



UNIVERSITAS INDONESIA

**VARIASI TEKNIK OPTIMASI PROBABILISTIC NEURAL
NETWORK MENGGUNAKAN GENETIC ALGORITHMS
DAN ORTHOGONAL ALGORITHMS DALAM APLIKASI
SISTEM PENGENAL WAJAH**

SKRIPSI

**STEPHEN
0706268032**

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



UNIVERSITAS INDONESIA

**VARIASI TEKNIK OPTIMASI PROBABILISTIC NEURAL
NETWORK MENGGUNAKAN GENETIC ALGORITHMS
DAN ORTHOGONAL ALGORITHMS DALAM APLIKASI
SISTEM PENGENAL WAJAH**

SKRIPSI

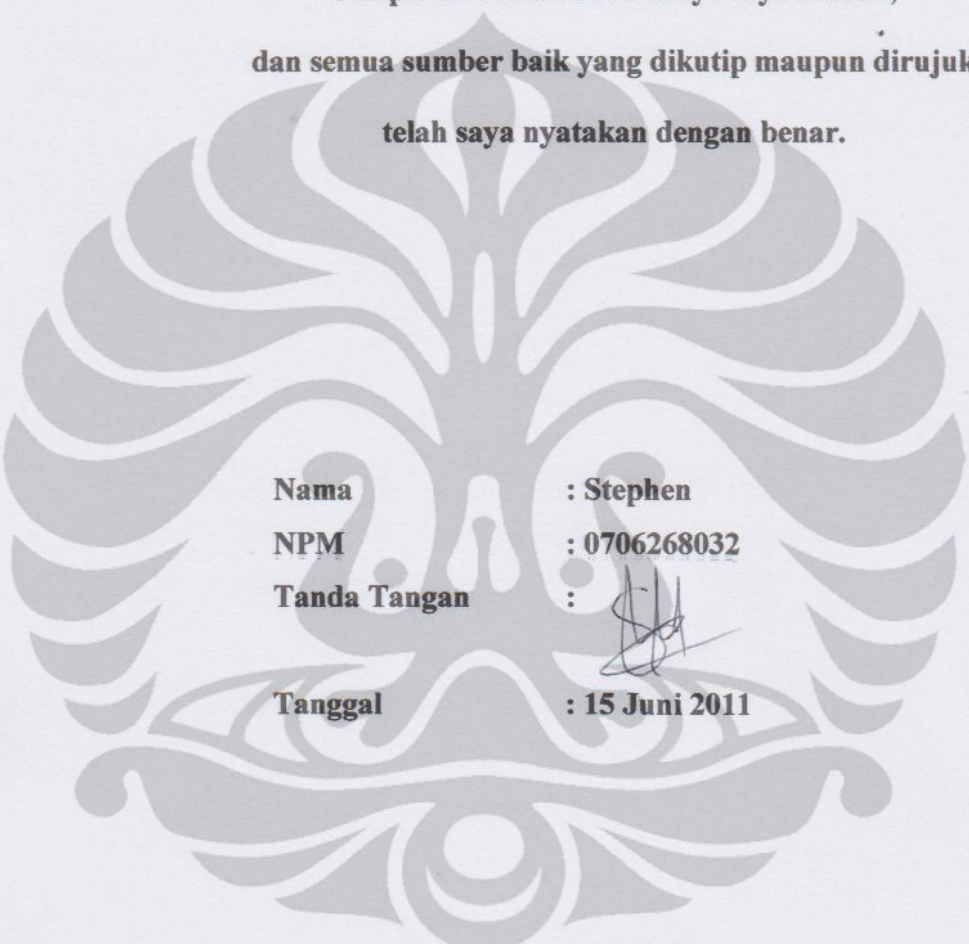
Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik


**STEPHEN
0706268032**

**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 : Stephen
NPM : 0706268032
Tanda Tangan : 
Tanggal : 15 Juni 2011


HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Stephen
NPM : 0706268032
Program Studi : Teknik Elektro
Judul Skripsi : Variasi Teknik Optimasi Probabilistic Neural Network Menggunakan Genetic Algorithms dan Orthogonal Algorithms Dalam Aplikasi Sistem Pengenal Wajah

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

Penguji :
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.

