan Bobot Algoritma NCL dengan Fungsi Error Cross-Entropy | 126 |
| C.1.Perubahan Bobot antara Lapisan Tersembunyi dan Lapisan Keluaran. | 127 |
| C.2.Perubahan Bobot antara Lapisan Masukan dan Lapisan Tersembunyi. | 128 |
| D. Penurunan Rumus Perubahan Bobot Algoritma NCL dengan Fungsi Error Kuadratis | 129 |
| D.1.Perubahan Bobot antara Lapisan Tersembunyi dan Lapisan Keluaran | 130 |
| D.2.Perubahan Bobot antara Lapisan Masukan dan Lapisan Tersembunyi | 131 |
| E. Dataset Pelatihan dan Pengujian | 132 |
| E.1.Dataset Balance scale | 134 |
| E.2.Dataset Breast cancer..... | 135 |
| E.3.Dataset Bupa | 136 |
| E.4.Dataset Credit card | 137 |
| E.5.Dataset Glass..... | 138 |
| E.6.Dataset Heart disease..... | 139 |
| E.7.Dataset Iris | 140 |
| E.8.Dataset Sonar | 142 |
| E.9.Dataset Vowel..... | 142 |
| E.10.Dataset Face infrared | 145 |
| E.11.Dataset Face visible..... | 147 |
| E.12.Dataset Odor | 148 |
| DAFTAR REFERENSI | 150 |

DAFTAR TABEL

| | | |
|-------------|---|----|
| Tabel 2.1. | Perbedaan antara Boolean-based Computer dan Neural-based Computer..... | 34 |
| Tabel 3.1. | Properties Dataset UCI untuk Sistem Pengenal Pola | 42 |
| Tabel 3.2. | Perbandingan Error Hasil Pelatihan NN Tunggal dan NNE untuk Dataset UCI | 44 |
| Tabel 3.3. | Perbandingan Jumlah Epoch Hasil Pelatihan NN Tunggal dan NNE untuk Dataset UCI | 45 |
| Tabel 3.4. | Perbandingan Waktu Pelatihan NN Tunggal dan NNE untuk Dataset UCI | 45 |
| Tabel 3.5. | Perbandingan Recognition Data Training NN Tunggal dan NNE untuk Dataset UCI | 45 |
| Tabel 3.6. | Perbandingan Recognition Data Testing NN Tunggal dan NNE untuk Dataset UCI | 46 |
| Tabel 4.1. | Ringkasan Hasil Pelatihan dengan Fungsi Error Kuadratis untuk Citra Inframerah | 62 |
| Tabel 4.2. | Ringkasan Hasil Pelatihan dengan Fungsi Error Cross-entropy untuk Citra Inframerah | 62 |
| Tabel 4.3. | Perbandingan Error, Epoch, dan Waktu Pelatihan NN Tunggal dan NNE untuk Citra Inframerah | 62 |
| Tabel 4.4. | Perbandingan Recognition Rate NN Tunggal dan NNE untuk Citra Inframerah | 62 |
| Tabel 4.5. | Perbandingan Error Hasil Pelatihan NN Tunggal dan NNE untuk Citra Inframerah | 65 |
| Tabel 4.6. | Perbandingan Epoch Hasil Pelatihan NN Tunggal dan NNE untuk Citra Inframerah | 65 |
| Tabel 4.7. | Perbandingan Waktu Pelatihan NN Tunggal dan NNE untuk Citra Inframerah | 65 |
| Tabel 4.8. | Perbandingan Recognition Rate Data Training NN Tunggal dan NNE untuk Citra Inframerah | 65 |
| Tabel 4.9. | Perbandingan Recognition Rate Data Testing NN Tunggal dan NNE untuk Citra Inframerah | 65 |
| Tabel 4.10. | Pengaruh Nilai α terhadap Recognition Rate NNE (3 jaringan) untuk Dataset Citra Wajah Inframerah | 68 |
| Tabel 4.11. | Pengaruh Nilai α terhadap Recognition Rate NNE (5 jaringan) untuk Dataset Citra Wajah Inframerah | 68 |
| Tabel 4.12. | Hasil Recognition Rate tiap kelas untuk Pengenalan Data Testing Citra Wajah Inframerah | 71 |
| Tabel 4.13. | Ringkasan Hasil Pelatihan dengan Fungsi Error Kuadratis untuk Citra Cahaya Tampak | 71 |

| | | |
|-------------|--|-----|
| Tabel 4.14. | Ringkasan Hasil Pelatihan dengan Fungsi Eror Cross-entropy untuk Citra Cahaya Tampak..... | 71 |
| Tabel 4.15. | Perbandingan Eror, Epoch, dan Waktu Pelatihan untuk Citra Cahaya Tampak | 72 |
| Tabel 4.16. | Perbandingan Recognition Rate NN Tunggal dan NNE untuk Citra Cahaya Tampak | 72 |
| Tabel 4.17. | Pengaruh Nilai α terhadap Recognition Rate NNE (3 jaringan) untuk Dataset Citra Wajah Cahaya Tampak | 73 |
| Tabel 4.18. | Pengaruh Nilai α terhadap Recognition Rate NNE (5 jaringan) untuk Dataset Citra Wajah Cahaya Tampak | 74 |
| Tabel 4.19. | Hasil Recognition Rate tiap kelas untuk Pengenalan Data Testing Citra Wajah Cahaya Tampak | 76 |
| Tabel 4.20. | Perbandingan Eror Hasil Pelatihan NN Tunggal dan NNE untuk Dataset Wajah | 77 |
| Tabel 4.21. | Perbandingan Epoch Hasil Pelatihan NN Tunggal dan NNE untuk Dataset Wajah | 77 |
| Tabel 4.22. | Perbandingan Waktu Pelatihan NN Tunggal dan NNE untuk Dataset Wajah | 77 |
| Tabel 4.23. | Perbandingan Recognition Rate Data Training NN Tunggal dan NNE untuk Dataset Wajah | 78 |
| Tabel 4.24. | Perbandingan Recognition Rate Data Testing NN Tunggal dan NNE untuk Dataset Wajah | 78 |
| Tabel 5.1. | Perbandingan Eror Hasil Pelatihan NN Tunggal dan NNE untuk Dataset Odor | 89 |
| Tabel 5.2. | Perbandingan Jumlah Epoch Hasil Pelatihan NN Tunggal dan NNE untuk Dataset Odor | 89 |
| Tabel 5.3. | Perbandingan Waktu Pelatihan NN Tunggal dan NNE untuk Dataset Odor | 89 |
| Tabel 5.4. | Perbandingan Recognition Data Training NN Tunggal dan NNE untuk Dataset Odor | 89 |
| Tabel 5.5. | Perbandingan Recognition Data Testing NN Tunggal dan NNE untuk Dataset Odor | 89 |
| Tabel 5.6. | Ringkasan Hasil Recognition Rate tiap kelas untuk Pengenalan Data Training Odor | 95 |
| Tabel 5.7. | Ringkasan Hasil Recognition Rate tiap kelas untuk Pengenalan Data Testing Odor | 96 |
| Tabel 6.1. | Perbandingan Eror Hasil Pelatihan NN Tunggal dan NNE untuk Seluruh Dataset | 101 |
| Tabel 6.2. | Perbandingan Jumlah Epoch Hasil Pelatihan NN Tunggal dan NNE untuk Seluruh Dataset | 103 |
| Tabel 6.3. | Perbandingan Waktu Pelatihan Pelatihan NN Tunggal dan NNE untuk Seluruh Dataset | 105 |

| | | |
|------------|---|-----|
| Tabel 6.4. | Perbandingan Recognition Rate NN Tunggal dan NNE untuk Seluruh Dataset | 108 |
| Tabel 6.5. | Ringkasan Peningkatan Recognition Rate NN Tunggal dan NNE Berbasis Fungsi Error Cross Entropy terhadap NN Tunggal dan NNE Berbasis Fungsi Error Kuadratis | 110 |



DAFTAR GAMBAR

| | | |
|--------------|--|----|
| Gambar 1.1. | Tahapan Penelitian..... | 6 |
| Gambar 1.2. | Diagram Alir Sistem Pengenal Pola | 10 |
| Gambar 1.3. | Penelitian Sistem Pengenal Pola | 11 |
| Gambar 1.4. | Sistematika Laporan Penelitian | 12 |
| Gambar 2.1. | Susunan Syaraf Manusia..... | 14 |
| Gambar 2.2. | Struktur Jaringan Syaraf Tiruan | 16 |
| Gambar 2.3. | Arsitektur Neural Network Ensemble (NNE) | 22 |
| Gambar 2.4. | Fungsi Error Kuadratis dan Turunannya..... | 32 |
| Gambar 2.5. | Fungsi Error <i>Cross Entropy</i> dan Turunannya | 33 |
| Gambar 2.6. | Sudut Pandang Pengenal Pola | 35 |
| Gambar 2.7. | Contoh Pengklasifikasian Linear dengan Dua Dimensi | 36 |
| Gambar 3.1. | Diagram Skema Sistem Pengenal Pola | 43 |
| Gambar 3.2. | Distribusi Nilai Eigen 9 Dataset UCI | 44 |
| Gambar 3.3. | Perbandingan Error, Epoch, Waktu Pelatihan NN Tunggal dan NNE untuk Dataset UCI | 50 |
| Gambar 3.4. | Recognition Rate NN Tunggal dan NNE untuk Dataset UCI | 50 |
| Gambar 4.1. | Diagram Skema Sistem Pengenal Wajah | 56 |
| Gambar 4.2. | Sketsa dan Desain Kamera CCD Inframerah | 58 |
| Gambar 4.3. | Foto Inframerah 40x30 pixel yang Dituliskan ke dalam Bentuk Matriks | 58 |
| Gambar 4.4. | Matriks Gambar Berukuran 40x30 pixel yang Disusun menjadi Matriks Berukuran 1x1200 | 58 |
| Gambar 4.5. | Pembentukan Matriks Dataset Wajah Inframerah Berukuran 200x1200 (200 sampel x 1200 dimensi)..... | 59 |
| Gambar 4.6. | Distribusi Nilai Eigen Dataset Wajah Inframerah | 59 |
| Gambar 4.7. | Foto Cahaya Tampak 30x30 pixel yang Dituliskan ke dalam Bentuk Matriks | 60 |
| Gambar 4.8. | Matriks Gambar Berukuran 30x30 pixel yang Disusun menjadi Matriks Berukuran 1x900 | 60 |
| Gambar 4.9. | Pembentukan Matriks Dataset Wajah Cahaya Tampak Berukuran 100x900 (100 sampel x 900 dimensi)..... | 60 |
| Gambar 4.10. | Distribusi Nilai Eigen Dataset Wajah Cahaya Tampak | 61 |
| Gambar 4.11. | Distribusi Nilai Eigen Dataset Wajah Inframerah (yang telah dimodifikasi) | 64 |
| Gambar 4.12. | Perbandingan Error, Epoch, Waktu Pelatihan NN Tunggal dan NNE untuk Citra Wajah Inframerah..... | 66 |
| Gambar 4.13. | Perbandingan Recognition Rate NN Tunggal dan NNE untuk Citra Wajah Inframerah | 66 |
| Gambar 4.14. | Pengaruh α terhadap Recognition Rate Data Testing NNE (3 jaringan) untuk Dataset Citra Wajah Inframerah | 68 |

BAB I

PENDAHULUAN

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

1.1. Latar Belakang

Artificial neural network (ANN; untuk selanjutnya dalam skripsi ini akan disebut sebagai *neural network* atau NN) atau Jaringan Syaraf Tiruan (JST) adalah paradigma pemrosesan suatu informasi yang terinspirasi oleh sistem sel syaraf biologi, sama seperti otak yang memproses suatu informasi. Elemen mendasar dari paradigma tersebut adalah struktur yang baru dari sistem pemrosesan informasi. *Neural network* (NN), seperti manusia, belajar dari suatu contoh. NN dibentuk untuk memecahkan suatu masalah tertentu seperti pengenalan pola atau klasifikasi karena proses pembelajaran.

NN keluar dari penelitian kecerdasan buatan, terutama percobaan untuk menirukan *fault-tolerance* dan kemampuan untuk belajar dari sistem syaraf biologi dengan model struktur *low-level* dari otak. Otak terdiri dari sekitar 10 milyar (10.000.000.000) neuron sel syaraf yang saling berhubungan.

Perkembangan penelitian tentang NN yang begitu pesat sejak ditemukannya NN dengan algoritma *Backpropagation* [1] menghasilkan banyak eksperimen dalam rangka menyelesaikan permasalahan-permasalahan di dunia. Para ilmuwan dan insinyur terkemuka seperti Widrow (1962), Hopfield (1985), Grossberg (1987), dan Kohonen (1984, 1988) turut menyumbangkan berbagai penemuan menarik mengenai arsitektur serta aplikasi NN yang mendemonstrasikan potensi dari teknologi NN.

Salah satu metode NN yang masih terus dikembangkan di dunia saat ini adalah *Neural Network Ensemble* (NNE). Skripsi Stephen Roy Imantaka (2010) menerapkan metode NNE ini dan mencoba menggunakannya untuk mengenali wajah (*face recognition*) dalam citra yang diambil dari kamera infra merah. Dengan NNE yang teroptimasi, Stephen Roy Imantaka berhasil membuat sistem pengenalan wajah dengan akurasi mencapai 99.9% [2].

Berawal dari penelitian Stephen Roy Imantaka di atas, penulis tergerak untuk melakukan serangkaian penelitian dalam rangka membuat sistem pengenalan pola (*pattern recognition*) berbasis NNE dengan metode pembelajaran *Negative Correlated Learning* (NCL) dengan akurasi yang lebih baik. Dalam skripsi ini penulis mencoba memperbaiki kemampuan NNE dengan menerapkan fungsi *cross-entropy error* (CEE) yang memiliki keunggulan dibandingkan fungsi kuadratis atau *mean square error* (MSE) [3].

Pada penelitian Penulis memakai dua jenis dataset, yaitu dataset yang diambil dari *database* University of California at Irvine (UCI) repository [12] serta dataset citra wajah inframerah, citra wajah cahaya tampak dan campuran aroma yang diambil dari penelitian NN yang telah dilakukan sebelumnya [2, 29]. Database UCI ini telah banyak digunakan oleh berbagai penelitian di seluruh dunia terutama yang berkaitan dengan NN dan *machine learning*. Dengan demikian, penulis berharap kelak di masa depan hasil penelitian ini dapat menjadi *benchmark* bagi para peneliti lain yang menaruh minat serupa dalam bidang NN dan NNE untuk pengenalan pola.

Pada penelitian ini, Penulis membuat sistem pengenalan wajah dengan NN tunggal dan NNE. NNE mengkombinasikan beberapa NN untuk mengambil keputusan. Penulis membandingkan hasil performa jaringan dengan pendekatan fungsi eror kuadratis dan *cross entropy*. Sistem pengenalan pola meliputi 3 tahap pemrosesan, yaitu pra-pemrosesan (*pre-processing*), pembelajaran (*training*), dan pengujian (*testing*). Tahap pra-pemrosesan dilakukan dengan menggunakan membaca seluruh data kemudian menormalisasinya, sedangkan tahap pembelajaran dan pengujian menggunakan algoritma *backpropagation*.

NNE yang dilatih dengan NCL memiliki kemampuan generalisasi yang lebih baik dibanding sekedar NN tunggal [11]. Sedangkan fungsi eror *cross entropy* memiliki kelebihan berupa kemampuan generalisasi yang lebih baik dan jumlah epoch yang lebih sedikit dibandingkan dengan fungsi eror kuadratis [22,23,65]. Melalui penelitian ini penulis ingin mempelajari dan memahami karakteristik NNE yang dilatih dengan fungsi eror *cross-entropy* dengan terhadap kemampuan NN dan NNE yang dilatih dengan fungsi kuadratis dalam mengenali pola.

1.2. Pernyataan Masalah

Masalah yang akan dibahas pada laporan skripsi ini adalah studi karakteristik NNE dengan algoritma pembelajaran NCL berbasis fungsi eror *cross entropy* serta fungsi eror kuadratis untuk diterapkan dalam suatu sistem pengenalan pola.

1.3. Tujuan Penelitian

Berikut ini adalah tujuan penelitian yang ingin dilakukan:

1. Memahami dan menganalisis bagaimana NN tunggal dengan algoritma *backpropagation* serta NNE dengan skema pembelajaran NCL dapat digunakan untuk sistem pengenalan pola (*pattern recognition system*) yang dapat mempelajari pola yang telah ada dan mengklasifikasi pola tersebut ke dalam kelas-kelas.
2. Memahami dan menganalisis bagaimana NN tunggal dengan algoritma *backpropagation* serta NNE dengan skema pembelajaran NCL 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.
3. Membandingkan performa sistem pengenalan wajah untuk citra inframerah dan citra cahaya tampak
4. Memahami dan menganalisis bagaimana NN tunggal dengan algoritma *backpropagation* serta NNE dengan skema pembelajaran NCL dapat digunakan untuk sistem pengenalan aroma (*odor recognition system*) yang dapat mempelajari pola campuran aroma yang telah ada dan mengenali suatu campuran aroma yang lain
5. Membandingkan performa NN tunggal dengan NNE dalam mengenali berbagai pola dataset penelitian.
6. Membandingkan karakteristik dan tingkat pengenalan antara NN yang dilatih dengan fungsi eror kuadratis dengan NN yang dilatih dengan fungsi eror *cross entropy*.

1.4. Batasan Masalah

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

1. Penulis menggunakan 12 buah dataset yang terdiri dari 9 dataset pola yang didapat dari “*UCI Repository of Machine Learning Database*” [12], 2 dataset citra wajah frontal yang diambil dari kamera infra merah dan kamera cahaya tampak, serta 1 dataset campuran 3 aroma (jeruk, kenanga, melati) dengan alkohol berbagai konsentrasi. (Keterangan mendetail mengenai masing-masing data set terdapat pada bagian Lampiran).
2. Perbandingan antara jumlah data pelatihan (*data training*) dan jumlah data pengujian (*data testing*) adalah sama besar (1:1).
3. NN dilatih dengan algoritma standar *Backpropagation*. Sedangkan NNE dilatih menggunakan algoritma *Negative Correlated Learning* (NCL).
4. Setiap jaringan dalam NNE diberi bobot awal yang berbeda-beda dan dilatih secara terpisah hingga terbentuk bobot akhir yang berbeda-beda. Kemudian, pengambilan keputusan NNE dilakukan dengan metode rata-rata (*simple averaging*)
5. Pada saat pelatihan NNE, tidak digunakan nilai λ (konstanta penalti dalam metode NCL), melainkan digunakan parameter $\gamma = 0.5$ (besar parameter kekuatan penalti untuk skema NCL) [10], dimana berlaku:
$$\gamma = \lambda \left[\frac{\text{jumlah jaringan}}{2(\text{jumlah jaringan}-1)} \right]$$
 [32].
6. Selama penelitian digunakan variabel-variabel sebagai berikut ini:
 - a) Jumlah unit *input* neuron sebanyak dimensi masing-masing data set
 - b) Jumlah unit *hidden* neuron sebanyak setengah unit input neuron
 - c) Jumlah unit *output* neuron sebanyak kelas masing-masing data set
 - d) Inisialisasi bobot menggunakan metode Nguyen-Widrow
 - e) Epoch maksimum sebesar 20000 (kecuali disebutkan secara berbeda)
 - f) Nilai laju pembelajaran (*learning rate*) α sebesar 0.1
 - g) Nilai laju momentum μ sebesar 0.2
7. Proses pelatihan akan berhenti jika salah satu dari syarat berikut ini terpenuhi:
 - a) epoch telah mencapai epoch maksimum yang ditetapkan yaitu sebanyak 20.000 epoch; atau

- b) NN telah mencapai keadaan konvergen, dengan demikian errornya telah mencapai error minimum yang ditetapkan yaitu 0.01 .
8. Parameter keberhasilan dihitung berdasarkan tingkat *recognition rate* (RR) atau derajat pengenalan. RR merupakan persentase banyaknya data yang diklasifikasi secara benar oleh NN terhadap total data secara keseluruhan. Terdapat tiga RR yang digunakan yaitu apabila NN mampu mengenali paling tidak 65%, 75%, atau 85% dari target yang ingin dicapai. Yang dimaksud RR rata-rata dalam penelitian ini mengacu pada hasil rata-rata dari ketiga RR tersebut.
 9. Semua komputasi NN dilakukan dengan membuat program dalam format *editor file m.file* pada *software matlab 2009a*.

1.5. Manfaat Penelitian

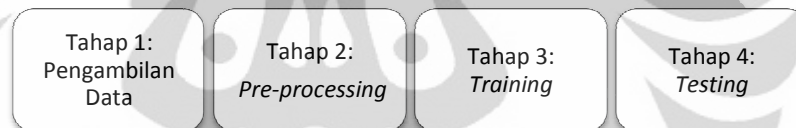
Berikut ini adalah manfaat penelitian pada skripsi ini:

1. Penelitian diharapkan agar pembaca dapat memahami penggunaan NN tunggal dengan algoritma *backpropagation* dan NNE sebagai sarana untuk mengenali dan mengklasifikasi pola.
2. Penelitian diharapkan agar pembaca dapat memahami penggunaan algoritma NCL dalam mengoptimalkan proses pembelajaran pada NNE.
3. Penelitian dapat digunakan sebagai bahan pembelajaran untuk peneliti yang ingin mempelajari sistem pengenalan pola berbasis NN secara umum ataupun sistem pengenalan pola dengan menggunakan dataset UCI.
4. Penelitian dapat digunakan sebagai bahan pembelajaran untuk peneliti yang ingin mempelajari sistem pengenalan wajah berbasis NN secara umum ataupun sistem pengenalan wajah dengan menggunakan citra inframerah dan citra cahaya tampak
5. Penelitian dapat digunakan sebagai bahan pembelajaran untuk peneliti yang ingin mempelajari sistem pengenalan aroma berbasis NN secara umum.
6. Penelitian diharapkan agar pembaca dapat memahami penggunaan fungsi eror *cross entropy* dan kuadratis untuk menghitung eror dalam proses pelatihan NNE.
7. Penelitian dapat digunakan sebagai bahan pembelajaran untuk peneliti yang ingin mempelajari karakteristik fungsi eror *cross entropy* dan kuadratis, serta

sebagai acuan atau referensi bagi penelitian tentang pengaruh pemilihan fungsi eror terhadap performa NNE.

8. Penelitian dapat dijadikan acuan atau referensi untuk pengembangan aplikasi sistem pengenalan pola (*pattern recognition*) dalam berbagai bidang.
9. Penelitian dapat dijadikan acuan atau referensi untuk pengembangan aplikasi sistem pengenalan wajah (*face recognition*) dalam berbagai bidang, misalnya sistem sekuriti, absensi pegawai atau pelajar, dan sebagainya.
10. Penelitian dapat dijadikan acuan atau referensi untuk pengembangan aplikasi sistem pengenalan aroma (*odor recognition*) dalam berbagai bidang, misalnya sistem pendeteksi narkoba, sistem pendeteksi kebocoran gas, maupun dalam aplikasi kontrol kualitas dalam industri makanan, minuman, parfum, dan sebagainya.

1.6 Tahapan Penelitian



Gambar 1.1 Tahapan Penelitian

1) Tahap Pengambilan data

Proses pengambilan data dilakukan dengan mendownload data “*UCI Repository of Machine Learning Database*” [12]. Terdapat 9 jenis data yang akan digunakan dalam penelitian ini yaitu data *balance scale, breast cancer, BUPA, credit, glass, heart disease, iris, sonar, vowel*. Data citra wajah frontal diambil dari kamera infra merah dan kamera cahaya tampak. Sedangkan data aroma didapatkan dari 8 buah sensor resonator kuarsa yang dilapisi membrane sensitif.

2) Tahap Pra-Pemrosesan (*Pre-Processing*)

Pada tahap pra-pemrosesan ini, data disusun ke dalam sebuah tabel dalam file Excel dengan banyak kolom sebanyak parameter dan jumlah baris sebanyak jumlah data. Data selanjutnya akan dibaca dari file Excel (dengan ekstensi .xls) tersebut melalui software Matlab

Setelah data dibaca dan disimpan dalam bentuk matriks, kemudian diterapkan *Principal Component Analysis* (PCA) dengan tujuan untuk mereduksi jumlah dimensi dengan tetap mempertahankan sebanyak mungkin varians pada data. Selanjutnya data dibagi dua dengan jumlah yang sama besar (perbandingan 1:1) untuk dijadikan data *training* (digunakan untuk tahap pelatihan) dan data *testing* (digunakan untuk tahap pengujian).

3) Tahap Pelatihan (*Training*)

Setelah melalui tahap pra-pemrosesan, tahap selanjutnya yang merupakan bagian utama sistem pengenalan pola adalah tahap pelatihan. NNE akan dilatih menggunakan algoritma NCL untuk mengenali pola-pola tersebut, yang disebut dengan tahap pelatihan (*training*).

Sistem dilatih untuk mengenali pola input database agar menghasilkan output yang sesuai dengan target tertentu. NN 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_i (*output layer*). Jumlah unit atau neuron pada lapisan input adalah sesuai dengan dimensi pada masing-masing data set.

NN ini menggunakan algoritma *backpropagation*, pada setiap pelatihan, terdapat 2 tahapan, yaitu propagasi maju (*feed forward*) dan propagasi balik (*backpropagation*). Modifikasi nilai bobot pada neuron-neuron terjadi pada tahap propagasi balik. Nilai bobot antara setiap neuron akan dimodifikasi terus-menerus sampai sistem ini menemukan bobot akhir yang konvergen yang mampu menggeneralisasi nilai inputan apapun dan membawa hasil akhir komputasi kepada salah satu kelas yang ada. Proses propagasi maju dan propagasi 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*) yaitu satu kali pelatihan dan nilai eror (apakah eror telah mencapai batas toleransi yang diizinkan).

Pada tahap ini NNE dilatih dengan menggunakan dua fungsi kesalahan yang berbeda, yaitu metode MSE dan CEE. Untuk membuktikan bahwa metode NNE lebih baik untuk kasus klasifikasi pola, penulis juga membandingkan performa NN tunggal dengan NNE yang terdiri dari 3 buah dan 5 buah untuk mengetahui hubungan antara penambahan *cost* (dengan bertambahnya jumlah jaringan) dan peningkatan performa yang dihasilkan oleh NN dalam mengenali pola. Setiap jaringan diberi bobot awal yang berbeda-beda dan dilatih secara bersama-sama (*dependent*) dengan algoritma NCL hingga terbentuk bobot akhir yang berbeda-beda. Pengambilan keputusan dilakukan dengan metode rata-rata (*simple averaging*). Setelah proses pelatihan selesai, jaringan ensemble tersebut digunakan untuk mengenali pola uji.

4) Tahap Pengujian (*Testing*)

Setelah selesai melakukan proses *training*, dilakukan pengujian atau verifikasi dengan menggunakan data *training* (yang dijadikan database penelitian) serta data *testing* (yang berada di luar database pelatihan). Proses ini disebut dengan tahap pengujian (*testing*).

Ketika pelatihan selesai dilakukan, jaringan kini memiliki bobot-bobot akhir yang telah konvergen dapat digunakan untuk pengenalan pola. Proses yang dilakukan untuk pengenalan pola ini hanyalah propagasi maju. Tahap pengujian ini berupa verifikasi untuk menguji kemampuan NN dalam mengklasifikasikan suatu pola ke dalam kelas tertentu yang menjadi targetnya.

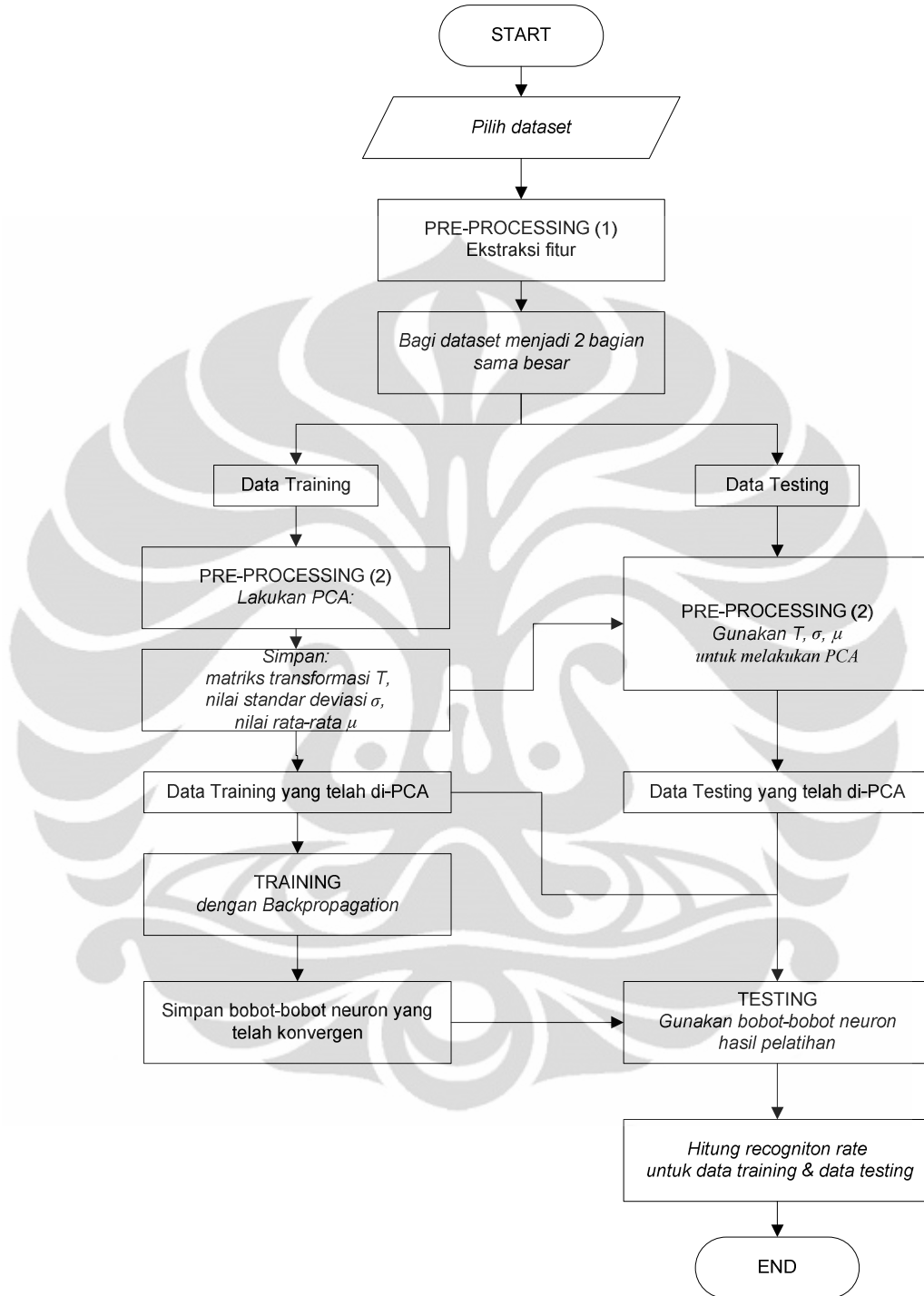
Dalam tahap pengujian, setiap data uji akan terlebih dahulu dijadikan ke dalam bentuk matriks seperti pada tahap *pre-processing*, kemudian dijalankan ke dalam NN tetapi hanya sampai pada tahap propagasi maju saja dengan menggunakan nilai bobot akhir yang telah konvergen dari hasil

pelatihan sebelumnya. Kemudian vektor data hasil perhitungan akhir dibandingkan dengan tiap target.

Output NN yang salah akan bernilai 0, sedangkan output yang benar (sesuai dengan target) akan bernilai 1. Dengan demikian *recogniton rate* pun dapat dihitung dengan memberikan toleransi sebesar ± 0.15 , ± 0.25 , serta ± 0.35 . Atau dengan kata lain, misalnya untuk toleransi sebesar ± 0.15 , maka output akan bernilai benar jika berada dalam range 0.85-1.00 dan bernilai salah jika berada dalam range 0.0-0.15. Demikian seterusnya, sehingga secara total kita menggunakan 3 buah limit yaitu limit 0.65, 0.75, dan 0.85.

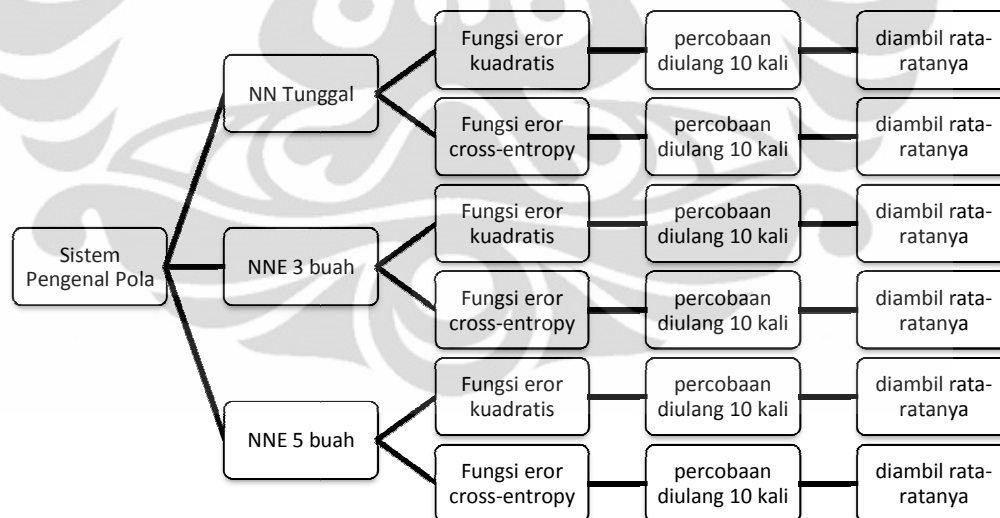
Untuk satu kali testing, persentase kesesuaian terhadap target, yang disebut *recognition rate* pun dihitung. *Recognition rate* merupakan persentase banyaknya data uji yang terklasifikasi secara benar oleh NN terhadap total keseluruhan data.

$$\text{Recognition rate} = \frac{\text{jumlah data yang terklasifikasi benar}}{\text{jumlah data total}} \times 100\% \quad (1.1)$$



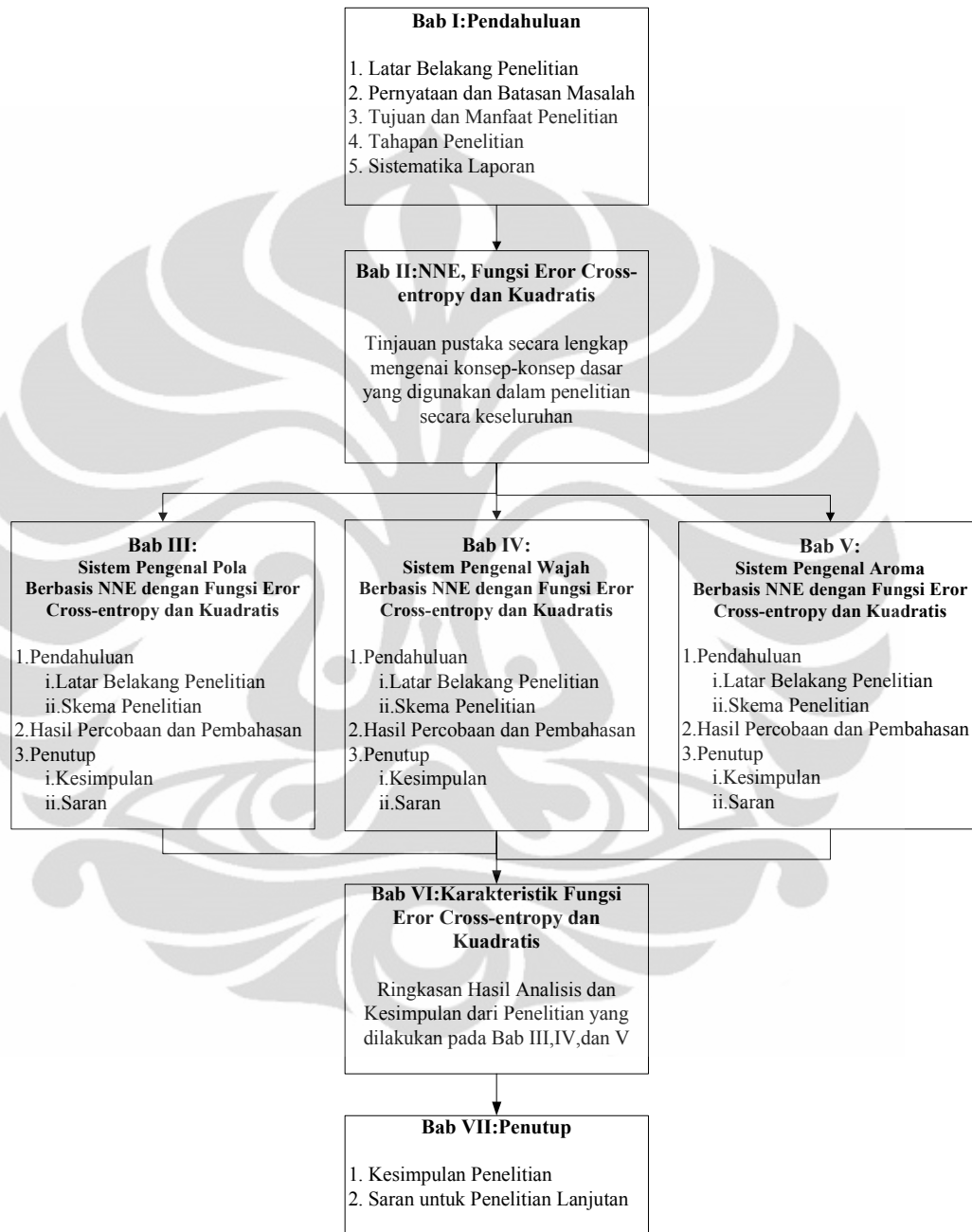
Gambar 1.2. Diagram Alir Sistem Pengenal Pola

Penelitian pada bab III, IV, dan V dilakukan dengan membangun sebuah sistem pengenalan pola untuk mendeteksi pola dataset yang berbeda-beda dengan menggunakan NN tunggal, NNE sejumlah 3 buah jaringan, dan NNE sejumlah 5 buah jaringan. Untuk masing-masing skema di atas (skema NN Tunggal, NNE 3 buah jaringan, NNE 5 buah jaringan), tiap jaringan tersebut akan dilatih menggunakan fungsi eror kuadratis dan fungsi eror *cross-entropy* secara terpisah. Tujuannya untuk membandingkan karakteristik NN dan NNE yang dilatih menggunakan fungsi eror kuadratis dan fungsi eror *cross-entropy* serta membandingkan performa jaringan mana yang lebih baik dilihat dari derajat pengenalan (*recognition rate*) yang dicapai. Untuk masing-masing fungsi eror di atas, pelatihan dilakukan sebanyak 10 kali percobaan kemudian diambil hasil rata-ratanya. Hal ini dilakukan karena hasil pelatihan memiliki nilai fluktuasi akibat inisialisasi bobot awal yang menggunakan nilai random, dengan mengambil rata-rata dari sepuluh percobaan diharapkan hasilnya cukup representatif.



Gambar 1.3. Penelitian Sistem Pengenal Pola

1.7. Sistematika Laporan Penelitian



Gambar 1.4. Sistematika Laporan Penelitian

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

Bab II, Dasar Teori, berisi tentang keseluruhan landasan konsep yang digunakan dalam penelitian, diantaranya yaitu konsep mengenai pengenalan pola (*pattern recognition*), konsep dasar NN dengan algoritma *backpropagation*, NNE dengan algoritma NCL, serta fungsi eror kuadratis dan *cross-entropy*.

Bab III, berisi hasil penelitian Sistem Pengenal Pola Berbasis NNE. Dalam penelitian ini digunakan dataset berupa 9 pola database UCI. Dalam penelitian ini akan dibahas mengenai performa NN tunggal serta NNE sebanyak 3 dan 5 buah yang dilatih dengan menggunakan fungsi eror kuadratis dan *cross-entropy*.

Bab IV, berisi hasil penelitian Sistem Pengenal Wajah Berbasis NNE. Dalam penelitian ini digunakan dua buah dataset yaitu *Face Infra-red* dan *Face Visible-light*, yaitu citra wajah frontal yang diambil dengan kamera infra merah dan kamera cahaya tampak. Bab ini bertujuan untuk membandingkan performa NN untuk mengenali citra inframerah dan citra cahaya tampak tersebut. Sama seperti bab sebelumnya, dalam penelitian ini akan dibahas mengenai performa NN tunggal serta NNE sebanyak 3 dan 5 buah yang dilatih dengan menggunakan fungsi eror kuadratis dan *cross-entropy*.

Bab V, berisi hasil penelitian Sistem Pengenal Aroma Berbasis NNE. Dalam penelitian ini digunakan dataset *Odor* yang berisi data yang diperoleh dari 8 sensor untuk campuran 3 macam aroma. Dalam penelitian ini akan dibahas mengenai performa NN tunggal serta NNE sebanyak 3 dan 5 buah yang dilatih dengan menggunakan fungsi eror kuadratis dan *cross-entropy* dalam mengenali campuran aroma tersebut.

Bab VI, Analisis perbandingan karakteristik fungsi eror *cross-entropy* dan kuadratis ditinjau dari eror, jumlah epoch, waktu pelatihan, dan derajat pengenalan. Bab ini berisi ringkasan hasil penelitian dari bab III hingga bab V.

Bab VII, Penutup. Bab ini berisi kesimpulan dari keseluruhan percobaan dan saran untuk penelitian lanjutan di masa depan.

BAB II

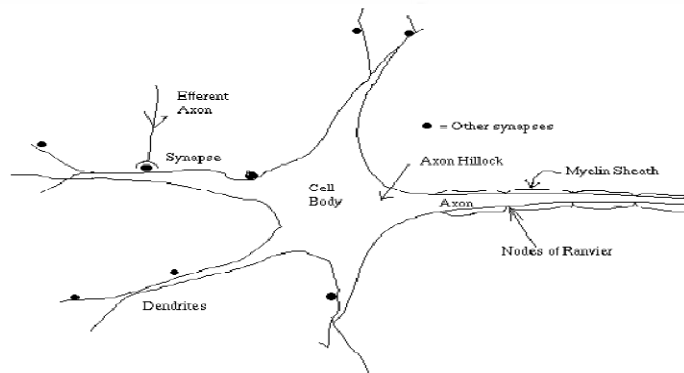
NEURAL NETWORK ENSEMBLES, FUNGSI EROR CROSS ENTROPY DAN KUADRATIS

Bab ini menjelaskan tentang beberapa teori dasar yang penulis gunakan dalam membangun sistem pengenalan pola, diantaranya mengenai konsep dasar *neural network* (NN) dengan algoritma standar *backpropagation*, *neural network ensemble* (NNE) dengan algoritma *Negative Correlated Learning* (NCL) yang merupakan perluasan dari algoritma *backpropagation*, fungsi eror *cross entropy* dan kuadratis, serta pembelajaran pada mesin (*machine learning*) untuk mengenali pola..

2.1. Neural Network dengan Algoritma Backpropagation

Otak manusia tersusun oleh lebih dari 100 jenis sel yang disebut neuron. Neuron-neuron tersebut terbagi-bagi ke dalam kumpulan yang saling terhubung yang disebut dengan jaringan (*network*). Setiap jaringan tersusun dari ribuan neuron yang terhubung satu sama lain. Dengan demikian, kita dapat menganggap bahwa otak merupakan kumpulan dari sel syaraf (*neural network*).

Sel syaraf mempunyai cabang struktur input (*dendrites*), sebuah inti sel dan percabangan struktur output (*axon*). *Axon* dari sebuah sel terhubung dengan *dendrites* yang lain melalui sebuah *synapse*. Sebuah sel syaraf akan aktif (*fired*) jika mendapatkan sinyal yang telah memenuhi batasan tertentu yang sering disebut dengan nilai ambang (*threshold*). Ketika sebuah sel syaraf aktif (*fired*), kemudian menimbulkan suatu sinyal electrochemical pada axon. Sinyal ini melewati *synapses* menuju ke sel syaraf yang lain.



Gambar 2.1. Susunan Syaraf Manusia

Sebelum memahami cara kerja NN diperlukan pengetahuan tentang konsep dasar yang menyusun sebuah NN. Berikut ini adalah penjelasan dari konsep tersebut.

a. Elemen pemroses (*neuron*)

Elemen pemroses merupakan satuan dasar dari sebuah NN. Setiap elemen pemroses menerima suatu data masukan, mengolahnya, kemudian menghasilkan sebuah keluaran tunggal yang dapat berupa hasil akhir (*final output*) maupun sebagai sebuah masukan ke elemen pemroses yang lain.

b. Jaringan (*network*)

Sebuah NN merupakan kumpulan elemen pemroses yang terhubung satu sama lain dan biasanya terkelompok sebagai lapisan-lapisan (*layers*).

c. Lapisan masukan (*input layer*)

Setiap elemen pemroses yang terletak pada lapisan masukan akan menerima suatu nilai masukan (*input value*) berbentuk bilangan biner atau kontinu yang merupakan representasi sebuah pola data dari dunia luar (*outside world*).

d. Lapisan tersembunyi (*hidden layer*)

Pada *neural network* dengan arsitektur *multilayer*, lapisan tersembunyi tidak berhubungan langsung dengan dunia luar, tetapi lapisan ini menambah derajat kerumitan sehingga NN mampu menyelesaikan masalah yang lebih kompleks dan non-linear.

e. Lapisan keluaran (*output layer*)

Setiap elemen pemroses yang terletak pada lapisan keluaran akan menghasilkan suatu nilai keluaran yang merupakan solusi dari masalah yang diselesaikan oleh NN.

f. Bobot (*weight*)

Bobot pada NN merepresentasikan kekuatan relatif ataupun nilai matematis dari berbagai koneksi antar neuron yang mengirimkan data dari satu lapisan ke lapisan berikutnya. Bobot antar neuron memegang fungsi yang penting dalam NN sebab bobot-bobot tersebut merupakan cara bagi NN untuk belajar sehingga mampu menghasilkan keluaran yang diinginkan.

g. Fungsi Penjumlahan (*summation function*)

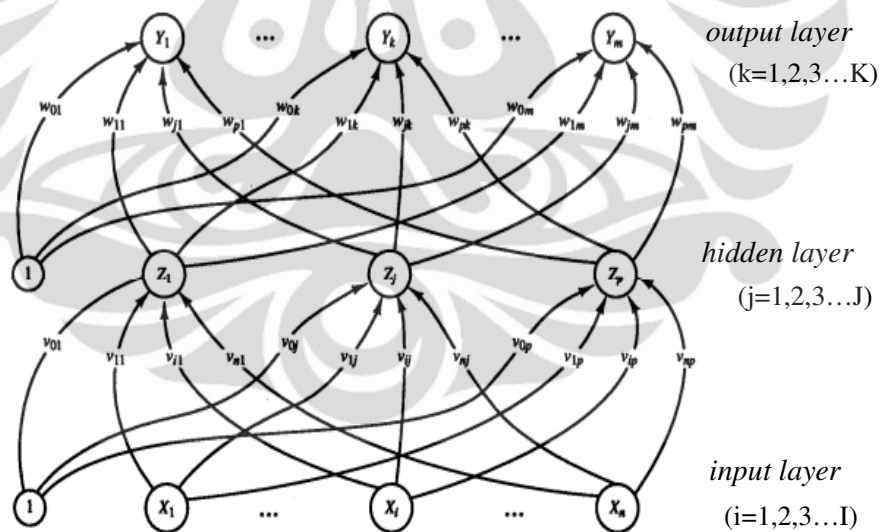
Fungsi penjumlahan merupakan rata-rata bobot dari seluruh elemen masukan. Sebuah fungsi penjumlahan sederhana akan mengalikan setiap nilai masukan (X_j) dengan bobotnya (W_{ij}) dan menjumlahkan totalnya sebagai jumlah bobot (S_i) dimana $S_i = \sum_j W_{ij} X_j$

h. Fungsi Aktivasi (*activation function*)

Salah satu fungsi aktivasi yang biasa digunakan adalah fungsi sigmoid yang dirumuskan sebagai $f(x) = \frac{1}{1 + e^{-x}}$

i. Pembelajaran (*learning*)

Terdapat dua macam metode pembelajaran yaitu (1) pembelajaran dengan pengarahannya (*supervised learning*) dimana pasangan input-output pada dataset diketahui dan bertindak sebagai “guru” dalam pembelajaran; dan (2) pembelajaran tanpa pengarahannya (*unsupervised learning*) dimana NN akan secara otomatis (*self-organize*) memilah-milah dataset ke dalam kategori-kategori.



Gambar 2.2. Struktur Jaringan Syaraf Tiruan

Mengadopsi esensi dasar dari prinsip kerja sistem syaraf biologi, jaringan syaraf tiruan dapat digambarkan sebagai berikut :

1. Elemen pemroses (*neurons*) menerima banyak sinyal masukan, baik dari data yang dimasukkan atau dari output sel syaraf pada jaringan syaraf.
2. Setiap sinyal yang diterima oleh sinapsis mempunyai sebuah bobot (*weight*). Elemen pemroses melakukan penjumlahan sinyal yang telah dikalikan dengan bobotnya.
3. Setiap elemen pemroses mempunyai sebuah nilai ambang (*threshold*). Apabila nilai masukan telah melampaui nilai tersebut, maka neuron akan aktif dan menyalurkan sebuah sinyal keluaran.
4. Sinyal keluaran akan menjaral melalui banyak cabang axon menuju ke neuron-neuron lain yang saling terhubung.

Terdapat tiga karakteristik penting yang perlu diperhatikan dalam mendesain sebuah NN yaitu [20]:

1. Arsitektur jaringan (*network architecture*); bagaimana hubungan koneksi antar neuron, jumlah lapisan yang digunakan serta jumlah neuron pada tiap lapisan. Dalam penelitian ini, digunakan arsitektur NN yang terdiri dari 3 lapisan yaitu lapisan input, lapisan tersembunyi, dan lapisan output.
2. Algoritma pembelajaran (*learning algorithm*); bagaimana NN belajar untuk mengubah bobot antar koneksi neuron, bagaimana neuron menghasilkan sinyal keluaran. Dalam penelitian ini digunakan algoritma pembelajaran *backpropagation*.
3. Fungsi aktivasi (*activation function*); bagaimana neuron menghitung penjumlahan dari informasi yang masuk yang menjadikan neuron tersebut dalam keadaan aktif. Pada umumnya setiap unit berbeda dalam suatu lapisan akan mempunyai fungsi aktivasi yang sama, dan fungsi aktivasi tersebut harus kontinu dan monoton tidak menurun [60]. Dalam penelitian ini digunakan fungsi sigmoid unipolar $f(net) = \frac{1}{1+e^{-net}}$, dengan net adalah hasil output komputasi setiap neuron.

Algoritma pelatihan *backpropagation* merupakan jenis pelatihan yang paling sering digunakan pada NN. Pelatihan *backpropagation* ini terdiri dari tiga tahap, yaitu tahap propagasi maju (*feedforward*), propagasi balik (*backward*), dan perubahan bobot.

Secara teoretis, NN dengan sebuah lapisan tersembunyi sudah cukup bagi *backpropagation* untuk mengenali sembarang perkawanan antara masukan dan target dengan tingkat ketelitian yang ditentukan. Penambahan lapisan tersembunyi tidak memberikan dampak yang signifikan selama proses pelatihan, serta meningkatkan kerumitan atau kompleksitas dalam perhitungan bobot, selain itu NN lebih cenderung untuk terjebak dalam *local minima* [13].

Algoritma *backpropagation* dapat dituliskan seperti berikut [4]:

1. Menentukan pola pada lapisan masukan (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}) serta bias (w_{0k} dan v_{0j}) 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.

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

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

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

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

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

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

Propagasi balik

6. Menghitung faktor δ neuron keluaran berdasarkan kesalahan di setiap neuron keluaran y_k . δ_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 α . Perubahan bobot ini akan digunakan untuk mengubah bobot w_{jk} .

$$\delta_k = (t_k - y_k) f'(y_{net_k}) = (t_k - y_k) y_k (1 - y_k) \quad (2.5)$$

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

7. Menghitung faktor δ neuron tersembunyi berdasarkan kesalahan di setiap neuron tersembunyi.

$$\delta_{net_j} = \sum_{k=1}^K \delta_k w_{jk} \quad (2.7)$$

$$\delta_j = \delta_{net_j} f'(z_{net_j}) = \delta_{net_j} z_j (1 - z_j) \quad (2.8)$$

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

Perubahan bobot

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

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

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

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

Terdapat beberapa faktor yang mempengaruhi keberhasilan dalam pembelajaran algoritma *backpropagation* diantaranya (1) inisialisasi bobot, (2) laju pembelajaran, dan (3) momentum [4].

Pertama, tentang inisialisasi bobot. Bobot awal akan mempengaruhi apakah NN mencapai titik minimum lokal (*local minima*) atau titik minimum global (*global minima*), dan seberapa cepat konvergensinya. Bobot yang menghasilkan nilai turunan aktivasi yang kecil sedapat mungkin dihindari karena akan menyebabkan perubahan bobotnya menjadi sangat kecil. Karena nilai bobot awal yang terlalu besar membuat nilai turunan fungsi aktivasinya menjadi sangat kecil juga, maka bobot dan bias awal sebaiknya diisi dengan bilangan acak kecil.

Nguyen dan Widrow mengusulkan metode inisialisasi bobot dan bias awal sehingga NN menghasilkan iterasi yang lebih cepat. Misalkan: n = jumlah unit masukan, p = jumlah unit tersembunyi, dan β = faktor skala = $0.7 \sqrt{p}$

Algoritma inisialisasi Nguyen-Widrow adalah sebagai berikut:

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

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

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

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

Kedua, mengenai laju pembelajaran. Laju pembelajaran (α) merupakan parameter NN yang berfungsi untuk mengendalikan proses penyesuaian bobot, nilainya bergantung pada kasus yang dihadapi. Laju pembelajaran yang terlalu kecil menyebabkan konvergensi NN menjadi lebih lambat. Namun laju pembelajaran yang terlalu besar dapat menyebabkan ketidakstabilan pada proses pelatihan.

Ketiga, yaitu mengenai momentum. Momentum digunakan untuk mempercepat pelatihan. Metode momentum melibatkan penyesuaian bobot ditambah dengan faktor momentum dari penyesuaian sebelumnya. Atau dengan kata lain, bobot baru pada waktu ke- $(t+1)$ didasarkan bobot pada waktu ke- (t) dan $(t-1)$. Jika μ adalah konstanta ($0 < \mu < 1$) yang menyatakan faktor momentum maka bobot baru dihitung berdasarkan persamaan:

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

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

Salah satu kekurangan utama NN dengan pelatihan *Backpropagation* adalah waktu pelatihan yang lama, maka pengurangan fitur dimensi dengan metode *Principal Component Analysis* (PCA) akan mempercepat proses pelatihan tanpa mengorbankan kekayaan informasi yang ada di dalam data [21].

PCA adalah suatu metode statistik yang dapat mengidentifikasi pola suatu data berdimensi tinggi, dan menyatakannya dengan memperhatikan kesamaan dan ketidaksamaannya[6]. Konsep utama dari PCA adalah mereduksi dimensi dari suatu set data yang terdiri dari banyak variabel ke dalam fitur dimensi-dimensi yang orthogonal, 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.

Secara matematis, PCA dilakukan melalui dekomposisi nilai eigen matriks data. PCA melakukan transformasi linear dari data X yang berdimensi D ke suatu data baru Y yang berdimensi d dan tidak berkorelasi, di mana $d \leq D$ [7]. 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 tahap *pre-processing* dalam pelatihan NN.

Namun demikian, salah satu kekurangan PCA yaitu bahwa PCA adalah salah satu prosedur pembelajaran tanpa pengarahan (*unsupervised learning*) yang tidak mempertimbangkan mengenai korelasi antara target output dengan dimensi input (*input features*). Sebagai tambahan, PCA adalah sebuah teknik reduksi dimensi secara linear, sehingga tidak sesuai untuk diterapkan pada suatu *problem* yang kompleks dengan struktur non-linear [51].

Langkah-langkah metode PCA adalah sebagai berikut [8]:

1. Ambil data yang akan diproses
2. Modifikasi nilai data sedemikian hingga nilai rata-ratanya sama dengan nol. Hal ini dapat dilakukan dengan mengurangi setiap elemen data dengan nilai rata-rata (*mean*) tiap dimensi atau dengan normalisasi nilai z .
3. Hitung nilai kovarians dari setiap dimensi yang satu dengan dimensi lainnya. Nilai-nilai kovarians ini kemudian dituliskan ke dalam suatu matriks kovarians.
4. Hitung vektor eigen dan nilai eigen dari matriks kovarians yang berpasangan dengannya. Semua vektor eigen suatu matriks adalah tegak lurus satu sama lain, dengan demikian data dapat diekspresikan ke dalam vektor-vektor eigen, alih-alih dalam sumbu koordinat.
5. Pilih *Principal Components* (PC) dari vektor eigen yang memiliki nilai eigen terbesar dari set data. Caranya adalah dengan mengurutkan nilai-nilai eigen dari yang terbesar hingga yang terkecil dan bentuklah matriks

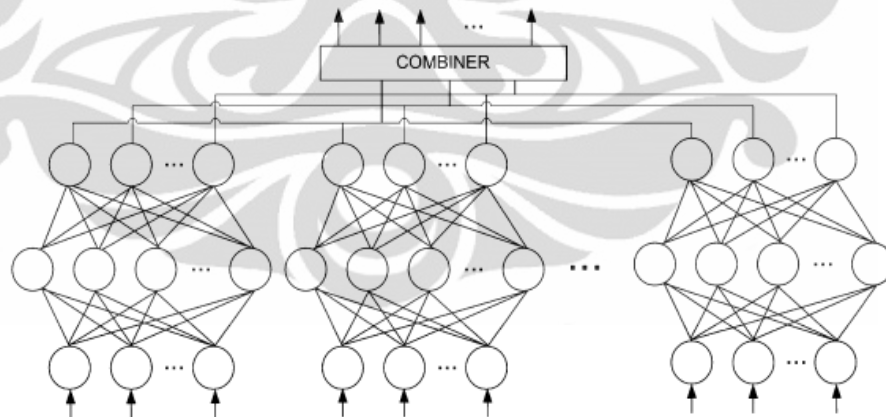
transformasi (*feature vector*) dengan memilih sejumlah PC yang memiliki nilai eigen terbesar.

6. Bentuk dataset yang baru berdasarkan PC yang telah dipilih dengan cara mengalikan dataset awal yang telah dinormalisasi dengan vektor-vektor eigen yang telah dipilih sebelumnya.

2.2. Neural Network Ensemble (NNE) dengan Algoritma NCL

Konsep *Neural Network Ensemble* (NNE) ini diajukan pada tahun 1990 oleh Lars Kai Hansen dan Peter Salamon. NNE mengkombinasikan beberapa NN tunggal untuk dilatih bersamaan dan mengambil keputusan bersama. Berdasarkan penelitian, NNE terbukti memiliki kemampuan generalisasi yang lebih baik daripada dalam mengenali pola dibandingkan dengan NN tunggal [10].

Sama seperti NN tunggal, setiap jaringan memiliki sebuah lapisan masukan, lapisan tersembunyi, dan lapisan keluaran. Sementara itu, pada NNE terdapat sebuah *combiner* yang berfungsi menentukan keputusan akhir yang didapat dari gabungan keputusan seluruh NN tunggal yang menyusun NNE tersebut. Untuk lebih jelasnya, perhatikan gambar 2.5. berikut ini.



Gambar 2.3. Arsitektur *Neural Network Ensemble* (NNE)

Serupa dengan prinsip jaringan syaraf individual, NNE juga menerapkan pembelajaran dengan pengarahannya (*supervised learning*). Tujuan utama dari pembelajaran dengan pengarahannya adalah untuk mengelompokkan pola (yang disebut dengan *instances*) ke dalam suatu kategori yang disebut

kelas atau label. Suatu set instances diberikan, atau biasa disebut *training set*. Ke dalam jaringan dimana label (kelas) dari masing-masing data pada *training set* juga diketahui. Dengan demikian, jaringan dibuat sedemikian hingga tercipta model baru yang dapat melabelkan (mengklasifikasikan) *instances-instances* baru (*testing set*) ke dalam kelas-kelas. Dalam hal ini jaringan sering disebut sebagai *classifier* [5]. 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. Namun demikian, tidak semua kumpulan (*crowds*) itu bijaksana. Surowiecki menyatakan untuk membuat kumpulan tersebut menjadi bijaksana, ada empat kriteria yang harus dipenuhi setiap kumpulan 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.

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

1. Metode rata-rata

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

2. Metode nilai maksimum

$$Y(x) = \max_{k=1,2,\dots,C} \left[\max[y_1(x), y_2(x), \dots, y_N(x)] \right] \quad (2.17)$$

3. Metode perkalian produk

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

4. Metode voting

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

Di mana C adalah jumlah kelas yang diurutkan di bawah kelas k pada jaringan ke- j . Perhitungan ini disebut dengan *Borda Count* [9].

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 NNE, 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 NNE yaitu dengan *Negative Correlated Learning* (NCL) [11].

Konsep di balik NCL bertujuan untuk melatih setiap jaringan individual berbeda di dalam NNE 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* yang terdapat pada fungsi errornya. NCL

mencoba melatih dan menggabungkan jaringan-jaringan individual pada suatu proses pelatihan yang sama [15].

Berikut ini adalah algoritma NCL. Pada awalnya, anggap sebuah *training set* yang didefinisikan sebagai berikut:

$$D = \{(\mathbf{x}(1), t(1), \mathbf{x}(2), t(2), \dots, \mathbf{x}(i), t(i))\} \quad (2.20)$$

di mana $\mathbf{x} \in R^p$ adalah input jaringan yang berupa vektor, t adalah output yang diharapkan (target) yang berupa skalar i 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- n memiliki keluaran y_n dan output dari keseluruhan jaringan pada set data ke- i disebut dengan $Y(i)$, yang dinyatakan sebagai berikut:

$$Y(i) = \left[\frac{1}{N} \sum_{n=1}^N y_n(i) \right] = \frac{1}{N} [y_1(i) + y_2(i) + y_3(i) + \dots + y_n(i)] \quad (2.21)$$

Pada persamaan (2.21), N 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:

$$E_n(i) = \sum_{i=1}^I \frac{1}{2} (y_n(i) - t(i))^2 + \sum_{i=1}^I \lambda p_n(i) \quad (2.22)$$

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

$$p_n(i) = [y_n(i) - Y(i)] \sum_{j \neq i}^N [y_n(i) - Y(i)] \quad (2.23)$$

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

$$\begin{aligned}
 \frac{\partial E_n(i)}{\partial y_n(i)} &= y_n(i) - t(i) + \lambda \frac{\partial p_n(i)}{\partial y_n(i)} \\
 &= y_n(i) - t(i) + \lambda \sum_{j \neq i} (y_n(i) - Y(i)) \\
 &= y_n(i) - t(i) + \lambda (NY(i) - y_n(i) - (N-1)Y(i)) \\
 &= y_n(i) - t(i) + \lambda (Y(i) - y_n(i)) \\
 &= y_n(i) - t(i) - \lambda (y_n(i) - Y(i)) \\
 &= (1 - \lambda)(y_n(i) - t(i)) + \lambda (Y(i) - t(i))
 \end{aligned} \tag{2.24}$$

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 halaman lampiran bagian. NCL dari persamaan tersebut tidak lain adalah perluasan sederhana dari algoritma *backpropagation*, yang hanya memerlukan perhitungan suku tambahan $\lambda(Y(i) - y_n(i))$ pada jaringan ke- n .

Algoritma pelatihan NNE metode NCL dengan dapat dituliskan seperti berikut:

1. Menentukan pola pada lapisan masukan (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 dengan metode Nguyen Widrow.
3. Tentukan kondisi *stopping*: epoh > epohMax atau errorTotal > 0.01
4. Selama kondisi *stopping* bernilai FALSE, maka lakukan Step 5-14 berikut ini.
5. Untuk setiap data yang ada, lakukan Proses Training sbb :

Komputasi Maju(Feed Forward):

Untuk setiap jaringan ($n = 1, 2, 3, \dots, N$) lakukan 6-8:

6. Setiap unit input (X_i , dimana $i = 1, 2, 3, \dots, I$):
 - a. menerima input x_i

- b. mengirimkannya ke semua unit layer di atasnya (*Hidden layer*).
7. Setiap unit hidden (Z_j , dimana $j = 1,2,3,\dots,J$)
- a. menghitung semua sinyal input dengan bobotnya :

$$z_net_j = v_{j0} + \sum_{i=1}^n x_i v_{ji} \quad (2.25)$$

- b. menghitung nilai aktivasi setiap unit hidden

$$z_j = f(z_net_j) = \frac{1}{1 + e^{-z_net_j}} \quad (2.26)$$

- c. mengirim nilai aktivasi sebagai input untuk unit output
8. Setiap unit output (Y_k , dimana $k = 1,2,3,\dots,K$)
- a. menghitung semua sinyal inputnya dengan bobotnya :

$$y_net_k = w_{k0} + \sum_{j=1}^n z_j w_{jk} \quad (2.27)$$

- b. menghitung nilai aktivasi setiap unit output sebagai output jaringan

$$y_k = f(y_net_k) = \frac{1}{1 + e^{-y_net_k}} \quad (2.28)$$

9. Ambil keputusan bersama dengan cara mengambil nilai rata-rata dari keluaran jaringan ke- n dimana $n = 1, 2, 3, \dots, N$.

$$Y = \left[\frac{1}{N} \sum_{n=1}^N y_n \right] = \frac{1}{N} [y_1 + y_2 + y_3 + \dots + y_n] \quad (2.29)$$

10. Mencari jumlah error NNE untuk 1 epoh.
- a. jika menggunakan fungsi eror kudratis, maka:

$$Error = \frac{1}{2}(Y - t)^2 \quad (2.30)$$

Ket: t = target, Y = keluaran NNE

- b. jika menggunakan fungsi eror cross entropy, maka:

$$Error = (-t)(\ln Y) - (1-t)(\ln(1-Y)) \quad (2.31)$$

Ket: t = target, Y = keluaran NNE

Perubahan Bobot (Backpropagation of error) :

Untuk setiap jaringan ke- n ($n = 1,2,3,\dots,N$) lakukan:

11. Setiap unit output (y_k , dimana $k = 1, 2, 3, \dots, K$) :

a. menerima pola target yang bersesuaian dengan pola input

b. menghitung informasi error :

i. jika menggunakan fungsi eror kudratis, maka:

$$\delta_k = [(1 - \lambda)(t - y_k) + \lambda(t - Y)]y_k(1 - y_k) \quad (2.32)$$

ii. jika menggunakan fungsi eror *cross entropy*, maka:

$$\delta_k = [(t - y_k) + \lambda(Y - y_k)]y_k(1 - y_k) \quad (2.33)$$

c. menghitung koreksi bobot unit output :

$$\Delta w_{jk} = \alpha \delta_k z_j + \mu \Delta w_{jk} \quad (2.34)$$

d. menghitung koreksi bias output :

$$\Delta w_{0k} = \alpha \delta_k + \mu \Delta w_{0k} \quad (2.35)$$

e. mengirimkan δ_k ke unit-unit yang ada pada layer di bawahnya

12. Setiap unit hidden (Z_j , dimana $j = 1, 2, 3, \dots, J$) :

a. menghitung semua koreksi error:

$$\delta_{in_j} = \sum \delta_k w_{jk} \quad (2.36)$$

b. menghitung nilai aktivasi koreksi error:

$$\delta_j = \delta_{in_j} f'(z_{in_j}) \quad (2.37)$$

c. menghitung koreksi bobot unit hidden :

$$\Delta v_{ij} = \alpha \delta_j x_i + \mu \Delta v_{ij} \quad (2.38)$$

d. menghitung koreksi error bias unit hidden:

$$\Delta v_{0j} = \alpha \delta_j + \mu \Delta v_{0j} \quad (2.39)$$

13. Update bobot pada unit output dan hidden

a. untuk setiap unit output (Y_k , dimana $k = 1, 2, 3, \dots, K$) : meng-update bobot dan biasnya (Z_j dimana $j = 0, 1, 2, \dots, J$)

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (2.40)$$

$$w_{0k}(\text{baru}) = w_{0k}(\text{lama}) + \Delta w_{0k} \quad (2.41)$$

b. untuk setiap unit hidden (Z_j dimana $j = 0, 1, 2, \dots, J$) : meng-update bobot dan biasnya (X_i dimana $i = 0, 1, 2, \dots, I$) :

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (2.42)$$

$$v_{0j}(\text{baru}) = v_{0j}(\text{lama}) + \Delta v_{0j} \quad (2.43)$$

14. Simpan bobot V_{ij} , V_{0j} , W_{jk} dan W_{0k} yang didapat dari proses training.
15. Hitung jumlah error total dan waktu training yang diperlukan.

Secara garis besar, terdapat 3 pendekatan dalam melatih NN tunggal dalam suatu NNE, yaitu *independent training*, *simultaneous training*, dan *sequential training* [34]. Dalam *independent training*, masing-masing NN tunggal dalam NNE dilatih secara *independent* untuk meminimalisasi eror antara keluaran NN dengan target yang ingin dicapai. Salah satu kelemahan utama dalam pendekatan ini adalah kurangnya interaksi yang terjadi selama proses pelatihan. Sehingga, NN tunggal mungkin saja menghasilkan output yang *positively correlated*, sehingga tidak berguna untuk memperbaiki performa NNE.

Dalam *sequential training*, NN tunggal dalam NNE dilatih secara berurutan satu per satu, tidak hanya untuk meminimalisasi eror antara keluaran NN, tapi juga untuk mendekorelasikan eror terhadap jaringan-jaringan tunggal yang terlebih dahulu *ditraining*. Dalam pendekatan ini, bobot antar koneksi dari terhadap jaringan-jaringan tunggal yang telah dilatih sebelumnya akan dibekukan ketika NN tunggal yang sekarang sedang dilatih. Sebagai hasilnya, melatih NN tunggal dalam NNE secara *sequential* tidak memberikan efek pada jaringan-jaringan tunggal yang telah dilatih pada waktu terdahulu. Dengan begitu, eror NN tunggal yang satu dengan yang lain tidak *negatively correlated*, padahal penelitian menunjukkan bahwa korelasi negatif antar NN tunggal dalam sebuah NNE penting untuk memperbaiki performa NNE.

Dalam *simultaneous training*, setiap NN tunggal dalam NNE akan dilatih secara simultan dengan jumlah epoch yang sama. Kelebihan dari pendekatan ini adalah metode ini mampu menghasilkan jaringan-jaringan tunggal yang bias dimana eror-erornya *negatively correlated* satu dengan yang lain. Kelemahan dari pendekatan ini adalah semua NN tunggal dalam NNE memperdulikan eror keseluruhan NNE, sehingga hal ini mungkin dapat menyebabkan adanya kompetisi antar NN tunggal dalam NNE. Selain itu,

untuk dapat memilih jumlah epoch yang tepat untuk semua NN tunggal dalam NNE merupakan suatu hal yang sulit untuk dilakukan.

Perlu dicatat bahwa tidak ada satupun dari ketiga pendekatan di atas yang telah dapat memecahkan isu mengenai desain arsitektur NNE. Sampai saat ini, jumlah NN tunggal serta bentuk topologi NN tunggal dalam suatu NNE biasanya ditetapkan terlebih dahulu ataupun dengan proses percobaan secara *trial-error*. Namun demikian, terdapat beberapa kriteria untuk membantu menentukan metode *ensemble* yang paling sesuai untuk tujuan (kebutuhan) tertentu. Kesulitan dalam memilih metode *ensemble* merupakan suatu problema pengambilan keputusan dengan banyak kriteria (*Multiple Criteria Decision Making/ MCDM*), artinya terdapat hubungan antar kriteria yang saling bertabrakan (apabila suatu kriteria terpenuhi dengan baik, maka kemungkinan kriteria lain dikorbankan) dan beberapa kriteria tidak dapat diukur dalam satuan-satuan yang setaraf atau sepadan (*comensurate units*). Oleh karena itu, dalam rangka menentukan metode yang benar (sesuai) secara sistematis dapat digunakan salah satu teknik MCDM seperti misalnya AHP (*Analytic Hierarchy Process*) [50].

Kriteria utama yang harus dipilih meliputi:

1. Akurasi (*accuracy*) — akurasi klasifikasi yang dilakukan NNE.
2. Biaya komputasi (*computational cost*) — biaya kompleksitas untuk membangun NNE serta waktu yang dibutuhkan untuk mengenali suatu objek.
3. Penskalaan (*scalability*) — kemampuan metode NNE untuk menskala dataset yang berjumlah besar.
4. Kelenturan (*flexibility*) — kemampuan untuk menggunakan inducer apapun (*inducer independant*), atau combiner apapun (*combiner independent*) untuk memberikan solusi terhadap tugas klasifikasi yang bervariasi, misalnya tidak hanya terbatas pada tugas klasifikasi biner yang hanya terdiri dari 2 kelas saja.
5. Kegunaan (*usability*) — menggunakan suatu set parameter terkontrol yang komperhensif atau menyeluruh dan dapat *dituning* dengan mudah

6. Keterjemahan (*interpretability*) — kemudahan bagi pengguna untuk menginterpretasi atau memahami hasil ensemble
7. Ketersediaan perangkat lunak (*software availability*) — seberapa banyak paket perangkat lunak yang mendukung metode ensemble tersebut. Ketersediaan yang tinggi (*high availability*) berarti bahwa pengguna dapat berpindah dari satu perangkat lunak ke perangkat lunak yang lain tanpa harus mengganti metode ensemble tersebut.

2.3. Fungsi Error Kuadratis dan Cross-Entropy

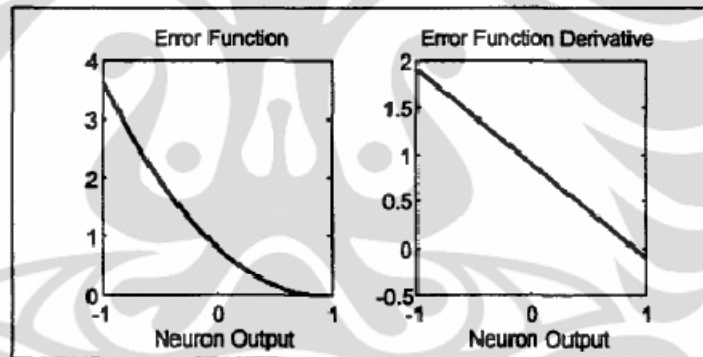
Feed-forward NN merupakan arsitektur NN yang paling sering digunakan dalam aplikasi kehidupan nyata. NN tersebut dilatih dengan metode pengajaran (*supervised learning*), dimana algoritma *Backpropagation* merupakan model yang paling sering digunakan daripada jenis algoritma pelatihan yang lain [58]. Ide di balik algoritma *Backpropagation* sesungguhnya adalah menuruni lembah permukaan error (*gradient descent*) untuk meminimalkan fungsi error dengan meng-*update* bobot-bobot neuron [4].

Sebuah fungsi error (disebut juga *loss function* atau *objective function*) adalah fungsi yang digunakan untuk membandingkan parameter-parameter ketika proses penyesuaian (*fitting*) antara keluaran NN dengan target yang diinginkan yang terjadi pada saat proses pelatihan berlangsung. Fungsi error menggambarkan selisih perbedaan (nilai atau jarak) antara keluaran NN dengan target [61]. Itulah mengapa, turunan dari fungsi error merupakan salah satu faktor dalam persamaan perubahan (*update*) bobot. Oleh karena itulah, untuk dapat berhasil, penting untuk melatih jaringan dengan fungsi error yang menyerupai tujuan dari masalah yang ingin diselesaikan.

Fungsi error yang paling umum dipakai, dan diterapkan secara luas dalam statistik, adalah fungsi error kuadratis. Fungsi error kuadratis menghitung selisih kuadrat antara target (t) dengan output jaringan (y), secara matematis dituliskan sebagai berikut

$$E = \frac{1}{2} \sum_k (t_k - y_k)^2 \quad (2.44)$$

Karakteristik fungsi eror kuadratis adalah penekanan yang dilakukan untuk mengurangi eror yang besar daripada eror yang kecil sebagai efek dari pengkuadratan (*squaring*) yang digunakan untuk menghitung selisih antara output NN dengan target [65]. Hal ini dapat menjadi keuntungan dalam banyak kasus, tetapi tidak selalu demikian. Dalam beberapa kasus, hal ini dapat menjadi kerugian. Di samping itu, akibat penjumlahan total (*summation*) eror-eror dari seluruh input *data training*, maka apabila sebuah kelas tidak terrepresentasikan dengan baik dalam suatu sampel populasi dan ternyata memiliki eror yang kecil, terdapat kemungkinan anggota kelas tersebut akan diabaikan oleh NN pada saat pelatihan [65]. Akibatnya anggota kelas tersebut memiliki *recognition rate* yang lebih rendah pada saat pengujian dibandingkan kelas lain yang terwakili dengan baik dan memiliki eror yang besar.



Gambar 2.4. Fungsi Eror Kuadratis dan Turunannya [65]

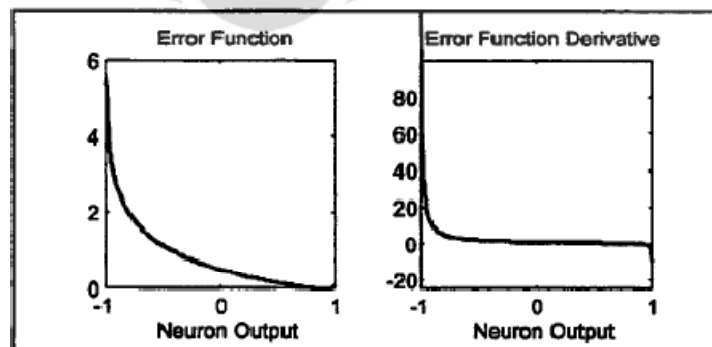
Salah satu kelemahan algoritma *Backpropagation* adalah waktu pelatihan yang lama. Ketika pelatihan berlangsung, terdapat tahap dimana penurunan eror kuadratis berlangsung sangat lambat. Dalam aplikasi pengenalan pola, output yang diharapkan adalah salah satu dari 2 nilai ekstrim fungsi aktivasi sigmoid. Keadaan yang disebut “tersaturasi secara salah (*incorrectly saturated*)” terjadi jika jumlah bobot-bobot output neuron mendekati nilai ekstrim yang salah. Jika hal ini terjadi, maka perubahan bobot hanya akan terjadi secara minim akibat kecilnya gradien fungsi sigmoid, dan akibatnya perubahan eror pun hampir tidak terjadi. Period stagnasi inilah yang mendominasi lamanya waktu pelatihan [63].

Selain membutuhkan waktu pelatihan yang lama, kekurangan NN algoritma pelatihan *Backpropagation* yang lain yaitu bahwa terdapat kemungkinan untuk terjebak dalam *local minima*. Hal ini disebabkan karena aktivasi ekstrim pada neuron ketika nilainya mendekati *asymptote* fungsi alih atau fungsi aktivasi [65]. Untuk menghindari permasalahan di atas, maka alih-alih menggunakan fungsi eror kuadratis yang umum dipakai, digunakanlah fungsi eror *cross-entropy* [22,23]. Fungsi eror *cross-entropy* secara matematis dituliskan sebagai berikut:

$$E = \sum_k (-t_k) \ln(y_k) - (1-t_k) \ln(1-y_k) \quad (2.45)$$

Khusus untuk masalah klasifikasi, dapat dikatakan bahwa fungsi eror *cross-entropy* merupakan fungsi yang lebih cocok digunakan daripada fungsi eror kuadratis, dimana fungsi ini membutuhkan epoch yang lebih sedikit untuk persentase *recognition rate* yang sama. Hal ini terutama berlaku pada problem yang kompleks, dimana fungsi eror *cross-entropy* ini lebih stabil digunakan dalam pelatihan daripada fungsi eror yang lainnya, termasuk fungsi eror kuadratis [65]. Metode fungsi eror *cross-entropy* terbukti memberikan solusi optimal yang konvergen pada sampel data yang besar [62].

Namun demikian, selain kelebihan yang disebutkan di atas, fungsi eror *cross-entropy* juga memiliki kekurangan berupa “spesialisasi berlebih (*overspecialization*)” untuk pola data *training* akibat sinyal eror yang terlalu kuat [64].



Gambar 2.5. Fungsi Error *Cross Entropy* dan Turunannya [65]

2.4. Neural Network untuk Pengenalan Pola

Otak manusia sangat baik dalam melakukan tugas-tugas yang berkaitan dengan penginderaan (*sensory*) seperti contohnya penglihatan (*vision*), pengenalan perkataan (*speech recognition*), pengenalan aroma (*odor recognition*) dan lain lain. Otak manusia dapat menyelesaikan tugas-tugas ini dengan usaha yang minimum dikarenakan struktur otak yang telah berkembang dan tersusun dalam sistem-sistem parallel dimana kumpulan elemen pemroses membagi (*share*) tugas untuk melakukan suatu pekerjaan besar dan bukannya membuat satu buah elemen pemroses bekerja sendiri untuk menyelesaikan semua pekerjaan.

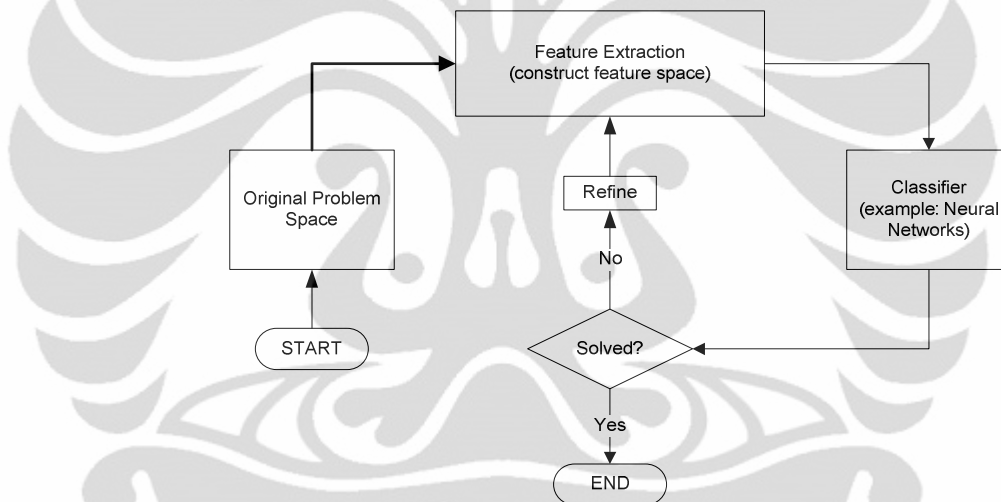
Namun demikian, komputer (atau mesin) memiliki struktur yang sangat berbeda dengan otak manusia. Komputer terdiri dari sebuah elemen pemroses yang mampu melakukan pekerjaan sederhana berulang-berulang seperti penambahan atau perkalian angka dengan sangat baik, namun komputer sangat buruk dalam melakukan pekerjaan yang membutuhkan pemrosesan beragam jenis data seperti penglihatan atau pendengaran misalnya.

Tabel 2.1. Perbedaan antara Boolean-based Computer dan Neural-based Computer

| Boolean-based computer | Neural-based computer |
|---|--|
| Baik digunakan untuk tugas logika matematika | Baik digunakan untuk tugas pengenalan pola |
| Proses berkecepatan tinggi (<i>high speed processing</i>) | Proses berkecepatan rendah (<i>low speed processing</i>) |
| Proses dilakukan secara satu per satu (<i>sequential</i>) | Proses dilakukan secara bersamaan (<i>parallel</i>) |

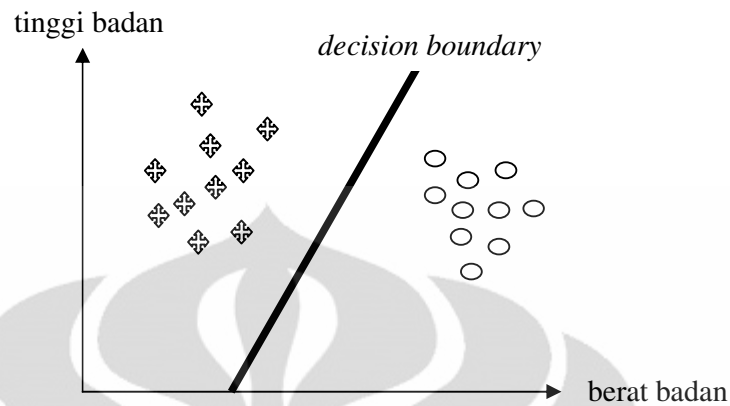
Tujuan mendasar dari pengenalan pola adalah pengklasifikasian (*classification*) suatu data masukan ke dalam suatu kategori kelas. Secara umum, sistem pengenalan pola terdiri dari dua tahap. Tahap pertama berupa ekstraksi fitur (*feature extraction*), sedangkan tahap berikutnya merupakan tahap pengklasifikasian.

Untuk melakukan pengenalan pola, pertama-tama dilakukan serangkaian pengukuran untuk dapat membedakan input ke dalam kategori kelas. Jika pengukuran dilakukan sebanyak n pengukuran dengan masing-masing pengukuran merupakan fitur yang unik, maka kita dapat menggunakan notasi aljabar untuk menciptakan suatu set fitur-fitur tersebut yang disebut dengan *feature vector*. Banyaknya dimensi dari vector tersebut membentuk sebuah n dimensi *feature space*. Selanjutnya adalah menentukan sebuah garis batas pemisah (*boundary*) antar kategori kelas yang disebut dengan fungsi diskriminan (*discriminating function*). Dengan demikian pengklasifikasian dilakukan dengan menentukan tergolong ke dalam kategori kelas manakah masing-masing input berdasarkan ciri-ciri fitur yang dimiliki.



Gambar 2.6. Sudut Pandang Pengenalan Pola (*Pattern Recognition View Point*)

Contoh sederhana untuk memvisualisasikan pengklasifikasian pola adalah membedakan antara penari balet dari pemain rugby. Dalam hal ini kita memutuskan untuk menggunakan 2 pengukuran untuk membedakan masing-masing kategori, yaitu dengan mengukur vektor tinggi badan dan berat badan dan membuat sebuah plot dari data pengukuran tersebut ke dalam ruang *Euclidean*. Dengan demikian, kita akan membentuk sebuah ruang vektor dua dimensi (x_1, x_2) . Selanjutnya kita akan melakukan pengklasifikasian pada tiap-tiap data tersebut dengan menggambarkan sebuah garis pemisah yang membedakan antara kelompok penari balet dari kelompok pemain rugby.



Gambar 2.7. Contoh Pengklasifikasian Linear dengan Dua Dimensi

- ⊕ penari balet
- pemain rugby

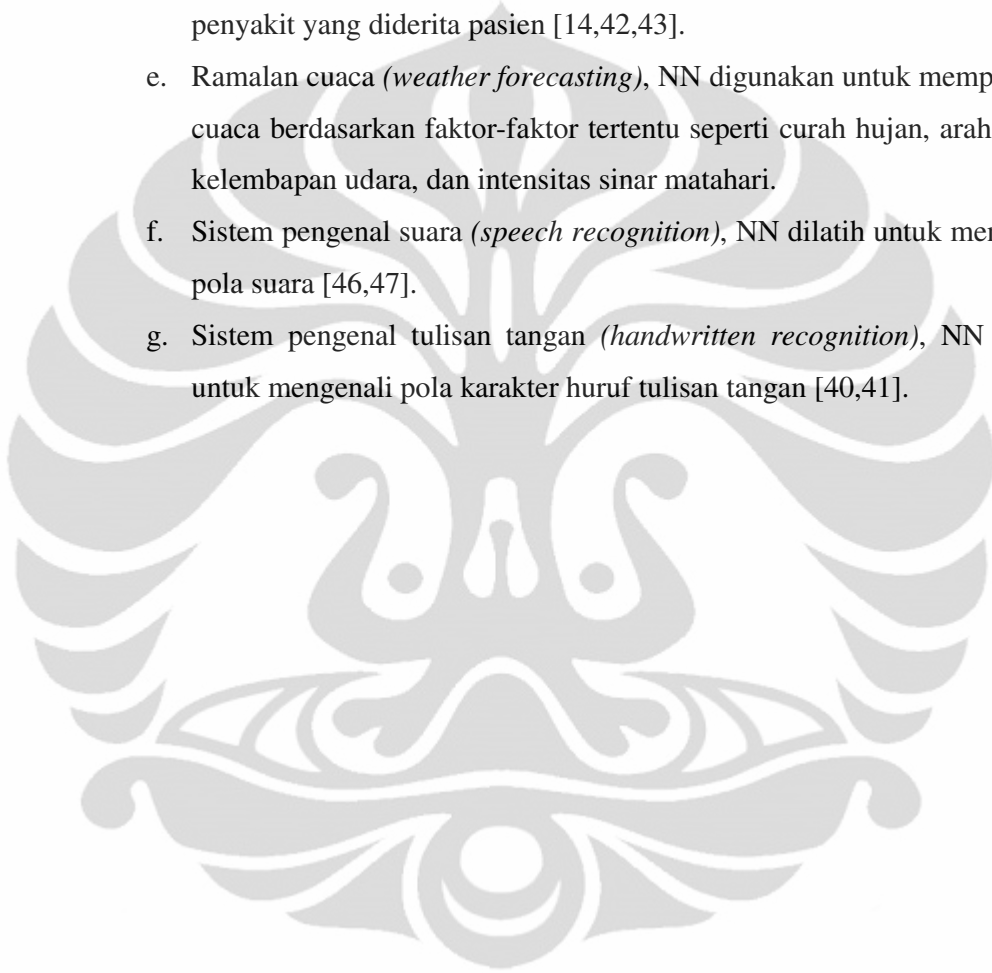
Klasifikasi dan pengenalan pola (*pattern recognition*) merupakan area yang mendominasi berbagai penelitian di bidang NN. NN untuk *problem* klasifikasi dapat dilihat sebagai sebuah fungsi pemetaan (*mapping function*), $F: R^d \rightarrow R^M$, dimana input x dengan dimensi sebanyak d dijadikan masukan ke dalam NN dan output y dengan vektor sebanyak M dihasilkan untuk membuat keputusan klasifikasi [51].

Suatu keunggulan NN yang tidak dimiliki oleh teknologi konvensional lain di bidang kecerdasan buatan (*artificial intelligence, AI*) adalah bahwa NN mampu untuk bekerja dengan data yang tidak lengkap (*incomplete*) ataupun data dengan gangguan (*noise*). Selain itu NN mampu mengolah data di luar data yang secara eksplisit dipelajari selama proses pelatihan. Dalam hal ini NN mampu mengeneralisasi atau membuat keputusan berdasarkan kemiripan yang paling dekat (*closest match*).

Aplikasi NN dalam klasifikasi dan pengenalan pola dapat dimanfaatkan dalam berbagai bidang, diantaranya:

- a. Jasa keuangan (*financial service*), contohnya untuk mengidentifikasi pola pada pasar saham (*stock market*), strategi penjualan surat berharga (*bond trading*), prediksi *bankruptcy* [38,39]

- b. Kegiatan industri, contohnya monitoring atau inspeksi pada alat-alat produksi di pabrik maupun produk yang dihasilkan [48,49].
- c. Diagnosis pada mesin, dengan melatih NN untuk mengidentifikasi kegagalan (*faults identification*) berdasarkan data sensor [44,45].
- d. Diagnosis kesehatan, dengan melatih NN berdasarkan data riwayat penyakit yang diderita pasien [14,42,43].
- e. Ramalan cuaca (*weather forecasting*), NN digunakan untuk memprediksi cuaca berdasarkan faktor-faktor tertentu seperti curah hujan, arah angin, kelembapan udara, dan intensitas sinar matahari.
- f. Sistem pengenalan suara (*speech recognition*), NN dilatih untuk mengenali pola suara [46,47].
- g. Sistem pengenalan tulisan tangan (*handwritten recognition*), NN dilatih untuk mengenali pola karakter huruf tulisan tangan [40,41].



BAB III
SISTEM PENGENAL POLA
BERBASIS NEURAL NETWORK ENSEMBLE
DENGAN FUNGSI EROR CROSS-ENTROPY DAN KUADRATIS

Bab ini menjelaskan tentang eksperimen yang dilakukan dengan membangun sistem pengenalan pola berbasis *Neural Network Ensemble* (NNE) yang dilatih dengan algoritma *Negative Correlated Learning* (NCL) dengan penghitungan eror berdasarkan fungsi eror kuadratis atau *Mean square error (MSE)* dan penghitungan eror berdasarkan fungsi eror cross-entropy.

3.1. Pendahuluan

Bagian Pendahuluan diawali dengan latar belakang penelitian pengenalan pola, kemudian diikuti dengan penjelasan skema penelitian (*experimental setup*) sistem pengenalan pola yang dibangun oleh Penulis.

3.1.1. Latar Belakang Penelitian

Dua isu paling penting mengenai pembelajaran dengan pengarahannya (*supervised learning*) yaitu performa generalisasi dan efisiensi. Generalisasi bertujuan membangun sebuah sistem pengenalan pola adaptif yang mencapai performa optimal pada sampel data yang tidak diikutsertakan pada data pelatihan (*training set*). Sedangkan efisiensi berkaitan dengan kompleksitas ruang (*space complexity*) dan kompleksitas waktu (*time complexity*) dari sebuah sistem pengenalan pola. Kompleksitas ruang mengacu kepada ukuran sistem, sedangkan kompleksitas waktu mengacu pada waktu komputasi yang dibutuhkan untuk membangun sistem tersebut.

Karakteristik performa generalisasi dari sistem dengan pembelajaran dengan pengarahannya ditandai dengan eror generalisasinya. Untuk masalah klasifikasi, eror generalisasi adalah probabilitas terjadinya misklasifikasi dari sampel acak yang dilakukan oleh sebuah sistem pengklasifikasi (*classifier*). Karena ukuran (kompleksitas ruang) dari sistem pengklasifikasi berkaitan dengan kemampuan generalisasi sistem, maka yang menjadi isu di sini adalah bagaimana menentukan

ukuran yang sesuai untuk sistem pengenalan pola sedemikian hingga eror generalisasi yang diinginkan pun dapat tercapai.

Terdapat banyak masalah di dalam dunia nyata (*real-world problems*) yang terlalu luas dan terlalu kompleks untuk dipecahkan oleh sebuah sistem tunggal sendirian. Ada banyak contoh baik sistem alami (*natural*) maupun buatan (*artificial*) yang menunjukkan bahwa sebuah sistem terpadu yang terdiri dari beberapa subsistem di dalamnya, terbukti dapat mengurangi kompleksitas sistem secara keseluruhan dan mampu memecahkan masalah yang lebih sulit. Kesuksesan NNE dalam memperbaiki kemampuan generalisasi dari *classifier* adalah salah satu contohnya [33]. Namun demikian, peningkatan kemampuan generalisasi pada NNE tersebut diiringi dengan biaya komputasi (*computational cost*) yang juga turut meningkat seiring dengan semakin besarnya ukuran sistem.

Terdapat beberapa metode untuk membangun sebuah NNE. Banyak diantaranya terdiri dari 2 tahapan, (1) membangun beberapa jaringan tunggal, lalu (2) menggabungkan atau mengkombinasikan jaringan-jaringan tersebut. Biasanya, jaringan tunggal akan dilatih secara independen satu sama lain. Kelemahan dari metode ini adalah hilangnya interaksi antara jaringan tunggal yang satu dengan jaringan tunggal yang lain selama proses pembelajaran berlangsung. Pada tahap penggabungan (kombinasi), tidak ada umpan balik (*feedback*) kepada performa jaringan tunggal. Dengan demikian terdapat kemungkinan bahwa ada beberapa jaringan tunggal yang tidak memberikan banyak kontribusi terhadap keseluruhan NNE.

Berangkat dari permasalahan itulah, maka digunakan metode pembelajaran *Negative Correlation Learning* (NCL) dalam membangun sebuah NNE. Ide di balik NCL adalah mendorong jaringan tunggal yang berbeda dalam sebuah kumpulan jaringan (*ensemble*) untuk mempelajari aspek atau bagian yang berbeda dari data *training* sehingga NNE dapat mempelajari keseluruhan data *training* dengan lebih baik [11].

Klasifikasi secara umum dapat didefinisikan sebagai proses identifikasi kesamaan dalam suatu dataset dengan tujuan untuk mendiskriminasi suatu grup (kelas) dari suatu grup yang lain [10]. Sedangkan pengenalan lebih mengacu kepada

apakah hasil klasifikasi pola ke dalam kelas-kelas tersebut memberikan hasil yang bernilai benar ataukah hasil yang bernilai salah.

Dalam menyelesaikan masalah klasifikasi, NN memberikan estimasi langsung mengenai probabilitas posterior (*posterior probability*) [52]. NN berguna untuk memetakan sejumlah input- x dengan d -dimensi ke dalam sejumlah vektor output- y untuk membuat keputusan klasifikasi. Namun demikian, fungsi pemetaan yang dilakukan NN tidaklah sempurna. Hal ini disebabkan karena adanya masalah *local minima*, arsitektur jaringan yang tidak optimal, serta terbatasnya jumlah sampel data untuk proses pelatihan [51].

Secara teori, dibutuhkan sebuah jaringan yang besar dan sampel data yang banyak untuk mendapatkan aproksimasi yang memuaskan. Namun demikian, secara empiris, ditemukan bahwa ukuran sampel atau banyaknya data sampel dalam pelatihan berpengaruh besar terhadap performa NN dalam melakukan klasifikasi, sedangkan jumlah neuron tersembunyi (*hidden neuron*) dan arsitektur jaringan tidaklah terlalu berpengaruh terhadap performa [53].

Output NN yang berupa estimasi kuadrat terkecil (*least square*) dari probabilitas posterior Bayesian tidak hanya digunakan untuk fungsi eror kuadratis saja, tetapi juga valid untuk tipe fungsi eror yang lain, seperti misalnya fungsi eror *cross-entropy* [54]. Fungsi eror *cross-entropy*, dengan karakteristik output biner yang dimilikinya, dapat menjadi kriteria yang lebih sesuai daripada fungsi eror kuadratis untuk digunakan dalam pelatihan NN untuk masalah klasifikasi [55]. Selain itu, fungsi eror *cross-entropy* terbukti menghasilkan peningkatan performa serta pengurangan lama waktu pelatihan [56].

Dalam beberapa literatur tentang penelitian NN untuk klasifikasi, hanya sedikit studi yang mempertimbangkan biaya kesalahan klasifikasi (*misclassification cost*) dalam pengambilan keputusan klasifikasi. Dengan kata lain, peneliti secara eksplisit ataupun implisit mengasumsikan konsekuensi biaya kesalahan klasifikasi yang setara (*equal*). Meskipun mengasumsikan fungsi biaya 0-1 dapat menyederhanakan pengembangan model, asumsi biaya yang setara tidak merepresentasikan banyak masalah dunia nyata (*real world problems*) seperti contohnya dalam bidang diagnosis medis, analisis resiko kredit, asuransi kualitas,

sampling penerimaan, dan prediksi kebangkrutan dimana biaya kesalahan klasifikasi yang tidak seimbang (*uneven*) lebih sesuai digunakan.

Dalam situasi-situasi tersebut, jumlah anggota kelompok dalam populasi seringkali tidak seimbang jumlahnya, sehingga kesalahan klasifikasi dapat menimbulkan konsekuensi signifikan yang berbeda-beda pada kelompok yang satu dan kelompok yang lain. Biaya kesalahan klasifikasi yang lebih tinggi akan terjadi pada subjek dalam kelompok yang lebih kecil (yang anggotanya jauh lebih sedikit). Dengan asumsi konsekuensi misklasifikasi yang setara, sebuah *classifier* akan cenderung berat sebelah (*bias*) terhadap kelompok yang ukurannya lebih besar yang memiliki jumlah sampel lebih banyak dalam data pelatihan.

Untuk membantu mengurangi keadaan *bias* tersebut, dapat dilakukan dengan cara memperbesar jumlah sampel pada data pelatihan. Dengan demikian terdapat cukup banyak anggota dari kelompok yang lebih kecil yang terwakili dalam sampel data. Alternatif lainnya, yaitu dengan memilih set data pelatihan (dan data pengujian tentunya) dengan jumlah anggota yang seimbang dari tiap kelompok yang ada [57].

Pada penelitian skripsi ini, Penulis membangun sebuah sistem pengenalan pola berbasis NN. Untuk meningkatkan derajat pengenalan NN tunggal, maka Penulis melakukan penelitian berbasis NNE dengan jumlah jaringan sebanyak 3 dan 5 buah. Untuk mengetahui perbandingan karakteristik fungsi eror kuadratis dan fungsi eror *cross-entropy* terhadap NNE, maka proses pelatihan dilakukan dengan kedua jenis fungsi eror tersebut (secara terpisah) dan kemudian hasil yang diperoleh akan dibandingkan untuk mengetahui fungsi eror mana yang lebih baik untuk diterapkan dalam sistem pengenalan pola.