Penyusunan skripsi ini tidak semata-mata dihasilkan dari usaha penulis saja, tetapi juga berkat bantuan, dukungan, dan saran yang diberikan orang-orang lain di sekeliling penulis mulai dari awal masa perkuliahan hingga saat ini. Oleh karena itu, penulis bermaksud untuk 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 untuk terus menuntut ilmu;
- 3) Alifia Fithritama, Dimas Adityamurthi, dan Harry Bian Pramudia sebagai rekan satu bimbingan yang telah memberi masukan, ide, semangat, dan saling mendukung dari awal hingga akhir penyusunan skripsi ini;
- 4) Para dosen Departemen Teknik Elektro Universitas Indonesia atas pengetahuan yang diberikan selama ini; dan
- 5) Teman-teman, rekan-rekan, dan pihak-pihak lain yang baik secara langsung ataupun tidak langsung membantu penulis dalam menyelesaikan skripsi ini.

Penulis menyadari bahwa dalam penulisan skripsi ini tidak terlepas dari kekurangan dan kesalahan. Oleh karena itu, penulis mohon maaf atas kekurangan atau kesalahan tersebut. Semoga semua hal yang telah dilakukan ini dapat memberikan hal positif bagi semua pihak yang terkait.

Depok, 15 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 : Stephen
NPM : 0706268032
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:

**VARIASI TEKNIK OPTIMASI PROBABILISTIC NEURAL
NETWORK MENGGUNAKAN GENETIC ALGORITHMS
DAN ORTHOGONAL ALGORITHMS DALAM APLIKASI
SISTEM PENGENAL WAJAH**

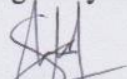
beserta perangkat yang ada. Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai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

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 15 Juni 2011

Yang menyatakan


(Stephen)

ABSTRAK

Nama : Stephen
Program Studi : Teknik Elektro
Judul : Variasi Teknik Optimasi Probabilistic Neural Network
Menggunakan Genetic Algorithms dan Orthogonal Algorithms
Dalam Aplikasi Sistem Pengenal Wajah

Proses optimasi pada *Probabilistic Neural Network* (PNN) dapat dilakukan terhadap nilai *smoothing parameter* maupun struktur neuron. Setiap permasalahan memiliki nilai *smoothing parameter* optimal yang berbeda. Optimasi struktur neuron bertujuan untuk mereduksi banyak neuron yang digunakan sehingga dapat mempersingkat waktu komputasi. Skripsi ini membahas proses pencarian nilai *smoothing parameter* optimal menggunakan algoritma genetika dan struktur neuron optimal menggunakan algoritma ortogonal dalam sistem pengenal wajah. Terdapat dua jenis teknik optimasi yang akan dibahas, lalu membandingkan hasilnya dengan PNN struktur utuh dan backpropagation. Data wajah yang digunakan berupa foto infra merah dan cahaya tampak.

Kata kunci:

Probabilistic Neural Network, *smoothing parameter*, struktur neuron, algoritma genetika, algoritma ortogonal

ABSTRACT

Name : Stephen
Study Program: Electrical Engineering
Title : Development of Probabilistic Neural Network Optimization
Technique Using Genetic Algorithms and Orthogonal Algorithms
and its Application in Face Recognition System

Optimization of Probabilistic Neural Network (PNN) can be performed to the value of smoothing parameter and neuron structure. Every problem has different value of smoothing parameter. Optimization of neuron structure aims to reduce the number of neurons used, in order to shorten computation time. This thesis discusses the process of finding the optimal value of smoothing parameter using genetic algorithms and optimal neuron structure using orthogonal algorithms in face recognition system. Two types of optimization techniques which will be discussed, then the results are compared with full structure PNN and backpropagation. Face data used in the form of infrared and visible light images.

Keywords:

Probabilistic Neural Network, smoothing parameter, neuron structure, genetic algorithms, orthogonal algorithms

DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS	ii
HALAMAN PENGESAHAN.....	iii
KATA PENGANTAR.....	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS	v
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL	xiv
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Tujuan Penelitian.....	3
1.3. Pembatasan Masalah	3
1.4. Metode Penulisan	4
1.5. Sistematika Penulisan.....	4
BAB 2 PROBABILISTIC NEURAL NETWORK, TEKNIK OPTIMASI, DAN SKEMA PERCOBAAN	6
2.1. Data Penelitian	6
2.1.1. Principal Component Analysis (PCA).....	6
2.1.2. Algoritma Kompensasi dan Normalisasi Iluminasi.....	9
2.1.3. Pengolahan Data Penelitian	13
2.2. Jaringan Saraf Tiruan Algoritma Probabilistic Neural Network.....	18
2.2.1. Klasifikasi Bayesian dan Estimator PDF Parzen.....	18
2.2.2. Probabilistic Neural Network	21
2.3. Optimasi Struktur Neuron JST PNN	24
2.3.1. Proses Gram-Schmidt	24
2.3.2. Metode Pencarian Struktur Neuron Optimal dengan Orthogonal Algorithms	26
2.4. Optimasi Nilai Smoothing Parameter JST PNN	30