3.1.2. Skema Penelitian

Untuk meneliti performa sistem pengenalan pola yang dibangun, maka Penulis melakukan eksperimen dengan menggunakan 9 buah dataset yang tersedia dari University of California at Irvine (UCI) repository [12] dengan *properties* seperti yang dijelaskan pada tabel 4.1. berikut ini.

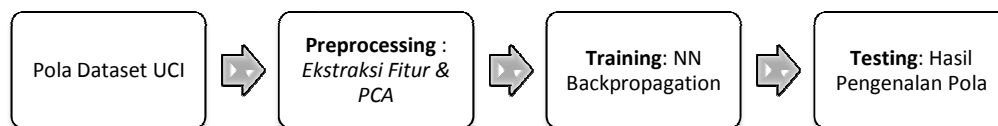
Perlu diketahui bahwa pada tahap *pre-processing* dilakukan PCA untuk semua dataset meskipun dimensinya cukup rendah (<50 dimensi), hal ini bertujuan agar komposisi data lebih terstruktur dimana vektor-vektor data akan disusun

berurutan sesuai dengan besarnya nilai eigen yang berpasangan dengan vektor tersebut. Hal ini secara tidak langsung akan mempercepat komputasi yang dilakukan oleh perangkat lunak. Sedangkan untuk dataset yang berdimensi tinggi (>50), maka PCA dilakukan untuk mereduksi dimensi, seperti contohnya pada dataset Sonar yang direduksi menjadi 30 dimensi (menjadi setengah dari jumlah dimensi aslinya).

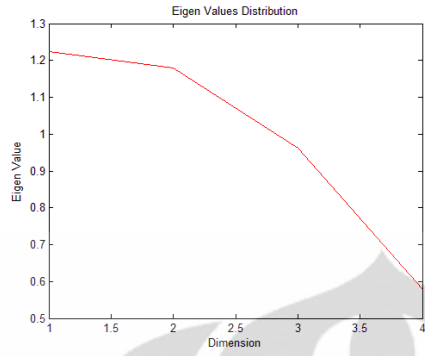
Selain itu perlu dicatat bahwa dalam penelitian ini semua dataset disusun sedemikian rupa sehingga jumlah anggota per kelas bernilai sama (*equal*). Hal ini untuk menghindari ketimpangan biaya kesalahan klasifikasi (*misclassification cost*) antar kelas seperti yang telah dijelaskan pada bagian Pendahuluan.

Tabel 3.1. *Properties* Dataset UCI untuk Penelitian Sistem Pengenal Pola

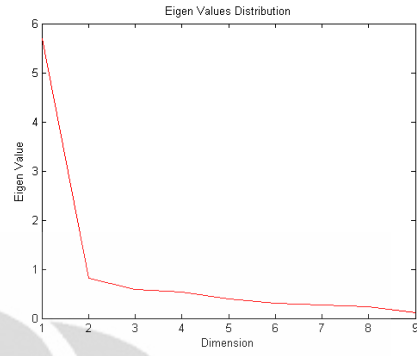
| No | Tipe Data | Jumlah Dimensi (asli/ sebelum PCA) | Jumlah Dimensi (setelah PCA) | Jumlah Kelas | Jumlah Data |
|----|---------------|------------------------------------|------------------------------|--------------|-------------|
| 1 | Balance Scale | 4 | 4 | 3 | 147 |
| 2 | Breast cancer | 10 | 10 | 2 | 482 |
| 3 | Bupa | 6 | 6 | 2 | 290 |
| 4 | Credit | 15 | 15 | 2 | 614 |
| 5 | Glass | 10 | 10 | 6 | 54 |
| 6 | Heart disease | 13 | 13 | 2 | 240 |
| 7 | Iris | 4 | 4 | 3 | 150 |
| 8 | Sonar | 60 | 30 | 2 | 194 |
| 9 | Vowel | 12 | 12 | 11 | 990 |



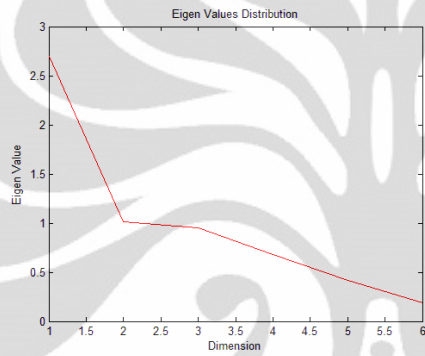
Gambar 3.1. Diagram Skema Sistem Pengenal Pola



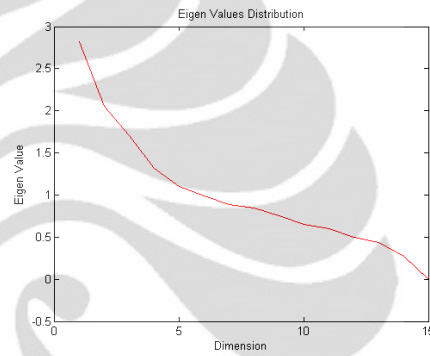
3.2.a



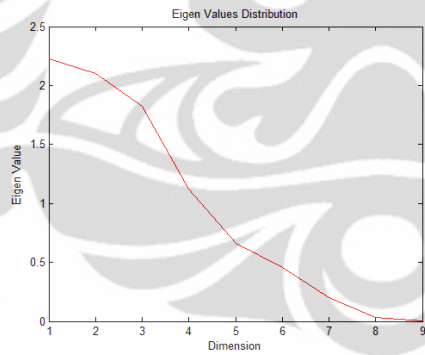
3.2.b



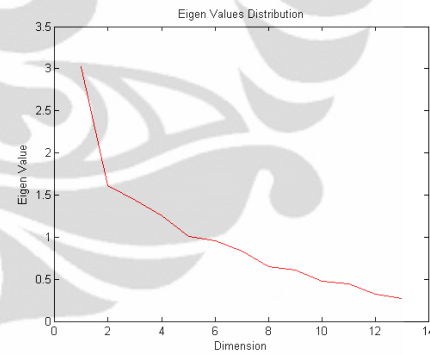
3.2.c



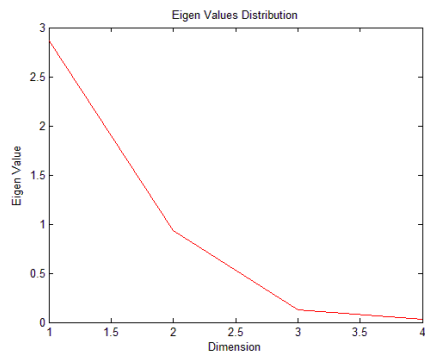
3.2.d



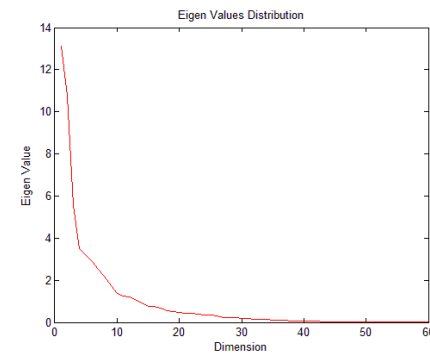
3.2.e



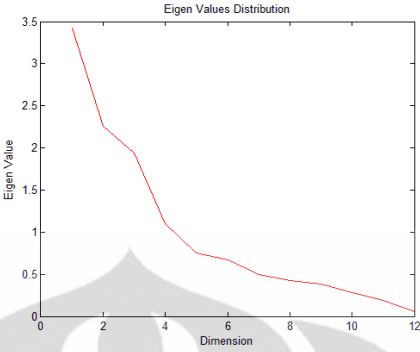
3.2.f



3.2.g



3.2.h



3.2.i

Gambar 3.2. Distribusi Nilai Eigen 9 Dataset UCI: (a) Balance-scale (b) Breast-cancer (c) Bupa (d) Credit-card (e) Glass (f) Heart-disease (g) Iris (h) Sonar (i) Vowel

3.2. Hasil Penelitian dan Pembahasan

Pada penelitian ini digunakan NN tunggal, NNE sebanyak 3 buah jaringan, serta NNE sebanyak 5 buah jaringan untuk mengetahui pengaruh penambahan jumlah jaringan terhadap karakteristik NN dan NNE yang dilatih dengan fungsi eror kuadratis dan *cross-entropy*. Pada bagian ini, fokus utama adalah mengetahui karakteristik fungsi eror kuadratis dan *cross-entropy* terhadap pengaruh penambahan jumlah jaringan (dari NN tunggal menjadi NNE). Setelah mengetahui karakteristik tersebut, kemudian akan ditarik kesimpulan mengenai keseluruhan penelitian sistem pengenalan pola pada bab ini. Tabel 3.2 - 3.6 berisi ringkasan perbandingan eror, jumlah epoch, waktu pelatihan, *recognition rate* data *training*, serta *recognition rate* data *testing* hasil pelatihan NN dan NNE untuk pengenalan pola data UCI.

Tabel 3.2. Perbandingan Error Hasil Pelatihan NN Tunggal dan NNE untuk Dataset UCI

| Data set | NN tunggal | | NNE 3 jaringan | | NNE 5 jaringan | |
|------------------|-------------|--------------|----------------|--------------|----------------|--------------|
| | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy |
| Balance-scale | 2.73 | 22.31 | 0.01 | 0.01 | 0.01 | 0.01 |
| Breast-cancer | 3.80 | 0.01 | 1.64 | 0.01 | 1.39 | 0.01 |
| Bupa | 5.67 | 6.61 | 4.17 | 40.54 | 4.28 | 35.85 |
| Credit-card | 5.97 | 0.01 | 4.69 | 10.17 | 4.63 | 8.92 |
| Glass | 0.01 | 0.02 | 0.02 | 0.02 | 0.01 | 0.02 |
| Heart-disease | 0.71 | 0.01 | 0.35 | 0.20 | 0.54 | 0.01 |
| Iris | 0.01 | 9.01 | 0.74 | 9.05 | 9.04 | 9.04 |
| Sonar | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 |
| Vowel | 4.06 | 0.04 | 3.17 | 0.04 | 2.71 | 0.04 |
| Rata-rata | 2.55 | 4.22 | 1.64 | 6.67 | 2.51 | 5.99 |

Tabel 3.3. Perbandingan Jumlah Epoch Hasil Pelatihan NN Tunggal dan NNE untuk Dataset UCI

| Data set | NN tunggal | | NNE 3 jaringan | | NNE 5 jaringan | |
|------------------|------------------|---------------------|------------------|---------------------|------------------|---------------------|
| | <i>Kuadratis</i> | <i>Cros-entropy</i> | <i>Kuadratis</i> | <i>Cros-entropy</i> | <i>Kuadratis</i> | <i>Cros-entropy</i> |
| Balance-scale | 20000 | 20000 | 5474 | 15726 | 4738 | 4738 |
| Breast-cancer | 20000 | 7585 | 20000 | 9950 | 20000 | 10651 |
| Bupa | 20000 | 20000 | 20000 | 20000 | 20000 | 20000 |
| Credit-card | 20000 | 1346 | 20000 | 11728 | 20000 | 20000 |
| Glass | 9703 | 20000 | 11929 | 20000 | 14473 | 20000 |
| Heart-disease | 14038 | 8408 | 16768 | 17772 | 20000 | 20000 |
| Iris | 4461 | 20000 | 20000 | 20000 | 20000 | 20000 |
| Sonar | 401 | 1070 | 629 | 1097 | 605 | 1088 |
| Vowel | 20000 | 20000 | 20000 | 20000 | 20000 | 20000 |
| Rata-rata | 14289 | 11157 | 14978 | 15141 | 15535 | 15164 |

Tabel 3.4. Perbandingan Waktu Pelatihan NN Tunggal dan NNE untuk Dataset UCI

| Data set | NN tunggal | | NNE 3 jaringan | | NNE 5 jaringan | |
|------------------|------------------|---------------------|------------------|---------------------|------------------|---------------------|
| | <i>Kuadratis</i> | <i>Cros-entropy</i> | <i>Kuadratis</i> | <i>Cros-entropy</i> | <i>Kuadratis</i> | <i>Cros-entropy</i> |
| Balance-scale | 366.73 | 385.15 | 147.87 | 772.32 | 116.30 | 116.30 |
| Breast-cancer | 643.34 | 648.95 | 2190.87 | 1267.67 | 6349.74 | 4288.16 |
| Bupa | 321.55 | 307.80 | 806.50 | 1135.44 | 1169.00 | 1768.09 |
| Credit-card | 1137.98 | 128.14 | 1870.68 | 1020.38 | 6439.54 | 7410.83 |
| Glass | 33.45 | 162.78 | 85.77 | 227.66 | 206.58 | 286.91 |
| Heart-disease | 187.52 | 298.45 | 1366.26 | 1521.28 | 1486.11 | 1932.25 |
| Iris | 361.41 | 332.61 | 848.53 | 863.91 | 568.74 | 589.61 |
| Sonar | 7.26 | 15.64 | 24.67 | 129.24 | 88.97 | 159.97 |
| Vowel | 2192.23 | 3935.25 | 6521.56 | 6143.33 | 11820.09 | 11776.59 |
| Rata-rata | 583.50 | 690.53 | 1540.30 | 1453.47 | 3138.34 | 3147.63 |

Tabel 3.5. Perbandingan Recognition Data Training NN Tunggal dan NNE untuk Dataset UCI

| Data set | NN tunggal | | NNE 3 jaringan | | NNE 5 jaringan | |
|------------------|------------------|---------------------|------------------|---------------------|------------------|---------------------|
| | <i>Kuadratis</i> | <i>Cros-entropy</i> | <i>Kuadratis</i> | <i>Cros-entropy</i> | <i>Kuadratis</i> | <i>Cros-entropy</i> |
| Balance-scale | 96.00% | 96.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| Breast-cancer | 98.43% | 100.00% ■ | 99.18% | 100.00% | 99.11% | 100.00% |
| Bupa | 94.18% ■ | 92,52% | 88.73% ■ | 85.07% | 83.70% | 85.46% ■ |
| Credit-card | 97.86% | 96.99% | 96.37% | 96.73% | 95.28% | 96.11% |
| Glass | 100.00% | 100.00% | 99.72% | 100.00% | 99.58% | 100.00% |
| Heart-disease | 99.42% | 100.00% | 98.27% | 99.92% ■ | 96.55% | 100.00% ■ |
| Iris | 100.00%■ | 98.67% | 98.14% | 98.67% | 98.67% | 98.67% |
| Sonar | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| Vowel | 98.36% | 100.00% ■ | 97.59% | 100.00% ■ | 95.86% | 100.00% ■ |
| Rata-rata | 98.25% | 98.96% | 97.56% | 97.82% | 96.53% | 97.80% ■ |

Tabel 3.6. Perbandingan Recognition Data Testing NN Tunggal dan NNE untuk Dataset UCI

| Data set | NN tunggal | | NNE 3 jaringan | | NNE 5 jaringan | |
|---------------|------------|--------------|----------------|--------------|----------------|--------------|
| | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy |
| Balance-scale | 88.89%■ | 78.89% | 50.39% | 63.55%■ | 50.62% | 50.44% |
| Breast-cancer | 95.53%■ | 90.45% | 89.64% | 90.08% | 89.13% | 88.85% |
| Bupa | 58.68% | 59.25% | 45.39% ■ | 41.10% | 40.82% | 43.86% ■ |
| Credit-card | 76.21%■ | 72.20% | 67.23% | 67.53% | 66.17% | 67.03% |
| Glass | 54.86%■ | 46.22% | 41.11% | 45.55% ■ | 40.66% | 47.67% ■ |
| Heart-disease | 72.51% | 77.03%■ | 73.08% | 75.28% ■ | 70.47% | 72.89% ■ |
| Iris | 91.91% | 96.00%■ | 94.93% | 96.00% ■ | 96.00% | 96.00% |
| Sonar | 39.69% | 64.19%■ | 62.44% | 64.97% ■ | 62.14% | 64.56% ■ |
| Vowel | 20.65% | 23.24%■ | 13.20% | 17.92% ■ | 12.15% | 15.39% ■ |
| Rata-rata | 66.55% | 67.55%■ | 59.71% | 62.44% ■ | 58.68% | 60.74% ■ |

Keterangan: tanda (■) menunjukkan recognition rate yang lebih tinggi (peningkatan $\geq 1\%$)

Dari tabel 3.2 yang berisi perbandingan eror hasil pelatihan NN tunggal dan NNE, terlihat bahwa tidak semua jaringan mampu mencapai kondisi konvergen kecuali pada kasus pengenalan dataset *Sonar* yang cukup mudah dimana seluruh NN tunggal dan NNE baik yang dilatih dengan fungsi eror kuadratis maupun *cross entropy* mampu mencapai kondisi konvergen. Untuk pengenalan dataset *Balance-scale*, pelatihan dengan NN tunggal tidak mampu mencapai kondisi konvergen, tetapi pelatihan dengan NNE mampu mereduksi eror pelatihan dan mencapai keadaan konvergen. Sedangkan untuk dataset *Breast-cancer*, hanya NN tunggal dan NNE (dengan 3 dan 5 jaringan) yang dilatih dengan fungsi eror *cross entropy* yang mampu mencapai keadaan konvergen, sedangkan NN tunggal dan NNE yang dilatih dengan fungsi eror kuadratis tidak mampu menuju kondisi konvergen meskipun eror yang dicapainya sudah cukup rendah.

Dilihat dari rata-rata eror pelatihan yang dihasilkan, dari tabel 3.2 dapat disimpulkan bahwa pelatihan menggunakan fungsi eror *cross entropy* akan menghasilkan eror yang lebih tinggi daripada pelatihan dengan menggunakan fungsi eror kuadratis. Tetapi hal ini dikarenakan terutama oleh eror dataset *Bupa* yang secara mencolok jauh lebih tinggi melebihi eror dataset-dataset lainnya, hal ini mengimplikasikan bahwa masalah *Bupa* lebih cocok didekati dengan metode fungsi eror kuadratis. Sedangkan untuk kasus dataset lainnya, fungsi eror *cross entropy*

menghasilkan lebih banyak jaringan yang konvergen (baik NN tunggal maupun NNE) daripada metode fungsi error kuadratis.

Dari tabel 3.3, dapat dilihat bahwa jaringan yang mampu mencapai kondisi konvergen akan membutuhkan epoch yang lebih sedikit daripada jaringan yang tidak. Hal ini dikarenakan proses pelatihan berhenti ketika error telah mencapai minimum (sebesar 0.01), yang berarti bahwa jaringan telah mampu mengenali pola input data pelatihan dengan baik. Sebaliknya, jaringan yang tidak mampu mencapai error minimum harus mengulang pelatihan berkali-kali hingga proses pelatihan akhirnya dihentikan akibat jumlah epoch yang telah mencapai batas maksimum (20.000 epoch).

Dilihat dari rata-rata jumlah epoch pelatihan yang diperlukan, dari tabel 3.3 dapat disimpulkan bahwa pelatihan menggunakan fungsi error *cross entropy* akan membutuhkan epoch yang setara (kurang lebih sama) daripada pelatihan dengan menggunakan fungsi error kuadratis. Bahkan untuk kasus NN tunggal dan NNE 5 jaringan, pelatihan menggunakan fungsi error *cross entropy* membutuhkan epoch yang lebih sedikit daripada pelatihan dengan menggunakan fungsi error kuadratis.

Hasil perbandingan waktu pelatihan pada tabel 3.18 menunjukkan bahwa lamanya waktu pelatihan dipengaruhi oleh jumlah jaringan, ukuran dataset, serta jumlah epoch pelatihan. Semakin banyak jumlah jaringan yang menyusun NNE, maka semakin lama waktu pelatihan yang diperlukan. Demikian pula, semakin besar ukuran dataset (semakin banyak jumlah sampel), maka waktu pelatihan yang diperlukan pun semakin meningkat.

Pengaruh jumlah jaringan, ukuran dataset, serta jumlah epoch pelatihan terhadap peningkatan waktu pelatihan dapat dijelaskan sebagai berikut. Dalam pelatihan NNE dengan NCL, NNE dilatih secara bersama-sama (*simultaneous*). Dalam prosesnya, satu jaringan berinteraksi dengan jaringan yang lainnya melalui suatu *penalty term* untuk terus menerus mengubah bobot-bobot mereka agar error yang dihasilkan oleh NNE mencapai minimum memperoleh keputusan akhir yang benar. Dengan demikian, untuk melatih NNE yang terdiri dari 3 buah jaringan, maka pelatihan yang dibutuhkan akan menjadi 3 kali lipat waktu pelatihan yang dibutuhkan untuk melatih NN tunggal pada dataset yang sama. Dan, jika kita menggunakan 5 buah jaringan untuk membentuk NNE, maka waktu pelatihan pun akan menjadi 5 kali lipat.

Sedangkan pengaruh ukuran dataset terhadap lamanya waktu pelatihan dapat dilihat pada contoh kasus dataset *Vowel* yang membutuhkan waktu pelatihan paling lama dibandingkan dataset-dataset lainnya. Satu hal yang perlu diketahui bahwa *Vowel* memiliki ukuran dataset yang terbesar diantara dataset lainnya (sebanyak 990 data, 90 data per kelasnya). Urutan kedua dan ketiga terbesar ditempati oleh dataset *Credit-card* sebanyak 614 data, 307 data per kelasnya) dan *Breast-cancer* (sebanyak 482 data, 241 data per kelasnya) yang membutuhkan waktu lebih lama dibandingkan dataset lainnya.

Terakhir mengenai jumlah epoch yang berpengaruh terhadap waktu pelatihan. Seperti yang telah diketahui dari tabel 3.3 sebelumnya bahwa jaringan yang mampu mencapai kondisi konvergen akan membutuhkan epoch yang lebih sedikit daripada jaringan yang tidak. Banyaknya epoch secara langsung berkaitan dengan lamanya waktu pelatihan yaitu bahwa semakin banyak epoch berarti semakin lama waktu pelatihan yang dibutuhkan oleh jaringan untuk menuju keadaan konvergen. Dengan demikian, jaringan yang tidak mampu menuju konvergen otomatis akan membutuhkan epoch maksimum (sebanyak 20.000 epoch) untuk menghentikan proses pelatihan dan karenanya membutuhkan waktu pelatihan yang lebih lama dibandingkan jaringan lain yang telah mencapai kondisi konvergen sebelum epoch pelatihan mencapai 20.000.

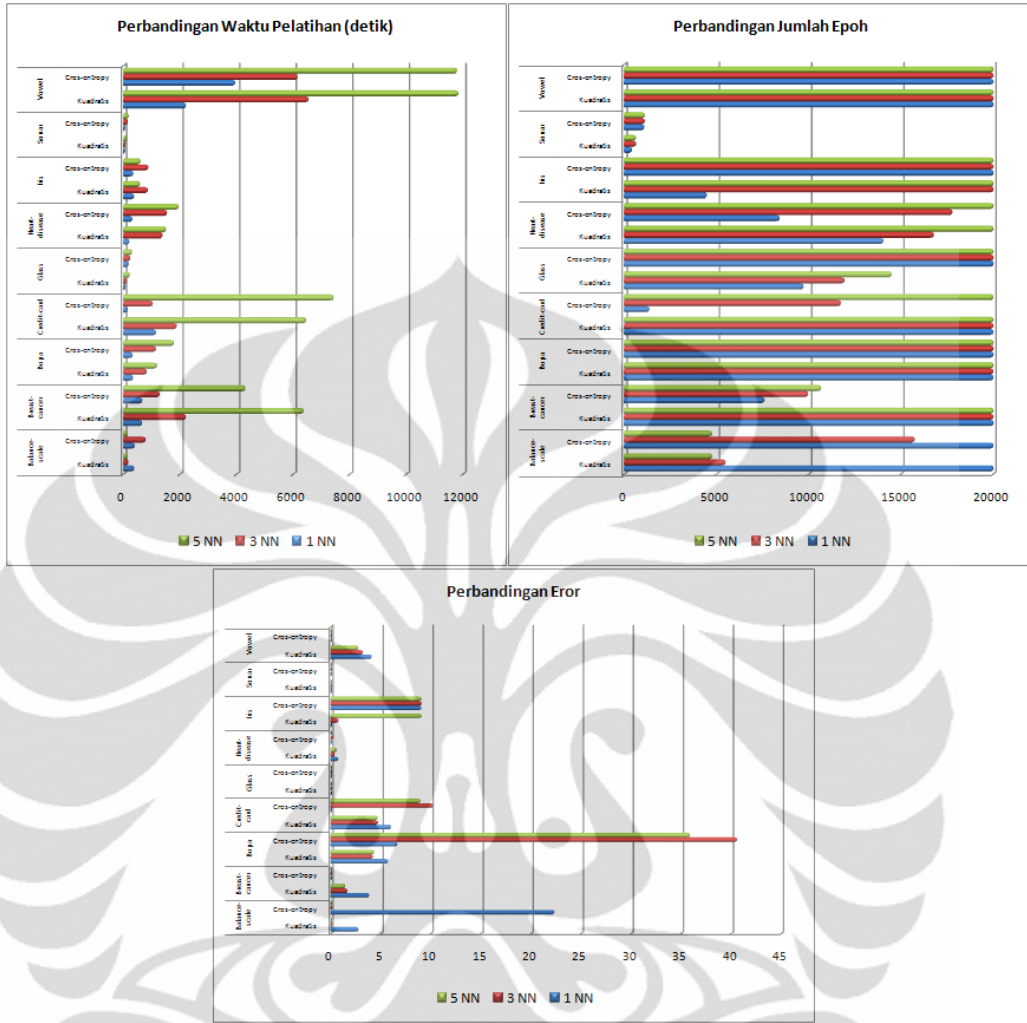
Dilihat dari rata-rata lama waktu pelatihan yang diperlukan, dari tabel 3.4 dapat disimpulkan bahwa pelatihan menggunakan fungsi eror *cross entropy* akan membutuhkan waktu pelatihan yang cukup setara (kurang lebih sama) daripada pelatihan dengan menggunakan fungsi eror kuadratis. Perlu diketahui bahwa hasil penelitian menunjukkan pelatihan NN tunggal dengan fungsi eror *cross entropy* membutuhkan waktu pelatihan lebih lama daripada NN tunggal dengan fungsi eror kuadratis, tetapi sebaliknya pelatihan NNE 3 jaringan dengan fungsi eror *cross entropy* membutuhkan waktu pelatihan lebih singkat daripada NNE dengan fungsi eror kuadratis.

Perbandingan derajat pengenalan data *training* yang disajikan pada tabel 3.5 menunjukkan bahwa secara umum NN tunggal dan NNE telah mampu mengenali data *training* dengan baik. Hal ini dibuktikan dengan pencapaian *recognition rate* yang sangat tinggi (berkisar antara 95-100%) untuk semua pengenalan dataset. Untuk

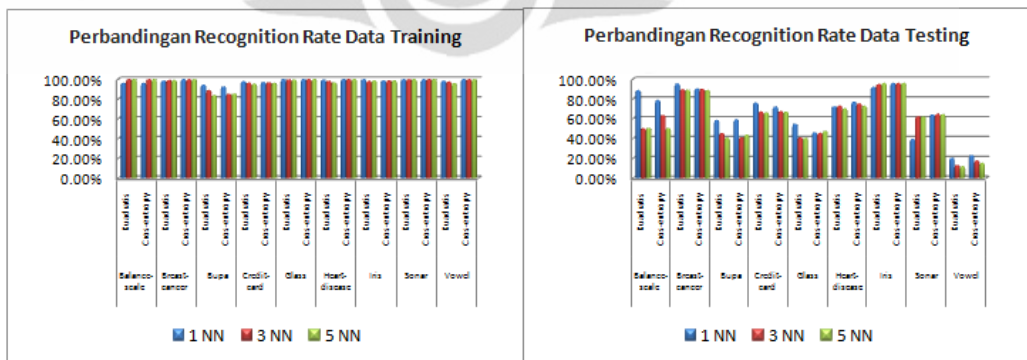
dataset *Sonar*, pengenalan data *training* selalu mencapai angka 100% untuk semua kasus NN tunggal dan NNE. Sedangkan dataset *Breast-cancer*, *Glass*, dan *Vowel* mampu mencapai *recognition rate* 100% apabila jaringan (NN tunggal dan NNE) dilatih dengan fungsi eror *cross entropy*, tetapi *recognition rate* tersebut akan menurun jika jaringan dilatih dengan fungsi eror kuadratis. Lain halnya dengan dataset *Balance-scale* yang hanya mampu mencapai *recognition rate* 100% apabila digunakan NNE (dengan 3 maupun 5 jaringan), tetapi tidak mampu mencapai *recognition rate* tersebut jika NN tunggal yang digunakan. Dilihat dari rata-rata *recognition rate* data *training* yang dihasilkan, dari tabel 3.5 dapat disimpulkan bahwa pelatihan menggunakan fungsi eror *cross entropy* akan menghasilkan *recognition rate* yang lebih tinggi daripada pelatihan dengan menggunakan fungsi eror kuadratis dan hal ini berlaku baik untuk NN tunggal maupun NNE.

Perbandingan derajat pengenalan data *testing* yang disajikan pada tabel 3.6 menunjukkan bahwa secara umum terjadi penurunan *recognition rate* pada semua dataset apabila dibandingkan dengan hasil pengenalan data *training* seperti yang tersaji dalam tabel 3.5 sebelumnya. Meskipun *recognition rate* data *testing* yang dicapai masih cukup tinggi (di atas 50%) pada mayoritas dataset (*Balance-scale*, *Breast-cancer*, *Credit-card*, *Heart-disease*, *Iris*, dan *Sonar*), tetapi pada dataset yang lain (*Bupa*, *Glass*, *Vowel*) penurunan *recognition rate* terbilang cukup signifikan. Dataset *Heart*, *Sonar* dan *Vowel* mampu mencapai *recognition rate* yang lebih tinggi apabila NN tunggal dan NNE dilatih dengan fungsi eror *cross entropy*, dibandingkan *recognition rate* dengan fungsi eror kuadratis. Lain halnya dengan dataset *Glass* yang hanya mengalami peningkatan *recognition rate* apabila digunakan NNE dengan fungsi eror *cross entropy*, tetapi peningkatan tersebut tidak terjadi pada NN tunggal.

Dilihat dari rata-rata *recognition rate* data *testing* yang dihasilkan, dari tabel 3.6 dapat disimpulkan bahwa pelatihan menggunakan fungsi eror *cross entropy* akan menghasilkan *recognition rate* yang lebih tinggi daripada pelatihan dengan menggunakan fungsi eror kuadratis dan hal ini berlaku baik untuk NN tunggal maupun NNE. Hal ini membuktikan bahwa jaringan yang dilatih dengan fungsi eror *cross entropy* memiliki kemampuan generalisasi yang lebih tinggi daripada jaringan yang dilatih dengan fungsi eror kuadratis, karena itu jaringan mampu mengenali pola data baru meskipun pola data tersebut tidak diikutkan dalam proses pelatihan.



Gambar 3.3. Perbandingan Error, Epoch, Waktu Pelatihan NN Tunggal dan NNE untuk Dataset UCI



Gambar 3.4. Recognition Data Training NN Tunggal dan NNE untuk Dataset UCI

3.3. Penutup

Bagian Penutup ini diawali dengan kesimpulan penelitian sistem pengenalan pola (*pattern recognition system*) yang dilakukan Penulis, kemudian diikuti dengan saran dari Penulis untuk penelitian selanjutnya terkait dengan masalah sistem pengenalan pola terutama penelitian yang menggunakan *benchmark* dataset UCI.

3.3.1. Kesimpulan Penelitian

Penelitian ini mengemukakan mengenai sistem pengenalan pola menggunakan NN tunggal dan NNE (3 dan 5 buah jaringan) berbasis fungsi eror *cross entropy* dan fungsi eror kuadratis. Terdapat 9 pola dataset UCI yang digunakan dalam penelitian ini yaitu *Balance-scale*, *Breast-cancer*, *Bupa*, *Credit-card*, *Glass*, *Heart-disease*, *Iris*, *Sonar*, dan *Vowel*. NN tunggal dilatih dengan algoritma standar *Backpropagation*, sedangkan NNE dilatih dengan algoritma NCL dengan pengambilan keputusan akhir berupa metode rata-rata.

Dari percobaan diketahui bahwa pelatihan menggunakan fungsi eror *cross entropy* akan menghasilkan eror rata-rata yang lebih tinggi daripada pelatihan dengan menggunakan fungsi eror kuadratis. Hal ini dikarenakan terutama oleh eror dataset *Bupa* yang secara mencolok jauh lebih tinggi melebihi eror dataset-datset lainnya. Meskipun demikian, fungsi eror *cross entropy* menghasilkan lebih banyak jaringan yang konvergen (baik NN tunggal maupun NNE) daripada metode fungsi eror kuadratis. Jaringan yang mampu mencapai kondisi konvergen akan membutuhkan epoch yang lebih sedikit daripada jaringan yang tidak. Penambahan jumlah epoch tidak berpengaruh signifikan pada penurunan eror apabila grafik eror telah mencapai keadaan saturasi.

Dari hasil penelitian diketahui bahwa pelatihan menggunakan fungsi eror *cross entropy* membutuhkan epoch dan waktu pelatihan yang setara (kurang lebih sama) jika dibandingkan dengan pelatihan menggunakan fungsi eror kuadratis. Peningkatan durasi waktu pelatihan dipengaruhi oleh peningkatan jumlah jaringan, besarnya ukuran dataset, serta banyaknya epoch pelatihan.

Dari percobaan pengenalan pola data UCI yang dilakukan, tingkat pengenalan yang dihasilkan cukup tinggi. Pola dataset yang paling mudah dikenali adalah *Iris*, sedangkan pola yang paling sulit dikenali adalah *Vowel*. Tingkat

pengenalan rata-rata untuk data *training* adalah 97.45% (dengan fungsi eror kuadratis) dan 98.19% (dengan fungsi eror *cross entropy*). Tingkat pengenalan rata-rata untuk data *testing* adalah 61.65% (dengan fungsi eror kuadratis) dan 63.58% (dengan fungsi eror *cross entropy*). Dengan demikian, dapat disimpulkan bahwa derajat pengenalan yang dihasilkan oleh NN dan NNE yang dilatih dengan fungsi eror *cross entropy* mengungguli derajat pengenalan yang dihasilkan oleh NN dan NNE yang dilatih dengan fungsi eror kuadratis.

3.3.2.Saran untuk Penelitian Selanjutnya

Penelitian selanjutnya mengenai sistem pengenal pola berbasis NNE adalah untuk mencari jumlah populasi dataset serta epoch pelatihan yang akan mengoptimalkan keseluruhan performa dataset, terutama dataset *Vowel* yang menghasilkan *recognition rate* paling rendah diantara dataset UCI lainnya, dan karenanya memerlukan penelitian lebih lanjut. Selain itu, penelitian untuk menentukan jumlah optimal NN tunggal yang membentuk NNE juga perlu dilakukan. Hal ini dikarenakan tingkat kesulitan dataset UCI yang berbeda-beda. Untuk problem dataset yang mudah, NN tunggal sebenarnya telah cukup digunakan untuk menghasilkan *recognition rate* yang tinggi. Namun untuk problem dataset yang lebih sulit, perlu dilakukan penambahan jumlah jaringan hingga membentuk sebuah NNE untuk memperbaiki kemampuan NN dalam mengenali pola .

Salah satu hal menarik yang perlu dicatat, yaitu bahwa aplikasi penelitian sistem pengenal pola dengan dataset UCI dapat diterapkan secara luas dalam berbagai bidang. Contohnya, dataset *Breast cancer*, *Bupa*, dan *Heart disease* untuk klasifikasi penyakit kanker payudara, kelainan hati, serta penyakit jantung dapat dilakukan secara langsung oleh para dermatologis tanpa memerlukan peralatan yang canggih ataupun mahal, cukup menggunakan PC (*personal computer*) yang harganya terjangkau. Pengenalan pola dataset *Credit-card* dengan NN dapat digunakan di bank-bank untuk menentukan apakah aplikasi kartu kredit dari seorang nasabah akan diterima atau ditolak. Pengenalan pola dataset *Glass* untuk klasifikasi jenis kaca untuk tujuan investigasi kriminologi jenis kaca yang tertinggal sebagai barang bukti di TKP (Tempat Kejadian Perkara). Selain itu, pengenalan pola dataset *Vowel* dapat diaplikasikan secara luas untuk kebutuhan pengenalan suara (*speech recognition*).

BAB IV

SISTEM PENGENAL WAJAH

BERBASIS NEURAL NETWORK ENSEMBLE

DENGAN FUNGSI EROR CROSS-ENTROPY DAN KUADRATIS

Bab ini menjelaskan tentang eksperimen yang dilakukan Penulis dengan membangun sebuah sistem pengenalan wajah (*face recognition system*) berbasis *Neural Network Ensemble* (NNE) yang dilatih dengan algoritma *Negative Correlated Learning* (NCL) dengan penghitungan eror berdasarkan fungsi eror kuadratis atau *Mean square error* (MSE) dan penghitungan eror berdasarkan fungsi eror *cross-entropy*.

4.1. Pendahuluan

Bagian Pendahuluan diawali dengan latar belakang penelitian pengenalan wajah dengan citra wajah inframerah dan citra wajah cahaya tampak, kemudian diikuti dengan penjelasan skema penelitian (*experimental setup*) sistem pengenalan wajah yang dibangun oleh Penulis.

4.1.1. Latar Belakang Penelitian

Manusia mampu membedakan dan mengenali ribuan wajah yang berbeda. Di masa moderen ini mesin dan komputer pun dirancang dengan suatu sistem cerdas agar mampu melakukan pengenalan wajah seperti halnya yang dilakukan oleh otak manusia. Teknologi pengenalan wajah semakin dimanfaatkan dan dikembangkan dengan aplikasi yang luas cakupannya, seperti misalnya sistem keamanan (*security system*) yang dapat mengizinkan orang tertentu mengakses area terbatas (*restricted area*), sistem presensi pegawai, serta dalam bidang militer seperti contohnya kamera *surveillance* yang dipasang dalam pesawat tanpa awak (*Unmanned Aerial Vehicle* (UAV)).

Dalam penglihatan komputer (*computer vision*), sistem pengenalan biasanya terdiri dari beberapa sensor, sebuah galeri model dari objek-objek, serta sebuah kemampuan membuat keputusan (*decision-making ability*). Ketika sebuah objek

dilihat oleh sebuah sensor, citra digital akan diproses sedemikian hingga citra tersebut direpresentasikan dengan cara yang serupa dengan model dalam galeri *database*. Kemudian, algoritma pengenalan akan mencari pasangan model input yang paling cocok (mirip atau sesuai) dengan model objek.

Secara umum, ada tiga macam tugas utama yang dilakukan oleh pengenalan wajah [35], yaitu:

1. Verifikasi: “*Am I who I say I am?*”

Seseorang memberikan biometriknya dan mengklaim identitasnya. Sistem kemudian membandingkan biometrik yang diberikan dengan biometrik yang tersimpan dalam database dari identitas yang diklaim tersebut. Berdasarkan hasil perbandingan biometrik tersebut, sistem akan memutuskan apakah ia menerima atau menolak klaim orang tersebut.

2. Identifikasi: “*Who am I?*”

Sebuah gambar dari seseorang yang tidak dikenali (*unknown person*) diberikan ke sistem pengenal wajah, dengan asumsi bahwa kita telah tahu terlebih dahulu bahwa orang tersebut ada di dalam database. Sistem kemudian membandingkan antara gambar dari orang tak dikenal tersebut dengan gambar orang dikenal (*known person*) yang ada di dalam database.

3. Daftar Pengawasan: “*Are you looking for me?*”

Sebuah sistem pengenal wajah harus terlebih dahulu mendeteksi apakah seseorang ada atau tidak di dalam daftar pengawasan (*watch list*). Jika orang tersebut ada di dalam daftar pengawasan, maka sistem harus mengidentifikasi orang tersebut dengan benar.

Dalam penelitian ini, Penulis fokus pada tugas yang kedua, yaitu tugas identifikasi. Penulis membangun sebuah Sistem Pengenal Wajah yang berbasis NNE dengan algoritma NCL dan membandingkan performa antara sistem pengenal wajah dengan menggunakan gambar dari kamera inframerah (*infra red*) dengan sistem pengenal wajah yang menggunakan gambar kamera cahaya tampak (*visible-light*).

Pengenalan wajah (*face recognition*) menggunakan metode kecerdasan komputasi (*computational intelligence*) merupakan sebuah subjek aktivitas riset yang aktif di kalangan komunitas peneliti penglihatan komputer (*computer vision*). Namun demikian, pengenalan wajah dalam domain kamera infra merah mendapatkan perhatian yang lebih sedikit dari kalangan peneliti dibandingkan dengan citra wajah dalam cahaya tampak. Citra kamera cahaya tampak sangat terpengaruh terhadap variasi kondisi penerangan (*lighting*) sehingga hal tersebut tampaknya menjadi hambatan bagi *robustness* algoritma pengenalan wajah. Lain halnya dengan kamera infra-merah yang mengukur emisivitas panas dari permukaan wajah dan citra yang dihasilkan relative stabil terhadap variasi iluminasi. Oleh karena itu, infra-merah mempunyai kemampuan untuk identifikasi dalam segala tingkat penerangan, baik pagi hari (*day-time*) maupun malam hari (*night time*), termasuk dalam keadaan gelap total [17]. Pengenalan dengan citra cahaya tampak lebih baik dibandingkan dengan citra inframerah jika gambar diambil di dalam ruangan (*indoor*). Namun jika gambar diambil di luar ruangan (*outdoor*), maka citra inframera menunjukkan derajat pengenalan yang lebih baik [19]. Hal ini dikarenakan gambar yang diambil di luar ruangan memiliki variasi cahaya yang lebih besar daripada variasi cahaya di dalam ruangan. Namun demikian, kamera infra-merah memiliki kekurangan dalam hal resolusi yang lebih rendah daripada kamera dengan spektrum cahaya tampak [18].

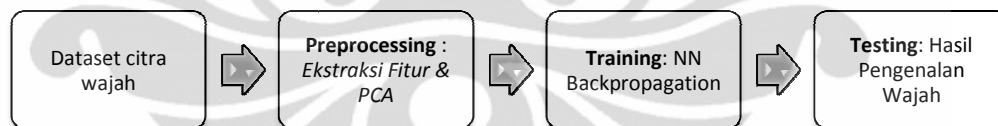
Penelitian awal mengenai pengembangan sistem pengenalan wajah dengan citra inframerah telah dilakukan pada tahun 2010 di Laboratorium Teknik Kendali, Departemen Teknik Elektro, Fakultas Teknik Universitas Indonesia. Dalam penelitian tersebut, sistem pengenalan wajah dibangun menggunakan NNE 4 skema yang berbeda, yaitu (1) algoritma backpropagation, (2) algoritma NCL, (3) parsialisasi data, dan (4) gabungan kedua algoritma tersebut. Hasil yang ditinjau dari *recognition rate* menunjukkan NNE, yang terdiri dari sejumlah NN, memberikan performa yang lebih baik dibandingkan sebuah NN tunggal. Selain itu, semakin banyak jumlah NN yang digunakan untuk menyusun NNE, maka semakin tinggi *recognition rate* yang dihasilkan [2].

Dalam rangka menguji kemampuan sistem pengenalan wajah untuk menyelesaikan persoalan di atas, maka penulis mengajukan dua pendekatan yang akan dibahas dan diimplementasikan dalam penelitian skripsi ini. Pendekatan

pertama dilakukan dengan menambah dataset penelitian, tidak hanya menggunakan gambar dari kamera inframerah saja, tetapi juga menggunakan gambar dari kamera cahaya tampak dengan tujuan untuk membandingkan performa diantara keduanya. Pendekatan yang kedua berkaitan dengan metode penghitungan fungsi eror yang dipakai. Dalam penelitian sebelumnya, hanya digunakan fungsi eror kuadratis dalam proses pelatihan NN. Sedangkan dalam penelitian ini, Penulis menggunakan baik fungsi eror kuadratis maupun fungsi eror *cross-entropy*. Selain itu, jika pada penelitian sebelumnya percobaan hanya dilakukan satu kali, pada penelitian ini percobaan dilakukan sebanyak 10 kali kemudian diambil hasil rata-ratanya.

4.1.2. Skema Penelitian

Permasalahan dalam penelitian ini berkaitan dengan penggunaan NN tunggal dengan algoritma *Backpropagation* dan NNE (3 buah 5 buah) dengan algoritma NCL sebagai sistem pengenalan wajah untuk mendiskriminasi citra wajah inframerah dan citra wajah cahaya tampak ke dalam kelas-kelas yang sesuai. Masing-masing NN akan dilatih (secara terpisah) dengan menggunakan dua fungsi eror yaitu fungsi eror kuadratis maupun fungsi eror *cross-entropy* untuk mendapatkan perbandingan karakteristik diantara keduanya. Masing-masing pelatihan akan diulang sebanyak 10 kali untuk kemudian diambil nilai rata-ratanya.



Gambar 4.1. Diagram Skema Sistem Pengenal Wajah

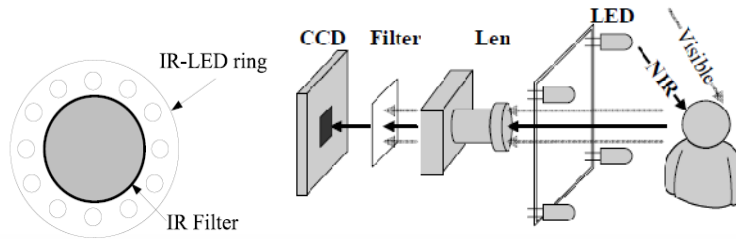
Seperti yang telah disebutkan sebelumnya, dalam penelitian ini, digunakan dua buah dataset, yaitu dataset citra wajah yang dihasilkan oleh kamera inframerah serta dataset citra wajah dari kamera cahaya tampak. Kedua dataset tersebut berisi foto wajah frontal dari 10 orang objek (berikutnya disebut sebagai *kelas*) dengan berbagai ekspresi, antara lain ekspresi netral, sedih, senang dengan posisi bibir terbuka (senyum) maupun bibir tertutup. (Keterangan lengkap mengenai kedua dataset wajah ini dapat dilihat di bagian Lampiran).

Perlu dicatat, bahwa ukuran pixel serta jumlah sampel pada kedua dataset tersebut tidak sama. Citra inframerah berukuran 40x30 pixel dan terdiri dari 200 sampel gambar sedangkan citra cahaya tampak berukuran 30x30 pixel dan hanya terdiri dari 100 sampel gambar. Selain itu, perlu diketahui bahwa meskipun kedua dataset terdiri dari jumlah kelas yang sama, yaitu 10 kelas, namun kesepuluh orang yang dijadikan objek gambar dalam dataset inframerah berbeda dengan objek gambar dalam dataset cahaya tampak.

Seperti pada penelitian pada bab sebelumnya, pada tahap *pre-processing* akan dilakukan dengan metode PCA untuk mereduksi dimensi dataset. Metode PCA pertama kali dijelaskan untuk representasi citra wajah oleh Kirby dan Sirovich [36] dan kemudian diadaptasi untuk pengenalan wajah oleh Turk dan Pentland [37]. Dimensi dataset citra infra merah yang berjumlah 1200 (40x30 pixel) dan dimensi dataset citra cahaya tampak yang berdimensi 900 (30x30 pixel) akan direduksi menjadi 50 dengan tujuan untuk penyederhanaan yang akan mempercepat waktu komputasi pada proses pelatihan. Berikut ini akan dipaparkan mengenai *experimental setup* untuk masing-masing dataset.


4.1.2.1. Skema Penelitian Citra Wajah Inframerah

Data citra wajah inframerah diambil menggunakan kamera CCD near IR (CCTV) yang 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 [30]. Supaya memperoleh kondisi pencahayaan yang stabil, kamera CCD ini juga dilengkapi dengan Infrared Filter untuk menghalangi cahaya tampak [31]. Jika dalam keadaan yang terang, gambar yang ditangkap kamera ini berwarna seperti layaknya kamera cahaya tampak (*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. Peralatan kamera CCD Infrared yang digunakan dalam pengambilan database citra wajah inframerah dapat dilihat pada gambar 5.1 di bawah ini [2] dan tidak dibahas lebih lanjut dalam penelitian skripsi ini.



Gambar 4.2. Sketsa dan Desain Kamera CCD Inframerah

Pada tahap *pre-processing*, gambar yang diperoleh di bentuk ke dalam suatu matriks di mana setiap elemen 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 untuk tujuan penyederhanaan. Dengan demikian setiap gambar yang berukuran 40x30 pixel tadi akan berubah menjadi matriks berukuran 40 x 30 di mana setiap elemen matriksnya adalah nilai input gambar. Selanjutnya, untuk memudahkan komputasi berikutnya maka setiap matriks yang berukuran 40 x 30 tersebut disusun ulang menjadi sebuah matriks yang berukuran 1 x 1200 tanpa mengubah nilai-nilai di dalamnya. Dengan demikian setiap gambar inframerah tersebut memiliki 1200 nilai fitur (dimensi).



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

Gambar 4.3. Foto Inframerah 40x30 pixel yang Dituliskan ke dalam Bentuk Matriks

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

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

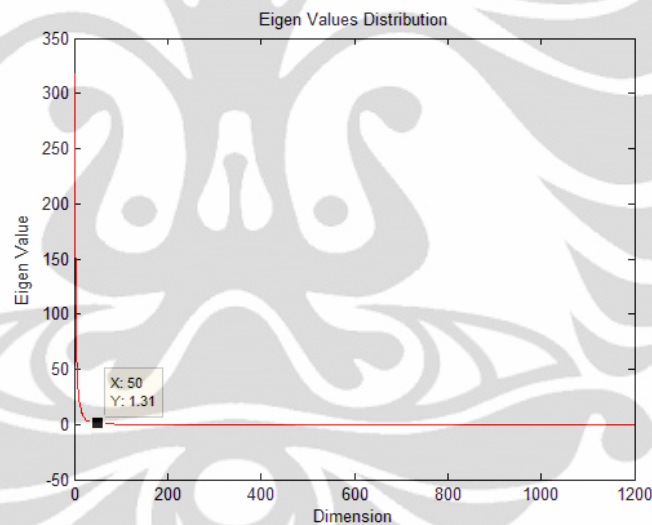
Gambar 4.4 Matriks Gambar Berukuran 40x30 Pixel yang Disusun menjadi Matriks Berukuran 1x1200



$$\begin{bmatrix}
 x_{1,1} & x_{1,2} & \cdots & x_{1,1200} \\
 x_{2,1} & x_{2,2} & \cdots & x_{2,1200} \\
 \vdots & \vdots & \ddots & \vdots \\
 x_{200,1} & x_{200,2} & \cdots & x_{200,1200}
 \end{bmatrix}$$

Gambar 4.5. Pembentukan Matriks Dataset Wajah Inframerah
Berukuran 200x1200 (200 sample x 1200 dimensi)

Setelah tahap ekstraksi fitur selesai, maka proses selanjutnya adalah mereduksi dimensi input yang besar 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 1200 dimensi dapat direduksi menjadi 50 dimensi.



Gambar 4.6. Distribusi Nilai Eigen Dataset Wajah Inframerah


Untuk keperluan tahap pelatihan dan pengujian, maka dataset citra wajah inframerah yang berjumlah 200 foto tadi dibagi menjadi 2 bagian sama besar, 100 foto digunakan sebagai data *training* dan 100 foto lainnya digunakan sebagai data *testing*.

4.1.2.2. Skema Penelitian Citra Wajah Cahaya Tampak

Data citra wajah cahaya tampak diambil menggunakan kamera cahaya tampak. Jika dalam keadaan yang terang, gambar yang ditangkap kamera ini

berwarna, namun ketika keadaan gelap, kamera ini tidak dapat menangkap gambar dengan jelas, apalagi jika kondisi pencahayaan di luar gelap total. Oleh karena itu, dapat dikatakan bahwa kualitas gambar yang dihasilkan sangat bergantung pada intensitas cahaya pada saat pengambilan gambar.

Sama seperti sebelumnya, pada tahap *pre-processing*, gambar yang diperoleh dikonversikan ke intensitas *Gray Scale* yang hanya memiliki satu dimensi nilai untuk tujuan penyederhanaan. Dengan demikian setiap gambar yang berukuran 30x30 pixel tadi akan berubah menjadi matriks berukuran 30 x 30 di mana setiap elemen matriksnya adalah nilai input gambar. Selanjutnya, untuk memudahkan komputasi berikutnya maka setiap matriks yang berukuran 30 x 30 tersebut disusun ulang menjadi sebuah matriks yang berukuran 1 x 900 tanpa mengubah nilai-nilai di dalamnya. Dengan demikian setiap gambar inframerah tersebut memiliki 900 nilai fitur (dimensi).




$$= \begin{bmatrix} x_{11} & \cdots & x_{1,30} \\ \vdots & \ddots & \vdots \\ x_{30,1} & \cdots & x_{30,30} \end{bmatrix}$$

Gambar 4.7. Foto Cahaya Tampak 30x30 pixel yang Dituliskan ke dalam Bentuk Matriks

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

$$= [x_{1,1} \cdots x_{1,900}]$$

Gambar 4.8. Matriks Gambar Berukuran 30x30 Pixel yang disusun menjadi Matriks Berukuran 1x900

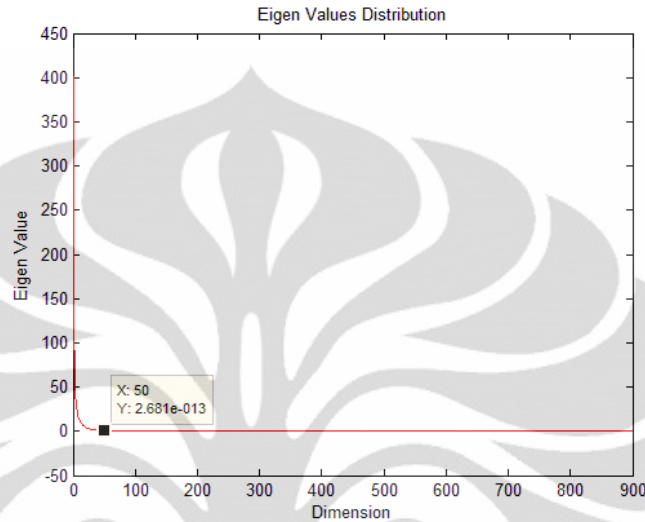


$$= \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,900} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,900} \\ \vdots & \vdots & \ddots & \vdots \\ x_{100,1} & x_{100,2} & \cdots & x_{100,900} \end{bmatrix}$$

Gambar 4.9. Pembentukan Matriks Dataset Wajah Cahaya Tampak berukuran 100x900 (100 sample x 900 dimensi)

Setelah tahap ekstraksi fitur selesai, maka proses selanjutnya adalah mereduksi dimensi input yang besar 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 50 dimensi.



Gambar 4.10. Distribusi Nilai Eigen Dataset Wajah Cahaya Tampak

Untuk keperluan tahap pelatihan dan pengujian, maka dataset citra wajah cahaya tampak yang berjumlah 100 foto tadi dibagi menjadi 2 bagian sama besar, 50 foto digunakan sebagai data *training* dan 50 foto lainnya digunakan sebagai data *testing*.

4.2. Hasil Penelitian dan Pembahasan

Pada bagian ini akan diuraikan mengenai hasil penelitian mengenai sistem pengenalan wajah. Tujuan utama penelitian ini yaitu membandingkan performa (peningkatan atau penurunan derajat pengenalan yang terjadi) antara jaringan yang dilatih menggunakan fungsi eror kuadratis dan jaringan yang dilatih menggunakan fungsi eror *cross-entropy*. Pada penelitian ini digunakan NN tunggal serta NNE sebanyak 3 dan 5 buah jaringan untuk mengetahui pengaruh penambahan jumlah jaringan terhadap karakteristik dari fungsi eror kuadratis dan fungsi eror *cross-entropy*. Tiap percobaan dilakukan sebanyak 10 kali kemudian diambil nilai rata-ratanya. Tabel yang disajikan dalam bagian Pembahasan ini merupakan ringkasan nilai rata-rata yang didapat dari hasil percobaan.

4.2.1. Hasil Penelitian dengan Citra Wajah Infra-merah

Berikut ini adalah hasil pelatihan NN tunggal, NNE dengan 3 jaringan, dan NNE dengan 5 jaringan yang dilatih untuk mengenali pola dataset citra wajah inframerah.

Tabel 4.1. Ringkasan Hasil Pelatihan dengan Fungsi Error Kuadratis untuk Citra Inframerah

| Jumlah NN | Error | Epo | Time | Recognition Data Training | | | Recognition Data Testing | | |
|-----------|-------|--------|----------|---------------------------|------------|------------|--------------------------|------------|------------|
| | | | | Limit 0.85 | Limit 0.75 | Limit 0.65 | Limit 0.85 | Limit 0.75 | Limit 0.65 |
| 1 buah | 0.01 | 1044.2 | 25.8133 | 100.00% | 100.00% | 100.00% | 88.80% | 91.60% | 93.00% |
| 3 buah | 0.01 | 1390.6 | 231.7617 | 100.00% | 100.00% | 100.00% | 94.30% | 97.10% | 98.40% |
| 5 buah | 0.07 | 9735 | 2320.072 | 98.00% | 100.00% | 100.00% | 93.30% | 97.80% | 99.20% |

(Keterangan: hasil di atas merupakan hasil rata-rata setelah dilakukan 10 kali percobaan)

Tabel 4.2. Ringkasan Hasil Pelatihan dengan Fungsi Error Cross-entropy untuk Citra Inframerah

| Jumlah NN | Error | Epo | Time | Recognition Data Training | | | Recognition Data Testing | | |
|-----------|-------|--------|-----------|---------------------------|------------|------------|--------------------------|------------|------------|
| | | | | Limit 0.85 | Limit 0.75 | Limit 0.65 | Limit 0.85 | Limit 0.75 | Limit 0.65 |
| 1 buah | 0.01 | 5707.8 | 327.6578 | 100.00% | 100.00% | 100.00% | 98.00% | 98.60% | 99.30% |
| 3 buah | 0.01 | 5736.6 | 925.482 | 100.00% | 100.00% | 100.00% | 98.60% | 99.50% | 100.00% |
| 5 buah | 0.01 | 5724.6 | 1484.6962 | 100.00% | 100.00% | 100.00% | 99.50% | 99.90% | 100.00% |

(Keterangan: hasil di atas merupakan hasil rata-rata setelah dilakukan 10 kali percobaan)

Tabel 4.3. Perbandingan Jumlah Error, Epo, dan Waktu Pelatihan untuk Citra Inframerah

| Jumlah NN | Error | | Jumlah Epo | | Waktu Pelatihan (detik) | |
|-----------|-----------|--------------|------------|--------------|-------------------------|--------------|
| | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy |
| 1 buah | 0.01 | 0.01 | 1044 | 5708 | 25.81 | 327.66 |
| 3 buah | 0.01 | 0.01 | 1391 | 5737 | 231.76 | 925.48 |
| 5 buah | 0.07 | 0.01 | 9735 | 5725 | 2320.07 | 1484.70 |

Tabel 4.4. Perbandingan Recognition Rate NN Tunggal untuk Citra Inframerah

| Jumlah NN | Recognition rata-rata Data Training | | Recognition rata-rata Data Testing | |
|-----------|-------------------------------------|--------------|------------------------------------|--------------|
| | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy |
| 1 buah | 100% | 100% | 91.13% | 98.63% ■ |
| 3 buah | 100% | 100% | 96.60% | 99.36% ■ |
| 5 buah | 99.13% | 100% | 96.76% | 99.80% ■ |

Keterangan: tanda (■) menunjukkan recognition rate yang lebih tinggi (peningkatan $\geq 1\%$)

Berdasarkan tabel 4.3, terlihat bahwa semua NN dan NNE yang dilatih untuk mengenali pola dataset citra wajah inframerah mampu mencapai kondisi konvergen, kecuali NNE (5 buah jaringan) yang dilatih dengan fungsi eror kuadratis yang rata-rata erornya masih mencapai 0.07 dengan 9735 epoh.

Apabila jaringan mampu mencapai kondisi konvergen, penambahan jumlah jaringan tampaknya tidak mempengaruhi jumlah epoh, tetapi mempengaruhi lamanya waktu pelatihan. Hal ini dapat dijelaskan sebagai berikut. Untuk NN tunggal, 1 epoh pelatihan hanya dilakukan untuk sebuah jaringan. Sedangkan pada NNE, pada setiap epoh nya dilakukan pelatihan beberapa jaringan sekaligus (secara *simultaneous*). Dengan demikian waktu yang dibutuhkan oleh NNE untuk melalui 1 epoh akan berbanding lurus dengan jumlah jaringan yang menyusunnya. Semakin banyak jumlah jaringan, maka waktu pelatihan yang diperlukan akan semakin lama. Lebih lanjut lagi, jika dibandingkan, ternyata NN yang dilatih dengan fungsi eror kuadratis membutuhkan epoh yang relatif lebih sedikit daripada NN yang dilatih dengan fungsi eror *cross entropy*.

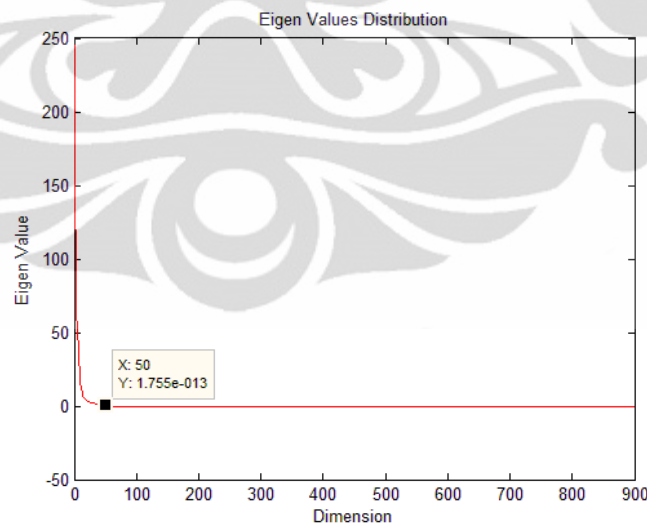
Jika ditinjau dari derajat pengenalan data *training* seperti tampak pada tabel 4.4, tampak bahwa semua NN tunggal dan NNE, baik yang dilatih dengan fungsi eror kuadratis maupun *cross entropy*, mampu mengenali data *training* dengan sangat baik dengan *recognition rate* setinggi 99-100%.

Masih mengenai tabel 4.4, tampak jelas bahwa fungsi eror *cross entropy* mengungguli fungsi eror kuadratis dalam hal pengenalan data *testing*. Hal ini terjadi baik pada kasus NN tunggal maupun NNE dengan 3 dan 5 jaringan. Namun demikian, apabila ditinjau dari perbedaan *recognition rate* yang dicapai maka dapat dikatakan bahwa perbedaan yang signifikan terjadi pada NN tunggal, dimana fungsi eror *cross entropy* menghasilkan *recognition rate* 7.5% lebih tinggi dibanding fungsi eror kuadratis, sedangkan pada NNE selisih *recognition rate* yang dihasilkan hanya sebesar 3%.

Meskipun secara umum peningkatan jumlah jaringan berdampak pada peningkatan *recognition rate*, tetapi peningkatan ini signifikan pada saat penambahan dari 1 ke 3 jaringan, sedangkan penambahan dari 3 ke 5 jaringan tidak membuahkan peningkatan yang signifikan, apalagi jika kita turut mempertimbangkan *computational cost* berupa waktu pelatihan yang berlipat ganda.

Seperti yang disebutkan pada bagian sebelumnya, Penulis bermaksud untuk membandingkan antara hasil penelitian pengenalan wajah dengan citra inframerah dan citra cahaya tampak. Namun demikian, hal ini sulit dilakukan karena 3 hal, yaitu: (1) Perbedaan ukuran pixel foto, citra inframerah berukuran 40x30 pixel sedangkan citra cahaya tampak berukuran 30x30 pixel; (2) Perbedaan ukuran dataset, dataset citra inframerah terdiri dari 200 sampel gambar sedangkan citra cahaya tampak hanya terdiri dari 100 sampel gambar, (3) Perbedaan objek yang diteliti, meskipun masing-masing dataset terdiri dari gambar 10 orang, tetapi 10 orang yang dijadikan objek gambar tersebut berbeda satu sama lain.

Karena pengambilan data ulang sulit dilakukan, oleh karena itu agar perbandingan menjadi lebih adil, maka Penulis akan mengatur ulang ukuran gambar (*resize*) citra infra merah agar ukurannya menjadi sama dengan citra cahaya tampak, yaitu 30x30 pixel. Selain itu, Penulis juga mengurangi jumlah sampel pada dataset menjadi setengah dari jumlah aslinya (100 gambar), agar jumlah sampel ini setara dengan jumlah sampel pada dataset citra cahaya tampak. Gambar 4.16 di bawah ini menunjukkan persebaran nilai eigen untuk dataset citra inframerah yang telah dimodifikasi agar berukuran 30x30 pixel dan berjumlah 100 sampel gambar.



Gambar 4.11. Distribusi Nilai Eigen Dataset Wajah Inframerah (yang telah dimodifikasi)

Di samping tujuan yang telah disebutkan di atas, percobaan ini juga bertujuan untuk mengetahui pengaruh ukuran dataset dan jumlah pixel pada gambar terhadap derajat pengenalan yang dicapai. Berikut ini adalah perbandingan hasil pelatihan NN untuk dataset citra wajah inframerah asli dan citra wajah inframerah modifikasi.

Tabel 4.5. Perbandingan Eror Hasil Pelatihan NN Tunggal dan NNE untuk Citra Inframerah

| Jumlah NN | Citra Wajah Inframerah (asli) | | Citra Wajah Inframerah (modifikasi) | |
|-----------|-------------------------------|--------------|-------------------------------------|--------------|
| | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy |
| 1 buah | 0.01 | 0.01 | 0.01 | 0.01 |
| 3 buah | 0.01 | 0.01 | 0.01 | 0.01 |
| 5 buah | 0.07 | 0.01 | 0.15 | 0.01 |

Tabel 4.6. Perbandingan Epoch hasil Pelatihan NN Tunggal dan NNE untuk Citra Inframerah

| Jumlah NN | Citra Wajah Inframerah | | Citra Wajah Inframerah (modifikasi) | |
|-----------|------------------------|--------------|-------------------------------------|--------------|
| | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy |
| 1 buah | 1044 | 5708 | 1546 | 5913 |
| 3 buah | 1391 | 5737 | 1928 | 5782 |
| 5 buah | 9735 | 5725 | 20000 | 5805 |

Tabel 4.7. Perbandingan Waktu Pelatihan (detik) NN Tunggal dan NNE untuk Citra Inframerah

| Jumlah NN | Citra Wajah Inframerah | | Citra Wajah Inframerah (modifikasi) | |
|-----------|------------------------|--------------|-------------------------------------|--------------|
| | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy |
| 1 buah | 25.81 | 327.66 | 43.52 | 168.67 |
| 3 buah | 231.76 | 925.48 | 65.92 | 198.93 |
| 5 buah | 2320.07 | 1484.70 | 1127.78 | 324.45 |

Tabel 4.8. Perbandingan Recognition Data Training NN Tunggal dan NNE untuk Citra Inframerah

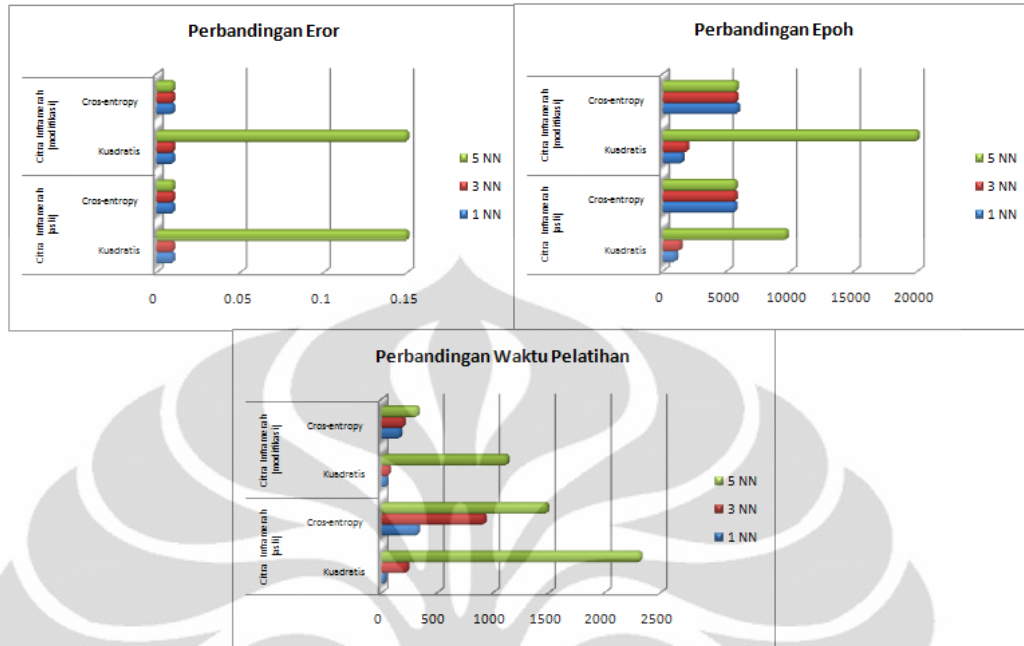
| Jumlah NN | Citra Wajah Inframerah | | Citra Wajah Inframerah (modifikasi) | |
|-----------|------------------------|--------------|-------------------------------------|--------------|
| | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy |
| 1 buah | 100% | 100% | 100.00% | 100.00% |
| 3 buah | 100% | 100% | 100.00% | 100.00% |
| 5 buah | 99.13% | 100% | 95.33% | 100.00% ■ |

Keterangan: tanda (■) menunjukkan recognition rate yang lebih tinggi (peningkatan $\geq 1\%$)

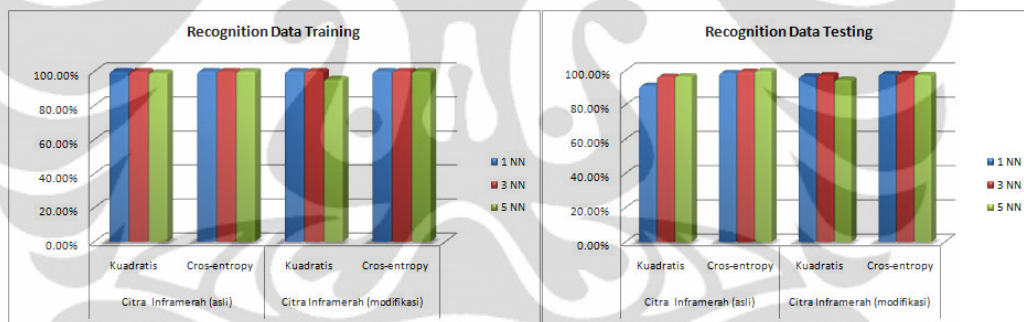
Tabel 4.9. Perbandingan Recognition Data Testing NN Tunggal dan NNE untuk Citra Inframerah

| Jumlah NN | Citra Wajah Inframerah | | Citra Wajah Inframerah (modifikasi) | |
|-----------|------------------------|--------------|-------------------------------------|--------------|
| | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy |
| 1 buah | 91.13% | 98.63% ■ | 96.67% | 98.00% ■ |
| 3 buah | 96.60% | 99.36% ■ | 97.40% | 98.13% |
| 5 buah | 96.76% | 99.80% ■ | 94.67% | 97.47% ■ |

Keterangan: tanda (■) menunjukkan recognition rate yang lebih tinggi (peningkatan $\geq 1\%$)



Gambar 4.12. Perbandingan Waktu Pelatihan NN Tunggal dan NNE untuk Citra Inframerah



Gambar 4.13. Recognition Rate Data Training NN Tunggal dan NNE untuk Citra Inframerah

Dari tabel 4.5. di atas terlihat bahwa penurunan jumlah sampel membuat eror pelatihan yang dicapai semakin besar pada NNE 5 jaringan yang dilatih dengan fungsi eror kuadratis. Sedangkan pada NN tunggal dan NNE lainnya yang telah mampu mencapai kondisi konvergen, penurunan jumlah sampel tidak berpengaruh terhadap eror yang dicapai.

Dari tabel 4.6 tampak bahwa penurunan jumlah sampel data dan penurunan resolusi piksel menyebabkan NN dan NNE membutuhkan epoch yang lebih banyak untuk dapat mencapai kondisi konvergen. Pada NN tunggal dan NNE yang dilatih dengan fungsi eror kuadratis, penambahan jumlah epoch cukup signifikan (sekitar 2 kali lipat). Sedangkan pada NN tunggal dan NNE yang dilatih dengan fungsi eror *cross entropy*, peningkatan jumlah epoch ini tidaklah signifikan jumlahnya. Hal ini

menunjukkan bahwa fungsi eror *cross entropy* lebih stabil terhadap penurunan jumlah sampel serta penurunan resolusi piksel gambar.

Dari tabel 4.7 tampak bahwa penurunan jumlah sampel data dan penurunan resolusi piksel menyebabkan NN dan NNE menghabiskan waktu pelatihan yang lebih sedikit daripada sebelumnya. Hal ini disebabkan karena penurunan jumlah sampel berdampak langsung pada berkurangnya pola data masukan yang harus dipelajari oleh jaringan. Semakin sedikit pola yang dipelajari, maka semakin cepat proses pelatihan berlangsung.

Dari tabel 4.8 dan 4.9 tampak bahwa penurunan jumlah sampel data dan penurunan resolusi piksel berdampak pada penurunan *recognition rate* baik pada data *training* maupun pada data *testing*. Hal ini dikarenakan penurunan jumlah sampel berdampak langsung pada berkurangnya pola data masukan yang harus dipelajari oleh jaringan. Semakin sedikit pola yang dipelajari, maka semakin berkurang kemampuan generalisasi jaringan dalam mengenali pola baru yang tidak diikutsertakan dalam pelatihan. Selain itu, penurunan resolusi gambar juga menyebabkan gambar menjadi semakin kabur. Akibatnya gambar menjadi lebih sulit dikenali daripada gambar asli yang memiliki resolusi lebih tinggi. Dari tabel 4.9, dapat dikatakan bahwa *recognition rate* yang dicapai oleh fungsi eror *cross entropy* mampu mengungguli fungsi eror kuadratis.

Pada tabel 4.9 tampak bahwa saat jumlah jaringan bertambah (dari 3 ke 5 jaringan), maka *recognition rate* untuk data *testing* tidak meningkat tetapi justru menurun. Penurunan *recognition rate* ini signifikan terjadi pada NNE dengan fungsi eror kuadratis. Penulis berhipotesis bahwa penurunan *recognition rate* tersebut berkaitan dengan nilai parameter penalti (γ) pada NCL yang pada awal penelitian ini telah ditentukan senilai $\gamma = 0.5$. Untuk menguji hipotesis tersebut, Penulis melakukan percobaan untuk mengetahui pengaruh perubahan nilai γ terhadap *recognition rate* yang dihasilkan. Dalam percobaan kali ini, inisialisasi bobot neuron dengan Nguyen-Widrow hanya dilakukan 1 kali pada awal percobaan, kemudian bobot-bobot tersebut disimpan untuk percobaan-percobaan berikutnya. Dengan demikian, faktor bobot random yang menyebabkan fluktuasi hasil *recognition rate* pun dapat dihilangkan sehingga perbedaan hasil *recognition rate* hanya hanya dipengaruhi oleh perbedaan nilai γ . Hasil pelatihan NNE dapat dilihat pada tabel 4.10 dan 4.11 di bawah ini.

Tabel 4.10. Pengaruh Nilai γ terhadap Recognition Rate NNE (3 jaringan) untuk Citra Inframerah

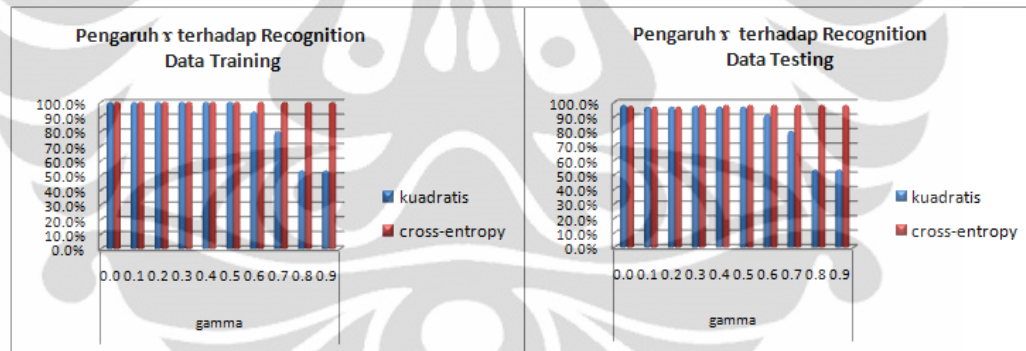
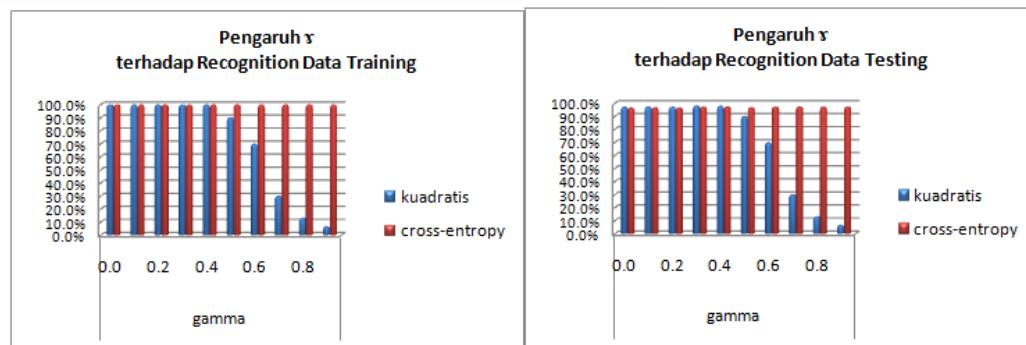
| Recognition | Fungsi Error | γ | | | | | | | | |
|---------------|---------------|----------|-------|-------|-------|-------|--------|--------|--------|--------|
| | | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| Data Training | kuadratis | 100% | 100% | 100% | 100% | 100% | 93.3% | 80% | 53.3% | 53.3% |
| | cross-entropy | 100% | 100% | 100% | 100% | 100% | 100% ■ | 100% ■ | 100% ■ | 100% ■ |
| Data Testing | kuadratis | 96.7% | 96.7% | 97.3% | 96.7% | 96.7% | 91.3% | 80% | 53.3% | 53.3% |
| | cross-entropy | 96.7% | 96.7% | 98% ■ | 98% ■ | 98% ■ | 98% ■ | 98% ■ | 98% ■ | 98% ■ |

Keterangan: tanda (■) menunjukkan recognition rate yang lebih tinggi (peningkatan $\geq 1\%$)

Tabel 4.11. Pengaruh Nilai γ terhadap Recognition Rate NNE (5 jaringan) untuk Citra Inframerah

| Recognition | Fungsi Error | γ | | | | | | | | |
|---------------|---------------|----------|-------|-------|-------|---------|---------|---------|---------|---------|
| | | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| Data Training | kuadratis | 100% | 100% | 100% | 100% | 90% | 70% | 30% | 13.3% | 6.7% |
| | cross-entropy | 100% | 100% | 100% | 100% | 100% ■ | 100% ■ | 100% ■ | 100% ■ | 100% ■ |
| Data Testing | kuadratis | 97.3% | 97.3% | 98% | 98% | 90% | 70% | 30% | 13.3% | 6.7% |
| | cross-entropy | 96.7% | 96.7% | 97.3% | 97.3% | 96.7% ■ | 97.3% ■ | 97.3% ■ | 97.3% ■ | 97.3% ■ |

Keterangan: tanda (■) menunjukkan recognition rate yang lebih tinggi (peningkatan $\geq 1\%$)

**Gambar 4.14.** Pengaruh γ terhadap Recognition Rate NNE (3 jaringan) untuk Citra Wajah Inframerah**Gambar 4.15.** Pengaruh γ terhadap Recognition Rate NNE (5 jaringan) untuk Citra Wajah Inframerah

Dari tabel 4.10, dapat dilihat bahwa nilai γ yang optimum untuk pelatihan NNE (3 jaringan) dengan dataset citra wajah inframerah didapatkan pada nilai $\gamma = 0.3$. Sedangkan dari tabel 4.11, dapat dilihat bahwa nilai γ yang optimum untuk pelatihan NNE (5 jaringan) didapatkan pada nilai $\gamma = 0.3$ dan $\gamma = 0.4$. Nilai ini menghasilkan *recognition rate* data testing yang paling tinggi dibandingkan nilai γ yang lain, baik untuk NNE dengan fungsi eror kuadratis maupun NNE dengan fungsi eror *cross entropy*. Dengan demikian, dapat disimpulkan bahwa semakin banyak jumlah jaringan yang menyusun NNE, maka semakin tinggi nilai γ yang diperlukan untuk mencapai *recognition rate* optimum. Hal ini disebabkan karena makin banyak jumlah jaringan, maka makin banyak bobot yang harus diperbaiki, untuk itulah diperlukan nilai γ yang semakin besar seiring dengan penambahan jumlah jaringan.

Secara umum peningkatan nilai γ tidak begitu berpengaruh terhadap variasi *recognition rate* data *training* untuk jaringan yang dilatih dengan fungsi eror *cross entropy*. NNE yang dilatih dengan fungsi *cross entropy* menghasilkan *recognition rate* data *training* yang sempurna (100%) berapapun nilai γ yang digunakan. Sedangkan NNE yang dilatih dengan fungsi kuadratis hanya mencapai *recognition rate* 100% jika digunakan $\gamma \leq 0.5$ (untuk NNE 3 jaringan) dan $\gamma \leq 0.4$ (untuk NNE 5 jaringan), lebih dari nilai tersebut maka *recognition rate* akan menurun drastis, jauh di bawah *recognition rate* yang dihasilkan oleh NNE dengan fungsi eror *cross entropy*.

Dari hasil percobaan pada tabel 4.10 dan 4.11 didapatkan bahwa nilai γ yang besar ($\gamma > 0.5$ untuk NNE 3 jaringan dan $\gamma > 0.4$ untuk NNE 5 jaringan) menghasilkan penurunan *recognition rate* yang signifikan, baik pada pengenalan data *training* maupun data *testing*. Namun hal ini tampaknya hanya berlaku pada NNE yang dilatih dengan fungsi kuadratis, sedangkan NNE yang dilatih dengan fungsi *cross entropy* menghasilkan *recognition rate* yang stabil dan tidak terpengaruh terhadap perubahan nilai γ . Hal ini juga menjelaskan tentang fenomena penurunan *recognition rate* yang terjadi pada saat penambahan jumlah jaringan dari NN tunggal ke NNE yang dilatih dengan fungsi eror kuadratis pada percobaan sebelumnya yang menggunakan nilai γ yang telah diset terlebih dahulu sebesar 0.5.

Berdasarkan percobaan di atas, kini diketahui bahwa NNE 3 jaringan akan menghasilkan *recognition rate* optimum sebesar 97.3% (untuk fungsi eror kuadratis) dan 98% (untuk fungsi eror *cross entropy*) pada nilai $\alpha = 0.3$ sedangkan NNE 5 jaringan menghasilkan *recognition rate* optimum sebesar 98% (untuk fungsi eror kuadratis) dan 97.3% (untuk fungsi eror *cross entropy*) baik pada nilai $\alpha = 0.3$ maupun $\alpha = 0.4$.

Dari hasil ini, dapat dikatakan bahwa penambahan jumlah jaringan tidaklah selalu berdampak terhadap peningkatan *recognition rate* (data testing). Hal ini disebabkan karena pada keputusan akhir pada NNE mempertimbangkan keputusan individual dari masing-masing NN tunggal yang menyusunnya. Metode pengambilan keputusan pada NNE dalam percobaan ini dilakukan dengan menggunakan metode rata-rata (*simple averaging*) sehingga tiap jaringan yang menyusun NNE memiliki hak suara yang sama terhadap keputusan akhir NNE, tidak peduli apakah jaringan tersebut membuat keputusan benar sehingga berkontribusi membuat keputusan akhir NNE menjadi benar, ataupun jika jika jaringan tersebut membuat keputusan salah sehingga berkontribusi membuat keputusan akhir NNE menjadi salah.

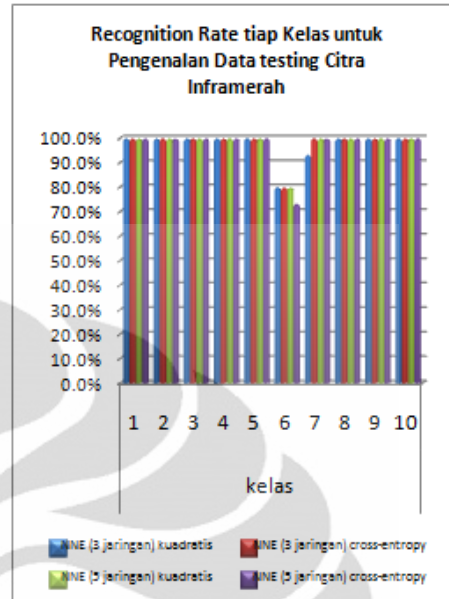
Dengan demikian, jika mayoritas jaringan dalam NNE memberikan keputusan yang benar, maka keputusan akhir NNE akan menjadi lebih benar, dan *recognition rate* pun meningkat. Namun sebaliknya, jika mayoritas jaringan membuat keputusan yang salah, maka keputusan akhir NNE pun akan menjadi salah, akibatnya *recognition rate* yang dihasilkan NNE pun akan menurun dibandingkan dengan *recognition rate* yang dihasilkan NN tunggal sendirian.

Untuk lebih jelasnya, di bawah ini akan disajikan tabel 4.12 yang berisi *recognition rate* per kelas yang didapatkan dari percobaan pengaruh perubahan α terhadap *recognition rate* yang dilakukan di atas. Penurunan *recognition rate* terjadi karena terdapat 2 kelas (kelas ke-6 dan ke-7) yang sulit dikenali oleh NNE 3 jaringan. Ketika jumlah jaringan ditambah, NNE 5 jaringan mampu mengenali kelas ke-7 dengan sempurna, tetapi kelas ke-6 tetap sulit dikenali. NNE 5 jaringan dengan fungsi eror kuadratis mengenali 80% dari anggota kelas ke-6, sedangkan NNE 5 jaringan dengan fungsi eror *cross entropy* hanya mengenali 73.3% anggota dari kelas ke-6. Hal inilah yang menyebabkan *recognition rate* NNE (5 jaringan) fungsi eror *cross entropy* menurun sebesar 0.7% daripada NNE (3 jaringan).

Tabel 4.12. Recognition Rate tiap kelas untuk Pengenalan Data Testing Citra Inframerah

| Kelas | NNE 3 jaringan ($\gamma = 0.3$) | | NNE 5 jaringan ($\gamma = 0.3$ dan 0.4) | |
|-----------|-----------------------------------|--------------|---|--------------|
| | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy |
| 1 | 100% | 100% | 100% | 100% |
| 2 | 100% | 100% | 100% | 100% |
| 3 | 100% | 100% | 100% | 100% |
| 4 | 100% | 100% | 100% | 100% |
| 5 | 100% | 100% | 100% | 100% |
| 6 | 80% | 80% | 80% | 73.3% |
| 7 | 93.3% | 100% | 100% | 100% |
| 8 | 100% | 100% | 100% | 100% |
| 9 | 100% | 100% | 100% | 100% |
| 10 | 100% | 100% | 100% | 100% |
| Rata-rata | 97.3% | 98% | 98% | 97.3% |

Keterangan: tanda (■) menunjukkan recognition rate yang lebih tinggi (peningkatan $\geq 1\%$)



Gambar 4.16. Recognition Rate tiap Kelas untuk Data Testing Citra Wajah Inframerah

4.2.2. Hasil Penelitian dengan Citra Wajah Cahaya Tampak

Berikut ini adalah hasil pelatihan NN tunggal, NNE dengan 3 jaringan, dan NNE dengan 5 jaringan yang dilatih untuk mengenali pola dataset citra wajah cahaya tampak.

Tabel 4.13. Ringkasan Hasil Pelatihan dengan Fungsi Error Kuadratis untuk Citra Cahaya Tampak

| Jumlah NN | Error | Epoh | Time | Recognition Data Training | | | Recognition Data Testing | | |
|-----------|-------|----------|--------|---------------------------|------------|------------|--------------------------|------------|------------|
| | | | | Limit 0.85 | Limit 0.75 | Limit 0.65 | Limit 0.85 | Limit 0.75 | Limit 0.65 |
| 1 buah | 0.01 | 1799.4 | 18.28 | 100.00% | 100.00% | 100.00% | 99.00% | 99.20% | 100.00% |
| 3 buah | 0.01 | 2583.2 | 69.57 | 100.00% | 100.00% | 100.00% | 92.20% | 93.20% | 95.00% |
| 5 buah | 0.06 | 14949.40 | 657.56 | 95.00% | 100.00% | 100.00% | 87.80% | 93.60% | 94.80% |

(Keterangan: hasil di atas merupakan hasil rata-rata setelah dilakukan 10 kali percobaan)

Tabel 4.14. Ringkasan Hasil Pelatihan dengan Fungsi Cross-entropy untuk Citra Cahaya Tampak

| Jumlah NN | Error | Epoh | Time | Recognition Data Training | | | Recognition Data Testing | | |
|-----------|-------|---------|---------|---------------------------|------------|------------|--------------------------|------------|------------|
| | | | | Limit 0.85 | Limit 0.75 | Limit 0.65 | Limit 0.85 | Limit 0.75 | Limit 0.65 |
| 1 buah | 0.01 | 9832.1 | 231.61 | 100.00% | 100.00% | 100.00% | 95.60% | 96.10% | 96.60% |
| 3 buah | 0.01 | 9858.7 | 626.73 | 100.00% | 100.00% | 100.00% | 95.80% | 96.10% | 97.00% |
| 5 buah | 0.01 | 10002.7 | 1819.85 | 100.00% | 100.00% | 100.00% | 94.60% | 95.40% | 95.60% |

(Keterangan: hasil di atas merupakan hasil rata-rata setelah dilakukan 10 kali percobaan)

Tabel 4.15. Perbandingan Jumlah Error, Epoch, dan Waktu Pelatihan Citra Cahaya Tampak

| Jumlah NN | Error | | Jumlah Epoch | | Waktu Pelatihan (detik) | |
|-----------|-----------|--------------|--------------|--------------|-------------------------|--------------|
| | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy |
| 1 buah | 0.01 | 0.01 | 1799 | 9832 | 18.28 | 231.61 |
| 3 buah | 0.01 | 0.01 | 2583 | 9859 | 69.57 | 626.73 |
| 5 buah | 0.06 | 0.01 | 14949 | 10003 | 657.56 | 1819.85 |

Tabel 4.16. Perbandingan Recognition Rate NN Tunggal dan NNE untuk Citra Cahaya Tampak

| Jumlah NN | Recognition rata-rata Data Training | | Recognition rata-rata Data Testing | |
|-----------|-------------------------------------|--------------|------------------------------------|--------------|
| | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy |
| 1 buah | 100.00% | 100.00% | 99.40% ■ | 96.10% |
| 3 buah | 100.00% | 100.00% | 93.46% | 96.30% ■ |
| 5 buah | 98.33% | 100.00% ■ | 92.07% | 95.20% ■ |

Keterangan: tanda (■) menunjukkan recognition rate yang lebih tinggi (peningkatan $\geq 1\%$)

Berdasarkan tabel 4.15, terlihat bahwa semua NN dan NNE yang dilatih untuk mengenali pola dataset citra wajah cahaya tampak mampu mencapai kondisi konvergen, kecuali NNE (5 buah jaringan) yang dilatih dengan fungsi error kuadratis yang rata-rata erornya masih mencapai 0.06 dengan epoch rata-rata sebanyak 14949.

Apabila jaringan mampu mencapai kondisi konvergen, penambahan jumlah jaringan tidak mempengaruhi jumlah epoch, tetapi mempengaruhi lamanya waktu pelatihan. Hal ini dapat dijelaskan sebagai berikut. Untuk NN tunggal, 1 epoch pelatihan hanya dilakukan untuk sebuah jaringan. Sedangkan pada NNE, pada setiap epoch nya dilakukan pelatihan beberapa jaringan sekaligus (secara *simultaneous*). Dengan demikian waktu yang dibutuhkan oleh NNE untuk melalui 1 epoch akan berbanding lurus dengan jumlah jaringan yang menyusunnya. Semakin banyak jumlah jaringan, maka waktu pelatihan yang diperlukan akan semakin lama. Lebih lanjut lagi, jika dibandingkan, ternyata NN yang dilatih dengan fungsi error kuadratis membutuhkan epoch yang relatif lebih sedikit daripada NN yang dilatih dengan fungsi error *cross entropy*.

Jika ditinjau dari derajat pengenalan data *training* seperti tampak pada tabel 4.16, tampak bahwa semua NN tunggal dan NNE, baik yang dilatih dengan fungsi error kuadratis maupun *cross entropy*, mampu mengenali data *training* dengan sangat baik dengan *recognition rate* setinggi 98-100%. Pada kasus NN tunggal, fungsi error kuadratis memberikan derajat pengenalan yang lebih tinggi daripada fungsi error

cross entropy. Namun demikian, pada NNE tampak bahwa fungsi eror *cross entropy* mengungguli fungsi eror kuadratis dalam hal pengenalan data *testing*. Hal ini dikarenakan NN mempunyai kemampuan generalisasi yang lebih baik daripada NN tunggal

Meskipun secara umum peningkatan jumlah jaringan berdampak pada peningkatan *recognition rate*, tetapi peningkatan ini signifikan. Bahkan pada kasus NN dan NNE yang dilatih dengan fungsi eror kuadratis, penurunan *recognition rate* justru terjadi saat jumlah jaringan ditambah, baik penambahan dari 1 ke 3 jaringan maupun dari 3 ke 5 jaringan. Hal ini disebabkan pada NNE pengambilan keputusan dilakukan dengan menggunakan metode rata-rata, dengan demikian jika mayoritas jaringan dalam NNE memberikan keputusan yang salah, maka keputusan akhir NNE pun akan menjadi salah, akibatnya *recognition rate* pun menurun.

Pada tabel 4.16, saat jumlah jaringan bertambah (dari 1 ke 3 jaringan ataupun dari 3 ke 5 jaringan), maka *recognition rate* untuk data *testing* tidak meningkat tetapi justru menurun. Penurunan *recognition rate* ini signifikan pada NNE dengan fungsi eror kuadratis. Penulis berhipotesis bahwa penurunan *recognition rate* tersebut berkaitan dengan nilai parameter penalti (γ) pada NCL yang pada awal penelitian ini telah ditentukan senilai $\gamma=0.5$. Untuk menguji hipotesis tersebut, Penulis melakukan percobaan untuk mengetahui pengaruh perubahan nilai γ terhadap *recognition rate* yang dihasilkan. Dalam percobaan ini, inialisasi bobot neuron dengan Nguyen-Widrow hanya dilakukan 1 kali pada awal percobaan, kemudian bobot-bobot tersebut disimpan untuk percobaan-percobaan berikutnya. Dengan demikian, faktor bobot random yang menyebabkan fluktuasi hasil *recognition rate* pun dapat dihilangkan sehingga perbedaan hasil *recognition rate* hanya dipengaruhi oleh perbedaan nilai γ . Hasil pelatihan NNE dapat dilihat pada tabel 4.17 dan 4.18 di bawah ini.

Tabel 4.17. Pengaruh Nilai γ terhadap Recognition NNE (3 jaringan) untuk Citra Cahaya Tampak

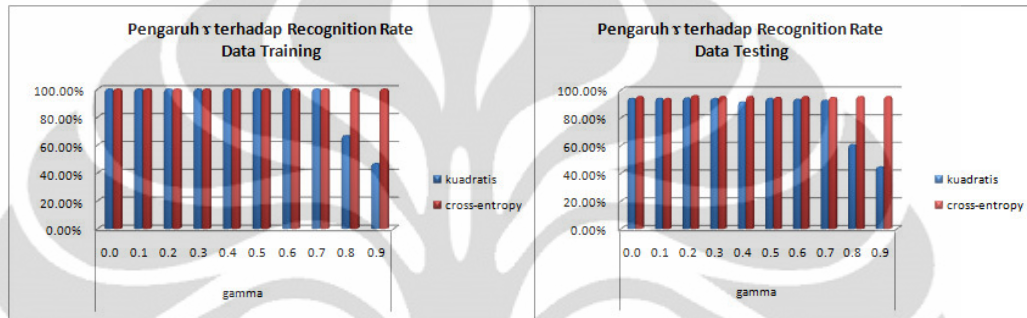
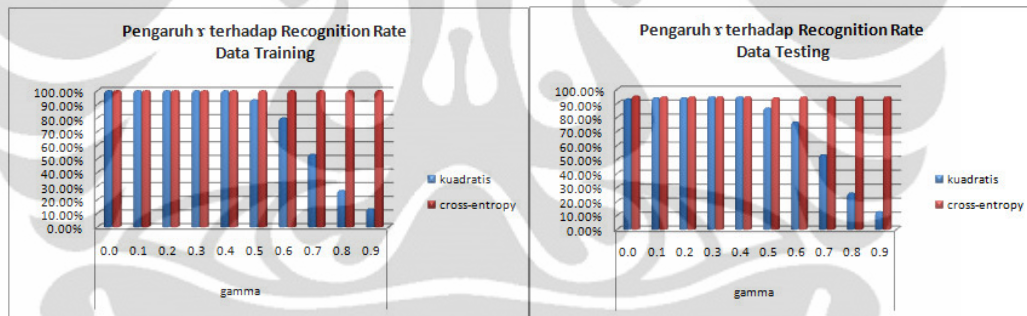
| Recognition | Fungsi Error | γ | | | | | | | | |
|---------------|---------------|----------|------|--------|--------|-------|--------|--------|--------|--------|
| | | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| Data Training | kuadratis | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 66.7% | 46.7% |
| | cross-entropy | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100%■ | 100%■ |
| Data Testing | kuadratis | 94% | 94% | 94% | 91.3% | 94% | 93.3% | 92.7% | 60.7% | 44.7% |
| | cross-entropy | 94% | 96%■ | 95.3%■ | 95.3%■ | 94.7% | 95.3%■ | 94.7%■ | 95.3%■ | 95.3%■ |

Keterangan: tanda (■) menunjukkan *recognition rate* yang lebih tinggi (peningkatan $\geq 1\%$)

Tabel 4.18. Pengaruh Nilai γ terhadap Recognition NNE (5 jaringan) untuk Citra Cahaya Tampak

| Recognition | Fungsi Error | γ | | | | | | | | |
|---------------|---------------|----------|-------|-------|-------|-------|-------|-------|-------|-------|
| | | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| Data Training | kuadratis | 100% | 100% | 100% | 100% | 93.3% | 80.0% | 53.3% | 26.7% | 13.3% |
| | cross-entropy | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% |
| Data Testing | kuadratis | 94.0% | 94.0% | 94.7% | 94.7% | 86.7% | 76.7% | 53.3% | 26.0% | 12.7% |
| | cross-entropy | 94.7% | 94.7% | 94.7% | 94.7% | 94.0% | 94.7% | 94.7% | 94.7% | 94.7% |

Keterangan: tanda (■) menunjukkan recognition rate yang lebih tinggi (peningkatan $\geq 1\%$)

**Gambar 4.17.** Pengaruh γ terhadap Recognition Rate NNE (3 jaringan) untuk Citra Cahaya Tampak**Gambar 4.18.** Pengaruh γ terhadap Recognition Rate NNE (5 jaringan) untuk Citra Cahaya Tampak

Dari tabel 4.17, diketahui bahwa nilai γ optimum untuk NNE (3 jaringan) didapatkan pada nilai $\gamma = 0.2$. Sedangkan dari tabel 4.18, dapat dilihat bahwa nilai γ optimum untuk pelatihan NNE (5 jaringan) didapatkan pada nilai $\gamma = 0.3$ dan $\gamma = 0.4$. Nilai ini menghasilkan *recognition rate* data *testing* yang paling tinggi dibandingkan nilai γ yang lain, baik untuk NNE dengan fungsi eror kuadratis maupun *cross entropy*. Dengan demikian, dapat disimpulkan bahwa semakin banyak jumlah jaringan yang menyusun NNE, maka semakin tinggi nilai γ yang diperlukan untuk mencapai *recognition rate* optimum. Hal ini disebabkan karena makin banyak jumlah jaringan, maka makin banyak bobot yang harus diperbaiki, untuk itulah diperlukan nilai γ yang semakin besar seiring dengan pertambahan jumlah jaringan.