2.4.1.	Genetic Algorithms.....	30
2.4.2.	Metode Pencarian Nilai Smoothing Parameter Optimal dengan Genetic Algorithms.....	35
2.5.	Skema Percobaan	38
BAB 3 NILAI SMOOTHING PARAMETER OPTIMAL DENGAN GENETIC ALGORITHMS 43		
3.1.	Tujuan Percobaan	43
3.2.	Prosedur Percobaan	44
3.3.	Hasil dan Analisis Percobaan	45
3.3.1.	Hasil dan Analisis Percobaan dengan Data IR-TNI.....	47
3.3.2.	Hasil dan Analisis Percobaan dengan Data IR-NI.....	49
3.3.3.	Analisis Pengaruh Proses NI pada Data IR	52
3.3.4.	Hasil dan Analisis Percobaan dengan Data VL-TNI.....	54
3.3.5.	Hasil dan Analisis Percobaan dengan Data VL-NI	58
3.3.6.	Analisis Pengaruh Proses NI pada Data VL.....	60
3.4.	Kesimpulan.....	62
BAB 4 TEKNIK OPTIMASI PROBABILISTIC NEURAL NETWORK 63		
4.1.	Tujuan Percobaan	63
4.2.	Teknik Optimasi PNN Berorientasi Nilai Smoothing Parameter.....	64
4.2.1.	Prosedur Percobaan	64
4.2.2.	Teknik Optimasi PNN Berorientasi Nilai Smoothing Parameter dengan Parameter GA Menurut Jurnal	66
4.2.3.	Teknik Optimasi PNN Berorientasi Nilai Smoothing Parameter dengan Parameter GA Modifikasi Jurnal	75
4.2.4.	Perbandingan Teknik Optimasi PNN Berorientasi Nilai Smoothing Parameter dengan Parameter GA Menurut Jurnal dan Modifikasi Jurnal	84
4.3.	Teknik Optimasi PNN Berorientasi Struktur Neuron	85
4.3.1.	Prosedur Percobaan	85
4.3.2.	Hasil dan Analisis Percobaan	88
4.3.3.	Analisis Teknik Optimasi PNN Berorientasi Struktur Neuron	95
4.4.	Kesimpulan.....	96

BAB 5 PERBANDINGAN HASIL PERCOBAAN.....	98
5.1. Perbandingan PNN, OPNN_SP_Jurnal, OPNN_SP, dan OPNN_SN.....	98
5.1.1. Perbandingan Hasil Percobaan Data IR-TNI.....	98
5.1.2. Perbandingan Hasil Percobaan Data VL-NI.....	107
5.1.3. Analisis Perbandingan	115
5.2. Kesimpulan.....	118
BAB 6 KESIMPULAN DAN SARAN	119
6.1. Kesimpulan.....	119
6.2. Saran.....	120
DAFTAR PUSTAKA	121
LAMPIRAN.....	123



DAFTAR GAMBAR

Gambar 2.1. Foto berukuran 30x30 <i>pixel</i> yang dituliskan ke dalam bentuk matriks	7
Gambar 2.2. Langkah pengolahan data tanpa NI.....	16
Gambar 2.3. Langkah pengolahan data dengan NI.....	17
Gambar 2.4. Struktur PNN.....	21
Gambar 2.5. Ilustrasi pencarian vektor v_2	24
Gambar 2.6. Ilustrasi pencarian vektor v_3	25
Gambar 2.7. <i>Orthogonal Algorithms</i>	29
Gambar 2.8. Metode pencarian struktur neuron optimal dengan OA.....	30
Gambar 2.9. <i>Flowchart</i> algoritma genetika	32
Gambar 2.10. Ilustrasi <i>crossover</i>	34
Gambar 2.11. Metode pencarian nilai <i>smoothing parameter</i> dengan GA.....	37
Gambar 2.12. Skema Percobaan 1	39
Gambar 2.13. Skema Percobaan 2	40
Gambar 2.14. Skema Percobaan 3	41
Gambar 2.15. Perbedaan letak <i>loop</i> GA dan OA pada Percobaan 2 dan 3.....	42
Gambar 3.1. Prosedur Percobaan 1	45
Gambar 3.2. Hasil RR perulangan Percobaan 1 data IR-TNI [20,30,50]	47
Gambar 3.3. Hasil RR perulangan Percobaan 1 data IR-TNI [30,20,50]	48
Gambar 3.4. Hasil RR perulangan Percobaan 1 data IR-TNI [40,10,50]	48
Gambar 3.5. Hasil RR perulangan Percobaan 1 data IR-NI [20,30,50].....	50
Gambar 3.6. Hasil RR perulangan Percobaan 1 data IR-NI [30,20,50].....	51
Gambar 3.7. Hasil RR perulangan Percobaan 1 data IR-NI [40,10,50].....	51
Gambar 3.8. Hasil RR perulangan Percobaan 1 data VL-TNI [20,30,50].....	55
Gambar 3.9. Hasil RR perulangan Percobaan 1 data VL-TNI [30,20,50].....	55
Gambar 3.10. Hasil RR Percobaan 1 data VL-TNI [40,10,50].....	56
Gambar 3.11. Hasil RR Percobaan 1 data VL-NI [20,30,50]	58
Gambar 3.12. Hasil RR Percobaan 1 data VL-NI [30,20,50]	59
Gambar 3.13. Hasil RR Percobaan 1 data VL-NI [40,10,50]	59
Gambar 4.1. Prosedur Percobaan 2	66