Secara umum peningkatan nilai α tidak begitu berpengaruh terhadap variasi *recognition rate* data *training*, baik untuk jaringan yang dilatih dengan fungsi eror kuadratis maupun jaringan yang dilatih dengan fungsi *cross entropy*. NNE yang dilatih dengan fungsi *cross entropy* menghasilkan *recognition rate* data *training* yang sempurna (100%) berapapun nilai α yang digunakan. Sedangkan NNE yang dilatih dengan fungsi kuadratis mencapai *recognition rate* 100% jika digunakan $\alpha \leq 0.7$ (untuk NNE 3 jaringan) dan $\alpha \leq 0.4$ (untuk NNE 5 jaringan), lebih dari nilai tersebut maka *recognition rate* akan menurun drastis.

Dari hasil percobaan pada tabel 4.17 dan 4.18 didapatkan bahwa nilai α yang besar ($\alpha > 0.7$ untuk NNE 3 jaringan dan $\alpha > 0.4$ untuk NNE 5 jaringan) menghasilkan penurunan *recognition rate* yang signifikan, baik pada pengenalan data *training* maupun data *testing*. Hal ini terutama berlaku pada NNE dengan fungsi kuadratis, sedangkan NNE dengan fungsi *cross entropy* menghasilkan *recognition rate* yang lebih stabil terhadap perubahan nilai α . Hal ini menjelaskan tentang fenomena penurunan *recognition rate* yang terjadi pada saat penambahan jumlah jaringan dari NN tunggal ke NNE yang dilatih dengan fungsi eror kuadratis pada percobaan sebelumnya yang menggunakan nilai α yang telah diset terlebih dahulu sebesar 0.5.

Berdasarkan percobaan di atas, kini diketahui bahwa NNE 3 jaringan akan menghasilkan *recognition rate* optimum sebesar 94% (untuk fungsi eror kuadratis) dan 96% (untuk fungsi eror *cross entropy*) pada nilai $\alpha = 0.2$ sedangkan NNE 5 jaringan menghasilkan *recognition rate* optimum sebesar 94.7% (untuk fungsi eror kuadratis dan *cross entropy*) baik pada nilai $\alpha = 0.3$ maupun $\alpha = 0.4$.

Dari hasil ini, dapat dikatakan bahwa penambahan jumlah jaringan tidaklah selalu berdampak terhadap peningkatan *recognition rate* (data *testing*). Hal ini disebabkan karena pada keputusan akhir pada NNE mempertimbangkan keputusan individual dari masing-masing NN tunggal yang menyusunnya dengan metode pengambilan keputusan berdasarkan rata-rata. Jika mayoritas jaringan dalam NNE memberikan keputusan yang benar, maka keputusan akhir NNE akan menjadi lebih benar, dan *recognition rate* pun meningkat. Namun, jika mayoritas jaringan membuat keputusan yang salah, maka keputusan akhir NNE pun akan menjadi salah, akibatnya *recognition rate* yang dihasilkan NNE pun akan menurun dibandingkan dengan *recognition rate* yang dihasilkan NN tunggal sendirian.

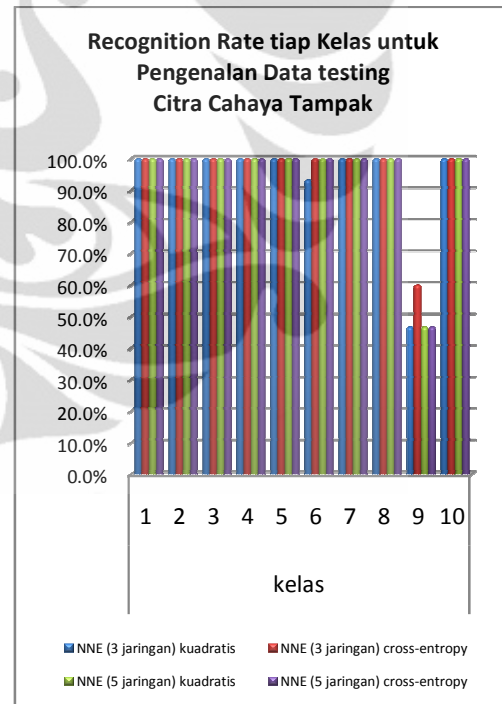
Untuk lebih jelasnya, di bawah ini akan disajikan tabel 4.19 yang berisi *recognition rate* (untuk pengenalan data *testing*) per kelas hasil pelatihan NNE (3 jaringan dan 5 jaringan) pada percobaan pengaruh perubahan γ terhadap *recognition rate* yang dilakukan di atas. Dari tabel 4.19 terlihat bahwa terdapat 2 kelas (kelas ke-6 dan ke-9) yang sulit dikenali oleh NNE (3 jaringan) dengan fungsi eror kuadratis dan hanya kelas ke-9 yang sulit dikenali oleh NNE (3 jaringan) fungsi *cross entropy*. Ketika jumlah jaringan ditambah, NNE (5 jaringan) telah mampu mengenali kelas ke-6 dengan sempurna, tetapi bagaimanapun kelas ke-9 tetap sulit untuk dikenali. NNE 5 jaringan hanya mampu mengenali 46.67 % dari anggota kelas ke-9 (setara dengan pengenalan NNE 3 jaringan fungsi eror kuadratis), tetapi pada kasus NNE yang dilatih dengan fungsi eror *cross entropy*, NNE (3 jaringan) sebelumnya telah mampu mengenali 60% anggota dari kelas ke-9. Hal inilah yang menyebabkan *recognition rate* NNE (5 jaringan) fungsi eror *cross entropy* menurun sebesar 1.3% daripada NNE (3 jaringan).

Tabel 4.19. Recognition Rate tiap kelas untuk Pengenalan Data Testing

Citra Cahaya Tampak

| Kelas | NNE 3 jaringan ($\gamma = 0.2$) | | NNE 5 jaringan ($\gamma = 0.3$ dan 0.4) | |
|------------------|--------------------------------------|--------------|--|--------------|
| | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy |
| 1 | 100% | 100% | 100% | 100% |
| 2 | 100% | 100% | 100% | 100% |
| 3 | 100% | 100% | 100% | 100% |
| 4 | 100% | 100% | 100% | 100% |
| 5 | 100% | 100% | 100% | 100% |
| 6 | 93.3% | 100%■ | 100% | 100% |
| 7 | 100% | 100% | 100% | 100% |
| 8 | 100% | 100% | 100% | 100% |
| 9 | 46.7% | 60%■ | 46.7% | 46.7% |
| 10 | 100% | 100% | 100% | 100% |
| Rata-rata | 94% | 96%■ | 94.7% | 94.7% |

Keterangan: tanda (■) menunjukkan *recognition rate* yang lebih tinggi (peningkatan $\geq 1\%$)



Gambar 4.19. Recognition Rate tiap Kelas untuk Data Testing Citra Cahaya Tampak

4.2.3. Ringkasan Hasil Penelitian Citra Inframerah dan Cahaya Tampak

Seperti yang telah disebutkan sebelumnya, bahwa perbandingan performa antara kedua dataset (secara langsung) sulit dilakukan karena perbedaan *properties* dari dataset citra wajah inframerah dan citra wajah cahaya tampak yang digunakan dalam penelitian ini. Diantaranya yaitu fakta bahwa kesepuluh orang (kelas) yang dijadikan objek dalam gambar pada dataset citra wajah inframerah berbeda dengan kesepuluh orang objek pada dataset citra wajah cahaya tampak. Sedangkan perbedaan lainnya yaitu mencakup perbedaan jumlah sampel data dan resolusi gambar dapat diatasi dengan cara membuang sebagian sampel agar jumlahnya menjadi sama (yaitu sebanyak 100 sampel foto) serta mengurangi resolusi dengan cara mengatur ulang ukuran gambar (*resize*) agar gambar sama-sama berukuran 30x30 piksel. Oleh karena itu perlu diketahui bahwa perbandingan yang dilakukan di sini bersifat relatif, hanya sebatas untuk mengetahui gambaran umum mengenai hasil penelitian yang telah dilakukan Penulis, dan bukanlah perbandingan yang bersifat secara umum.

Tabel 4.20. Perbandingan Error Hasil Pelatihan NN Tunggal dan NNE untuk Dataset Wajah

| Data set | NN tunggal | | NNE 3 jaringan | | NNE 5 jaringan | |
|---------------|------------|--------------|----------------|--------------|----------------|--------------|
| | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy |
| Inframerah | 0.01 | 0.01 | 0.01 | 0.01 | 0.15 | 0.01 |
| Cahaya Tampak | 0.01 | 0.01 | 0.01 | 0.01 | 0.06 | 0.01 |
| Rata-rata | 0.01 | 0.01 | 0.01 | 0.01 | 0.11 | 0.01 |

Tabel 4.21. Perbandingan Epoch Hasil Pelatihan NN Tunggal dan NNE untuk Dataset Wajah

| Data set | NN tunggal | | NNE 3 jaringan | | NNE 5 jaringan | |
|---------------|------------|--------------|----------------|--------------|----------------|--------------|
| | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy |
| Inframerah | 1546 | 5913 | 1928 | 5782 | 20000 | 5805 |
| Cahaya Tampak | 1799 | 9832 | 2583 | 9859 | 14949 | 10003 |
| Rata-rata | 1672.5 | 7872.5 | 2255.5 | 7820.5 | 17474.5 | 7904 |

Tabel 4.22. Perbandingan Waktu Pelatihan (detik) NN Tunggal dan NNE untuk Dataset Wajah

| Data set | NN tunggal | | NNE 3 jaringan | | NNE 5 jaringan | |
|---------------|------------|--------------|----------------|--------------|----------------|--------------|
| | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy |
| Inframerah | 43.52 | 168.67 | 65.92 | 198.93 | 1127.78 | 324.45 |
| Cahaya Tampak | 18.28 | 231.61 | 69.57 | 626.73 | 657.56 | 1819.85 |
| Rata-rata | 30.9 | 200.14 | 67.75 | 412.83 | 892.67 | 1072.15 |

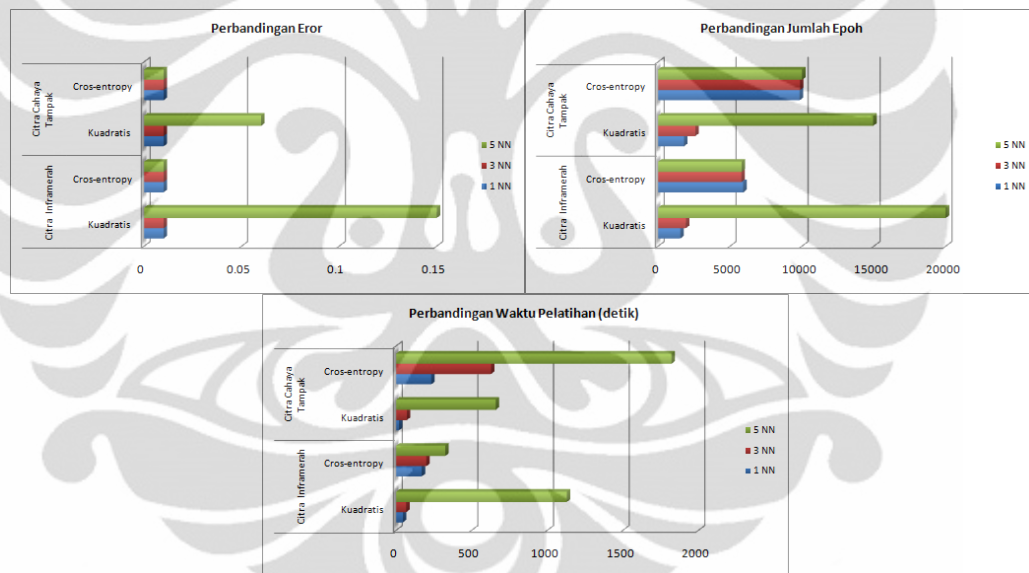
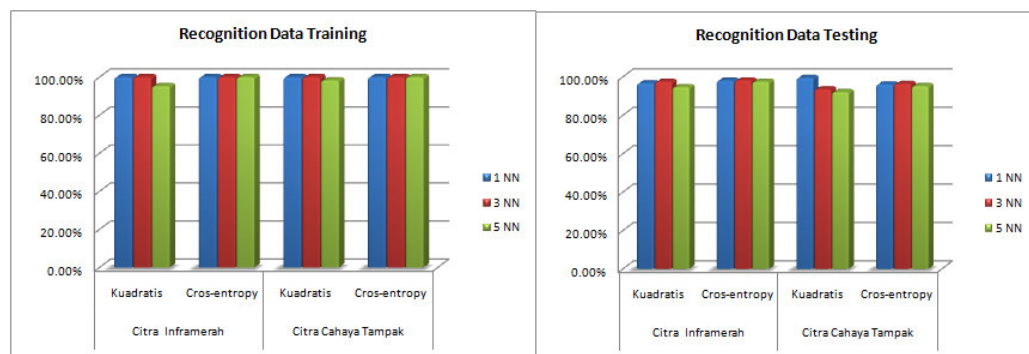
Tabel 4.23. Perbandingan Recognition Data Training NN Tunggal dan NNE untuk Dataset Wajah

| Data set | NN tunggal | | NNE 3 jaringan | | NNE 5 jaringan | |
|------------------|-------------|--------------|----------------|--------------|----------------|--------------|
| | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy |
| Inframerah | 100% | 100% | 100% | 100% | 95.33% | 100%■ |
| Cahaya Tampak | 100% | 100% | 100% | 100% | 98.33% | 100%■ |
| Rata-rata | 100% | 100% | 100% | 100% | 96.83% | 100%■ |

Tabel 4.24. Perbandingan Recognition Data Testing NN Tunggal dan NNE untuk Dataset Wajah

| Data set | NN tunggal | | NNE 3 jaringan | | NNE 5 jaringan | |
|------------------|---------------|---------------|----------------|----------------|----------------|----------------|
| | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy |
| Inframerah | 96.67% | 98.00%■ | 97.40% | 98.13% | 94.67% | 97.47%■ |
| Cahaya Tampak | 99.40%■ | 96.10% | 93.46% | 96.30%■ | 92.07% | 95.20%■ |
| Rata-rata | 98.03% | 97.05% | 95.43% | 97.22%■ | 93.37% | 96.34%■ |

Keterangan: tanda (■) menunjukkan recognition rate yang lebih tinggi (peningkatan $\geq 1\%$)

**Gambar 4.20.** Perbandingan Error, Epoch, Waktu Pelatihan NN dan NNE untuk Dataset Wajah**Gambar 4.40.** Perbandingan Recognition Rate NN dan NNE untuk Dataset Wajah

Dari tabel 4.20 di atas tampak bahwa hampir semua NN tunggal dan NNE yang dilatih untuk mengenali pola dataset citra wajah inframerah dan citra wajah cahaya tampak mampu menuju kondisi konvergen selama pelatihan. Untuk jaringan yang tidak mampu menuju konvergen pun, eror yang dicapai relatif kecil.

Dari tabel 4.21 didapatkan bahwa NN tunggal dan NNE yang dilatih menggunakan fungsi eror *cross entropy* membutuhkan epoch yang jumlahnya lebih banyak dibandingkan NN tunggal dan NNE yang dilatih menggunakan fungsi eror kuadratis. Secara relatif, peningkatan epoch terjadi secara lebih signifikan pada pengenalan dataset citra wajah cahaya tampak dibandingkan pada dataset citra wajah inframerah.

Hasil yang didapat dari tabel 4.21 mengenai jumlah epoch pelatihan dapat diperluas ke dalam hasil yang diperoleh pada tabel 4.22 mengenai waktu pelatihan yang diperlukan. Seperti diketahui bersama bahwa jumlah epoch sangat berkaitan dengan waktu pelatihan yang diperlukan. Semakin banyak epoch yang dilakukan selama pelatihan berarti bahwa proses pelatihan membutuhkan waktu yang semakin lama. Oleh karena itu kesimpulan yang didapat mengenai peningkatan epoch yang diperlukan untuk melatih NN tunggal dan NNE dengan fungsi eror *cross entropy* berdampak pada peningkatan lamanya waktu pelatihan yang diperlukan untuk melatih NN tunggal dan NNE dengan fungsi eror *cross entropy* dibandingkan dengan fungsi eror kuadratis.

Pengenalan terhadap data *training* menunjukkan hasil yang sempurna (100%) seperti tampak pada tabel 4.23. Hal ini menunjukkan bahwa NN dan NNE telah mampu mengenali pola data pelatihan dengan sangat baik. Pengecualian terjadi pada NNE 5 jaringan yang dilatih dengan fungsi eror kuadratis yang hanya mampu mengenali 95% dan 98% dari data *training* citra inframerah dan citra cahaya tampak. Selanjutnya, pengenalan terhadap data *testing* citra inframerah dan citra cahaya tampak menunjukkan hasil yang baik, dimana terlihat bahwa mayoritas *recognition rate* yang dicapai oleh NN tunggal dan NNE yang dilatih dengan fungsi eror *cross entropy* mengungguli fungsi eror kuadratis. Hal ini membuktikan bahwa NN dan NNE yang dilatih dengan fungsi eror *cross entropy* memiliki kemampuan generalisasi yang lebih baik dibandingkan yang dilatih dengan fungsi eror kuadratis.

Penambahan jumlah jaringan tidak selalu membuahkan peningkatan *recognition rate* pada NNE. Hal ini disebabkan karena pengaruh nilai α pada pelatihan NCL, metode pengambilan keputusan akhir pada NNE, serta kompleksitas masalah yang dihadapi. Dari hasil percobaan menggunakan dataset citra wajah inframerah didapatkan bahwa NNE 3 jaringan akan menghasilkan *recognition rate* optimum pada nilai $\alpha = 0.3$ sedangkan NNE 5 jaringan menghasilkan *recognition rate* optimum pada nilai $\alpha = 0.3$ maupun $\alpha = 0.4$. Di sisi lain, dari hasil percobaan menggunakan dataset citra wajah cahaya tampak didapatkan bahwa NNE 3 jaringan akan menghasilkan *recognition rate* optimum pada nilai $\alpha = 0.2$ sedangkan NNE 5 jaringan menghasilkan *recognition rate* optimum pada nilai $\alpha = 0.3$ maupun $\alpha = 0.4$.

4.3. Penutup

Bagian Penutup ini diawali dengan kesimpulan penelitian sistem pengenalan wajah (*face recognition system*) yang dilakukan Penulis, kemudian diikuti dengan saran dari Penulis untuk penelitian selanjutnya terkait dengan masalah sistem pengenalan wajah dengan citra inframerah dan citra cahaya tampak.

4.3.1. Kesimpulan Penelitian

Penelitian ini mengemukakan mengenai sistem pengenalan wajah dengan citra inframerah dan citra cahaya tampak menggunakan NN tunggal dan NNE (3 dan 5 buah jaringan) berbasis fungsi eror *cross entropy* dan fungsi eror kuadratis. NN tunggal dilatih dengan algoritma standar *Backpropagation*, sedangkan NNE dilatih dengan algoritma NCL dengan pengambilan keputusan akhir berupa metode rata-rata.

Dari percobaan yang dilakukan, tingkat pengenalan yang dihasilkan cukup tinggi. Tingkat pengenalan rata-rata untuk data *training* citra wajah inframerah adalah 98.44% (dengan fungsi eror kuadratis) dan 100% (dengan fungsi eror *cross entropy*). Tingkat pengenalan rata-rata untuk data *testing* citra wajah inframerah adalah 96.25% (dengan fungsi eror kuadratis) dan 97.87% (dengan fungsi eror *cross entropy*). Sedangkan tingkat pengenalan rata-rata untuk data *training* citra wajah cahaya tampak adalah 99.44% (dengan fungsi eror kuadratis) dan 100% (dengan fungsi eror *cross entropy*). Tingkat pengenalan rata-rata untuk data *testing* citra

wajah cahaya tampak adalah 94.98% (dengan fungsi eror kuadratis) dan 95.87% (dengan fungsi eror *cross entropy*). Dengan demikian, dapat disimpulkan bahwa derajat pengenalan yang dihasilkan oleh NN dan NNE yang dilatih dengan fungsi eror *cross entropy* mengungguli derajat pengenalan yang dihasilkan oleh NN dan NNE yang dilatih dengan fungsi eror kuadratis.

Dari percobaan pengenalan pola citra wajah inframerah diketahui bahwa penurunan jumlah sampel serta resolusi gambar berdampak pada peningkatan eror pelatihan dan peningkatan jumlah epoch pelatihan. Hal ini terutama berlaku pada NN tunggal dan NNE yang dilatih dengan fungsi eror kuadratis. Namun demikian penurunan jumlah sampel serta resolusi gambar membuat waktu pelatihan semakin cepat, hal ini disebabkan karena waktu yang diperlukan untuk melakukan 1 kali epoch lebih sedikit daripada sebelumnya.

Hasil penelitian menunjukkan bahwa *recognition rate* pada NNE yang dilatih dengan algoritma NCL dipengaruhi oleh besarnya nilai konstanta penalti (λ). Semakin banyak jumlah jaringan yang menyusun NNE, maka semakin tinggi nilai λ yang diperlukan untuk mencapai *recognition rate* optimum. Namun demikian terdapat batas nilai λ tertentu yang membuat perubahan drastis pada *recognition rate* yang dicapai. Umumnya berlaku bahwa nilai λ yang terlalu besar (>0.5) justru membuat *recognition rate* semakin menurun.

NNE yang dilatih dengan fungsi eror kuadratis lebih rentan terhadap perubahan nilai λ , sebaliknya NNE yang dilatih dengan fungsi eror *cross entropy* lebih stabil dan cenderung tidak terpengaruh terhadap perubahan nilai λ . Umumnya nilai λ menimbulkan dampak fluktuasi yang lebih besar terhadap *recognition rate* data *testing*, dibandingkan dengan *recognition rate* data *training* yang lebih stabil.

4.3.2. Saran untuk Penelitian Selanjutnya

Untuk dapat diaplikasikan ke dalam kehidupan sehari-hari, misalnya dalam sistem keamanan (*security system*) perlu adanya pengembangan lebih lanjut mengenai sistem pendeteksi wajah (*face detection system*) agar dapat dijalankan bersama dengan sistem pengenalan wajah (*face recognition system*) secara *real-time*.

Untuk dapat lebih meningkatkan *recognition rate*, perlu dilakukan penelitian menggunakan algoritma pelatihan NN selain *Backpropagation*. Hal ini juga penting

dilakukan untuk menekan *computation cost* mengingat algoritma *Backpropagation* yang membutuhkan proses pelatihan dalam waktu yang cukup lama. Penulis juga berharap ke depannya akan ada pengembangan penelitian lanjutan mengenai sistem pengenalan wajah dengan citra wajah inframerah dan citra wajah cahaya tampak yang dilakukan dengan mengkombinasikan kedua citra tersebut. Metode kombinasi ini mungkin akan meningkatkan *recogniton rate* daripada metode citra inframerah saja maupun metode citra cahaya tampak saja secara terpisah.

Selain itu, penelitian lebih lanjut dengan menggunakan skenario jangka-waktu (*time-lapse*) perlu dilakukan. Skenario ini menggunakan data uji (*data testing*) yang diambil setelah jangka waktu tertentu (misalnya 1 minggu, 1 bulan, atau bahkan 1 tahun) dari pengambilan data pelatihan (*data training*). Dengan demikian kita dapat mengetahui seberapa toleransi NN dalam mengenali gambar objek pada data uji yang mungkin akan tampak sedikit berbeda daripada gambar objek (yang sama) pada data pelatihan. Umumnya penelitian sistem pengenalan wajah mengambil data pelatihan dan data uji pada saat yang bersamaan, padahal sebenarnya skenario jangka-waktu ini lebih relevan untuk diaplikasikan ke dalam dunia nyata daripada skenario tanpa jangka-waktu.

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 dan citra cahaya tampak terutama sistem pengenalan wajah yang berbasis NNE.

BAB V

SISTEM PENGENAL AROMA

BERBASIS NEURAL NETWORK ENSEMBLE

DENGAN FUNGSI EROR CROSS-ENTROPY DAN KUADRATIS

Bab ini menjelaskan tentang eksperimen yang dilakukan Penulis dengan membangun sebuah sistem pengenalan aroma (*odor recognition system*) berbasis *Neural Network Ensemble* (NNE) yang dilatih dengan algoritma *Negative Correlated Learning* (NCL). Pada pelatihan, penghitungan eror dilakukan berdasarkan fungsi eror kuadratis atau *Mean square error* (MSE) dan *cross-entropy*.

5.1. Pendahuluan

Bagian Pendahuluan diawali dengan latar belakang penelitian pengenalan aroma, kemudian diikuti dengan penjelasan skema penelitian (*experimental setup*) sistem pengenalan aroma yang dibangun oleh Penulis.

5.1. 1. Latar Belakang Penelitian

Sistem deteksi gas sangat diperlukan untuk bidang industri, seperti industri makanan, minuman, dan kosmetik. Sistem yang berkaitan dengan deteksi gas ini juga sangat penting untuk memonitor masalah lingkungan hidup maupun bidang klinis. Seperti diketahui, faktor kualitas yang dominan pada minuman beralkohol adalah aromanya, dan kontrol kualitas dalam pembuatan kosmetik atau minuman berbahan dasar alkohol diperlukan untuk menjaga mutu produk yang dihasilkan.

Hingga saat ini, kontrol kualitas ini dilakukan oleh manusia yang memiliki keahlian sangat khusus. Akan tetapi, manusia pakar ini mempunyai keterbatasan yang berkaitan dengan masalah kesehatan dan perasaan sesaat (*mood*), sehingga mempengaruhi kinerja proses kontrol kualitas tersebut. Untuk dapat menaggulangi kelemahan ini, maka sistem deteksi gas yang dapat melakukan kontrol kualitas hasil produk secara otomatis dengan kinerja yang konsisten menjadi kebutuhan yang sangat penting.

Metode pengenalan pola untuk klasifikasi aroma dalam Sistem penciuman Elektronik pertama kali dilakukan dengan menggunakan sensor semikonduktor [25]. Akan tetapi, karena sensor ini sangat sulit untuk dapat mendeteksi gas dengan konsentrasi yang sangat rendah, maka beberapa peneliti mencoba menggunakan resonator kuarsa sebagai sensornya [26] dan piranti sel elektrokimia[27].

Penelitian awal mengenai pengembangan sistem penciuman elektronik ini telah dilakukan di Laboratorium Kecerdasan Komputasional Fakultas Ilmu Komputer Universitas Indonesia [28]. Dalam penelitian tersebut, sistem deteksi gas dibuat menggunakan resonator kuarsa yang dipadukan dengan teknik identifikasi aroma menggunakan metode pengenalan pola NN. Sistem Penciuman Elektronik ini menggunakan 4 buah jenis sensor kuarsa terlapis membran sensitif, dan dalam aplikasinya digunakan untuk membuat klasifikasi beberapa jenis aroma produk kosmetik *Martha Tilaar* serta beberapa tingkat konsentrasi alkohol yang berbeda. Akan tetapi, sistem penciuman elektronik yang dikembangkan ini tidak mampu mengenali aroma campuran yang terdiri dari gabungan beberapa aroma. Apabila digunakan untuk dapat membuat klasifikasi aroma 2 campuran, maka tingkat pengenalannya berada di bawah 40% dan akan semakin menurun lagi apabila digunakan 3 campuran aroma [29].

Dalam rangka meningkatkan kemampuan sistem penciuman elektronik untuk menyelesaikan persoalan di atas, maka penulis mengajukan dua pendekatan yang akan dibahas dan diimplementasikan dalam penelitian skripsi ini. Pendekatan pertama dilakukan dengan meningkatkan jumlah sensor hingga dua kali lipat yaitu menjadi 8 buah sensor secara paralel. Peningkatan jumlah sensor ini akan menyebabkan peningkatan jumlah dimensi ruang untuk dapat mengenali pola-pola masukan yang akan diolah, yang akan meningkatkan kemampuan NN dalam mengenali pola-pola masukan tersebut. Pendekatan yang kedua berkaitan dengan metode sistem klasifikasi yang digunakan. Dalam sistem pengenalan aroma yang lama hanya digunakan sebuah NN tunggal untuk mengklasifikasi pola aroma, sedangkan dalam penelitian ini selain digunakan NN tunggal, akan digunakan juga NNE yang terdiri dari kumpulan 3 dan 5 buah NN tunggal sebagai perbandingan.

Peningkatan jumlah jaringan NN ini diharapkan akan meningkatkan *recognition rate* sistem pengenalan aroma.

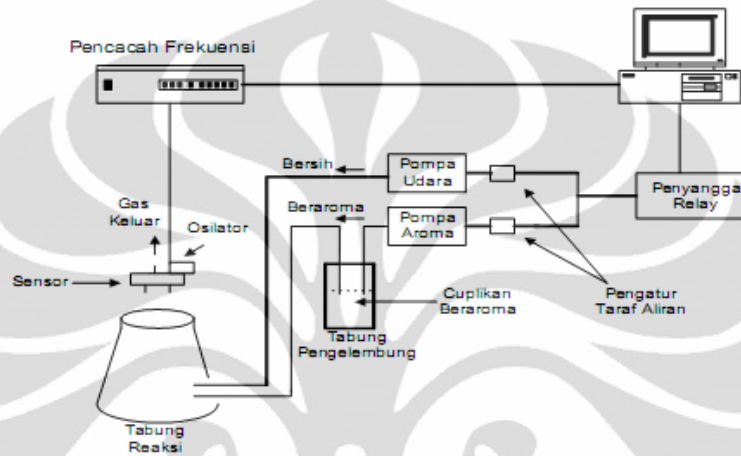
Permasalahan dalam penelitian ini berkaitan dengan penggunaan NN dan NNE algoritma *Backpropagation* sebagai sistem pengenalan pola dalam diskriminasi aroma. Algoritma *Backpropagation* digunakan karena *Backpropagation* merupakan peralatan (*tool*) standar untuk membangun hubungan-hubungan atau keterkaitan (*relationship*) antar data dalam dunia nyata [59]. Sistem penciuman elektronik menggunakan sensor kuarsa untuk menggantikan fungsi sel reseptor dalam hidung manusia, sedangkan NN yang digunakan untuk meniru sistem syaraf manusia-pakar yang memiliki kemampuan khusus mengenali aroma.

5.1. 2. Skema Penelitian

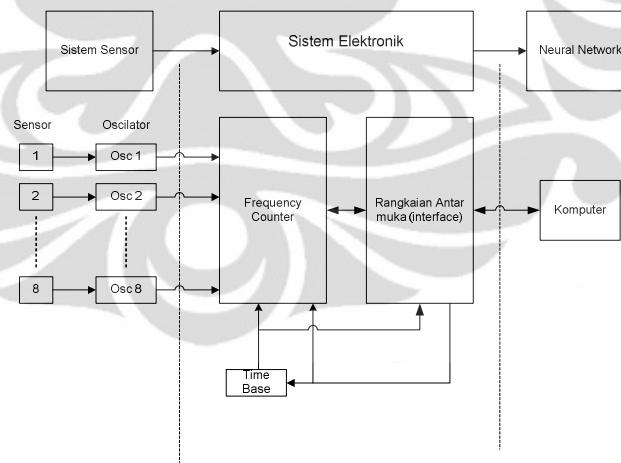
Sistem penciuman elektronik terdiri dari subsistem sensor yang mengubah besaran aroma menjadi besaran fisik serta subsistem elektronik yang mengukur besaran perubahan frekuensi sensor dan menyimpan data ke computer. Sistem penciuman elektronik dalam penelitian ini digunakan untuk mendeteksi dan menggolongkan campuran 3 buah aroma produk kosmetik *Martha Tilaar* dengan konsentrasi alkohol yang berbeda-beda. (Keterangan lengkap mengenai dataset yang digunakan dapat dilihat di bagian Lampiran). Yang dimaksudkan dengan aroma 3-campuran di sini berarti bahwa aroma tersebut diperoleh dengan komposisi 33.33% aroma #1, 33.33% aroma #2, dan 33.33% alkohol dengan konsentrasi antara 0-70%. Pengenalan terhadap aroma campuran ini mempunyai tingkat kesulitan yang lebih tinggi dibandingkan dengan penggunaan aroma dasar seperti yang dilakukan pada eksperimen yang lalu.

Dataset odor ini terdiri dari 3 buah subset (yaitu subset A,B,dan C), yang merupakan turunan dari dataset yang asli. Seperti yang telah disebutkan sebelumnya, bahwa dataset odor ini terdiri dari campuran 3 aroma (jeruk, kenanga, melati) dan alkohol dengan konsentrasi antara 0-70%. Secara keseluruhan dataset odor ini terdiri dari 18 kelas yang terpisah menjadi 3 tipe (kategori). Tipe 1 merupakan merupakan campuran jeruk-kenanga-alkohol, tipe 2 merupakan campuran jeruk-melati-alkohol, sedangkan tipe 3 merupakan campuran melati-kenanga-alkohol. Masing-masing subset (A,B, dan C) akan terdiri dari 12 kelas yang merupakan gabungan dari 2 tipe.

Misalnya, subset A yang merupakan gabungan tipe 1 dan 2 (6 kelas dari tipe 1 dan 6 kelas lainnya dari tipe 2), subset B merupakan gabungan tipe 2 dan 3 (6 kelas dari tipe 2 dan 6 kelas lainnya dari tipe 3), sedangkan subset C merupakan gabungan tipe 1 dan 3 (6 kelas dari tipe 1 dan 6 kelas lainnya dari tipe 3). Dengan demikian, subset yang satu dengan subset yang lain memiliki anggota yang beririsan.

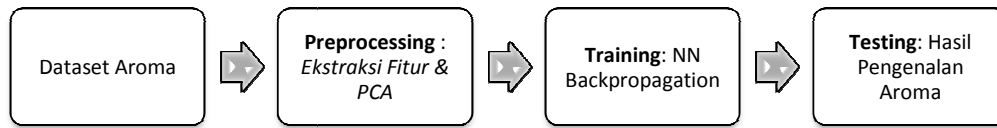


Gambar 5.1 Peralatan Sistem Pengujian Pengenal Aroma



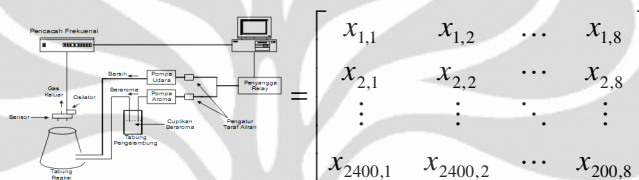
Gambar 5.2. Skematik Diagram Sistem Penciuman Elektronik

Peralatan sistem pengujian pengenal aroma dan skematik diagram penggunaan sistem penciuman elektronik dapat dilihat pada gambar di atas [24], dan tidak dibahas dalam penelitian skripsi ini.



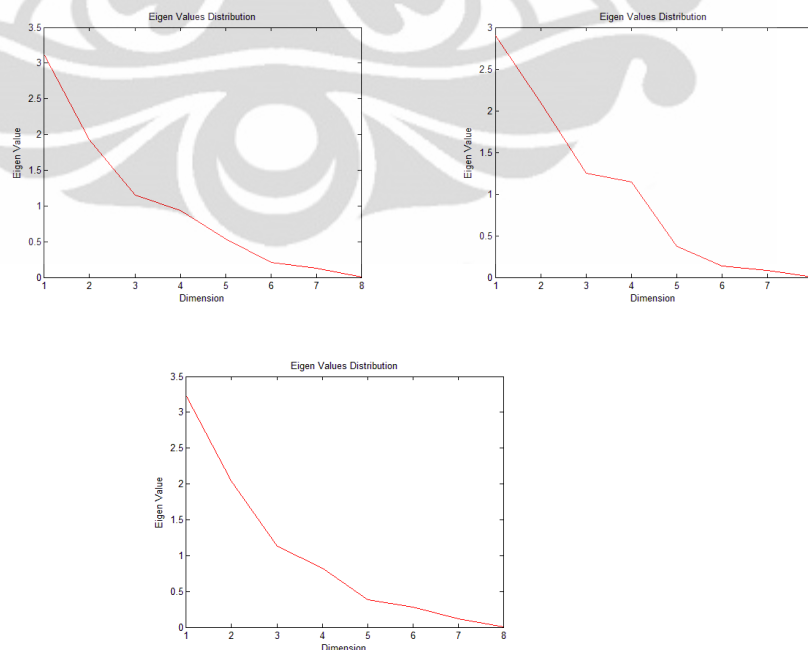
Gambar 5.3. Diagram Skema Sistem Pengenal Aroma

Pada tahap *pre-processing*, data sebanyak 8 dimensi x 2400 sample (12 kelas@200 sample) akan berubah menjadi matriks berukuran 8 x 2400 di mana setiap elemen matriksnya adalah nilai input data dari masing-masing sensor.



Gambar 5.4. Pembentukan Matriks Dataset Odor berukuran 2400x 8 (2400 sample x 8 dimensi)

Setelah tahap ekstraksi fitur selesai, maka proses selanjutnya adalah menurunkan set data baru dengan nilai eigen yang telah diurutkan dari besar ke kecil dengan menggunakan PCA. Karena dimensinya sudah cukup rendah, hanya berjumlah 8 buah, maka PCA tidak digunakan untuk mereduksi dimensi.



Gambar 5.5. Distribusi Nilai Eigen Dataset *Odor A* , *Odor B*, dan *Odor C*

Untuk keperluan tahap pelatihan dan pengujian, maka masing-masing dataset odor yang berjumlah 2400 data tadi dibagi menjadi 2 bagian sama besar, 1200 data digunakan sebagai data *training* dan 1200 data lainnya digunakan sebagai data *testing*.

5.2. Hasil Penelitian dan Pembahasan

Pada bagian ini akan diuraikan mengenai hasil penelitian mengenai sistem pengenal aroma menggunakan NN tunggal, NNE sebanyak 3 buah jaringan, serta NNE sebanyak 5 buah jaringan untuk mengetahui pengaruh penambahan jumlah jaringan terhadap performa derajat pengenalan (*recognition rate*) yang dicapai. Selain itu, tujuan utama penelitian ini yaitu membandingkan performa (peningkatan atau penurunan derajat pengenalan yang terjadi) antara jaringan yang dilatih menggunakan fungsi eror kuadratis dan jaringan yang dilatih menggunakan fungsi eror *cross-entropy* untuk menyelesaikan masalah pengenalan aroma yang tergolong sebagai masalah yang kompleks dengan tingkat kesulitan tinggi akibat kemiripan antara campuran aroma yang satu dengan campuran aroma yang lainnya.

Perlu diketahui bahwa berbeda dengan skema percobaan pada dua penelitian sebelumnya (penelitian sistem pengenal pola dengan data UCI dan sistem pengenal wajah dengan citra wajah inframerah dan citra wajah cahaya tampak), karena pelatihan dengan dataset *odor* ini membutuhkan waktu yang cukup lama, demi alasan keterbatasan waktu, maka percobaan hanya dilakukan sebanyak 1 kali dan epoh maksimum dicapai setelah 10.000 kali.

Di bawah ini akan disajikan tabel-tabel berisi ringkasan hasil pelatihan pengenalan aroma dengan NN Tunggal, NNE (3 dan 5 jaringan). Analisis ditinjau dari perbandingan eror minimum, jumlah epoh, serta waktu pelatihan yang dibutuhkan antara jaringan berbasis fungsi eror *cross entropy* dan kuadratis, diikuti dengan perbandingan derajat pengenalan (*recognition rate*) pada data *training* dan derajat pengenalan pada data *testing* untuk kedua jenis fungsi eror. Selain itu, akan dibahas mengenai derajat pengenalan pada masing-masing kelas. Terakhir, berdasarkan data hasil percobaan tersebut, ditariklah kesimpulan mengenai hasil penelitian secara keseluruhan.

Tabel 5.1. Perbandingan Error Hasil Pelatihan NN Tunggal dan NNE untuk Dataset Odor

| Data set | NN tunggal | | NNE 3 jaringan | | NNE 5 jaringan | |
|------------------|--------------|---------------|----------------|---------------|----------------|---------------|
| | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy |
| Odor A | 1.20 | 0.22 | 0.12 | 0.18 | 0.10 | 0.19 |
| Odor B | 57.24 | 404.55 | 54.25 | 347.04 | 51.14 | 326.06 |
| Odor C | 0.06 | 0.01 | 1.24 | 2.56 | 0.77 | 1.61 |
| Rata-rata | 19.50 | 134.92 | 18.54 | 116.59 | 17.33 | 109.29 |

Tabel 5.2. Perbandingan Jumlah Epoch Hasil Pelatihan Tunggal dan NNE untuk Dataset Odor

| Data set | NN tunggal | | NNE 3 jaringan | | NNE 5 jaringan | |
|------------------|--------------|--------------|----------------|--------------|----------------|--------------|
| | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy |
| Odor A | 10000 | 10000 | 10000 | 10000 | 10000 | 10000 |
| Odor B | 10000 | 10000 | 10000 | 10000 | 10000 | 10000 |
| Odor C | 10000 | 7546 | 10000 | 10000 | 10000 | 10000 |
| Rata-rata | 10000 | 9182 | 10000 | 10000 | 10000 | 10000 |

Tabel 5.3. Perbandingan Waktu Pelatihan (detik) Tunggal dan NNE untuk Dataset Odor

| Data set | NN tunggal | | NNE 3 jaringan | | NNE 5 jaringan | |
|------------------|----------------|----------------|----------------|----------------|----------------|----------------|
| | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy |
| Odor A | 1623.87 | 1640.31 | 4456.30 | 5158.03 | 7578.44 | 7391.52 |
| Odor B | 1633.65 | 1715.93 | 5299.69 | 5299.69 | 6835.02 | 7521.38 |
| Odor C | 1660.17 | 1367.99 | 4657.69 | 4776.14 | 6996.33 | 6893.40 |
| Rata-rata | 1639.23 | 1574.74 | 4804.56 | 5077.95 | 7136.60 | 7268.77 |

Tabel 5.4. Perbandingan Recognition Rate Data Training NN Tunggal dan NNE untuk Dataset Odor

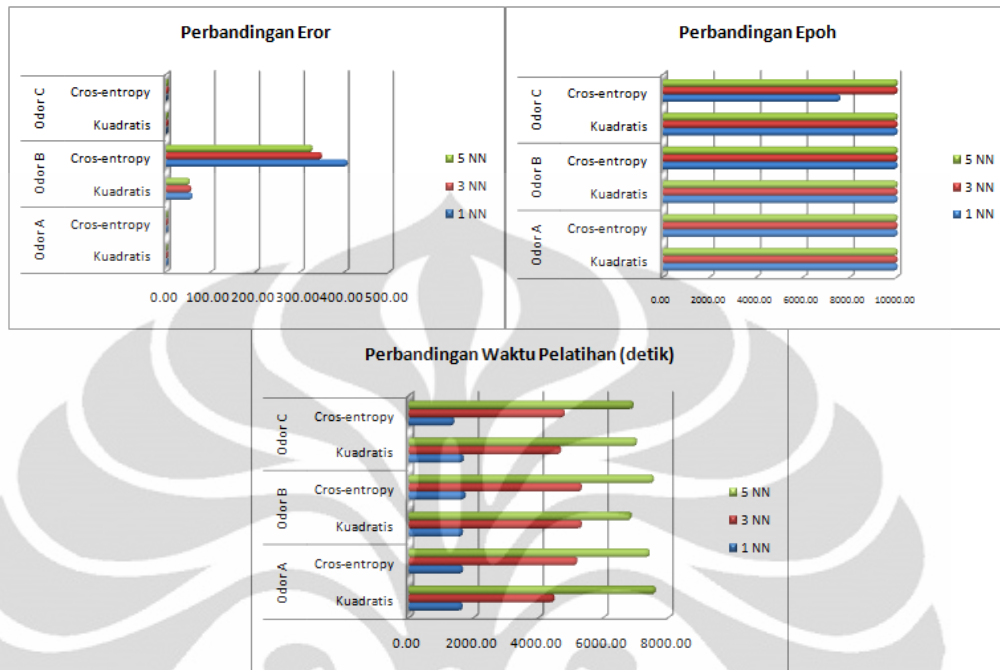
| Data set | NN tunggal | | NNE 3 jaringan | | NNE 5 jaringan | |
|------------------|---------------|---------------|----------------|---------------|----------------|---------------|
| | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy |
| Odor A | 99.67% | 100% | 99.95% | 100.00% | 100.00% | 100.00% |
| Odor B | 82.92%■ | 80.20% | 79.97% | 79.97% | 80.94% | 82.39%■ |
| Odor C | 100.00% | 100.00% | 99.42% | 99.83% | 99.33% | 99.92% |
| Rata-rata | 94.20% | 93.40% | 93.11% | 93.27% | 93.42% | 94.10% |

Keterangan: tanda (■) menunjukkan recognition rate yang lebih tinggi (peningkatan $\geq 1\%$)

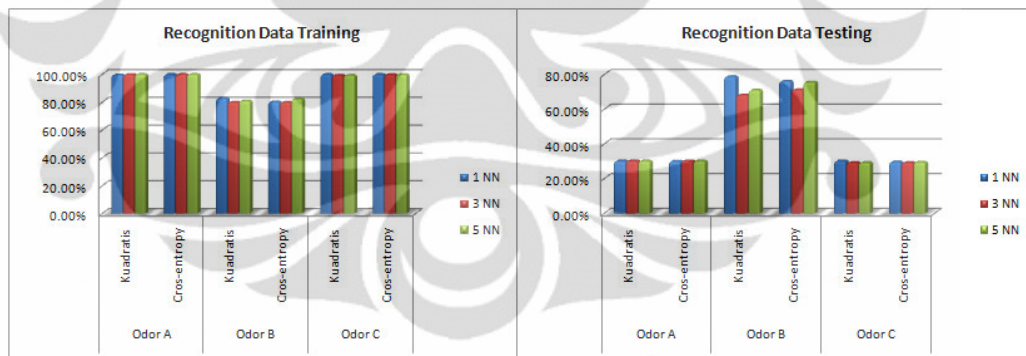
Tabel 5.5. Perbandingan Recognition Rate Data Testing NN Tunggal dan NNE untuk Dataset Odor

| Data set | NN tunggal | | NNE 3 jaringan | | NNE 5 jaringan | |
|------------------|----------------|---------------|----------------|----------------|----------------|----------------|
| | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy |
| Odor A | 30.17% | 30.11% | 30.25% | 30.22% | 30.14% | 30.17% |
| Odor B | 78.83%■ | 76.14% | 68.17% | 71.25%■ | 71.03% | 75.72%■ |
| Odor C | 30.22% | 29.61% | 29.28% | 29.28% | 29.25% | 29.47% |
| Rata-rata | 46.41%■ | 45.29% | 42.57% | 43.58%■ | 43.47% | 45.12%■ |

Keterangan: tanda (■) menunjukkan recognition rate yang lebih tinggi (peningkatan $\geq 1\%$)



Gambar 5.6. Perbandingan Error, Epoh, Waktu Pelatihan NN Tunggal dan NNE untuk Dataset Odor



Gambar 5.7 Perbandingan Recognition Rate NN Tunggal dan NNE untuk Dataset Odor

Berdasarkan tabel 5.1, secara umum terlihat bahwa penambahan jumlah jaringan akan berdampak terhadap penurunan jumlah error, baik pada jaringan yang dilatih dengan fungsi error kuadratis maupun jaringan yang dilatih dengan fungsi error *cross entropy*. Namun demikian, terdapat sedikit keanehan pada pelatihan dengan dataset *Odor C* jika dibandingkan dengan pelatihan dataset *Odor A* dan *Odor B*, yaitu bahwa penambahan jumlah jaringan dari NN tunggal ke NNE 3 jaringan menyebabkan peningkatan error bukan sebaliknya. Akan tetapi, saat jaringan

ditambahkan menjadi 5 buah, maka eror pun kembali menurun walaupun tidak serendah eror yang dihasilkan oleh NN tunggal. Fenomena ini disebabkan oleh penghitungan eror pada NNE berdasarkan keluaran akhir NNE yang merupakan nilai rata-rata dari keluran sejumlah NN tunggal yang menyusun NNE tersebut. Besar kemungkinan pada saat pelatihan NNE 3 jaringan, mayoritas NN tunggal dalam NNE menghasilkan eror yang besar, sehingga keluaran NNE pun memiliki eror yang besar. Namun, pada saat pelatihan NNE 5 jaringan, mayoritas NN tunggal dalam NNE tersebut mampu mempelajari pola-pola masukan dengan baik dan mencapai keadaan konvergen sehingga eror NNE pun lambat laun akan menurun.

Apabila ditinjau dari jumlah eror rata-rata, jaringan yang dilatih dengan fungsi eror kuadratis menghasilkan eror sebesar 18.46 sedangkan jaringan dengan fungsi eror *cross entropy* menghasilkan eror sebesar 120.27. Dengan demikian, dapat disimpulkan bahwa pada pengenalan dataset *Odor*, fungsi eror kuadratis menghasilkan nilai eror yang lebih rendah daripada eror yang dihasilkan oleh NN dan NNE yang berbasis fungsi eror *cross entropy*.

Dari tabel 5.2 terlihat bahwa hampir semua jaringan (baik NN tunggal maupun NNE) yang dilatih untuk semua dataset *OdorA*, *Odor B*, maupun *Odor C*, membutuhkan pelatihan sebanyak 10.000 epoh. Hal ini berkaitan dengan hasil yang didapatkan pada tabel 5.1 sebelumnya, bahwa dari keseluruhan pelatihan NN tunggal dan NNE yang dilakukan hampir semua jaringan tidak mampu mencapai kondisi konvergen, dan hanya terdapat 1 buah NN yang mampu mencapai kondisi konvergen, yaitu NN tunggal yang dilatih dengan fungsi eror *cross entropy*. Sedangkan NN tunggal dan NNE yang lain tidak mampu mencapai kondisi konvergen, dan karenanya proses pelatihan berhenti karena epoh telah mencapai batas maksimum (sebanyak 10.000) bukan karena jaringan telah mencapai toleransi eror minimum yang ditentukan. Proses pelatihan yang berhenti tanpa mencapai kondisi konvergen menunjukkan bahwa NN belum mampu mengenali pola-pola masukan, atau pola-pola tersebut memang terlalu kompleks dan sulit dipetakan sehingga mustahil erornya mencapai 0.01. Dengan demikian, dari tabel 5.2. dapat disimpulkan bahwa pelatihan NN tunggal dan NNE dengan fungsi eror kuadratis maupun fungsi eror *cross entropy* tidak mampu mencapai kondisi konvergen karena tingkat kesulitan dataset *Odor* yang terlalu tinggi.