Gambar 4.2. Hasil RR Percobaan 2A data IR-TNI [20,30,50]	67
Gambar 4.3. Hasil RR Percobaan 2A data IR-TNI [30,20,50]	67
Gambar 4.4. Hasil RR Percobaan 2A data IR-TNI [40,10,50]	68
Gambar 4.5. Hasil RR Percobaan 2A data VL-NI [20,30,50]	71
Gambar 4.6. Hasil RR Percobaan 2A data VL-NI [30,20,50]	72
Gambar 4.7. Hasil RR Percobaan 2A data VL-NI [40,10,50]	72
Gambar 4.8. Hasil RR Percobaan 2B data IR-TNI [20,30,50]	75
Gambar 4.9. Hasil RR Percobaan 2B data IR-TNI [30,20,50]	76
Gambar 4.10. Hasil RR Percobaan 2B data IR-TNI [40,10,50]	76
Gambar 4.11. Hasil RR Percobaan 2B data VL-NI [20,30,50].....	80
Gambar 4.12. Hasil RR Percobaan 2B data VL-NI [30,20,50].....	80
Gambar 4.13. Hasil RR Percobaan 2B data VL-NI [40,10,50].....	81
Gambar 4.14. Prosedur Percobaan 3	88
Gambar 4.15. Hasil RR Percobaan 3 data IR-TNI [20,30,50]	89
Gambar 4.16. Hasil RR Percobaan 3 data IR-TNI [30,20,50]	89
Gambar 4.17. Hasil RR Percobaan 3 data IR-TNI [40,10,50]	90
Gambar 4.18. Hasil RR Percobaan 3 data VL-NI [20,30,50]	92
Gambar 4.19. Hasil RR Percobaan 3 data VL-NI [30,20,50]	93
Gambar 4.20. Hasil RR Percobaan 3 data VL-NI [40,10,50]	93
Gambar 5.1. Waktu <i>training</i> rata-rata PNN dan OPNN data IR-TNI.....	99
Gambar 5.2. Waktu <i>training</i> rata-rata PNN, OPNN, dan BP data IR-TNI.....	100
Gambar 5.3. Waktu <i>testing</i> rata-rata PNN dan OPNN data IR-TNI.....	101
Gambar 5.4. Waktu <i>testing</i> rata-rata PNN, OPNN, dan BP data IR-TNI.....	101
Gambar 5.5. Banyak neuron aktif rata-rata data IR-TNI	102
Gambar 5.6. <i>RR_tr</i> rata-rata PNN dan OPNN data IR-TNI.....	103
Gambar 5.7. <i>RR_trs</i> rata-rata PNN dan OPNN data IR-TNI.....	104
Gambar 5.8. <i>RR_ts</i> rata-rata PNN dan OPNN data IR-TNI.....	105
Gambar 5.9. <i>RR_ts</i> rata-rata PNN, OPNN, dan BP data IR-TNI.....	106
Gambar 5.10. Waktu <i>training</i> rata-rata PNN dan OPNN data VL-NI.....	107
Gambar 5.11. Waktu <i>training</i> rata-rata PNN, OPNN, dan BP data VL-NI.....	108
Gambar 5.12. Waktu <i>testing</i> rata-rata PNN dan OPNN data VL-NI.....	109
Gambar 5.13. Waktu <i>testing</i> rata-rata PNN, OPNN, dan BP data VL-NI.....	110