Dari tabel 5.3. dapat diamati bahwa semakin banyak jaringan yang digunakan, maka semakin lama waktu yang dibutuhkan untuk mempelajari pola-pola data masukan sehingga proses pembelajaran pun akan berlangsung lebih lama. Hal ini dikarenakan banyaknya jaringan akan membuat komputasi menjadi lebih banyak sehingga waktu yang dibutuhkan pun cukup lama. Sebagai contohnya jika menggunakan NNE yang terdiri dari gabungan 3 jaringan, maka waktu pelatihan yang dibutuhkan adalah 3 kali lipat waktu pelatihan NN tunggal. Demikian pula berlaku pada NNE yang terdiri dari 5 jaringan, yang membutuhkan waktu lima kali lipat dibandingkan waktu pelatihan untuk sebuah NN tunggal.

Apabila ditinjau dari waktu pelatihan rata-rata NN tunggal dan NNE seperti yang tampak pada tabel 5.3., dapat disimpulkan bahwa pada pelatihan NN tunggal untuk pengenalan dataset *Odor*, fungsi eror *cross entropy* membutuhkan waktu pelatihan yang lebih singkat daripada pelatihan NN tunggal dengan fungsi eror kuadratis. Sedangkan pada pelatihan NNE, fungsi eror *cross entropy* membutuhkan waktu pelatihan yang lebih lama daripada pelatihan NNE dengan fungsi eror kuadratis.

Secara umum, dapat dikatakan bahwa peningkatan jumlah jaringan berpengaruh terhadap peningkatan derajat pengenalan, tetapi peningkatan tersebut tidaklah terlalu signifikan. Bahkan pada beberapa kasus justru terjadi penurunan *recognition rate* ketika jumlah jaringan ditambah. Hal ini disebabkan karena pengambilan keputusan NNE yang menggunakan metode rata-rata. Apabila dalam NNE terdapat jaringan tunggal yang memiliki keputusan salah, maka keputusan tersebut akan digabungkan dengan keputusan dari jaringan-jaringan lainnya, sehingga kemungkinan terjadinya kesalahan dalam keputusan akhir NNE menjadi kecil, dengan asumsi bahwa jaringan-jaringan lainnya memberikan keputusan yang benar. Namun sebaliknya, jika mayoritas jaringan dalam NNE memberikan keputusan yang salah, maka keputusan akhir NNE pun akan menjadi salah dan *recognition rate* pun menurun.

Jika ditinjau dari tingkat pengenalan berdasarkan jenis fungsi eror yang digunakan, maka dari tabel 5.4. terlihat jelas bahwa pengenalan data *training* menunjukkan hasil yang serupa untuk kedua jenis eror, kecuali untuk NNE (5 buah jaringan) yang dilatih untuk dataset B. Pada NNE ini, fungsi eror *cross entropy*

memberikan peningkatan sebesar 1.45% dibandingkan fungsi eror kuadratis. Sedangkan untuk pengenalan data *testing* pada tabel 5.5., NN dan NNE yang dilatih untuk mengenali dataset A dan C menunjukkan hasil yang serupa untuk kedua jenis eror. Namun untuk dataset B, NN tunggal dengan fungsi eror kuadratis menghasilkan pengenalan yang lebih tinggi dengan selisih 2.7%. Namun ketika jumlah jaringan ditambah, menjadi 3 dan 5 buah, maka fungsi eror *cross entropy* lebih baik dipakai.

Apabila ditinjau dari *recognition rate* rata-rata NN tunggal dan NNE untuk pengenalan pola dataset *Odor* seperti yang tampak pada tabel 5.4, dapat disimpulkan bahwa pada NN tunggal dan NNE yang dilatih fungsi eror *cross entropy* menghasilkan *recognition rate* data *training* yang setara dengan NN tunggal dan NNE yang dilatih dengan fungsi eror kuadratis. Namun demikian, dari hasil percobaan yang dilakukan, terdapat kecenderungan bahwa pelatihan NN tunggal berbasis fungsi eror kuadratis menghasilkan *recognition rate* lebih tinggi daripada NN tunggal berbasis fungsi eror *cross entropy*. Sebaliknya, pelatihan NNE berbasis fungsi eror *cross entropy* menghasilkan *recognition rate* lebih tinggi daripada NNE berbasis fungsi eror kuadratis.

Penurunan *recognition rate* terjadi pada pengenalan data *testing* untuk dataset A dan C, dimana pengenalan data *training* yang sebelumnya dapat dikenali dengan sangat baik (pengenalan mencapai 99-100%), tetapi menurun drastis menjadi 29-30% untuk pengenalan data *testing*. Hasil ini berlaku untuk kedua jenis fungsi eror. Bahkan penambahan jumlah jaringan pun tidak memberikan dampak yang cukup signifikan. Hal ini kemungkinan disebabkan karena terjadinya *overfitting* pada proses pelatihan. Seperti kita ketahui sebelumnya, bahwa NN dan NNE yang dilatih untuk mengenali pola dataset A dan C mampu mencapai eror yang cukup rendah (berkisar antara 0.01-1.2), yang berarti bahwa NN dan NNE mampu mengenali pola data *training* dengan cukup baik. Salah satu akibat fenomena *overfitting* adalah rendahnya kemampuan generalisasi, yaitu bahwa jaringan mampu mengenali data yang sama persis yang digunakan dalam pelatihan (data *training*), tetapi tidak mampu mengenali pola data baru (data *testing*) di luar data pelatihan meskipun data tersebut mirip atau serupa dengan data pelatihan.

Namun keadaan ini berbeda dengan apa yang terjadi pada pelatihan NN dan NNE untuk dataset B yang menghasilkan eror yang masih terbilang sangat tinggi

(berkisar antara 51-326) dan terlihat mencolok jika dibandingkan dengan eror serupa yang dicapai untuk kasus dataset A dan C. Eror yang tinggi dapat diartikan bahwa NN dan NNE belum dapat memetakan hubungan antara pola-pola data pelatihan dengan baik, dikarenakan karena tingkat kesulitan yang tinggi dalam membedakan campuran aroma yang satu dengan campuran aroma yang lain yang memang sangat mirip. Secara langsung, eror yang tinggi tersebut mengakibatkan rendahnya pengenalan pada data *training*. Namun, di sisi lain, tingginya eror tersebut mengimplikasikan bahwa data tersebut mengandung banyak informasi di dalamnya, karena sebaliknya jika data tersebut mudah diprediksi dengan benar artinya kandungan informasi di dalamnya rendah. Mempelajari data ini merupakan suatu keuntungan yang terhadap kemampuan generalisasi NN dan NNE, sehingga jaringan tidak hanya mampu mengenali pola yang sama persis dengan dataset pelatihan tetapi juga mampu mengenali pola baru yang mirip dengan data pelatihan. Hal ini dibuktikan dengan derajat pengenalan data *testing* untuk dataset B yang mencapai 68-78% yang jauh lebih tinggi daripada dataset A dan C yang hanya mencapai angka $\pm 30\%$, dan tentu saja *recognition rate* ini hampir serupa baiknya jika dibandingkan dengan pengenalan data *training* (80-83%) .

Apabila ditinjau dari *recognition rate* rata-rata seperti yang tampak pada tabel 5.5, dapat disimpulkan bahwa pelatihan NN tunggal berbasis fungsi eror kuadratis menghasilkan *recognition rate* data *testing* yang lebih tinggi daripada NN tunggal berbasis fungsi eror *cross entropy*. Sebaliknya, pelatihan NNE berbasis fungsi eror *cross entropy* menghasilkan *recognition rate* data *testing* lebih tinggi daripada NNE berbasis fungsi eror kuadratis. Hal ini tampaknya sejalan dengan kesimpulan sebelumnya yang kita dapatkan dari tabel 5.4.

Tabel 5.6. Ringkasan Hasil Recognition tiap kelas untuk Pengenalan Data Training Odor

| | Kelas | Dataset Odor A | | Dataset Odor B | | Dataset Odor C | |
|-------------------------|---------|----------------|------------------|-----------------|-----------------|----------------|----------------|
| | | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy |
| NN Tunggal | 1 | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| | 2 | 100.00% | 100.00% | 0.00% | 0.00% | 100.00% | 100.00% |
| | 3 | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| | 4 | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| | 5 | 100.00% | 100.00% | 98.00% | 88.00% | 100.00% | 100.00% |
| | 6 | 100.00% | 100.00% | 77.33% | 74.00% | 100.00% | 100.00% |
| | 7 | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| | 8 | 100.00% | 100.00% | 29.00% | 31.00% | 100.00% | 100.00% |
| | 9 | 100.00% | 100.00% | 98.00% | 70.33% | 100.00% | 100.00% |
| | 10 | 100.00% | 100.00% | 98.67% | 99.00% | 100.00% | 100.00% |
| | 11 | 97.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| | 12 | 99.00% | 100.00% | 94.00% | 100.00% | 100.00% | 100.00% |
| Rata-rata : | | 99.67% | 100.00% ■ | 82.92% ■ | 80.20% | 100.00% | 100.00% |
| ENN (3 buah) | Kelas | Dataset Odor A | | Dataset Odor B | | Dataset Odor C | |
| | | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy |
| | 1 | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| | 2 | 100.00% | 100.00% | 0.00% | 0.00% | 100.00% | 100.00% |
| | 3 | 100.00% | 100.00% | 98.67% | 98.67% | 100.00% | 100.00% |
| | 4 | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| | 5 | 100.00% | 100.00% | 99.00% | 99.00% | 100.00% | 100.00% |
| | 6 | 100.00% | 100.00% | 84.00% | 84.00% | 100.00% | 100.00% |
| | 7 | 100.00% | 100.00% | 81.33% | 81.33% | 68.67% | 100.00% |
| | 8 | 100.00% | 100.00% | 8.33% | 8.33% | 100.00% | 100.00% |
| | 9 | 100.00% | 100.00% | 94.33% | 94.33% | 100.00% | 100.00% |
| | 10 | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| 11 | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | |
| 12 | 99.33% | 100.00% | 94.00% | 94.00% | 95.00% | 98.00% | |
| Rata-rata : | | 99.95% | 100.00% | 79.97% | 79.97% | 99.42% | 99.83% |
| ENN (5 buah) | Kelas | Dataset A | | Dataset B | | Dataset C | |
| | | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy |
| | 1 | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| | 2 | 100.00% | 100.00% | 0.00% | 0.00% | 100.00% | 100.00% |
| | 3 | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| | 4 | 100.00% | 100.00% | 99.33% | 100.00% | 100.00% | 100.00% |
| | 5 | 100.00% | 100.00% | 98.00% | 94.67% | 100.00% | 100.00% |
| | 6 | 100.00% | 100.00% | 80.00% | 79.00% | 100.00% | 100.00% |
| | 7 | 100.00% | 100.00% | 94.67% | 100.00% | 100.00% | 100.00% |
| | 8 | 100.00% | 100.00% | 18.67% | 29.00% | 100.00% | 100.00% |
| | 9 | 100.00% | 100.00% | 90.33% | 89.33% | 100.00% | 100.00% |
| | 10 | 100.00% | 100.00% | 96.33% | 98.67% | 99.33% | 100.00% |
| 11 | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | |
| 12 | 100.00% | 100.00% | 94.00% | 98.00% | 92.67% | 99.00% | |
| Rata-rata : | | 100.00% | 100.00% | 80.94% | 82.39% ■ | 99.33% | 99.92% |

Keterangan: tanda (■) menunjukkan recognition rate yang lebih tinggi (peningkatan $\geq 1\%$)

Tabel 5.7. Ringkasan Hasil Recognition Rate tiap kelas untuk Pengenalan Data Testing Odor

| | Kelas | Dataset Odor A | | Dataset Odor B | | Dataset Odor C | |
|-----------------|--------|----------------|--------------|----------------|--------------|----------------|--------------|
| | | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy |
| NN Tunggal | 1 | 31.00% | 28.33% | 100.00% | 100.00% | 31.00% | 31.00% |
| | 2 | 31.00% | 31.00% | 0.00% | 0.00% | 31.00% | 31.00% |
| | 3 | 31.00% | 31.00% | 98.00% | 99.00% | 31.00% | 31.00% |
| | 4 | 31.00% | 31.00% | 96.67% | 98.33% | 31.00% | 31.00% |
| | 5 | 31.00% | 31.00% | 97.33% | 85.00% | 31.00% | 31.00% |
| | 6 | 30.00% | 30.00% | 86.00% | 87.67% | 30.00% | 30.00% |
| | 7 | 30.00% | 30.00% | 90.00% | 95.00% | 30.00% | 30.00% |
| | 8 | 30.00% | 30.00% | 12.00% | 13.00% | 30.00% | 30.00% |
| | 9 | 30.00% | 30.00% | 95.33% | 64.00% | 30.00% | 30.00% |
| | 10 | 29.00% | 29.00% | 82.33% | 83.00% | 28.00% | 20.33% |
| | 11 | 29.00% | 30.00% | 97.33% | 100.00% | 29.67% | 30.00% |
| | 12 | 29.00% | 30.00% | 91.00% | 84.33% | 30.00% | 30.00% |
| Rata-rata : | | 30.17% | 30.11% | 78.83%■ | 76.14% | 30.22% | 29.61% |
| ENN (3 buah) | Kelas | Dataset Odor A | | Dataset Odor B | | Dataset Odor C | |
| | | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy |
| | 1 | 30.67% | 31.00% | 98.67% | 98.67% | 31.00% | 31.00% |
| | 2 | 31.00% | 31.00% | 0.00% | 0.00% | 31.00% | 31.00% |
| | 3 | 31.00% | 31.00% | 93.00% | 93.00% | 31.00% | 30.33% |
| | 4 | 31.00% | 31.00% | 96.33% | 96.33% | 31.00% | 31.00% |
| | 5 | 31.00% | 31.00% | 85.33% | 85.33% | 31.00% | 31.00% |
| | 6 | 30.00% | 30.00% | 65.33% | 65.33% | 30.00% | 30.33% |
| | 7 | 30.00% | 30.00% | 69.67% | 69.67% | 29.67% | 30.33% |
| | 8 | 30.00% | 30.00% | 3.33% | 3.33% | 30.00% | 30.33% |
| | 9 | 30.00% | 30.00% | 88.33% | 88.33% | 30.00% | 30.33% |
| | 10 | 28.33% | 27.67% | 73.33% | 73.33% | 20.67% | 20.33% |
| 11 | 30.00% | 30.00% | 94.00% | 94.00% | 29.67% | 27.33% | |
| 12 | 30.00% | 30.00% | 88.33% | 88.00% | 28.33% | 29.33% | |
| Rata-rata : | | 30.25% | 30.22% | 68.17% | 71.25%■ | 29.28% | 29.28% |
| ENN (5 buah) | Kelas | Dataset Odor A | | Dataset Odor B | | Dataset Odor C | |
| | | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy |
| | 1 | 30.67% | 31.00% | 99.00% | 99.67% | 30.00% | 31.00% |
| | 2 | 31.00% | 31.00% | 0.00% | 0.00% | 31.00% | 31.00% |
| | 3 | 29.33% | 31.00% | 95.67% | 96.33% | 30.67% | 30.67% |
| | 4 | 31.00% | 31.00% | 98.33% | 97.33% | 31.00% | 31.00% |
| | 5 | 31.00% | 31.00% | 85.00% | 90.33% | 30.00% | 31.00% |
| | 6 | 30.00% | 30.00% | 60.67% | 88.00% | 30.00% | 30.00% |
| | 7 | 30.00% | 30.00% | 79.67% | 90.00% | 30.00% | 30.00% |
| | 8 | 30.00% | 30.00% | 7.67% | 10.00% | 30.00% | 30.00% |
| | 9 | 30.00% | 30.00% | 92.33% | 85.00% | 30.00% | 30.00% |
| | 10 | 29.00% | 27.00% | 63.00% | 74.00% | 22.00% | 21.00% |
| 11 | 30.00% | 30.00% | 90.67% | 93.33% | 28.67% | 28.33% | |
| 12 | 30.00% | 30.00% | 91.00% | 84.67% | 27.67% | 29.67% | |
| Rata-rata : | | 30.14% | 30.17% | 71.03% | 75.72%■ | 29.25% | 29.47% |

Keterangan: tanda (■) menunjukkan recognition rate yang lebih tinggi (peningkatan $\geq 1\%$)

5.3. Penutup

Bagian Penutup ini diawali dengan kesimpulan penelitian sistem pengenalan aroma, kemudian diikuti dengan saran dari Penulis untuk penelitian selanjutnya terkait dengan masalah sistem pengenalan aroma.

5.3.1. Kesimpulan Penelitian

Penelitian ini mengemukakan mengenai pengembangan sistem pengenalan campuran aroma berbasis NNE dengan menggunakan sistem penciuman elektronik berbasis 8 sensor kuarsa terlapis membran untuk membedakan 12 macam aroma 3-campuran dengan 6 konsentrasi alkohol yang berbeda (0%, 15%, 25%, 35%, 45%, 70%).

Dari percobaan yang dilakukan, diketahui bahwa penambahan jumlah jaringan akan berdampak terhadap penurunan jumlah eror. Secara umum fungsi eror kuadratis menghasilkan nilai eror yang lebih rendah daripada eror yang dihasilkan oleh NN tunggal dan NNE yang berbasis fungsi eror *cross entropy*.

Meskipun pelatihan NN tunggal dan NNE dengan fungsi eror kuadratis maupun fungsi eror *cross entropy* tidak mampu mencapai kondisi konvergen karena kompleksitas pola dataset *Odor*, tetapi tingkat pengenalan yang dihasilkan cukup tinggi mengingat tingkat kesulitan masalah yang dihadapi. Tingkat pengenalan rata-rata data *training* mencapai 93.8% untuk NN tunggal, 93.2% untuk NNE dengan 3 buah jaringan, dan 93.76% untuk NNE dengan 5 buah jaringan. Sedangkan Tingkat pengenalan rata-rata data *testing* mencapai 45.85% untuk NN tunggal, 43.1% untuk NNE dengan 3 buah jaringan, dan 44.3% untuk NNE dengan 5 buah jaringan.

Semakin banyak jaringan yang digunakan, maka semakin lama waktu yang dibutuhkan untuk mempelajari pola-pola data masukan sehingga proses pembelajaran pun akan berlangsung lebih lama. Pelatihan NN tunggal berbasis fungsi eror *cross entropy* membutuhkan waktu pelatihan yang lebih singkat daripada pelatihan NN tunggal dengan fungsi eror kuadratis. Sedangkan pada pelatihan NNE, fungsi eror *cross entropy* membutuhkan waktu pelatihan yang lebih lama daripada pelatihan NNE dengan fungsi eror kuadratis.

Hasil penelitian menunjukkan bahwa derajat pengenalan data *training* yang dihasilkan antara fungsi eror *cross entropy* dan fungsi eror kuadratis cukup serupa.

Namun demikian, terdapat kecenderungan bahwa pelatihan NN tunggal berbasis fungsi eror kuadratis menghasilkan *recognition rate* lebih tinggi daripada NN tunggal berbasis fungsi eror *cross entropy*. Sebaliknya, pelatihan NNE berbasis fungsi eror *cross entropy* menghasilkan *recognition rate* lebih tinggi daripada NNE berbasis fungsi eror kuadratis.

Sedangkan untuk pengenalan data *testing*, NN tunggal berbasis fungsi eror kuadratis menghasilkan *recognition rate* data *testing* yang lebih tinggi daripada NN tunggal berbasis fungsi eror *cross entropy*. Sebaliknya, pelatihan NNE berbasis fungsi eror *cross entropy* menghasilkan *recognition rate* data *testing* lebih tinggi daripada NNE berbasis fungsi eror kuadratis

fungsi eror *cross entropy* menunjukkan superioritasnya dibandingkan fungsi eror kuadratis apabila diterapkan pada NNE.

5.3.2. Saran untuk Penelitian Selanjutnya

Terdapat banyak cara lain yang dapat dilakukan pada tahap pelatihan untuk mengoptimalkan performa jaringan, misalnya dengan mencoba algoritma lain pada NNE, eksperimen lanjutan dapat dilakukan dengan optimasi algoritma NN selain *Backpropagation*, seperti misalnya NNE berbasis *fuzzy* atau *genetic algorithm*.

Optimalisasi tidak hanya dapat diterapkan pada tahap pelatihan tetapi juga dapat dilakukan pada tahap *pre-processing*, misalnya dengan mencoba menggunakan pereduksi dimensi selain PCA yang mampu diterapkan pada data yang bersifat non-linear seperti contohnya metode *Kernel-PCA*. Optimalisasi pada tahap pengujian, misalnya dengan mencoba menggunakan metode-metode pengambilan keputusan yang lain, seperti metode voting, nilai maksimum, atau perkalian produk.

Untuk dapat lebih meningkatkan derajat pengenalan (*recognition rate*) sistem terhadap aroma yang lebih sulit, dapat dilakukan penambahan jumlah sensor. Peningkatan jumlah sensor diperlukan apabila kita harus mengenal aroma campuran dengan tingkat perbedaan yang dangat rendah. Peningkatan jumlah sensor bertujuan meningkatkan fitur dimensi pola masukan aroma, sebagai akibatnya akan terjadi peningkatan derajat pengenalan aroma. Penulis berharap dari penelitian ini, muncul penelitian-penelitian lain yang dapat melanjutkan dan mengoptimalkan performa sistem pengenal aroma multikelas berbasis NNE.

BAB VI

KARAKTERISTIK NEURAL NETWORK ENSEMBLE

BERBASIS FUNGSI EROR CROSS-ENTROPY DAN KUADRATIS

Bab ini berisi tentang analisis perbandingan hasil penelitian sistem pengenalan pola berbasis *Neural Network Ensemble* (NNE) berdasarkan hasil yang diperoleh dari bab III, IV, dan V secara keseluruhan. Setelah melakukan eksperimen dengan menggunakan 9 pola dataset UCI, dataset citra wajah inframerah dan citra wajah cahaya tampak, serta dataset *Odor*, Penulis membuat ringkasan perbandingan antara karakteristik NNE yang dilatih dengan fungsi eror *cross entropy* dan kuadratis yang ditinjau dari jumlah eror, epoh, waktu pelatihan, dan derajat pengenalan (*recognition rate*).

6.1. Nilai Eror Pelatihan

Dari tabel 6.1. diketahui bahwa pelatihan menggunakan fungsi eror *cross entropy* akan menghasilkan eror rata-rata yang lebih tinggi daripada pelatihan dengan menggunakan fungsi eror kuadratis. Hal ini dikarenakan terutama oleh eror dataset *Bupa* dan *Odor* yang secara mencolok jauh lebih tinggi melebihi eror dataset-dataset lainnya. Problem dataset *Odor* memiliki tingkat kesulitan yang tinggi dan ukuran dataset yang paling besar diantara dataset lainnya (sebanyak 2400 sampel data), sehingga pola tersebut sangat sulit dikenali oleh NN karena kemiripan antara campuran aroma yang satu dengan campuran aroma yang lain, akibatnya tidak ada satupun dari NN yang mampu menuju kondisi konvergen, meskipun jumlah jaringan telah ditambah sebanyak 5 buah sekalipun.

Apabila diperhatikan, dataset dengan ukuran yang besar (jumlah sampel data yang banyak) menghasilkan eror pelatihan yang lebih tinggi daripada dataset dengan ukuran kecil atau sedang. Sebagai contohnya dataset *Vowel* (990 sampel data) dan *Odor* (2400 sampel data) yang menghasilkan eror yang tinggi dan tidak pernah mencapai kondisi konvergen selama pelatihan. Hal ini dapat dijelaskan sebagai berikut. Pada setiap 1 kali epoh pelatihan, jaringan akan mempelajari seluruh pola data masukan yang ada. Setelah proses pengubahan bobot selesai untuk satu kali epoh, maka jaringan akan menghitung selisih (eror) antara keluaran jaringan dan

target untuk setiap data. Besarnya eror tersebut akan dijumlahkan untuk keseluruhan data untuk mengecek apakah kondisi *stopping* pelatihan telah terpenuhi atau belum. Dengan demikian semakin banyak jumlah data, otomatis semakin besar pula eror yang dihasilkan karena eror tersebut merupakan jumlah total dari selisih antara keluaran jaringan dan target untuk seluruh data.

Dari tabel 6.1. diperoleh bahwa walaupun secara rata-rata fungsi eror *cross entropy* menghasilkan eror yang lebih tinggi, tetapi fungsi eror *cross entropy* membuat jaringan yang konvergen (baik NN tunggal maupun NNE) pada mayoritas dataset daripada metode fungsi eror kuadratis. Dari 72 kasus pelatihan yang diamati (12 dataset x 3 model NN x 2 jenis fungsi eror), terdapat 17 kasus dimana fungsi eror *cross entropy* membuat jaringan yang konvergen, tetapi hanya 11 kasus dimana fungsi eror kuadratis mampu membuat jaringan konvergen. Hal ini jelas menunjukkan keunggulan fungsi eror *cross entropy* terhadap fungsi eror kuadratis.

Apabila diperhatikan, semakin banyak jumlah jaringan yang dilatih, maka semakin sedikit NN yang mampu mencapai kondisi konvergen. Seperti dapat dilihat pada tabel 6.1. pada kasus NN tunggal terdapat 11 NN yang konvergen (6 dilatih dengan fungsi eror *cross entropy* dan 5 dengan fungsi eror kuadratis). Saat jaringan ditambah menjadi 3 buah, hanya terdapat 9 NNE yang mencapai keadaan konvergen (5 dilatih dengan fungsi eror *cross entropy* dan 4 dengan fungsi eror kuadratis). Kemudian setelah jaringan ditambah lagi menjadi 5 buah, maka jumlah NNE yang mencapai kondisi konvergen pun semakin berkurang, hanya 8 buah NNE saja (6 NNE diantaranya dilatih dengan fungsi eror *cross entropy* dan 2 lainnya dengan fungsi eror kuadratis). Hal ini disebabkan pada pelatihan NN tunggal, jaringan hanya memperbaiki bobot-bobot neuron untuk meminimalkan keluaran jaringan dengan target yang diinginkan. Sedangkan pada NNE, eror dihitung berdasarkan nilai rata-rata semua jaringan yang menyusun NNE tersebut. Dengan demikian, apabila mayoritas jaringan belum mencapai eror minimum, maka NNE pun tidak akan mampu mencapai kondisi konvergen.

Tabel 6.1. Perbandingan Error Hasil Pelatihan NN Tunggal dan NNE untuk Seluruh Dataset

| Data set | NN tunggal | | NNE 3 jaringan | | NNE 5 jaringan | |
|------------------|-------------|--------------|----------------|--------------|----------------|--------------|
| | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy |
| Balance-scale | 2.73 | 22.31 | 0.01■ | 0.01■ | 0.01■ | 0.01■ |
| Breast-cancer | 3.80 | 0.01■ | 1.64 | 0.01■ | 1.39 | 0.01■ |
| Bupa | 5.67 | 6.61 | 4.17 | 40.54 | 4.28 | 35.85 |
| Credit-card | 5.97 | 0.01■ | 4.69 | 10.17 | 4.63 | 8.92 |
| Glass | 0.01■ | 0.02 | 0.02 | 0.02 | 0.01■ | 0.02 |
| Heart-disease | 0.71 | 0.01■ | 0.35 | 0.20 | 0.54 | 0.01■ |
| Iris | 0.01■ | 9.01 | 0.74 | 9.05 | 9.04 | 9.04 |
| Sonar | 0.01■ | 0.01■ | 0.01■ | 0.01■ | 0.01■ | 0.01■ |
| Vowel | 4.06 | 0.04 | 3.17 | 0.04 | 2.71 | 0.04 |
| Face IR | 0.01■ | 0.01■ | 0.01■ | 0.01■ | 0.15 | 0.01■ |
| Face Visible | 0.01■ | 0.01■ | 0.01■ | 0.01■ | 0.06 | 0.01■ |
| Odor | 19.5 | 134.92 | 18.54 | 116.59 | 17.33 | 109.29 |
| Rata-rata | 3.54 | 14.41 | 2.78 | 14.72 | 3.35 | 13.60 |

Keterangan: tanda (■) menunjukkan jaringan yang mencapai kondisi konvergen (error = 0.01)

6.2. Jumlah Epoch Pelatihan

Dari hasil penelitian diketahui bahwa peningkatan jumlah jaringan dari NN tunggal ke NNE berdampak pada peningkatan jumlah epoch. Namun, peningkatan epoch rata-rata pada NNE dengan fungsi eror *cross entropy* tidak sebanyak peningkatan jumlah epoch yang dibutuhkan oleh NNE dengan fungsi eror kuadratis.

Dari tabel 6.2. dapat dihitung bahwa pelatihan menggunakan fungsi eror *cross entropy* membutuhkan epoch rata-rata sebanyak 12987 epoch sedangkan fungsi eror kuadratis membutuhkan rata-rata 13223 epoch. Dengan demikian, dapat disimpulkan bahwa pelatihan menggunakan fungsi eror *cross entropy* membutuhkan epoch rata-rata yang lebih sedikit jika dibandingkan dengan pelatihan menggunakan fungsi eror kuadratis.

Hal ini berkaitan dengan pembahasan pada bagian 6.1. sebelumnya, bahwa fungsi eror *cross entropy* menghasilkan lebih banyak jaringan yang konvergen daripada fungsi eror kuadratis. Jaringan yang konvergen akan membutuhkan jumlah epoch yang lebih sedikit daripada jaringan yang tidak konvergen, hal ini disebabkan jaringan yang konvergen telah mampu mengenali pola dengan baik sehingga pelatihan selesai karena eror telah mencapai minimum (sebesar 0.01). Sebaliknya

pada jaringan yang tidak mampu mencapai kondisi konvergen, pelatihan akan diulang terus menerus hingga pada akhirnya proses pelatihan dihentikan karena epoch telah mencapai batas maksimum (sebanyak 20.000 epoch).

Jumlah epoch yang dibutuhkan pada pelatihan berkaitan langsung dengan error yang dicapai. Untuk dataset yang mudah dikenali, epoch dengan jumlah sedikit telah cukup untuk membuat jaringan menuju konvergen (contohnya *Sonar*), tetapi sebaliknya pada masalah dataset yang sulit dikenali, epoch maksimum pun tetap tidak dapat menghasilkan error yang rendah. Dari percobaan yang dilakukan dengan data UCI pada bab III, diketahui bahwa meskipun jumlah epoch pelatihan diperbanyak menjadi 50.000 kali, penurunan error yang dihasilkan tidaklah signifikan karena grafik error telah mencapai keadaan saturasi. Dengan demikian dapat disimpulkan bahwa pada jaringan yang tidak konvergen, penambahan jumlah epoch tidak akan berpengaruh signifikan terhadap penurunan error karena pada batas epoch tertentu grafik error akan menuju keadaan saturasi, karenanya penurunan terjadi sangat lambat dan tidak seimbang jika dibandingkan dengan peningkatan *computation cost* yang dihasilkan

Apabila diperhatikan dengan seksama, dari perbandingan jumlah epoch yang terangkum dalam tabel 6.2. terlihat bahwa pelatihan NN tunggal dan NNE (3 jaringan) dengan fungsi error *cross entropy* membutuhkan epoch lebih banyak daripada NN tunggal dan NNE (3 jaringan) dengan fungsi error kuadratis. Namun ketika jumlah jaringan ditambah menjadi 5 buah jaringan, maka NNE (5 jaringan) berbasis fungsi error *cross entropy* membutuhkan epoch lebih sedikit daripada NNE (5 jaringan) berbasis fungsi error kuadratis. Dengan demikian, dapat disimpulkan bahwa untuk NNE dengan jumlah jaringan yang sedikit, maka pelatihan dengan fungsi error kuadratis membutuhkan jumlah epoch yang lebih sedikit daripada fungsi error *cross entropy*. Namun, untuk jumlah jaringan yang banyak, fungsi error *cross entropy* membutuhkan jumlah epoch yang lebih sedikit daripada fungsi error kuadratis. Hal ini menunjukkan keunggulan fungsi error *cross entropy* terhadap fungsi error kuadratis apabila diterapkan pada NNE dengan jumlah jaringan yang banyak.

Tabel 6.2. Perbandingan Jumlah Epoch Hasil Pelatihan NN Tunggal dan NNE untuk Seluruh Dataset

| Data set | NN tunggal | | NNE 3 jaringan | | NNE 5 jaringan | |
|------------------|------------------|---------------------|------------------|---------------------|------------------|---------------------|
| | <i>Kuadratis</i> | <i>Cros-entropy</i> | <i>Kuadratis</i> | <i>Cros-entropy</i> | <i>Kuadratis</i> | <i>Cros-entropy</i> |
| Balance-scale | 20000 | 20000 | 5474 | 15726 | 4738 | 4738 |
| Breast-cancer | 20000 | 7585 | 20000 | 9950 | 20000 | 10651 |
| Bupa | 20000 | 20000 | 20000 | 20000 | 20000 | 20000 |
| Credit-card | 20000 | 1346 | 20000 | 11728 | 20000 | 20000 |
| Glass | 9703 | 20000 | 11929 | 20000 | 14473 | 20000 |
| Heart-disease | 14038 | 8408 | 16768 | 17772 | 20000 | 20000 |
| Iris | 4461 | 20000 | 20000 | 20000 | 20000 | 20000 |
| Sonar | 401 | 1070 | 629 | 1097 | 605 | 1088 |
| Vowel | 20000 | 20000 | 20000 | 20000 | 20000 | 20000 |
| Face IR | 1546 | 5913 | 1928 | 5782 | 20000 | 5805 |
| Face Visible | 1799 | 9832 | 2583 | 9859 | 14949 | 10003 |
| Odor | 10000 | 9182 | 10000 | 10000 | 10000 | 10000 |
| Rata-rata | 11829 | 11945 | 12443 | 13493 | 15397 | 13524 |

6.3. Waktu Pelatihan

Lamanya waktu pelatihan dipengaruhi oleh jumlah jaringan, ukuran dataset, serta jumlah epoch pelatihan. Semakin banyak jumlah jaringan yang menyusun NNE, maka semakin lama waktu pelatihan yang diperlukan. Demikian pula, semakin besar ukuran dataset (semakin banyak jumlah sampel), maka waktu pelatihan yang diperlukan pun semakin meningkat. Dan tentu saja, semakin banyak epoch yang dilakukan dalam pelatihan, maka semakin lama waktu pelatihan yang dibutuhkan.

Pengaruh jumlah jaringan, ukuran dataset, serta jumlah epoch pelatihan terhadap peningkatan waktu pelatihan dapat dijelaskan sebagai berikut. Dalam pelatihan NNE dengan NCL, NNE dilatih secara bersama-sama (*simultaneous*). Dalam prosesnya, satu jaringan berinteraksi dengan jaringan yang lainnya melalui suatu *penalty term* untuk terus menerus mengubah bobot-bobot mereka agar eror yang dihasilkan oleh NNE mencapai minimum memperoleh keputusan akhir yang benar. Dengan demikian, untuk melatih NNE yang terdiri dari 3 buah jaringan, maka pelatihan yang dibutuhkan akan menjadi 3 kali lipat waktu pelatihan yang dibutuhkan untuk melatih NN tunggal pada dataset yang sama. Dan, jika kita menggunakan 5 buah jaringan untuk membentuk NNE, maka waktu pelatihan pun akan menjadi 5 kali lipat.

Sedangkan pengaruh ukuran dataset terhadap lamanya waktu pelatihan dapat dilihat pada contoh kasus dataset *Vowel* dan *Odor* yang membutuhkan waktu pelatihan paling lama dibandingkan dataset-dataset lainnya. Pelatihan untuk dataset *Vowel* membutuhkan waktu rata-rata 3063.74 detik (0.85 jam) untuk NN tunggal, 6332.44 detik (1.76 jam) untuk NNE 3 jaringan, dan 11798.34 detik (3.28 jam) untuk NNE 5 jaringan, jauh melebihi waktu yang dibutuhkan untuk melatih dataset UCI lainnya. Satu hal yang perlu diketahui bahwa *Vowel* memiliki ukuran dataset yang terbesar diantara dataset UCI lainnya (sebanyak 990 data, 90 data per kelasnya). Sedangkan dataset *Odor* yang merupakan dataset dengan ukuran terbesar diantara keduabelas dataset dalam penelitian ini, dengan jumlah sampel data sebanyak 2400 buah, membutuhkan waktu rata-rata 1606.98 detik (0.44 jam) untuk NN tunggal, 4941.25 detik (1.37 jam) untuk NNE 3 jaringan, dan 7202.68 detik (2 jam) untuk NNE 5 jaringan.

Jika diperhatikan, dataset *Vowel* membutuhkan rata-rata waktu pelatihan yang lebih lama daripada dataset *Odor*. Hal ini disebabkan karena pengaruh jumlah dimensi dan jumlah kelas dataset *Vowel* yang lebih tinggi daripada *Odor*. Dataset *Vowel* memiliki 12 dimensi dan 11 kelas, berarti terdapat 12 neuron *input* x 6 neuron *hidden* x 11 neuron *output* = 792 buah bobot neuron yang harus dimodifikasi terus-menerus selama pelatihan berlangsung. Sedangkan dataset *Odor* memiliki 8 dimensi dan 12 kelas, berarti terdapat 8 neuron *input* x 4 neuron *hidden* x 12 neuron *output* = 384 buah bobot neuron. Hal inilah yang menyebabkan pelatihan dengan dataset *Vowel* membutuhkan waktu paling lama diantara dataset lainnya.

Terakhir mengenai jumlah epoch yang berpengaruh terhadap waktu pelatihan. Seperti yang telah diketahui dari bagian 6.2 bahwa jaringan yang mampu mencapai kondisi konvergen akan membutuhkan epoch yang lebih sedikit daripada jaringan yang tidak. Banyaknya epoch secara langsung berkaitan dengan lamanya waktu pelatihan yaitu bahwa semakin banyak epoch berarti semakin lama waktu pelatihan yang dibutuhkan (untuk kasus pengenalan dataset yang sama). Jaringan yang tidak mampu menuju konvergen otomatis akan membutuhkan epoch maksimum (sebanyak 20.000 epoch) untuk menghentikan proses pelatihan dan karenanya membutuhkan waktu pelatihan yang lebih lama dibandingkan jaringan lain yang telah mencapai kondisi konvergen sebelum epoch pelatihan mencapai 20.000.

Dari hasil penelitian pada tabel 6.3 diketahui bahwa pelatihan menggunakan fungsi eror *cross entropy* membutuhkan waktu pelatihan rata-rata sebanyak 1803.23 detik, sedangkan fungsi eror kuadratis membutuhkan rata-rata 1747.83 detik. Dengan demikian, dapat disimpulkan bahwa pelatihan menggunakan fungsi eror *cross entropy* membutuhkan waktu pelatihan yang lebih lama jika dibandingkan dengan pelatihan menggunakan fungsi eror kuadratis. Secara sekilas hal ini tampaknya bertolak belakang dengan kesimpulan yang didapatkan sebelumnya dimana jaringan yang dilatih dengan fungsi eror *cross entropy* membutuhkan epoch yang lebih sedikit daripada jaringan yang dilatih dengan fungsi eror kuadratis. Namun sebenarnya hal ini menunjukkan bahwa jaringan dengan fungsi eror *cross entropy* membutuhkan waktu lebih lama untuk setiap satu kali epoch pelatihan dibandingkan jaringan dengan fungsi eror kuadratis.

Tabel 6.3. Perbandingan Waktu Pelatihan (detik) NN Tunggal dan NNE untuk Seluruh Dataset

| Data set | NN tunggal | | NNE 3 jaringan | | NNE 5 jaringan | |
|------------------|---------------|---------------|----------------|----------------|----------------|----------------|
| | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy |
| Balance-scale | 366.73 | 385.15 | 147.87 | 772.32 | 116.30 | 116.30 |
| Breast-cancer | 643.34 | 648.95 | 2190.87 | 1267.67 | 6349.74 | 4288.16 |
| Bupa | 321.55 | 307.80 | 806.50 | 1135.44 | 1169.00 | 1768.09 |
| Credit-card | 1137.98 | 128.14 | 1870.68 | 1020.38 | 6439.54 | 7410.83 |
| Glass | 33.45 | 162.78 | 85.77 | 227.66 | 206.58 | 286.91 |
| Heart-disease | 187.52 | 298.45 | 1366.26 | 1521.28 | 1486.11 | 1932.25 |
| Iris | 361.41 | 332.61 | 848.53 | 863.91 | 568.74 | 589.61 |
| Sonar | 7.26 | 15.64 | 24.67 | 129.24 | 88.97 | 159.97 |
| Vowel | 2192.23 | 3935.25 | 6521.56 | 6143.33 | 11820.09 | 11776.59 |
| Face IR | 43.52 | 168.67 | 65.92 | 198.93 | 1127.78 | 324.45 |
| Face Visible | 18.28 | 231.61 | 69.57 | 626.73 | 657.56 | 1819.85 |
| Odor | 1639.23 | 1574.74 | 4804.56 | 5077.95 | 7136.60 | 7268.77 |
| Rata-rata | 579.37 | 682.48 | 1566.89 | 1582.07 | 3097.25 | 3145.15 |

6.4. Derajat Pengenalan

Salah satu parameter keberhasilan NN yang sering dijadikan perhatian utama dan menjadi tolok ukur performa NN adalah derajat pengenalan (*recognition rate*). Derajat pengenalan dihitung dari persentase banyaknya data yang diklasifikasikan ke dalam kelas yang benar dibandingkan dengan jumlah total keseluruhan data.

Apabila dikaitkan dengan hasil eror pelatihan yang terdapat pada tabel 6.1., maka dapat disimpulkan bahwa derajat pengenalan data *training* berkaitan erat dengan eror yang dicapai saat pelatihan. Jaringan yang mampu mencapai kondisi konvergen (mencapai eror minimum = 0.01) akan menghasilkan derajat pengenalan yang lebih tinggi jika dibandingkan dengan jaringan yang tidak mencapai kondisi konvergen untuk kasus dataset yang sama. Bahkan apabila diperhatikan beberapa diantara jaringan yang mencapai konvergen tersebut mampu menghasilkan derajat pengenalan data *training* yang sempurna (100%).

Berbeda dengan pengenalan data *training* yang berkaitan erat dengan eror hasil pelatihan, pengenalan data *testing* lebih dipengaruhi oleh tingkat kesulitan dataset. Dataset dengan masalah yang lebih kompleks (contohnya *Vowel* dan *Odor*) akan menghasilkan derajat pengenalan data *testing* yang lebih rendah daripada dataset dengan masalah yang lebih sederhana (contohnya *Iris* dan *Breast-cancer*). Yang dimaksud dengan kompleksitas di sini berkaitan dengan jumlah dimensi, jumlah kelas, ukuran dataset, maupun persebaran data (nonlinear atau linear). Secara umum peningkatan kompleksitas masalah berbanding lurus dengan peningkatan jumlah dimensi, peningkatan jumlah kelas, peningkatan ukuran dataset, serta kenonlinearitasan.

Dari tabel 6.4. terlihat bahwa keduabelas dataset penelitian menghasilkan derajat pengenalan data *training* yang cukup tinggi (berkisar antara 85-100%). Jika dihitung terdapat 30 jaringan yang mampu mencapai tingkat pengenalan 100%, yaitu 19 jaringan diantaranya dilatih dengan fungsi eror *cross entropy* sedangkan 11 jaringan sisanya dilatih dengan fungsi eror kuadratis. 30 jaringan dengan tingkat pengenalan 100% tersebut terdiri dari 12 kasus NN tunggal, 9 kasus NNE 3 jaringan, dan 9 kasus NNE 5 jaringan.

Jika ditinjau dari derajat pengenalan data *testing*, keduabelas dataset penelitian menghasilkan derajat pengenalan yang bervariasi mulai dari yang terendah yaitu sebesar 12.15% (*Vowel*) hingga yang tertinggi yaitu sebesar 99.40% (*Face Visible*). Jika dikategorikan berdasarkan fungsi eror pelatihannya, maka fungsi eror *cross entropy* menghasilkan derajat pengenalan dengan kisaran antara 15.39% (*Vowel*, NNE 5 jaringan) hingga 98.13% (*Face Visible*, NNE 3 jaringan). Sedangkan

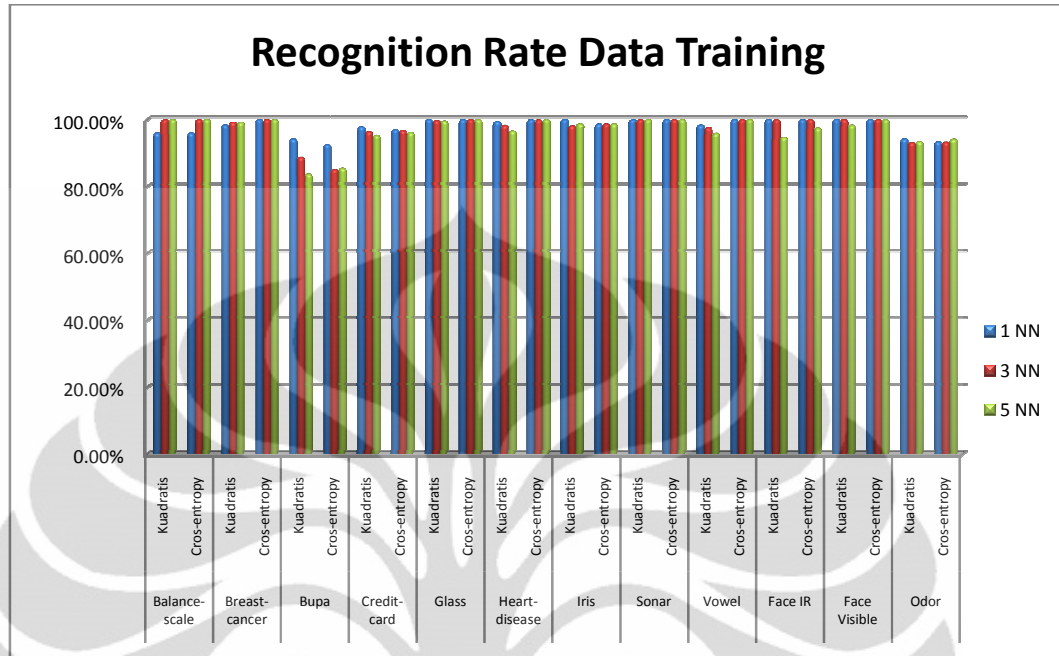
fungsi eror kuadratis menghasilkan derajat pengenalan dengan kisaran antara 12.15% (*Vowel*, NNE 5 jaringan) hingga 99.40% (*Face Visible*, NN tunggal).

Secara umum berlaku bahwa derajat pengenalan data pelatihan (data *training*) akan selalu lebih tinggi daripada pengenalan terhadap data uji (data *testing*). Oleh karena itu dalam analisis ini akan dibahas mengenai fungsi eror manakah yang mampu menghasilkan penurunan derajat pengenalan (antara data *training* dan data *testing*) yang lebih minimum. Dari hasil pada tabel 6.4 dapat dihitung bahwa pelatihan dengan fungsi eror *cross entropy* menghasilkan derajat pengenalan rata-rata sebesar 97.93% untuk data *training* dan 67.57 % untuk data *testing*, sedangkan fungsi eror kuadratis menghasilkan derajat pengenalan rata-rata sebesar 97.37% untuk data *training* dan 65.85 % untuk data *testing*. Dengan demikian, fungsi eror *cross entropy* menghasilkan selisih penurunan antara pengenalan data *training* dan data *testing* sebesar 30.36% (didapat dari 97.93% - 67.57 %), sedangkan fungsi eror kuadratis menghasilkan selisih penurunan sebesar 31.52% (didapat dari 97.37% - 65.85 %).

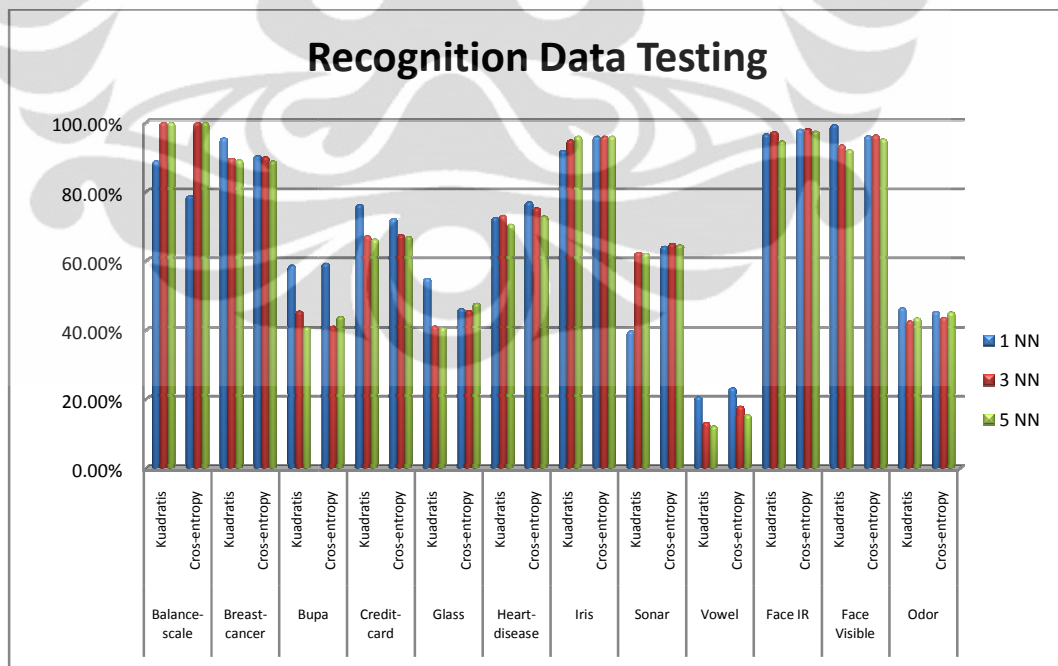
Dari perhitungan di atas dapat disimpulkan bahwa pelatihan menggunakan fungsi eror *cross entropy* menghasilkan derajat pengenalan rata-rata yang lebih tinggi dibandingkan dengan pelatihan menggunakan fungsi eror kuadratis, baik untuk pengenalan data *training* maupun pengenalan data *testing*. Selain itu, fungsi eror *cross entropy* menghasilkan penurunan tingkat pengenalan (antara data *training* dan data *testing*) yang lebih rendah daripada fungsi eror kuadratis. Hal ini berarti jaringan berbasis fungsi eror *cross entropy* memiliki kemampuan generalisasi yang lebih baik daripada jaringan berbasis fungsi eror kuadratis. Kemampuan generalisasi membuat NN mampu mengenali pola baru yang tidak diikutsertakan dalam pelatihan. Hal ini dibuktikan dengan hasil penelitian yang menunjukkan bahwa jaringan dengan fungsi eror *cross entropy* menghasilkan tingkat pengenalan yang lebih tinggi terhadap data *testing*.

Tabel 6.4. Perbandingan Recognition Rate NN Tunggal dan NNE untuk Seluruh Dataset

| Jumlah Jaringan | Data set | Recognition Data Training | | Recognition Data Testing | |
|--------------------|---------------|---------------------------|-----------------|--------------------------|-----------------|
| | | Kuadratis | Cros-entropy | Kuadratis | Cros-entropy |
| NN Tunggal | Balance-scale | 96.00% | 96.00% | 88.89%■ | 78.89% |
| | Breast-cancer | 98.43% | 100.00% ■ | 95.53%■ | 90.45% |
| | Bupa | 94.18% ■ | 92,52% | 58.68% | 59.25% |
| | Credit-card | 97.86% | 96.99% | 76.21%■ | 72.20% |
| | Glass | 100.00% | 100.00% | 54.86%■ | 46.22% |
| | Heart-disease | 99.42% | 100.00% | 72.51% | 77.03%■ |
| | Iris | 100.00%■ | 98.67% | 91.91% | 96.00%■ |
| | Sonar | 100.00% | 100.00% | 39.69% | 64.19%■ |
| | Vowel | 98.36% | 100.00% ■ | 20.65% | 23.24%■ |
| | Face IR | 100.00% | 100.00% | 96.67% | 98.00%■ |
| | Face Visible | 100.00% | 100.00% | 99.40%■ | 96.10% |
| | Odor | 94.20% | 93.40% | 46.41%■ | 45.29% |
| Rata-rata : | | 98.20% | 98.13% | 70.12% | 70.57% |
| NNE 3 jaringan | Balance-scale | 100.00% | 100.00% | 50.39% | 63.55%■ |
| | Breast-cancer | 99.18% | 100.00% | 89.64% | 90.08% |
| | Bupa | 88.73% ■ | 85.07% | 45.39% ■ | 41.10% |
| | Credit-card | 96.37% | 96.73% | 67.23% | 67.53% |
| | Glass | 99.72% | 100.00% | 41.11% | 45.55% ■ |
| | Heart-disease | 98.27% | 99.92% ■ | 73.08% | 75.28% ■ |
| | Iris | 98.14% | 98.67% | 94.93% | 96.00% ■ |
| | Sonar | 100.00% | 100.00% | 62.44% | 64.97% ■ |
| | Vowel | 97.59% | 100.00% ■ | 13.20% | 17.92% ■ |
| | Face IR | 100.00% | 100.00% | 97.40% | 98.13% |
| | Face Visible | 100.00% | 100.00% | 93.46% | 96.30% ■ |
| | Odor | 93.11% | 93.27% | 42.57% | 43.58% ■ |
| Rata-rata : | | 97.59% | 97.81% | 64.24% | 66.76% ■ |
| NNE 5 jaringan | Balance-scale | 100.00% | 100.00% | 50.62% | 50.44% |
| | Breast-cancer | 99.11% | 100.00% | 89.13% | 88.85% |
| | Bupa | 83.70% | 85.46% ■ | 40.82% | 43.86% ■ |
| | Credit-card | 95.28% | 96.11% | 66.17% | 67.03% |
| | Glass | 99.58% | 100.00% | 40.66% | 47.67% ■ |
| | Heart-disease | 96.55% | 100.00% ■ | 70.47% | 72.89% ■ |
| | Iris | 98.67% | 98.67% | 96.00% | 96.00% |
| | Sonar | 100.00% | 100.00% | 62.14% | 64.56% ■ |
| | Vowel | 95.86% | 100.00% ■ | 12.15% | 15.39% ■ |
| | Face IR | 95.33% | 100.00% | 94.67% | 97.47% ■ |
| | Face Visible | 98.33% | 100.00% | 92.07% | 95.20% ■ |
| | Odor | 93.42% | 94.10% | 43.47% | 45.12% ■ |
| Rata-rata : | | 96.32% | 97.86% ■ | 63.20% | 65.37% ■ |



Gambar 6.1. Perbandingan Recognition Data Training NN Tunggal dan NNE untuk Seluruh Dataset



Gambar 6.2. Perbandingan Recognition Data Testing NN Tunggal dan NNE untuk Seluruh Dataset

Untuk mengukur peningkatan performa derajat pengenalan yang dihasilkan oleh fungsi eror *cross entropy* terhadap fungsi eror kuadratis, Penulis menggunakan parameter berupa persentase peningkatan derajat pengenalan dengan rumus sebagai berikut:

$$Prr = 100 \frac{Prr_{kuadratis} - Prr_{crossentropy}}{Prr_{kuadratis}}$$

Dalam persamaan 6.1. di atas $Prr_{kuadratis}$ adalah persentase pengenalan NN yang dilatih dengan fungsi eror kuadratis dan $Prr_{crossentropy}$ adalah persentase pengenalan NN yang dilatih dengan fungsi eror cross-entropy .

Tabel 6.5. Persentase Peningkatan Recognition Rate NN Tunggal dan NNE Berbasis Fungsi Error Cross Entropy terhadap NN Tunggal dan NNE Berbasis Fungsi Error Kuadratis

| Data set | NN tunggal | | NNE 3 jaringan | | NNE 5 jaringan | |
|---------------|-----------------------------|----------------------------|-----------------------------|----------------------------|-----------------------------|----------------------------|
| | <i>Prr</i> Data Training | <i>Prr</i> Data Testing | <i>Prr</i> Data Training | <i>Prr</i> Data Testing | <i>Prr</i> Data Training | <i>Prr</i> Data Testing |
| Balance-scale | 0.00 | -11.25 | 0.00 | 26.12 | 0.00 | -0.36 |
| Breast-cancer | 1.60 | -5.32 | 0.83 | 0.49 | 0.90 | -0.31 |
| Bupa | -1.76 | 0.97 | -4.12 | -9.45 | 2.10 | 7.45 |
| Credit-card | -0.89 | -5.26 | 0.37 | 0.45 | 0.87 | 1.30 |
| Glass | 0.00 | -15.75 | 0.28 | 10.80 | 0.42 | 17.24 |
| Heart-disease | 0.58 | 6.23 | 1.68 | 3.01 | 3.57 | 3.43 |
| Iris | -1.33 | 4.45 | 0.54 | 1.13 | 0.00 | 0.00 |
| Sonar | 0.00 | 61.73 | 0.00 | 4.05 | 0.00 | 3.89 |
| Vowel | 1.67 | 12.54 | 2.47 | 35.76 | 4.32 | 26.67 |
| Face IR | 0.00 | 1.38 | 0.00 | 0.75 | 4.90 | 2.96 |
| Face Visible | 0.00 | -3.32 | 0.00 | 3.04 | 1.70 | 3.40 |
| Odor | -0.85 | -2.41 | 0.17 | 2.37 | 0.73 | 3.80 |

Keterangan: tanda (■/□) menunjukkan *Prr* yang bernilai (positif/negatif)

Nilai persentase peningkatan *recognition rate* dalam tabel 3.19 di atas berkisar antara nilai negatif, nol, dan nilai positif. Nilai negatif berarti bahwa *recognition rate* yang dihasilkan oleh fungsi eror *cross-entropy* lebih rendah daripada *recognition rate* yang dihasilkan oleh fungsi eror kuadratis. Nilai nol berarti bahwa *recognition rate* yang dihasilkan oleh fungsi eror *cross-entropy* sama dengan *recognition rate* yang dihasilkan oleh fungsi eror kuadratis. Sedangkan nilai positif berarti bahwa *recognition rate* yang dihasilkan oleh fungsi eror *cross-entropy* lebih tinggi dibandingkan *recognition rate* yang dihasilkan oleh fungsi eror kuadratis.

Dari tabel 6.5. di atas terlihat bahwa dari 72 kasus (12 dataset x 3 model NN x 2 jenis data) yang diteliti, terdapat 45 kasus peningkatan *recognition rate* oleh fungsi eror *cross-entropy* terhadap fungsi eror kuadratis, 14 kasus penurunan *recognition rate* oleh fungsi eror *cross-entropy* terhadap fungsi eror kuadratis, serta 13 kasus dimana kedua fungsi eror menunjukkan *recognition rate* yang setara. Dengan demikian, dapat disimpulkan bahwa secara umum pelatihan menggunakan fungsi eror *cross entropy* akan menghasilkan derajat pengenalan yang lebih tinggi baik pada pengenalan data *training* maupun pada pengenalan data *testing* daripada pelatihan dengan menggunakan fungsi eror kuadratis dan hal ini berlaku baik untuk NN tunggal maupun NNE. Dengan demikian, hasil ini semakin menguatkan kesimpulan yang diperoleh sebelumnya dari tabel 6.4.

6.4.1. Pengaruh Ukuran Dataset terhadap Derajat Pengenalan

Secara teori, untuk mendapatkan hasil *recognition rate* yang tinggi, diperlukan jumlah sampel yang cukup banyak sebagai data pelatihan. Hal ini penting terutama untuk masalah-masalah yang rumit dan nonlinear. Namun untuk masalah-masalah yang relatif sederhana (linear, dimensi rendah, jumlah kelas sedikit), maka penggunaan ukuran sampel yang kecil pun sudah mencukupi. Dengan demikian, dapat dikatakan bahwa untuk mendapatkan hasil *recognition rate* yang tinggi, maka dibutuhkan ukuran dataset yang proporsional terhadap tingkat kesulitan masalah yang dihadapi.

Dengan semakin banyaknya sampel pola data yang dipelajari dalam pelatihan, maka semakin banyak informasi yang didapatkan oleh NN, sebagai hasilnya NN pun dapat mengenali pola data uji dengan baik. Namun perlu diingat bahwa ukuran sampel yang besar menyebabkan eror pelatihan pun menjadi besar, sehingga memerlukan epoch yang lebih banyak untuk dapat menuju keadaan konvergen. Dalam hal ini kita dapat mengambil contoh kasus pelatihan dengan dataset *Odor* (terdiri dari 2400 sampel data) yang menghasilkan eror tinggi serta waktu pelatihan yang lebih lama jika dibandingkan dengan dataset-dataset lainnya.

6.4.2. Pengaruh Nilai Parameter Penalti NCL (α) terhadap Derajat Pengenalan

NCL merupakan metoda untuk menghasilkan diversitas antar jaringan dalam NNE, sebagai salah satu syarat terjadinya peningkatan performa NNE terhadap NN tunggal yaitu apabila terdapat perbedaan pendapat (*diversity of opinion*) dari anggota-anggotanya. Semakin besar diversitas, maka semakin baik performa NNE. Dengan algoritma NCL, yang merupakan ekstensi dari algoritma *Backpropagation* dengan penambahan faktor penalti pada penghitungan fungsi erornya, maka perubahan bobot antar lapisan akan dipengaruhi oleh faktor penalti ini. Selama proses pelatihan berlangsung, jaringan yang satu berinteraksi dengan jaringan yang lain melalui faktor penalti tersebut karena informasi error dihitung berdasarkan keluaran tiap jaringan, $y(n)$, dan keluaran akhir NNE, $Y(n)$, yang merupakan rata-rata keluaran tiap jaringan yang menyusun NNE. Dengan demikian, masing-masing jaringan tidak hanya meminimalisir perbedaan antara keluaran jaringan $y(n)$ dan target $t(n)$, tetapi juga selisih antara keluaran bersama $Y(n)$ dengan target $t(n)$.

Dari hasil percobaan yang dilakukan dengan menggunakan dataset citra wajah inframerah dan citra wajah cahaya didapatkan kesimpulan bahwa *recognition rate* pada NNE yang dilatih dengan NCL dipengaruhi oleh besarnya α . Dari percobaan diketahui bahwa semakin banyak jumlah jaringan, maka semakin besar nilai parameter α yang dibutuhkan untuk mencapai *recognition rate* optimum. Namun demikian terdapat batas nilai α tertentu yang membuat perubahan drastis pada *recognition rate* yang dicapai. Umumnya berlaku bahwa nilai α yang terlalu besar (>0.5) justru membuat *recognition rate* semakin menurun. Umumnya nilai α menimbulkan dampak fluktuasi yang lebih besar terhadap *recognition rate* data *testing*, dibandingkan dengan *recognition rate* data *training* yang lebih stabil.

Selain itu, dari hasil percobaan diketahui bahwa NNE yang dilatih dengan fungsi eror kuadratis lebih rentan terhadap perubahan nilai α , sebaliknya NNE yang dilatih dengan fungsi eror *cross entropy* lebih stabil dan cenderung tidak terpengaruh terhadap perubahan nilai α . Hal ini menunjukkan keunggulan dari fungsi eror *cross entropy* terhadap fungsi eror kuadratis untuk digunakan dalam pelatihan NNE dengan NCL.

6.4.3. Pengaruh Jumlah Jaringan terhadap Derajat Pengenalan

Merupakan suatu hal yang sulit untuk menyeimbangkan antara akurasi pengenalan dan diversivitas antara NN tunggal dalam NNE. Jumlah NN tunggal dalam suatu NNE memiliki pengaruh yang besar terhadap performa NNE. Jika jumlah NN tunggal terlalu sedikit, maka kemampuan NNE dalam mengeneralisasi menjadi buruk. Namun, sebaliknya jika jumlah NN tunggal dalam NNE terlalu banyak, maka *computational cost* untuk melatih NNE akan bertambah seiring dengan bertambahnya jumlah NN tunggal.

Secara teori, semakin banyak jumlah jaringan yang digunakan, maka semakin tinggi *recognition rate* yang dihasilkan. Hal ini disebabkan karena dengan adanya semakin banyak jaringan, maka semakin banyak pula vektor keluaran yang dijadikan sebagai referensi keluaran keluaran akhir pada NNE. Ketika jumlah jaringan hanya satu (NN tunggal), maka keputusan akhir diambil hanya berdasarkan jaringan tersebut saja. Jika keputusan jaringan tersebut salah, maka keputusan akhir pun salah. Namun ketika jumlah jaringan ditambah, membentuk sebuah NNE, maka keputusan akhir akan diambil berdasarkan jaringan-jaringan yang membentuk NNE tersebut. Jika NNE merupakan gabungan dari 3 buah NN tunggal, maka keputusan NNE merupakan hasil rata-rata dari ketiga jaringan tersebut. Demikian pula jika menggunakan NNE yang terdiri dari 5 buah NN tunggal, maka keputusan NNE merupakan hasil rata-rata dari kelima jaringan tersebut.

Namun demikian, dari tabel 6.4, terlihat bahwa penambahan jumlah jaringan tidak selalu membuahkan peningkatan *recognition rate*. Hal ini disebabkan karena pengaruh nilai γ pada pelatihan NCL, metode pengambilan keputusan akhir pada NNE, serta kompleksitas masalah yang dihadapi.

Seperti yang telah disebutkan pada bagian 6.4.2 tentang pengaruh nilai parameter penalti (γ) dalam pelatihan NNE dengan algoritma NCL, bahwa terdapat nilai γ tertentu yang akan menghasilkan *recognition rate* optimum. Karena berlaku persamaan $\gamma = \lambda \left[\frac{\text{jumlah jaringan}}{2(\text{jumlah jaringan} - 1)} \right]$, maka penambahan jumlah jaringan berbanding lurus terhadap peningkatan nilai γ untuk mencapai *recognition rate* yang optimum.

Sebagai implikasinya, kesalahan dalam pemilihan nilai α dapat menyebabkan penurunan *recognition rate* pada NNE.

Metode pengambilan keputusan akhir NNE pada penelitian ini dilakukan berdasarkan metode rata-rata (*simple-averaging*), sehingga setiap jaringan tunggal akan diperlakukan secara sama. Namun demikian, pada kenyataannya tidak semua jaringan tunggal sama pentingnya. Terdapat jaringan tertentu yang secara relatif lebih unggul daripada jaringan yang lain, dan demikian pula sebaliknya. Dengan menggunakan metode rata-rata, apabila dalam NNE terdapat jaringan tunggal yang memiliki keputusan salah, maka keputusan tersebut akan digabungkan dengan keputusan dari jaringan-jaringan lainnya, sehingga kemungkinan terjadinya kesalahan dalam keputusan akhir NNE menjadi kecil, dengan asumsi bahwa jaringan-jaringan lainnya memberikan keputusan yang benar. Namun sebaliknya, jika mayoritas jaringan dalam NNE memberikan keputusan yang salah, maka keputusan akhir NNE pun akan menjadi salah dan akibatnya *recognition rate* NNE pun justru akan menurun.

Apabila diperhatikan dengan seksama, dari tabel 6.4. terlihat bahwa *recognition rate* rata-rata untuk pengenalan data *training* yang dicapai NN tunggal dengan fungsi eror kuadratis (98.20%) lebih tinggi daripada *recognition rate* NN tunggal dengan fungsi eror *cross entropy* (98.13%) meskipun selisihnya hanya 0.07%. Ketika jumlah jaringan ditambah, *recognition rate* yang dihasilkan oleh NNE (3 jaringan dan 5 jaringan) justru lebih rendah daripada *recognition rate* yang dihasilkan oleh NN tunggal dan hal ini berlaku untuk kedua fungsi eror. Namun demikian, jika dibandingkan antara *recognition rate* yang dihasilkan antara kedua fungsi eror, maka NNE berbasis fungsi eror *cross entropy* menghasilkan *recognition rate* lebih tinggi daripada NNE berbasis fungsi eror kuadratis, dengan selisih sebesar 0.41% (untuk NNE 3 jaringan) dan 1.54% (untuk NNE 5 jaringan).

Hal yang serupa juga berlaku pada pengenalan data *testing*. Dari tabel 6.4. terlihat bahwa *recognition rate* rata-rata untuk pengenalan data *testing* yang dicapai NN tunggal dengan fungsi eror *cross entropy* (70.57%) lebih tinggi daripada *recognition rate* NN tunggal dengan fungsi eror kuadratis (70.12%) dengan selisih 0.45%. Sama seperti sebelumnya, ketika jumlah jaringan ditambah, *recognition rate* yang dihasilkan oleh NNE (3 jaringan dan 5 jaringan) justru lebih rendah daripada

recognition rate yang dihasilkan oleh NN tunggal dan hal ini berlaku untuk kedua fungsi eror. Namun demikian, jika dibandingkan antara *recognition rate* yang dihasilkan antara kedua fungsi eror, maka NNE berbasis fungsi eror *cross entropy* menghasilkan *recognition rate* lebih tinggi daripada NNE berbasis fungsi eror kuadratis, dengan selisih sebesar 2.52% (untuk NNE 3 jaringan) dan 2.17% (untuk NNE 5 jaringan).

Dari penjelasan di atas, maka dapat disimpulkan bahwa semakin banyak jaringan yang digunakan, maka fungsi eror *cross entropy* menghasilkan selisih *recognition rate* yang lebih besar terhadap *recognition rate* yang dihasilkan fungsi eror kuadratis dan hal ini berlaku baik pada pengenalan data *training* maupun data *testing*.

6.5. Kesimpulan

Pelatihan menggunakan fungsi eror *cross entropy* menghasilkan eror rata-rata yang lebih tinggi daripada fungsi eror kuadratis. Hal ini dikarenakan terutama oleh eror dataset *Bupa* dan *Odor*. Namun demikian, fungsi eror *cross entropy* mampu membuat jaringan yang konvergen pada mayoritas dataset. Oleh sebab itulah, fungsi eror *cross entropy* membutuhkan epoch rata-rata yang lebih sedikit jika dibandingkan dengan fungsi eror kuadratis. Hal ini dikarenakan jumlah epoch yang dibutuhkan pada pelatihan berkaitan langsung dengan eror yang dicapai.

Semakin banyak jumlah jaringan yang dilatih, maka semakin banyak epoch yang diperlukan dan semakin sulit jaringan tersebut mencapai kondisi konvergen. Pada jaringan yang tidak konvergen, penambahan jumlah epoch tidak akan berpengaruh signifikan terhadap penurunan eror karena pada batas epoch tertentu grafik eror akan menuju keadaan saturasi. Untuk NNE dengan jumlah jaringan yang sedikit, maka fungsi eror kuadratis membutuhkan jumlah epoch yang lebih sedikit, tetapi untuk jumlah jaringan yang banyak, fungsi eror *cross entropy* membutuhkan jumlah epoch yang lebih sedikit daripada fungsi eror kuadratis. Hal ini menunjukkan keunggulan fungsi eror *cross entropy* terhadap fungsi eror kuadratis apabila diterapkan pada NNE dengan jumlah jaringan yang banyak.

Peningkatan durasi waktu pelatihan dipengaruhi oleh peningkatan jumlah jaringan, besarnya ukuran dataset, serta banyaknya epoch pelatihan. Jaringan dengan

fungsi eror *cross entropy* membutuhkan waktu lebih lama untuk setiap satu kali epoch pelatihan dibandingkan jaringan dengan fungsi eror kuadratis.

Penambahan jumlah jaringan tidak selalu membuahkan peningkatan *recognition rate*. Hal ini disebabkan karena pengaruh nilai α pada pelatihan NCL, metode pengambilan keputusan akhir pada NNE, serta kompleksitas masalah yang dihadapi. NNE yang dilatih dengan fungsi eror kuadratis lebih rentan terhadap perubahan nilai α , sebaliknya NNE yang dilatih dengan fungsi eror *cross entropy* lebih stabil dan cenderung tidak terpengaruh terhadap perubahan nilai α . Hal ini menunjukkan keunggulan dari fungsi eror *cross entropy* terhadap fungsi eror kuadratis untuk digunakan dalam pelatihan NNE dengan NCL.

Derajat pengenalan data *training* berkaitan erat dengan eror yang dicapai saat pelatihan sedangkan pengenalan data *testing* lebih dipengaruhi oleh tingkat kesulitan dataset. Untuk mendapatkan hasil *recognition rate* yang tinggi, maka dibutuhkan ukuran dataset yang proporsional terhadap tingkat kesulitan masalah yang dihadapi.

Pelatihan menggunakan fungsi eror *cross entropy* menghasilkan derajat pengenalan rata-rata yang lebih tinggi dibandingkan dengan fungsi eror kuadratis, baik untuk pengenalan data *training* maupun data *testing*. Semakin banyak jaringan yang digunakan, maka fungsi eror *cross entropy* menghasilkan selisih *recognition rate* yang lebih besar terhadap *recognition rate* yang dihasilkan fungsi eror kuadratis. Selain itu, fungsi eror *cross entropy* menghasilkan penurunan tingkat pengenalan (antara data *training* dan data *testing*) yang lebih rendah daripada fungsi eror kuadratis. Hal ini menunjukkan keunggulan NNE berbasis fungsi eror *cross entropy* terhadap NNE berbasis fungsi eror kuadratis.

BAB VII PENUTUP

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

7.1. Kesimpulan

Penelitian ini membahas studi tentang karakteristik NNE berbasis fungsi eror *cross entropy* dan kuadratis yang diterapkan untuk masalah pengenalan pola. Penelitian ini dibagi menjadi 3 yaitu penelitian mengenai sistem pengenalan pola (*pattern recognition*), dimana NN berbasis fungsi eror *cross entropy* dan kuadratis digunakan untuk mengenali 9 pola dataset UCI dengan tingkat kompleksitas yang berbeda-beda. Penelitian kedua, adalah penggunaan NN berbasis fungsi eror *cross entropy* dan kuadratis pada sistem pengenalan wajah (*face recognition*) menggunakan dataset citra wajah inframerah dan citra wajah cahaya tampak. Yang ketiga yaitu penelitian terkait sistem pengenalan aroma (*odor recognition*) yang menggunakan NN berbasis fungsi eror *cross entropy* dan kuadratis.

Pada penelitian ini, selain digunakan NNE yang terdiri dari 3 buah jaringan dan 5 buah jaringan, penelitian juga diterapkan pada NN tunggal untuk melihat pengaruh perubahan karakteristik antara NN tunggal dan NNE terhadap pemilihan fungsi eror yang digunakan dalam pelatihan. Pada proses pelatihan, NNE dilatih menggunakan *Negative Correlation Learning* (NCL) yang merupakan perluasan dari algoritma standar *backpropagation*. NCL ini berguna untuk menghasilkan diversitas antar jaringan yang menyusun NNE.

Sebelum data pelatihan dimasukkan ke dalam jaringan, terlebih dahulu dilakukan tahap *pre-processing* data dengan metode *Principal Component Analysis* (PCA). Tujuannya yaitu agar data menjadi lebih sederhana sehingga proses komputasi pun dapat berjalan lebih cepat. Metode PCA berguna untuk mengurangi banyaknya dimensi data berdasarkan besarnya nilai eigen. Semakin besar nilai eigen, maka vektor eigen yang berpasangan dengannya berkontribusi lebih banyak dibandingkan vektor eigen dengan nilai eigen yang kecil.

Hasil penelitian yang dilakukan menunjukkan bahwa:

1. Pelatihan menggunakan fungsi eror *cross entropy* menghasilkan eror rata-rata yang lebih tinggi daripada pelatihan dengan menggunakan fungsi eror kuadratis. Namun, fungsi eror *cross entropy* mampu membuat jaringan menjadi konvergen (baik NN tunggal maupun NNE) pada mayoritas dataset daripada metode fungsi eror kuadratis. Semakin banyak jumlah jaringan yang digunakan, maka semakin sedikit NN yang mampu mencapai kondisi konvergen. Semakin besar ukuran dataset, semakin tinggi eror yang dihasilkan.
2. Pelatihan menggunakan fungsi eror *cross entropy* membutuhkan epoch rata-rata yang lebih sedikit jika dibandingkan dengan pelatihan menggunakan fungsi eror kuadratis. Jaringan yang konvergen akan membutuhkan jumlah epoch yang lebih sedikit daripada jaringan yang tidak konvergen. Meskipun peningkatan jumlah jaringan dari NN tunggal ke NNE berdampak pada peningkatan jumlah epoch, tetapi peningkatan epoch rata-rata pada NNE dengan fungsi eror *cross entropy* tidak sebanyak peningkatan jumlah epoch yang dibutuhkan oleh NNE dengan fungsi eror kuadratis.
3. Pelatihan menggunakan fungsi eror *cross entropy* membutuhkan waktu pelatihan rata-rata yang lebih lama jika dibandingkan dengan pelatihan menggunakan fungsi eror kuadratis. Peningkatan waktu pelatihan dipengaruhi oleh peningkatan jumlah jaringan, besarnya ukuran dataset, serta banyaknya jumlah epoch pelatihan. Jaringan yang konvergen akan membutuhkan waktu pelatihan yang lebih singkat daripada jaringan yang tidak konvergen.
4. Pelatihan menggunakan fungsi eror *cross entropy* menghasilkan derajat pengenalan rata-rata yang lebih tinggi dibandingkan fungsi eror kuadratis, baik untuk data *training* maupun data *testing*. Semakin banyak jaringan yang digunakan, maka fungsi eror *cross entropy* menghasilkan selisih *recognition rate* yang lebih besar terhadap *recognition rate* yang dihasilkan fungsi eror kuadratis. Selain itu, fungsi eror *cross entropy* menghasilkan selisih penurunan tingkat pengenalan (antara data *training* dan data *testing*) yang lebih rendah daripada fungsi eror kuadratis.

7.2. Saran

Penelitian mengenai NN dalam dekade terakhir telah membawa kemajuan yang signifikan baik secara teori maupun aplikasi praktis. NN telah mendemonstrasikan kemampuannya sebagai alternatif pengganti *classifier* tradisional untuk banyak masalah klasifikasi praktis. Namun demikian, walaupun NN tampaknya sangat menjanjikan, terdapat banyak isu yang masih belum terpecahkan. Penelitian lebih lanjut sebaiknya dicurahkan untuk mengembangkan metode yang lebih efektif dan efisien dalam identifikasi model NN, pemilihan fitur-fitur variabel (dimensi data), kombinasi NN (*ensemble*), dan perlakuan terhadap misklasifikasi kelompok yang tidak seimbang.

Secara khusus, Penulis berharap agar sistem pengenalan pola ini dapat diaplikasikan ke dalam kehidupan sehari-hari. Namun perlu adanya pengembangan lebih lanjut mengenai sistem pendeteksi pola dengan metode NNE untuk mempercepat komputasi sehingga dapat menurunkan *cost computationnya*. Penulis juga berharap hasil penelitian ini dapat dipakai sebagai referensi penulis-penulis lain yang ingin meneruskan riset dalam bidang sistem pengenalan pola dengan NNE untuk mengembangkan sistem ini agar performanya lebih optimal lagi.

Terdapat kesempatan yang sangat luas dalam berbagai aspek penelitian NN sebagai *classifier*. Penulis yakin bahwa penelitian multidisiplin-ilmu dengan NN akan menghasilkan lebih banyak lagi aktivitas riset dan membuahkan hasil yang bermanfaat di masa depan.

LAMPIRAN

A. Penurunan Rumus Perubahan Bobot untuk Fungsi Error Cross Entropy

Fungsi error *cross entropy* dinyatakan oleh:

$$E = \sum_k (-t_k) \cdot \ln(y_k) - (1 - t_k) \cdot \ln(1 - y_k) \quad (\text{A.1})$$

A.1. Perubahan bobot antara lapisan tersembunyi dengan lapisan keluaran

Perubahan bobot antara lapisan tersembunyi dengan lapisan keluaran dinyatakan sebagai berikut:

$$\Delta w_{jk} = -\alpha \frac{\partial E}{\partial w_{jk}} \quad \text{di mana} \quad \frac{\partial E}{\partial w_{jk}} = \frac{\partial E}{\partial y_k} \times \frac{\partial y_k}{\partial y_{in_k}} \times \frac{\partial y_{in_k}}{\partial w_{jk}} \quad (\text{A.2})$$

dan berikut ini adalah turunan parsial error E terhadap output y_k

$$\frac{\partial E}{\partial y_{ki}} = \frac{\sum_k -t_k}{y_k} + \frac{1 - t_k}{1 - y_k} = \frac{-\sum_k (t_k - y_k)}{y_k (1 - y_k)} \quad (\text{A.3})$$

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

$$y_k = f(y_{in_k}) = \frac{1}{1 + e^{-y_{in_k}}} \quad (\text{A.4})$$

Berikut ini adalah turunan parsial y_k terhadap y_{in_k}

$$\frac{\partial y_k}{\partial y_{in_k}} = f'(y_{in_k}) = y_k (1 - y_k) \quad (\text{A.5})$$

di mana y_{in_k} adalah fungsi keluaran, yang dinyatakan oleh:

$$y_{in_k} = w_{0k} + \sum z_j w_{jk} \quad (\text{A.6})$$

dan berikut adalah turunan parsial y_{in_k} terhadap w_{jk}

$$\frac{\partial y_{in_k}}{\partial w_{jk}} = z_j \quad (\text{A.7})$$

Substitusi (A.3), (A.5) dan (A.7) ke dalam (A.2), maka didapatkan:

$$\frac{\partial E}{\partial w_{jk}} = \frac{-\sum_k (t_k - y_k) \cdot y_k (1 - y_k) \cdot z_j}{y_k (1 - y_k)} = -\sum_k (t_k - y_k) z_j \quad (\text{A.8})$$

Suku $\sum_k (t_k - y_k)$ dapat dilambangkan dengan δ_k , sehingga persamaan di atas dapat disingkat menjadi:

$$\frac{\partial E}{\partial w_{jk}} = -\delta_k z_j \quad (\text{A.9})$$

Akhirnya, didapatkan rumus perubahan bobot antara lapisan tersembunyi dan lapisan keluaran (A.2) sebagai berikut ini:

$$\Delta w_{jk} = -\alpha(-\delta_k z_j) = \alpha \delta_k z_j \quad (\text{A.10})$$

A.2. Perubahan bobot antara lapisan masukan dengan lapisan tersembunyi

Perubahan bobot antara lapisan masukan dengan lapisan tersembunyi dinyatakan sebagai berikut:

$$\Delta v_{ij} = -\alpha \frac{\partial E}{\partial v_{ij}} \quad \text{di mana} \quad \Delta v_{ij} = \alpha \delta_j x_i \quad (\text{A.11})$$

dan berikut ini adalah turunan parsial error E terhadap output y_k

$$\frac{\partial E}{\partial y_k} = \frac{\sum_k -t_k}{y_k} + \frac{1-t_k}{1-y_k} = \frac{-\sum_k (t_k - y_k)}{y_k(1-y_k)} \quad (\text{A.12})$$

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

$$y_k = f(y_{in_k}) = \frac{1}{1 + e^{-y_{in_k}}} \quad (\text{A.13})$$

Berikut ini adalah turunan parsial y_k terhadap y_{in_k}

$$\frac{\partial y_k}{\partial y_{in_k}} = f'(y_{in_k}) = y_k(1-y_k) \quad (\text{A.14})$$

di mana y_{in_k} adalah fungsi keluaran neuron output, yang dinyatakan oleh:

$$y_{in_k} = w_{0k} + \sum z_j w_{jk} \quad (\text{A.15})$$

Berikut adalah turunan parsial y_{in_k} terhadap z_j

$$\frac{\partial y_{in_k}}{\partial z_j} = w_{jk} \quad (\text{A.16})$$

dan berikut ini adalah turunan parsial z_j terhadap z_in_j

$$\frac{\partial z_j}{\partial z_in_j} = f'(z_in_j) = z_j(1 - z_j) \quad (\text{A.17})$$

di mana z_in_j adalah fungsi keluaran neuron tersembunyi, yang dinyatakan oleh:

$$z_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (\text{A.18})$$

Sedangkan berikut ini adalah turunan parsial z_in_j terhadap v_{ij}

$$\frac{\partial z_in_j}{\partial v_{ij}} = x_i \quad (\text{A.19})$$

Substitusi (A.12), (A.14), (A.16) dan (A.19) ke dalam (A.11), maka didapatkan:

$$\frac{\partial E}{\partial v_{ij}} = \frac{-\sum_k (t_k - y_k) \cdot y_k (1 - y_k) \cdot w_{jk} \cdot z_j (1 - z_j) \cdot x_i}{y_k (1 - y_k)} = -\sum_k (t_k - y_k) w_{jk} z_j (1 - z_j) x_i \quad (\text{A.20})$$

Suku $\sum_k (t_k - y_k)$ dapat dilambangkan dengan δ_k , sehingga persamaan di atas dapat disingkat menjadi:

$$\frac{\partial E}{\partial v_{ij}} = -\sum_k \delta_k w_{jk} z_j (1 - z_j) x_i \quad (\text{A.21})$$

Suku $\sum_k \delta_k w_{jk} z_j (1 - z_j)$ dapat dilambangkan dengan δ_j , sehingga persamaan di atas dapat disingkat menjadi:

$$\frac{\partial E}{\partial v_{ij}} = -\alpha(-\delta_j x_i) = \alpha \delta_j x_i \quad (\text{A.22})$$

Akhirnya, didapatkan rumus perubahan bobot antara lapisan masukan dan lapisan tersembunyi (A.11) sebagai berikut ini:

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (\text{A.23})$$

B. Penurunan Rumus Perubahan Bobot untuk Fungsi Error Kuadratis

Fungsi error kuadratis dinyatakan oleh:

$$E = \frac{1}{2} \sum_k (t_k - y_k)^2 \quad (\text{B.1})$$

B.1. Perubahan bobot antara lapisan tersembunyi dengan lapisan keluaran

Perubahan bobot antara lapisan tersembunyi dengan lapisan keluaran dinyatakan sebagai berikut:

$$\Delta w_{jk} = -\alpha \frac{\partial E}{\partial w_{jk}} \quad \text{di mana} \quad \frac{\partial E}{\partial w_{jk}} = \frac{\partial E}{\partial y_k} \times \frac{\partial y_k}{\partial y_{in_k}} \times \frac{\partial y_{in_k}}{\partial w_{jk}} \quad (\text{B.2})$$

dan berikut ini adalah turunan parsial error E terhadap output y_k

$$\frac{\partial E}{\partial y_k} = -\sum_k (t_k - y_k) \quad (\text{B.3})$$

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

$$y_k = f(y_{in_k}) = \frac{1}{1 + e^{-y_{in_k}}} \quad (\text{B.4})$$

Berikut ini adalah turunan parsial y_k terhadap y_{in_k}

$$\frac{\partial y_k}{\partial y_{in_k}} = f'(y_{in_k}) = y_k(1 - y_k) \quad (\text{B.5})$$

di mana y_{in_k} adalah fungsi keluaran, yang dinyatakan oleh:

$$y_{in_k} = w_{0k} + \sum z_j w_{jk} \quad (\text{B.6})$$

dan berikut adalah turunan parsial y_{in_k} terhadap w_{jk}

$$\frac{\partial y_{in_k}}{\partial w_{jk}} = z_j \quad (\text{B.7})$$

Substitusi (B.3), (B.5) dan (B.7) ke dalam (B.2), maka didapatkan:

$$\frac{\partial E}{\partial w_{jk}} = -\sum_k (t_k - y_k) y_k (1 - y_k) z_j \quad (\text{B.8})$$

Suku $\sum_k (t_k - y_k) y_k (1 - y_k)$ dapat dilambangkan dengan δ_k , sehingga persamaan di atas dapat disingkat menjadi:

$$\frac{\partial E}{\partial w_{jk}} = -\delta_k z_j \quad (\text{B.9})$$

Akhirnya, didapatkan rumus perubahan bobot antara lapisan tersembunyi dan lapisan keluaran (B.2) sebagai berikut ini:

$$\Delta w_{jk} = -\alpha(-\delta_k z_j) = \alpha \delta_k z_j \quad (\text{B.10})$$

B.2. Perubahan bobot antara lapisan masukan dengan lapisan tersembunyi

Perubahan bobot antara lapisan masukan dengan lapisan tersembunyi dinyatakan sebagai berikut:

$$\Delta v_{ij} = -\alpha \frac{\partial E}{\partial v_{ij}} \quad \text{di mana} \quad \frac{\partial E}{\partial v_{ij}} = \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial y_{in_k}} \cdot \frac{\partial y_{in_k}}{\partial z_j} \cdot \frac{\partial z_j}{\partial z_{in_j}} \cdot \frac{\partial z_{in_j}}{\partial v_{ij}} \quad (\text{B.11})$$

dan berikut ini adalah turunan parsial error E terhadap output y_k

$$\frac{\partial E}{\partial y_k} = -\sum_k (t_k - y_k) \quad (\text{B.12})$$

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

$$y_k = f(y_{in_k}) = \frac{1}{1 + e^{-y_{in_k}}} \quad (\text{B.13})$$

Berikut ini adalah turunan parsial y_k terhadap y_{in_k}

$$\frac{\partial y_k}{\partial y_{in_k}} = f'(y_{in_k}) = y_k (1 - y_k) \quad (\text{B.14})$$

di mana y_{in_k} adalah fungsi keluaran neuron output, yang dinyatakan oleh:

$$y_{in_k} = w_{0k} + \sum z_j w_{jk} \quad (\text{B.15})$$

Berikut adalah turunan parsial y_{in_k} terhadap z_j

$$\frac{\partial y_{in_k}}{\partial z_j} = w_{jk} \quad (\text{B.16})$$

dan berikut ini adalah turunan parsial z_j terhadap z_{in_j}

$$\frac{\partial z_j}{\partial z_{in_j}} = f'(z_{in_j}) = z_j (1 - z_j)$$

(B.17)

di mana z_{in_j} adalah fungsi keluaran neuron tersembunyi, yang dinyatakan oleh:

$$z_{in_j} = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (\text{B.18})$$

Sedangkan berikut ini adalah turunan parsial z_{in_j} terhadap v_{ij}

$$\frac{\partial z_{in_j}}{\partial v_{ij}} = x_i \quad (\text{B.19})$$

Substitusi (B.12), (B.14), (B.16) dan (B.19) ke dalam (B.11), maka didapatkan:

$$\frac{\partial E}{\partial v_{ij}} = - \sum_k (t_k - y_k) y_k (1 - y_k) w_{jk} z_j (1 - z_j) x_i \quad (\text{B.20})$$

Suku $\sum_k (t_k - y_k) y_k (1 - y_k)$ dapat dilambangkan dengan δ_k , sehingga persamaan di atas dapat disingkat menjadi:

$$\frac{\partial E}{\partial v_{ij}} = - \sum_k \delta_k w_{jk} z_j (1 - z_j) x_i \quad (\text{B.21})$$

Suku $\sum_k \delta_k w_{jk} z_j (1 - z_j)$ dapat dilambangkan dengan δ_j , sehingga persamaan di atas dapat disingkat menjadi:

$$\frac{\partial E}{\partial v_{ij}} = -\alpha(-\delta_j x_i) = \alpha \delta_j x_i \quad (\text{B.22})$$

Akhirnya, didapatkan rumus perubahan bobot antara lapisan masukan dan lapisan tersembunyi (B.11) sebagai berikut ini:

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (\text{B.23})$$

C. Penurunan Rumus Perubahan Bobot Algoritma NCL dengan Fungsi Error Cross-Entropy

Fungsi error jaringan ke- i pada NCL dinyatakan oleh: (N = jumlah data)

$$E_i = \frac{1}{N} \sum_{n=1}^N E_i(n) = \frac{1}{N} \sum_{n=1}^N -t(n) \ln(y_i(n)) - (1-t(n)) \ln(1-y_i(n)) + \frac{1}{N} \sum_{n=1}^N \lambda p_i(n) \quad (C.1)$$

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

$$y_i(n) = f(y_{-in_k}) = \frac{1}{1 + e^{-y_{-in_k}}} \quad (C.2)$$

di mana y_{-in_k} adalah fungsi keluaran, yang dinyatakan oleh:

$$y_{-in_k} = w_{0k} + \sum_{j=1}^J z_j w_{jk} \quad (C.3)$$

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

$$z_j = f(z_{-in_j}) = \frac{1}{1 + e^{-z_{-in_j}}} \quad (C.4)$$

di mana z_{-in_j} adalah fungsi keluaran, yang dinyatakan oleh:

$$z_{-in_j} = v_{0j} + \sum_{i=1}^I x_i v_{ij} \quad (C.5)$$

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

$$\begin{aligned} \frac{\partial E_i(n)}{\partial y_i(n)} &= \frac{t(n) - y_i(n)}{y_i(n)(1 - y_i(n))} + \lambda \frac{\partial p_i(n)}{\partial y_i(n)} \\ &= \frac{t(n) - y_i(n)}{y_i(n)(1 - y_i(n))} + \lambda \sum_{j \neq i} (y_j(n) - Y(n)) \\ &= \frac{t(n) - y_i(n)}{y_i(n)(1 - y_i(n))} + \lambda (MY(n) - y_i(n) - (M - 1)Y(n)) \\ &= \frac{t(n) - y_i(n)}{y_i(n)(1 - y_i(n))} + \lambda (Y(n) - y_i(n)) \end{aligned} \quad (C.6)$$

C.1. Perubahan bobot antar lapisan tersembunyi dengan lapisan keluaran

Perubahan bobot antara lapisan tersembunyi dengan lapisan keluaran

dinyatakan sebagai berikut

$$\Delta w_{jk} = -\alpha \frac{\partial E_i}{\partial w_{jk}} \quad \text{di mana} \quad \frac{\partial E_i}{\partial w_{jk}} = \frac{\partial E_i}{\partial y_i} \times \frac{\partial y_i}{\partial y_{in_k}} \times \frac{\partial y_{in_k}}{\partial w_{jk}} \quad (\text{C.7})$$

Berikut ini adalah turunan parsial y_i terhadap y_{in_i}

$$\frac{\partial y_i}{\partial y_{in_i}} = f'(y_{in_i}) = y_i(n)(1 - y_i(n)) \quad (\text{C.8})$$

dan berikut adalah turunan parsial y_{in_i} terhadap w_{jk}

$$\frac{\partial y_{in_i}}{\partial w_{jk}} = z_j \quad (\text{C.9})$$

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

$$\begin{aligned} \Delta w_{kj} &= -\alpha \left\{ -\left[\frac{t(n) - y_i(n)}{y_i(n)(1 - y_i(n))} + \lambda(Y(n) - y_i(n)) \right] \right\} [y_i(n)(1 - y_i(n))] z_j \\ &= \alpha \left[\frac{t(n) - y_i(n)}{y_i(n)(1 - y_i(n))} + \lambda(Y(n) - y_i(n)) \right] [y_i(n)(1 - y_i(n))] z_j \end{aligned} \quad (\text{C.10})$$

Suku $\left[\frac{t(n) - y_i(n)}{y_i(n)(1 - y_i(n))} + \lambda(Y(n) - y_i(n)) \right] [y_i(n)(1 - y_i(n))]$ dapat dilambangkan

dengan δ_k , sehingga persamaan di atas dapat disingkat menjadi:

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (\text{C.11})$$

C.2. Perubahan bobot antar lapisan masukan dengan lapisan tersembunyi

Perubahan bobot antara lapisan masukan dengan lapisan tersembunyi dinyatakan sebagai berikut

$$\Delta v_{ij} = -\alpha \frac{\partial E_i}{\partial v_{ij}} \quad \text{di mana} \quad \frac{\partial E_i}{\partial v_{ij}} = \frac{\partial E_i}{\partial y_i} \times \frac{\partial y_i}{\partial y_{in_i}} \times \frac{\partial y_{in_i}}{\partial z_j} \times \frac{\partial z_j}{\partial z_{in_j}} \times \frac{\partial z_{in_j}}{\partial v_{ij}} \quad (\text{C.12})$$

Berikut ini adalah turunan parsial y_i terhadap y_{in_i} (sama seperti C.8)

$$\frac{\partial y_i}{\partial y_{in_i}} = f'(y_{in_i}) = y_i(n)(1 - y_i(n)) \quad (\text{C.13})$$

dan berikut adalah turunan parsial y_{in_i} terhadap w_{jk}

$$\frac{\partial y_{in_i}}{\partial z_j} = w_{jk} \quad (\text{C.14})$$

Sedangkan turunan parsial z_j terhadap z_{in_j} dinyatakan oleh:

$$\frac{\partial z_j}{\partial z_{in_j}} = f'(z_{in_j}) = z_j(n)(1 - z_j(n)) \quad (\text{C.15})$$

dan turunan parsial z_{in_j} terhadap v_{ij} dinyatakan oleh:

$$\frac{\partial z_{in_j}}{\partial v_{ik}} = x_i \quad (\text{C.16})$$

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

$$\begin{aligned} \Delta v_{ij} &= -\alpha \left\{ - \left[\frac{t(n) - y_i(n)}{y_i(n)(1 - y_i(n))} + \lambda(Y(n) - y_i(n)) \right] \right\} [y_i(n)(1 - y_i(n))] w_{jk} [z_j(n)(1 - z_j(n))] x_i \\ &= \alpha \left[\frac{t(n) - y_i(n)}{y_i(n)(1 - y_i(n))} + \lambda(Y(n) - y_i(n)) \right] [y_i(n)(1 - y_i(n))] w_{jk} [z_j(n)(1 - z_j(n))] x_i \\ &= \alpha \delta_k w_{jk} [z_j(n)(1 - z_j(n))] x_i \end{aligned} \quad (\text{C.17})$$

Suku $\delta_k w [z_j(n)(1 - z_j(n))] x_i$ dapat dilambangkan dengan δ_j sehingga persamaan

di atas dapat disingkat menjadi:

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (\text{C.18})$$

D. Penurunan Rumus Perubahan Bobot Algoritma NCL dengan Fungsi Error Kuadratis

Fungsi error jaringan ke-i pada NCL dinyatakan oleh: (N = jumlah data)

$$E_i = \frac{1}{N} \sum_{n=1}^N E_i(n) = \frac{1}{N} \sum_{n=1}^N \frac{1}{2} (t(n) - y_i(n))^2 + \frac{1}{N} \sum_{n=1}^N \lambda p_i(n) \quad (D.1)$$

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

$$y_i(n) = f(y_{-in_k}) = \frac{1}{1 + e^{-y_{-in_k}}} \quad (D.2)$$

di mana y_{-in_k} adalah fungsi keluaran, yang dinyatakan oleh:

$$y_{-in_k} = w_{0k} + \sum_{j=1}^p z_j w_{jk} \quad (D.3)$$

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

$$z_j = f(z_{-in_j}) = \frac{1}{1 + e^{-z_{-in_j}}} \quad (D.4)$$

di mana z_{-in_j} adalah fungsi keluaran, yang dinyatakan oleh:

$$z_{-in_j} = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (D.5)$$

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

$$\begin{aligned} \frac{\partial E_i(n)}{\partial y_i(n)} &= y_i(n) - t(n) + \lambda \frac{\partial p_i(n)}{\partial y_i(n)} \\ &= y_i(n) - t(n) + \lambda \sum_{j \neq i} (y_j(n) - Y(n)) \\ &= y_i(n) - t(n) + \lambda (MY(n) - y_i(n) - (M-1)Y(n)) \\ &= y_i(n) - t(n) + \lambda (Y(n) - y_i(n)) \\ &= y_i(n) - t(n) - \lambda (y_i(n) - Y(n)) \\ &= (1 - \lambda)(y_i(n) - t(n)) + \lambda (Y(n) - t(n)) \\ &= -[(1 - \lambda)(t(n) - y_i(n)) + \lambda (t(n) - Y(n))] \end{aligned} \quad (D.6)$$

D.1. Perubahan bobot antar lapisan tersembunyi dengan lapisan keluaran

Perubahan bobot antara lapisan tersembunyi dengan lapisan keluaran dinyatakan sebagai berikut

$$\Delta w_{jk} = -\alpha \frac{\partial E_i}{\partial w_{jk}} \quad \text{di mana} \quad \frac{\partial E_i}{\partial w_{jk}} = \frac{\partial E_i}{\partial y_i} \times \frac{\partial y_i}{\partial y_{in_i}} \times \frac{\partial y_{in_i}}{\partial w_{jk}} \quad (\text{D.7})$$

Berikut ini adalah turunan parsial y_i terhadap y_{in_i}

$$\frac{\partial y_i}{\partial y_{in_i}} = f'(y_{in_i}) = y_i(n)(1 - y_i(n)) \quad (\text{D.8})$$

dan berikut adalah turunan parsial y_{in_i} terhadap w_{jk}

$$\frac{\partial y_{in_i}}{\partial w_{jk}} = z_j \quad (\text{D.9})$$

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

$$\begin{aligned} \Delta w_{kj} &= -\alpha \left\{ - \left[(1 - \lambda)(t(n) - y_i(n)) + \lambda(t(n) - Y(n)) \right] \right\} \left[y_i(n)(1 - y_i(n)) \right] z_j \\ &= \alpha \left[(1 - \lambda)(t(n) - y_i(n)) + \lambda(t(n) - Y(n)) \right] \left[y_i(n)(1 - y_i(n)) \right] z_j \quad (\text{D.10}) \end{aligned}$$

Suku $\left[(1 - \lambda)(t(n) - y_i(n)) + \lambda(t(n) - Y(n)) \right] \left[y_i(n)(1 - y_i(n)) \right]$ dapat dilambangkan dengan δ_k , sehingga persamaan di atas dapat disingkat menjadi:

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (\text{D.11})$$

D.2. Perubahan bobot antar lapisan masukan dengan lapisan tersembunyi

Perubahan bobot antara lapisan masukan dengan lapisan tersembunyi dinyatakan sebagai berikut

$$\Delta v_{ij} = -\alpha \frac{\partial E_i}{\partial v_{ij}} \quad \text{di mana} \quad \frac{\partial E_i}{\partial v_{ij}} = \frac{\partial E_i}{\partial y_i} \times \frac{\partial y_i}{\partial y_{in_i}} \times \frac{\partial y_{in_i}}{\partial z_j} \times \frac{\partial z_j}{\partial z_{in_j}} \times \frac{\partial z_{in_j}}{\partial v_{ij}} \quad (\text{D.12})$$

Berikut ini adalah turunan parsial y_i terhadap y_{in_i} (sama seperti D.8)

$$\frac{\partial y_i}{\partial y_{in_i}} = f'(y_{in_i}) = y_i(n)(1 - y_i(n)) \quad (\text{D.13})$$

dan berikut adalah turunan parsial y_{in_i} terhadap w_{jk}

$$\frac{\partial y_{in_i}}{\partial z_j} = w_{jk} \quad (\text{D.14})$$

Sedangkan turunan parsial z_j terhadap z_{in_j} dinyatakan oleh:

$$\frac{\partial z_j}{\partial z_{in_j}} = f'(z_{in_j}) = z_j(n)(1 - z_j(n)) \quad (\text{D.15})$$

dan turunan parsial z_{in_j} terhadap v_{ij} dinyatakan oleh:

$$\frac{\partial z_{in_j}}{\partial v_{ik}} = x_i \quad (\text{D.16})$$

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

$$\begin{aligned} \Delta v_{ij} &= -\alpha \left\{ - \left[(1 - \lambda)(t(n) - y_i(n)) + \lambda(t(n) - Y(n)) \right] \left[y_i(n)(1 - y_i(n)) \right] w_{jk} \left[z_j(n)(1 - z_j(n)) \right] x_i \right. \\ &= \alpha \left[(1 - \lambda)(t(n) - y_i(n)) + \lambda(t(n) - Y(n)) \right] \left[y_i(n)(1 - y_i(n)) \right] w_{jk} \left[z_j(n)(1 - z_j(n)) \right] x_i \\ &= \alpha \delta_k w_{jk} \left[z_j(n)(1 - z_j(n)) \right] x_i \end{aligned} \quad (\text{D.17})$$

Suku $\delta_k w_{jk} \left[z_j(n)(1 - z_j(n)) \right] x_i$ dapat dilambangkan dengan δ_j sehingga persamaan di atas dapat disingkat menjadi:

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (\text{D.18})$$

E. Database Pelatihan dan Pengujian

Tabel 1 di bawah ini berisi fitur dataset penelitian yang berisi keterangan jumlah dimensi, jumlah kelas, serta jumlah data masing-masing dataset. Sedangkan tabel 2 berisi keterangan mengenai jenis bilangan pada masing-masing dataset

Tabel 1. Fitur Dataset Penelitian

| No | Type Data | Jumlah Dimensi | Jumlah Kelas | Jumlah Data |
|----|---------------|----------------|--------------|-------------|
| 1 | Balance Scale | 4 | 3 | 625 |
| 2 | Breast cancer | 10 | 2 | 699 |
| 3 | Bupa | 6 | 2 | 345 |
| 4 | Credit | 15 | 2 | 690 |
| 5 | Glass | 10 | 6 | 214 |
| 6 | Heart disease | 13 | 2 | 270 |
| 7 | Iris | 4 | 3 | 150 |
| 8 | Sonar | 60 | 2 | 208 |
| 9 | Vowel | 12 | 11 | 990 |
| 10 | Face IR | 1200 | 10 | 200 |
| 11 | Face visible | 900 | 10 | 100 |
| 12 | Odor | 8 | 12 | 2400 |

Tabel 2. Dimensi Dataset Penelitian beserta Jenis Bilangannya

| No | Type Data | Jumlah Dimensi | Real | Integer | Nominal |
|----|---------------|----------------|------|---------|---------|
| 1 | Balance Scale | 4 | 4 | 0 | 0 |
| 2 | Breast cancer | 10 | 0 | 9 | 0 |
| 3 | Bupa | 6 | 6 | 0 | 0 |
| 4 | Credit | 15 | 10 | 0 | 5 |
| 5 | Glass | 10 | 9 | 1 | 0 |
| 6 | Heart disease | 13 | 13 | 0 | 0 |
| 7 | Iris | 4 | 4 | 0 | 0 |
| 8 | Sonar | 60 | 60 | 0 | 0 |
| 9 | Vowel | 12 | 10 | 0 | 3 |
| 10 | Face IR | 1200 | 1200 | 0 | 0 |
| 11 | Face visible | 900 | 900 | 0 | 0 |
| 12 | Odor | 8 | 8 | 0 | 0 |

Jika terdapat nilai yang hilang atau *missing value* pada dataset UCI yang ditandai dengan tanda tanya (?), maka nilai tersebut tidak dihilangkan melainkan akan diubah menjadi 0.

Pada saat tahap pengambilan data, dataset akan dipotong jumlahnya sedemikian hingga jumlah populasi untuk tiap kelas bernilai sama. Dengan demikian, setelah tahap pengambilan data, maka jumlah populasi pada masing-masing dataset yang digunakan akan tampak seperti pada tabel 1.3 berikut.

Tabel 3. Jumlah Populasi Dataset yang digunakan dalam Penelitian

| No | Tipe Data | Jumlah kelas | Jumlah data per kelas | Jumlah data (total) |
|----|---------------|--------------|-----------------------|---------------------|
| 1 | Balance Scale | 3 | 49 | 147 |
| 2 | Breast cancer | 2 | 241 | 482 |
| 3 | Bupa | 2 | 145 | 290 |
| 4 | Credit | 2 | 307 | 614 |
| 5 | Glass | 6 | 9 | 54 |
| 6 | Heart disease | 2 | 120 | 240 |
| 7 | Iris | 3 | 50 | 150 |
| 8 | Sonar | 2 | 97 | 194 |
| 9 | Vowel | 11 | 90 | 990 |
| 10 | Face IR | 10 | 20 | 200 |
| 11 | Face visible | 10 | 10 | 100 |
| 12 | Odor | 12 | 200 | 2400 |

Berikut ini akan dijelaskan mengenai keseluruhan properties database pelatihan (*training*) terdiri dari 12 dataset yaitu dataset *Balance-scale*, dataset *Breast-cancer*, dataset *Bupa*, dataset *Credit-card*, dataset *Glass*, dataset *Heart-disease*, dataset *Iris*, dataset *Sonar*, dataset *Vowel*, dataset *Face infra-red*, dataset *Face visible*, dan dataset *Odor*.

E.1. Dataset *Balance scale*

Judul: *Balance Scale Weight & Distance Database*

Data set ini dihasilkan dari hasil eksperimen untuk memodelkan sistem timbangan (*balance scale*) secara fisik. Setiap sampel data diklasifikasikan sebagai condong ke kanan (R), condong ke kiri (L), atau seimbang (B). Pada data set ini terdapat sebanyak 625 buah data dengan atribut 5 buah yang terdiri dari 4 buah atribut parameter dan 1 buah atribut nama kelas.

Informasi atribut:

1. Nama kelas (*class name*) : 3 (L, B, R)
2. Bobot kiri (*left-weight*) : 5 (1, 2, 3, 4, 5)
3. Lengan kiri (*left-distance*) : 5 (1, 2, 3, 4, 5)
4. Bobot kanan (*right-weight*) : 5 (1, 2, 3, 4, 5)
5. Lengan kanan (*right-distance*) : 5 (1, 2, 3, 4, 5)

Distribusi data :

- 46.08 % atau sebanyak 288 data merupakan kelas L
- 07.84 % atau sebanyak 49 data merupakan kelas B
- 46.08 % atau sebanyak 288 data merupakan kelas R

Ringkasan Statistik data:

| <i>Dataset Balance-scale</i> | | | | |
|------------------------------|----------------|----------------|-------------|------------------------|
| Atribut | Minimum | Maximum | Mean | Standar Deviasi |
| <i>left-weight</i> | 1 | 5 | 3 | 1.415346 |
| <i>left-distance</i> | 1 | 5 | 3 | 1.415346 |
| <i>right-weight</i> | 1 | 5 | 3 | 1.415346 |
| <i>right-distance</i> | 1 | 5 | 3 | 1.415346 |

E.2. Data set *Breast cancer*

Judul: *Wisconsin Breast Cancer Database*

Data set kanker payudara (*breast cancer*) ini didapatkan dari Rumah Sakit Universitas Wisconsin, Madison (*University of Wisconsin Hospital*) dari Dr. William H. Wolberg (8 Januari 1991). Pada data set ini terdapat sebanyak 699 buah data dengan atribut 11 buah yang terdiri dari 1 buah atribut nomor kode sampel, 8 buah atribut parameter dan 1 buah atribut nama kelas. Setiap sampel data diklasifikasikan sebagai *Benign* (2), atau *Malignant* (4). Terdapat 16 buah nilai atribut yang hilang (*missing values*) yang dinotasikan dengan simbol tanda tanya (?)

Informasi atribut:

| | | | |
|------------------------------------|----------|--------------------------|----------|
| <i>Sample code number</i> | nomor id | <i>Bare Nuclei</i> | 1 - 10 |
| <i>Clump Thickness</i> | 1 - 10 | <i>Bland Chromatin</i> | 1 - 10 |
| <i>Uniformity of Cell Size</i> | 1 - 10 | <i>Normal Nucleoli</i> | 1 - 10 |
| <i>Uniformity of Cell Shape</i> | 1 - 10 | <i>Mitosis (mitoses)</i> | 1 - 10 |
| <i>Marginal Adhesion</i> | 1 - 10 | <i>Kelas (class)</i> | 2 atau 4 |
| <i>Single Epithelial Cell Size</i> | 1 - 10 | | |

Ringkasan Statistik data:

| <i>Dataset Breast-cancer</i> | | | | |
|------------------------------------|----------------|----------------|-------------|------------------------|
| Atribut | Minimum | Maximum | Mean | Standar Deviasi |
| <i>Clump thickness</i> | 1 | 10 | 4.41774 | 2.815741 |
| <i>Uniformity of cell size</i> | 1 | 10 | 3.134478 | 3.051459 |
| <i>Uniformity of cell shape</i> | 1 | 10 | 3.207439 | 2.971913 |
| <i>Marginal adhesion</i> | 1 | 10 | 2.806867 | 2.855379 |
| <i>Single epithelial cell size</i> | 1 | 10 | 3.216023 | 2.2143 |
| <i>Bare nuclei</i> | 1 | 10 | 3.544656 | 3.643857 |
| <i>Bland chromatin</i> | 1 | 10 | 3.437768 | 2.438364 |
| <i>Normal nucleoli</i> | 1 | 10 | 2.866953 | 3.053634 |
| <i>Mitoses</i> | 1 | 10 | 1.589413 | 1.715078 |

Distribusi data :

65.5% atau sebanyak 458 data merupakan kelas *Benign* (2).

34.5% atau sebanyak 241 data merupakan kelas *Malignant* (4).

E.3. Data set *bupa*

Judul: *BUPA liver disorders Database*

Data set ini didapatkan dari BUPA Medical Research Ltd. Lima variable pertama merupakan tes darah untuk mendeteksi kelainan hati (*liver disorder*) yang mungkin terjadi karena konsumsi alkohol yang berlebihan. Setiap baris data mewakili record dari seorang individu laki-laki yang dijadikan sampel. Pada data set ini terdapat sebanyak 345 buah data dengan atribut 7 buah yang terdiri dari 6 buah atribut parameter dan 1 buah atribut nama kelas. Setiap sampel data diklasifikasikan sebagai 1 atau 2.

Informasi atribut:

1. *mcv* (*mean corpuscular volume*)
2. *alkphos* (*alkaline phosphatase*)
3. *sgpt* (*alamine aminotransferase*)
4. *sgot* (*aspartate aminotransferase*)
5. *gammagt* (*gamma-glutamyl transpeptidase*)
6. *drinks* (jumlah minuman alkohol yang dikonsumsi per hari)
7. *selector field* (kelas)

Hasil eksperimen menunjukkan bahwa sampel yang memiliki kebiasaan minum lebih dari 5 gelas alkohol per hari menunjukkan kecenderungan memiliki kelainan hati dibandingkan sampel yang mengkonsumsi alkohol dengan jumlah yang lebih rendah.

Ringkasan Statistik data:

| <i>Dataset Bupa</i> | | | | |
|---------------------|----------------|----------------|-------------|------------------------|
| Atribut | Minimum | Maximum | Mean | Standar Deviasi |
| <i>mcv</i> | 7 | 103 | 89.71676 | 7.66249 |
| <i>alkphos</i> | 23 | 138 | 69.86957 | 18.34767 |
| <i>sgpt</i> | 4 | 155 | 30.4058 | 19.51231 |
| <i>sgot</i> | 5 | 82 | 24.64348 | 10.06449 |
| <i>gammagt</i> | 5 | 297 | 38.28406 | 39.25462 |
| <i>drinks</i> | 0 | 20 | 3.455072 | 3.337835 |

E.4. Data set *credit card*

Judul: *Credit Approval Database*

Data set ini merupakan aplikasi kartu kredit (*credit card*). Semua atribut nama dan nilai telah diubah menjadi simbol tak berarti untuk menjaga kerahasiaan data. Data set ini menarik karena terdiri dari campuran atribut bernilai kontinu, nominal dengan nilai yang kecil, serta nominal dengan nilai yang besar, dan ada pula beberapa nilai yang hilang (*missing values*). Jumlah data sebanyak 690 buah dengan atribut 16 buah yang terdiri dari 15 buah atribut parameter dan 1 buah atribut nama kelas. Setiap sampel data diklasifikasikan sebagai + atau -.

Informasi atribut:

| | | | |
|-----|--|------|-------------|
| A1: | b, a. | A9: | t, f. |
| A2: | kontinu. | A10: | t, f. |
| A3: | kontinu. | A11: | kontinu. |
| A4: | u, y, l, t. | A12: | t, f. |
| A5: | g, p, gg. | A13: | g, p, s. |
| A6: | c, d, cc, i, j, k, m, r, q, w, x, e, aa, ff. | A14: | kontinu. |
| A7: | v, h, bb, j, n, z, dd, ff, o. | A15: | kontinu |
| A8: | kontinu. | A16: | + - (kelas) |

Distribusi data :

44.5% atau sebanyak 307 data merupakan kelas +

55.5% atau sebanyak 383 data merupakan kelas -

Nilai yang hilang:

Terdapat 37 kasus dengan satu atau lebih nilai yang hilang (*missing values*) atau sekitar 5% yang dinotasikan dengan simbol tanda tanya (?) dengan rincian sebagai berikut: A1 = 12 buah, A2 = 12 buah, A4 = 6 buah, A5 = 6 buah, A6 = 9 buah, A7 = 9 buah, A14 = 13 buah

E.5.Data set *glass*

Judul: *Glass Identification Database*

Data set ini diciptakan oleh B. *German Central Research Establishment Home Office Forensic Science Service Aldermaston, Reading, Berkshire RG7 4PN* (September,1987). Studi tentang klasifikasi jenis kaca (*glass*) dimotivasi oleh investigasi kriminologi. Pada TKP (tempat kejadian perkara), kaca yang tertinggal dapat digunakan sebagai barang bukti jika dapat diidentifikasi dengan benar. Dalam data set ini terdapat sebanyak 214 buah data dengan atribut 11 buah yang terdiri dari 10 buah atribut parameter dan 1 buah atribut nama kelas. Setiap sampel data diklasifikasikan ke dalam 7 kelas yang berbeda yang merupakan representasi dari jenis kaca tertentu.

Informasi atribut:

| | | | | |
|-----|----|-------------------------|-----|--|
| 1. | ID | 1 - 214 | 11. | Tipe kaca (atribut kelas) |
| 2. | RI | <i>refractive index</i> | | <i>1: building_windows_float_processed</i> |
| 3. | Na | Sodium | | <i>2: building_windows_non_float_processed</i> |
| 4. | Mg | Magnesium | | <i>3: vehicle_windows_float_processed</i> |
| 5. | Al | Aluminum | | <i>4: vehicle_windows_non_float_processed</i> |
| 6. | Si | Silicon | | <i>5: containers</i> |
| 7. | K | Potassium | | <i>6: tableware</i> |
| 8. | Ca | Calcium | | <i>7:headlamps</i> |
| 9. | Ba | Barium | | |
| 10. | Fe | Iron | | |

Ringkasan Statistik data:

| <i>Dataset Glass</i> | | | | | |
|----------------------|---------|---------|---------|-----------------|-------------|
| Atribut | Minimum | Maximum | Mean | Standar Deviasi | Correlation |
| RI | 1.5112 | 1.5339 | 1.5184 | 0.0030 | -0.1642 |
| Na | 10.73 | 17.38 | 13.4079 | 0.8166 | 0.5030 |
| Mg | 0 | 4.49 | 2.6845 | 1.4424 | -0.7447 |
| Al | 0.29 | 3.5 | 1.4449 | 0.4993 | 0.5988 |
| Si | 69.81 | 75.41 | 72.6509 | 0.7745 | 0.1515 |
| K | 0 | 6.21 | 0.4971 | 0.6522 | -0.0100 |
| Ca | 5.43 | 16.19 | 8.9570 | 1.4232 | 0.0007 |
| Ba | 0 | 3.15 | 0.1750 | 0.4972 | 0.5751 |
| Fe | 0 | 0.51 | 0.0570 | 0.0974 | -0.1879 |

Distribusi data:

| Persentase | Jumlah data | Kelas |
|------------|-------------|------------------------------|
| 76.17 % | 163 | <i>window glass</i> |
| 40.65 % | 87 | <i>float processed glass</i> |
| 32.71 % | 70 | <i>building windows</i> |
| 7.94 % | 17 | <i>vehicle windows</i> |
| 35.51 % | 76 | <i>non-float processed</i> |
| 35.51 % | 76 | <i>building windows</i> |
| 0.00 % | 0 | <i>vehicle windows</i> |
| 23.83 % | 51 | <i>non-window glass</i> |
| 6.07 % | 13 | <i>containers</i> |
| 4.20 % | 9 | <i>tableware</i> |
| 13.55% | 29 | <i>headlamps</i> |

E.6. Data set *heart disease*

Data set ini merupakan hasil penelitian tes kesehatan untuk mengetahui tentang penyakit jantung (*heart disease*) yang diderita oleh pasien. Dalam data set ini terdapat sebanyak 270 buah data dengan atribut 14 buah yang terdiri dari 13 buah atribut parameter dan 1 buah atribut nama kelas. Setiap sampel data diklasifikasikan ke dalam 2 kelas, apakah sampel tidak memiliki penyakit jantung (1) atautkah sampel memiliki penyakit jantung (2).

Informasi Atribut:

1. Umur (*age*)
2. Jenis kelamin (*sex*)
3. Tipe nyeri dada (*chest pain type*): 4 macam
4. Tekanan darah saat istirahat (*resting blood pressure*)
5. *Serum cholestorol* : dalam mg/dl
6. Tekanan darah saat puasa (*fasting blood sugar*) > 120 mg/dl
7. Hasil *electrocardiograph*: bernilai 0,1,2
8. Detak jantung maksimum (*maximum heart rate*)
9. *Exercise induced angina*
10. *Oldpeak = ST depression induced by exercise relative to rest*
11. The slope of the peak exercise ST segment
12. Jumlah pembuluh utama (*major vessels*) : bernilai 0-3

13. Keadaan jantung:

3 = normal;

6 = cacat permanen (*fixed defect*);7 = cacat non-permanen (*reversible defect*);

14. Atribut kelas:

1 = tidak memiliki penyakit jantung (*absence of heart disease*)2 = memiliki penyakit jantung (*presence of heart disease*)

Tipe atribut:

Bilangan Real :1,4,5,8,10,12

Bilangan Urut :11

Bilangan Biner :2,6,9

Bilangan Nominal :7,3,13

Ringkasan Statistik data:

| <i>Dataset Heart-disease</i> | | | | |
|------------------------------|---------|---------|----------|-----------------|
| Atribut | Minimum | Maximum | Mean | Standar Deviasi |
| 1 | 29 | 77 | 54.43333 | 9.109067 |
| 2 | 0 | 1 | 0.677778 | 0.468195 |
| 3 | 1 | 4 | 3.174074 | 0.95009 |
| 4 | 94 | 200 | 131.3444 | 17.86161 |
| 5 | 126 | 564 | 249.6593 | 51.68624 |
| 6 | 0 | 1 | 0.148148 | 0.355906 |
| 7 | 0 | 2 | 1.022222 | 0.997891 |
| 8 | 71 | 202 | 149.6778 | 23.16572 |
| 9 | 0 | 1 | 0.32963 | 0.470952 |
| 10 | 0 | 6.2 | 1.05 | 1.14521 |
| 11 | 1 | 3 | 1.585185 | 0.61439 |
| 12 | 0 | 3 | 0.67037 | 0.943896 |
| 13 | 3 | 7 | 4.696296 | 1.940659 |

E.7. Data set *iris*Judul: *Iris Plants Database*

Data set ini diciptakan oleh R.A. Fisher (Juli, 1988) dan didonasikan oleh Michael Marshal. Dataset ini merupakan salah satu database terbaik untuk literatur pengenalan pola (*pattern recognition*).

Data set ini terdiri dari 3 kelas dengan 50 *instance* pada masing-masing kelas yang merepresentasikan jenis tanaman iris (*iris plant*). Terdapat satu kelas yang terpisah secara linier (*linearly separable*) dari dua kelas lainnya, sedangkan sisanya tidak terpisah secara linier dari kelas lainnya.

Dalam data set ini terdapat sebanyak 150 buah data dengan atribut 5 buah yang terdiri dari 4 buah atribut parameter bernilai numeric dan 1 buah atribut nama kelas. Setiap sampel data diklasifikasikan ke dalam 3 kelas yang berbeda yang merupakan representasi dari jenis tanaman iris yaitu *Iris Setosa* (1), *Iris Versicolour* (2), atau *Iris Virginica* (3).

Informasi atribut:

1. Panjang kelopak bunga (*sepal length*) dalam cm
2. Lebar kelopak bunga (*sepal width*) dalam cm
3. Panjang mahkota bunga (*petal length*) dalam cm
4. Lebar mahkota bunga (*petal width*) dalam cm
5. Atribut kelas:
 - 1 = *Iris Setosa*
 - 2 = *Iris Versicolour*
 - 3 = *Iris Virginica*

Ringkasan Statistik data:

| Dataset Iris | | | | | |
|---------------------|---------|---------|------|-----------------|-------------|
| Atribut | Minimum | Maximum | Mean | Standar Deviasi | Correlation |
| <i>sepal length</i> | 4.3 | 7.9 | 5.84 | 0.83 | 0.7826 |
| <i>sepal width</i> | 2.0 | 4.4 | 3.05 | 0.43 | -0.4194 |
| <i>petal length</i> | 1.0 | 6.9 | 3.76 | 1.76 | 0.9490 |
| <i>petal width</i> | 0.1 | 2.5 | 1.20 | 0.76 | 0.9565 |

Distribusi data :

- 33.3% atau sebanyak 50 data merupakan kelas *Iris Setosa* (1).
- 33.3% atau sebanyak 50 data merupakan kelas *Iris Versicolour* (2).
- 33.3% atau sebanyak 50 data merupakan kelas *Iris Virginica* (3).

E.8. Data set sonar

Judul: *Sonar, Mines vs. Rocks Database*

Data set ini merupakan koleksi benchmark oleh Terry Sejnowski, dari Salk Institute dan the University of California, San Deigo. Data set ini dikembangkan secara berkolaborasi dengan R. Paul Gorman dari *Allied-Signal Aerospace Technology Center*. Data set ini telah digunakan oleh Gorman dan Sejnowski dalam studi mereka tentang klasifikasi sinyal sonar menggunakan jaringan syaraf tiruan (*artificial neural network*). Tugas yang dilakukan adalah melatih JST untuk membedakan antara sinyal sonar yang dipantulkan oleh logam silinder (*metal cylinder*) dan sinyal sonar yang dipantulkan oleh batu silinder yang kasar (*roughly cylindrical rock*).

File "sonar.mines" terdiri dari 111 pola yang didapat dari sinyal sonar yang dipantulkan pada logam silinder pada berbagai sudut (dengan range 90 derajat) dan berbagai kondisi. File "sonar.rocks" terdiri dari pola yang didapat dari sinyal sonar yang dipantulkan pada batu silinder pada berbagai sudut (dengan range 180 derajat) dan berbagai kondisi. Sinyal sonar yang ditransmisikan merupakan chirp dengan frekuensi termodulasi. Setiap pola merupakan set dengan nilai antara 0.0 hingga 1.0. Setiap nomer merepresentasikan energy pada pita frekuensi (*frequency band*) tertentu. Label yang diasosiasikan dengan masing-masing record mengandung huruf "R" jika objeknya merupakan batu (*rock*) dan "M" jika objeknya merupakan logam (*metal*).

E.9. Data set vowel

Judul: *Vowel Recognition (Deterding data)*

Data set ini digunakan untuk pengenalan pembicara (*speaker*) yang terdiri dari 11 huruf hidup tunak (*steady state vowels*) bahasa Inggris (*British English*). Data set ini diciptakan oleh David Deterding, Mahesan Niranjan, Tony Robinson.

File "vowel.data" terdiri dari array 3 dimensi yang berisi data vowel: *speaker, vowel, input*. Pembicara diberi nomer urut bilangan bulat antara 0 hingga 89, dimana terdapat 15 orang individu pembicara (*speaker*) yang

mengucapkan setiap vowel sebanyak enam kali. Data vowel diberi index bilangan bulat antara 0 hingga 10. Sedangkan nilai input merupakan nilai *floating-point* antara 0 hingga 9.

Struktur File:

| Kolom | Deskripsi |
|-------|--------------------------------|
| 0 | Data Train atau Data Test |
| 1 | Nomer kode <i>pembicara</i> |
| 2 | Jenis kelamin <i>pembicara</i> |
| 3 | Fitur 0 |
| 4 | Fitur 1 |
| 5 | Fitur 2 |
| 6 | Fitur 3 |
| 7 | Fitur 4 |
| 8 | Fitur 5 |
| 9 | Fitur 6 |
| 10 | Fitur 7 |
| 11 | Fitur 8 |
| 12 | Fitur 9 |
| 13 | Kelas |

Kode pembicara:

| Kode | Nama Pembicara |
|------|----------------|
| 0 | Andrew |
| 1 | Bill |
| 2 | David |
| 3 | Mark |
| 4 | Jo |
| 5 | Kate |
| 6 | Penny |
| 7 | Rose |
| 8 | Mike |
| 9 | Nick |
| 10 | Rich |
| 11 | Tim |
| 12 | Sarah |
| 13 | Sue |
| 14 | Wendy |

Kode data set:

| Kode | Deskripsi |
|------|------------|
| 0 | Data Train |
| 1 | Data Test |

Kode jenis kelamin pembicara:

| Kode | Jenis kelamin |
|------|---------------|
| 0 | Laki-laki |
| 1 | Perempuan |

Kode atribut kelas:

| Kode | Kelas |
|------|-------|
| 0 | hid |
| 1 | hId |
| 2 | hEd |
| 3 | hAd |
| 4 | hYd |
| 5 | had |
| 6 | hOd |
| 7 | hod |
| 8 | hUd |
| 9 | hud |
| 10 | hed |

Kode pembicara:

| Kode | Pembicara | Jenis Kelamin | Data set |
|------|-----------|---------------|----------|
| 0 | Andrew | 0 | 0 |
| 1 | Bill | 0 | 0 |
| 2 | David | 0 | 0 |
| 3 | Mark | 0 | 0 |
| 4 | Jo | 1 | 0 |
| 5 | Kate | 1 | 0 |
| 6 | Penny | 1 | 0 |
| 7 | Rose | 1 | 0 |
| 8 | Mike | 0 | 1 |
| 9 | Nick | 0 | 1 |
| 10 | Rich | 0 | 1 |
| 11 | Tim | 0 | 1 |
| 12 | Sarah | 1 | 1 |
| 13 | Sue | 1 | 1 |
| 14 | Wendy | 1 | 1 |

Ringkasan Statistik data:

| <i>Dataset Vowel</i> | | | | |
|----------------------|---------|---------|----------|-----------------|
| Atribut | Minimum | Maximum | Mean | Standar Deviasi |
| 1 | 1 | 14 | 7 | 4.322678 |
| 2 | 0 | 1 | 0.466667 | 0.49914 |
| 3 | -5.211 | -0.941 | -3.20374 | 0.868987 |
| 4 | -1.274 | 5.074 | 1.881764 | 1.175272 |
| 5 | -2.487 | 1.431 | -0.50777 | 0.711948 |
| 6 | -1.409 | 2.377 | 0.515483 | 0.759261 |
| 7 | -2.127 | 1.831 | -0.30566 | 0.664602 |
| 8 | -0.836 | 2.327 | 0.630244 | 0.603871 |
| 9 | -1.537 | 1.403 | -0.00436 | 0.461927 |
| 10 | -1.293 | 2.039 | 0.336553 | 0.573302 |
| 11 | -1.613 | 1.309 | -0.30298 | 0.570162 |
| 12 | -1.68 | 1.396 | -0.07134 | 0.603985 |

E.10. Data set *Face Infra-red*

Merupakan dataset berisi 200 foto (10 orang@20) berdimensi 40x30 pixel dalam format .jpg yang diambil dari kamera infra-merah. 10 orang objek gambar 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 ± 50 cm. 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. Video tersebut dicropping secara manual untuk mendapatkan citra wajah frontal, kemudian gambar diresize sehingga berukuran 40x30 pixel.

Berikut ini gambar dengan ukuran aslinya:

- Foto orang ke-1 (anne)



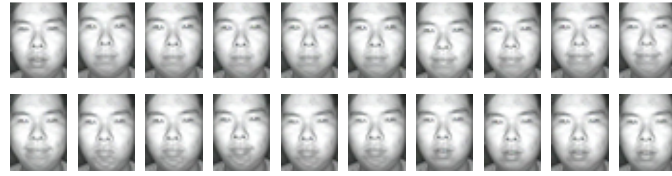
- Foto orang ke-2 (daus)



- Foto orang ke-3 (cindy)



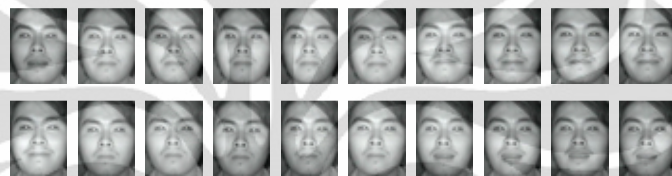
- Foto orang ke-4 (steve)



- Foto orang ke-5 (devi)



- Foto orang ke-6 (chris)



- Foto orang ke-7 (pelangi)



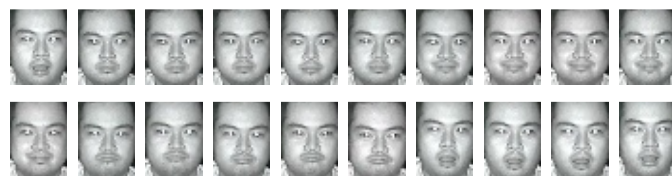
- Foto orang ke-8 (rudi)



- Foto orang ke-9 (eni)



- Foto orang ke-10 (pepsi)



E.11. Data set Face *Visible*

Merupakan dataset berisi 100 foto (10 orang@10) berdimensi 30x30 pixel yang diambil dari kamera cahaya tampak. Berikut ini gambar dengan ukuran aslinya:

- Foto orang ke-1



- Foto orang ke-2



- Foto orang ke-3



- Foto orang ke-4



- Foto orang ke-5



- Foto orang ke-6



- Foto orang ke-7



- Foto orang ke-8



- Foto orang ke-9



- Foto orang ke-10



E.12. Data set *Odor*

Merupakan dataset aroma produk kosmetik *Martha Tilaar* yang terdiri dari campuran tiga aroma (*odor*) yang terdiri dari aroma buah jeruk (*citrus*), bunga kenanga (*cannagga*), dan bunga melati (*jasmine*) yang ditambahkan dengan larutan alcohol dengan konsentrasi bervariasi sebesar 0%,15%, 25%, 35%, 45%, atau 70% .

Aroma tersebut diukur menggunakan 8 buah sensor resonator kuarsa yang dilapisi membrane sensitif dengan frekuensi dasar 20 MHz. Secara keseluruhan, data odor terdiri dari 18 macam campuran (kelas) secara keseluruhan. Masing- masing kelas terdiri dari 200 data, sehingga total terdapat 3600 data.

Pada penelitian ini, digunakan 3 buah dataset yang masing-masing terdiri dari 12 kelas dan merupakan yang merupakan gabungan dua per tiga bagian dari dataset odor yang asli. *Tipe 1* merupakan campuran jeruk-kenanga-alkohol, *tipe 2* merupakan campuran jeruk-melati-alkohol, sedangkan *tipe 3* merupakan campuran melati-kenanga-alkohol. Maka dataset A merupakan gabungan tipe 1 dan 2, dataset B merupakan gabungan tipe 2 dan 3, sedangkan dataset C merupakan gabungan tipe 1 dan 3. Berikut ini merupakan keterangan unsur campuran aroma dari masing-masing kelas.

| Dataset <i>Odor</i> A | Dataset <i>Odor</i> B | Dataset <i>Odor</i> C |
|---------------------------|----------------------------|----------------------------|
| Jeruk-Kenanga-Alkohol 0% | Jeruk-Melati-Alkohol 0% | Jeruk-Kenanga-Alkohol 0% |
| Jeruk-Kenanga-Alkohol 15% | Jeruk-Melati-Alkohol 15% | Jeruk-Kenanga-Alkohol 15% |
| Jeruk-Kenanga-Alkohol 25% | Jeruk-Melati-Alkohol 25% | Jeruk-Kenanga-Alkohol 25% |
| Jeruk-Kenanga-Alkohol 35% | Jeruk-Melati-Alkohol 35% | Jeruk-Kenanga-Alkohol 35% |
| Jeruk-Kenanga-Alkohol 45% | Jeruk-Melati-Alkohol 45% | Jeruk-Kenanga-Alkohol 45% |
| Jeruk-Kenanga-Alkohol 70% | Jeruk-Melati-Alkohol 70% | Jeruk-Kenanga-Alkohol 70% |
| Jeruk-Melati-Alkohol 0% | Kenanga-Melati-Alkohol 0% | Kenanga-Melati-Alkohol 0% |
| Jeruk-Melati-Alkohol 15% | Kenanga-Melati-Alkohol 15% | Kenanga-Melati-Alkohol 15% |
| Jeruk-Melati-Alkohol 25% | Kenanga-Melati-Alkohol 25% | Kenanga-Melati-Alkohol 25% |
| Jeruk-Melati-Alkohol 35% | Kenanga-Melati-Alkohol 35% | Kenanga-Melati-Alkohol 35% |
| Jeruk-Melati-Alkohol 45% | Kenanga-Melati-Alkohol 45% | Kenanga-Melati-Alkohol 45% |
| Jeruk-Melati-Alkohol 70% | Kenanga-Melati-Alkohol 70% | Kenanga-Melati-Alkohol 70% |

Untuk membantu mengilustrasikan dataset input eksperimen, berikut ini adalah atribut statistik data input dari akuisisi sensor data untuk setiap jenis dataset.

| <i>Dataset A</i> | | | | |
|------------------|---------|---------|----------|-----------------|
| Sensor | Minimum | Maximum | Mean | Standar Deviasi |
| 1 | 0.18350 | 0.20577 | 0.194946 | 0.004191 |
| 2 | 0.09859 | 0.13024 | 0.113532 | 0.008129 |
| 3 | 0.03004 | 0.04017 | 0.035411 | 0.001859 |
| 4 | 0.05035 | 0.08836 | 0.060773 | 0.011091 |
| 5 | 0.11833 | 0.13677 | 0.128192 | 0.003299 |
| 6 | 0.35267 | 0.40703 | 0.390037 | 0.010324 |
| 7 | 0.05233 | 0.07111 | 0.058621 | 0.003611 |
| 8 | 0.01327 | 0.02228 | 0.01849 | 0.001734 |

| <i>Dataset B</i> | | | | |
|------------------|----------|----------|----------|-----------------|
| Sensor | Minimum | Maximum | Mean | Standar Deviasi |
| 1 | 0.186240 | 0.205770 | 0.196127 | 0.003732 |
| 2 | 0.098590 | 0.125770 | 0.109549 | 0.007248 |
| 3 | 0.030900 | 0.037150 | 0.034965 | 0.000899 |
| 4 | 0.051080 | 0.088360 | 0.061286 | 0.011040 |
| 5 | 0.120110 | 0.136770 | 0.128966 | 0.003074 |
| 6 | 0.357650 | 0.407030 | 0.392379 | 0.009600 |
| 7 | 0.052970 | 0.062560 | 0.057581 | 0.001792 |
| 8 | 0.016170 | 0.022870 | 0.019146 | 0.001577 |

| <i>Dataset C</i> | | | | |
|------------------|----------|----------|----------|-----------------|
| Sensor | Minimum | Maximum | Mean | Standar Deviasi |
| 1 | 0.183500 | 0.205770 | 0.195596 | 0.004127 |
| 2 | 0.098600 | 0.130240 | 0.110712 | 0.007634 |
| 3 | 0.030040 | 0.040170 | 0.035257 | 0.001959 |
| 4 | 0.050350 | 0.088360 | 0.060904 | 0.011178 |
| 5 | 0.118330 | 0.136770 | 0.128656 | 0.003302 |
| 6 | 0.352670 | 0.407030 | 0.391169 | 0.010054 |
| 7 | 0.052330 | 0.071110 | 0.058437 | 0.003566 |
| 8 | 0.013270 | 0.022870 | 0.019271 | 0.002150 |

DAFTAR REFERENSI

- [1] Siang, Jong Jek. (2009). *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan MATLAB*, page 97. Yogyakarta: Penerbit ANDI.
- [2] Imantaka, Stephen Roy. (2010). *Skripsi: Sistem Pengenal Wajah Berbasis Neural Network Ensemble untuk Citra Infra Merah*. Depok: Program Studi Teknik Elektro, Universitas Indonesia.
- [3] Kline D. & Berardi V. (2005). "Revisiting Squared Error and Cross-Entropy Functions for Training Neural Network Classifiers", *Springer Neural Computing & Application* 14:310-318.
- [4] Kusumoputro, Benyamin. (2001). *Jaringan Neural Buatan: Bahan Belajar Mahasiswa*. Fakultas Ilmu Komputer Universitas Indonesia. Depok: Universitas Indonesia
- [5] Rokach, Lior (2009). "Ensemble-based Classifiers", *Springer Science+Business Media B.V. Artif.Intell.Rev* 2010, 33:1-39.
- [6] Jolliffe, I.T. (2002). *Principal Component Analysis (second edition)*. Springer Series in Statistics.
- [7] Buchala, Samarasena et al. Principal Component Analysis of Gender, Ethnicity, Age, and Identity of Face Images. <http://info.scopus.com>.
- [8] Smith, Lindsay I. (2002). A Tutorial on Principal Component Analysis. <http://info.scopus.com>.
- [9] Lee H., Hong S., & Kim E. (2009). "Neural Network Ensemble with Probabilistic Fusion and Its Application to Gait Recognition", *Elsevier Neurocomputing*, 72: 1557-1564.
- [10] Dam, H.H et al. (2008). "Neural-Based Learning Classifier Systems", *IEEE Transactions on Knowledge and Data Engineering*, Vol.20(1): 26-39.
- [11] Liu, Y. & X.Yao (1999). "Ensemble Learning via Negative Correlation", *Pergamon Neural Networks* 12, 1299-1404
- [12] Newman D., Hettich S, Blake C., Merz C. (1998). "UCI Repository of Machine Learning Databases", Dept. of Information and Computer Science, Univ. of California at Irvine, <http://www.ics.uci.edu/~mlern/MLRepository.html>

- [13] Villiers J. & Barnard E. (1992). "Backpropagation Neural Nets with One and Two Hidden Layers", *IEEE Transactions on Neural Networks*, Vol.4(1): 136-142.
- [14] Heden Bo et al. (1996). "Agreement between Artificial Neural Networks and Human Expert for the Electrocardiographic Diagnosis of Healed Myocardial Infarction", *Journal of the American College of Cardiology*, 28: 1012-1016.
- [15] Chan Z. & Kasabov N. (2005). "A Preliminary Study on Negative Correlation Learning via Correlation-Corrected Data (NCCD)", *Springer Neural Processing Letters*, 21:207-214.
- [16] Chen, Xin et al. (2005). "IR and Visible Light Face Recognition", *Elsevier Computer Vision and Image Understanding*, 99: 332-358.
- [17] Jain et al. (1999). "Biometrics: Personal Identification in Networked Society", Kluwer Academic Publishers.
- [18] Srivastava & X.Liu (2003). "Statistical Hypothesis Pruning for Identifying Faces from Infrared Images", *Image Vision Comput.* 21:651-661.
- [19] J.Wilder, P.J. Phillips, C.Jiang, S.Wiener. (1996). "Comparison of Visible and Infrared Imagery for Face Recognition", in *2nd International Conference on Automatic Face and Gesture Recognition, Killington*, page 182-187.
- [20] Fu, Limin. (1994). *Neural Networks in Computer Intelligence* (International edition, McGraw-Hill Series in Computer Science). New York: McGraw-Hill, Inc.
- [21] Kusumoputro, B & Aripin Ariyanto. (1996). "Neural Networks Diagnosis of Malignant Skin Cances Using Principal Component Analysis as a Preprocessor". Depok: Faculty of Computer Science, Universitas Indonesia.
- [22] Humpert. (1994). "Improving Backpropagation with a New Error Function", *Neural Networks*, Vol.7(8).
- [23] Matsuoka & J.Q.Yi. (1991). "Backpropagation based on the Logarithmic Error function and Elimination of Local Minima", *International Joint Conference on Neural Networks*, Vol.2.
- [24] Kusumoputro, B & Wisnu Jatmiko. (2002). "Pengembangan Sistem Penciuman Elektronik dengan 16 buah Sensor Kuarsa dan Algoritma Neural Propagasi Balik untuk Pengenalan Aroma Campuran". *Makara Sains Universitas Indonesia Vol.6(3):110-116*.

- [25] Ikegami, H.Arita, S.Iwanaga, M.Kaneyasu (1983). *Proceedings of the 4th European Hyb Microelectronics Conference*, page 122.
- [26] D.S. Balantine, S.L.Rose, J.W. Grante, H.Wohltjen. (1986). *Anal. Chem.* 58:3058.
- [27] J.R.Stetter, P.C.Jurs, S.L.Rose. (1986). *Anal. Chem.* 58:860.
- [28] Kusumoputro, B & M.Rivai. (1998). *Proceeding of International Conference on Computational Intelligence and Multimedia Applications, Melbourne, Australia*, page 264.
- [29] Jatmiko, Wisnu. (2000). *Thesis S2*. Depok: Program Studi Ilmu Komputer, Universitas Indonesia.
- [30] Xingming, Zhang. (2008). "A Biological Intelligence Access Control System Based on DSP and NIR Technology". *Springer-Verlag ICIC 2008,LNCS 5226*, page 58
- [31] Shuyan, Zhao.(2005). "An Automatic Face Recogniton System in the Near Infrared Spectrum". *Leipzig MLDM*, page 438.
- [32] Brown, G. (2004). "Diversity in Neural Network Ensembles". *PhD dissertation*. School of Computer Science University of Birmingham.
- [33] Yao,X & Y. Liu. (1998) "Making Use of Population Information in Evolutionary Artificial Neural Networks". *IEEE Transactions on Systems, Man, and Cybernetics.Part B: Cybernetics*, 28: 417-425.
- [34] Islam, M., Xin Yao, Kazuyuki Murase. (2003). "A Constructive Algorithm for Training Cooperative Neural Network Ensembles". *IEEE Transactions on Neural Networks*, Vol.14, No. 4, 820-834.
- [35] P.Phillips, P.Gother, R.Michealss, D.Blackburn, E.Tabassi, J.Bone .(2003). *Frvt 2002: Overview and Summary*.
- [36] Kirby, M. & L.Sirovich. (1990). "Application of the Karhunen-Loeve procedure for the characterization of Human Faces". *IEEE Transaction on Pattern Analysis And Machine Intelligence*, 12(1): 103-108.
- [37] Turk, M. & A.Pentland. (1991). "Face Recognition using Eigenfaces". *Comput. Vision Pattern Recognition*.
- [38] Leshno, M. & Y.Spector. (1996). "Neural network prediction Analysis: The bankruptcy case". *Neurocomput.* Vol.10: 125-147

- [39] G.Zhang, M.Y.Hu., E.B. Patuwo, D.Indro. (1999). "Artificial Neural Networks in Bankruptcy Prediction: General framework and cross-validation analysis". *Eur.J.Oper.res. Vol.116:16-32*
- [40] S.Knerr, L.Personnaz, G.Dreyfus. (1992). "Handwritten Digit Recognition by Neural Networks with Single-layer Training". *IEEE Transaction on Neural Networks. Vol 3: 962-968*
- [41] Guyon,I. (1991)."Applications of Neural Networks to Character Recognition". *Int.J.Pattern Recognition Artificial Intelligence Vol.5: 353-382*
- [42] Burke, H.B. (1994). "Artificial Neural Networks for Cancer Research: Outcome Prediction". *Sem.Surg. Oncol. Vol.10: 73-79*
- [43] Burke, H.B et al. (1997)." Artificial neural Networks Improve the Accuracy of Cancer Survival Prediction". *Cancer, Vol 79:857-862.*
- [44] Barlett, E.B. & R.E. Uhrig. (1992)." Nuclear Power Plant Status Diagnostics using Artificial Neural Networks". *Nuclear Technology, Vol.97: 271-281*
- [45] J.C.Hoskins, K.M. kaliyur, D.M. Himmelblau. (1990). "Incipient Fault Detection and Diagnosis using Artificial Neural networks". *Proceeding International Joint Conference on Neural networks : 81-86*
- [46] Bourlard, h. & N. Morgan. (1993). "Continuous Speech Recognition by Connectionist Statistical Methods". *IEEE Transaction on Neural networks, Vol.4: 893-909*
- [47] Lippmann, R.P. (1989). "Review of Neural Networks for Speech Recognition". *Neural Comput. Neural System, Vol.1:299-323.*
- [48] Petsche, T. (1998). "An autoassociator for on-line motor monitoring". *Industrial Applications of neural Networks, page 91-97, F.F. Soulie and P. Gallinari, Eds, Singapore: World Scientific.*
- [49] J.Lampinen, S.Smolander, M.Korhonen. (1998). "Wood Surface Inspection System based on Generic Visual Features". *Industrial Applications of Neural Networks, page 35-42. F.F. Soulie and P. Gallinari, Eds, Singapore: World Scientific.*

- [50] Saaty, X.(1996). “The Analytic Hierarchy Process: A 1993 Overview”. *Cent Eur J Oper Res Econ* 2(2): 11-137.
- [51] Zhang, Guoqiang P. (2000). “Neural Networks for Classification: A Survey”, *IEEE Transactions on System, MAN, and Cybernetics, Vol.30, No.4, 451-462.*
- [52] Wan, E. (1990). “Neural Network Classification: A Bayesian Interpretation”. *IEEE Transaction on Neural Networks, Vol.1 (4): 303-305*
- [53] M.S.Hung, M.Y.Hu, M.S.Shanker, B.E.Patuwo.(1996). “Estimating Posterior Probabilities In Classification Problems With Neural Networks”, *International Journal of Computational Intelligence and Organizations, Vol.1(1): 49-60*
- [54] Richard, M.D. & R.Lippmann. (1991). “Neural Network Classifiers estimate Bayesian a posteriori probabilities”. *Neural Computing Vol.3 : 461-483*
- [55] D.E. Rumelhart, R.Durbin. R.Golden, Y.Chauvin. (1995). “Backpropagation: The Basic Theory”. *Backpropagation: Theory, Architectures, and Applications, page 1-34.* Y.Chauvin & D.E. Rumelhart. Hillsdale, NJ: LEA.
- [56] Holt, M.J. & S.Semnani. (1990). “Convergence of backpropagation in Neural Networks using a Log-likelihood Cost Function”. *Electron. Letter, Vol.26 (23): 1964-1965.*
- [57] Barnard, E. & E.C. Botha. (1993). “Backpropagation uses prior information efficiently”. *IEEE Transactions on Neural Networks, Vol.4:794-802.*
- [58] Lippman, R.P. (1987). “An Introduction to Computing with Neural Nets”. *IEEE ASSP Magazine, page 4-22*
- [59] Pardo, M. & G.Sberveglieri. (2004). “Remarks on the use of Multilayer Perceptron for the Analysis of Chemical Sensor Array Data”. *IEEE Sensors Journals Vol.4 (3): 355-365.*
- [60] Rao, V.B. & H.V. Rao. (1993). *Neural network and Fuzzy Logic.* New York: MIS Press.

- [61] Miller J.W. & Goodman R. (1993). "On Loss Functions Which Minimize to Conditional Expected Values and Posterior Probabilities", *IEEE Transactions on Information Theory*, Vol.39, No.4, 1404-1408
- [62] Mannor S. et al. (2005). "The Cross Entropy Method for Classification", *Proceedings of the 22nd International Conference on Machine Learning, Bonn, Germany*.
- [63] Y.Lee, S.H.Oh, M.W. Kim.(1993). "An Analysis of Premature Saturation in backpropagation Learning". *Neural networks, Vol.6: 719-728*.
- [64] Oh, S.H. & Y.Lee. (1995). "A Modified Error Function to Improve the Error Backpropagation Algorithm for Multilayer Perceptrons". *ETRI Journal*, Vol.17(1): 11-22
- [65] Falas, T. & Andreas-G.Stafylopatis. (1999). "The Impact of the Error Selection in neural network-based Classifiers." *IEEE 0-7803-5529-6/99*, page 1799-1804.