Gambar 5.14. Banyak neuron aktif rata-rata data VL-NI	110
Gambar 5.15. <i>RR_tr</i> rata-rata PNN dan OPNN data VL-NI.....	111
Gambar 5.16. <i>RR_trs</i> rata-rata PNN dan OPNN data VL-NI	112
Gambar 5.17. <i>RR_ts</i> rata-rata PNN dan OPNN data VL-NI.....	113
Gambar 5.18. <i>RR_ts</i> rata-rata PNN, OPNN, dan BP data VL-NI.....	114



DAFTAR TABEL

Tabel 2.1. Data <i>infrared</i> dan <i>visible light</i> penelitian.....	14
Tabel 2.2. Data <i>infrared</i> dan <i>visible light</i> penelitian setelah normalisasi iluminasi	17
Tabel 2.3. Perbandingan parameter GA menurut jurnal dan modifikasi jurnal ...	40
Tabel 3.1. Banyak neuron struktur utuh data IR	46
Tabel 3.2. Banyak neuron struktur utuh data VL.....	46
Tabel 3.3. Hasil Percobaan 1 data IR-TNI.....	49
Tabel 3.4. Nilai σ Percobaan 1 data IR-TNI	49
Tabel 3.5. Hasil Percobaan 1 data IR-NI	52
Tabel 3.6. Nilai σ Percobaan 1 data IR-NI.....	52
Tabel 3.7. Perbandingan hasil Percobaan 1 data IR-TNI dan IR-NI	53
Tabel 3.8. Hasil Percobaan 1 data VL-TNI.....	57
Tabel 3.9. Nilai σ dan RR Percobaan 1 data VL-TNI [20,30,50]	57
Tabel 3.10. Nilai σ Percobaan 1 data VL-TNI.....	58
Tabel 3.11. Hasil Percobaan 1 data VL-NI.....	60
Tabel 3.12. Nilai σ Percobaan 1 data VL-NI	60
Tabel 3.13. Perbandingan hasil Percobaan 1 data VL-TNI dan VL-NI.....	61
Tabel 4.1. Hasil Percobaan 2A data IR-TNI	69
Tabel 4.2. Nilai <i>smoothing parameter</i> dan struktur neuron optimal Percobaan 2A data IR-TNI	69
Tabel 4.3. Urutan dan neuron aktif Percobaan 2A data IR-TNI [30,20,50], $\sigma =$ 57.118.....	70
Tabel 4.4. Urutan dan neuron aktif Percobaan 2A data IR-TNI [30,20,50], $\sigma =$ 10.675.....	70
Tabel 4.5. Hasil Percobaan 2A data VL-NI	73
Tabel 4.6. Nilai <i>smoothing parameter</i> dan struktur neuron optimal Percobaan 2A data VL-NI	73
Tabel 4.7. Urutan dan neuron aktif Percobaan 2A data VL-NI [40,10,50], $\sigma =$ 22.882.....	74

Tabel 4.8. Urutan dan neuron aktif Percobaan 2A data VL-NI [40,10,50], $\sigma =$ 12.358.....	74
Tabel 4.9. Hasil Percobaan 2B data IR-TNI.....	77
Tabel 4.10. Nilai <i>smoothing parameter</i> dan struktur neuron optimal Percobaan 2B data IR-TNI.....	77
Tabel 4.11. Urutan dan neuron aktif Percobaan 2B data IR-TNI [40,10,50], $\sigma =$ 62.115.....	78
Tabel 4.12. Urutan dan neuron aktif Percobaan 2B data IR-TNI [40,10,50], $\sigma =$ 3.194.....	79
Tabel 4.13. Hasil Percobaan 2B data VL-NI.....	81
Tabel 4.14. Nilai <i>smoothing parameter</i> dan struktur neuron optimal Percobaan 2B data VL-NI.....	82
Tabel 4.15. Urutan dan neuron aktif Percobaan 2B data VL-NI [30,20,50], $\sigma =$ 75.500.....	83
Tabel 4.16. Urutan dan neuron aktif Percobaan 2B data VL-NI [30,20,50], $\sigma =$ 26.000.....	83
Tabel 4.17. Perbandingan waktu <i>training</i> , konsistensi, dan konvergensi Percobaan 2A dan 2B.....	85
Tabel 4.18. Hasil Percobaan 3 data IR-TNI.....	90
Tabel 4.19. Nilai <i>smoothing parameter</i> dan struktur neuron optimal Percobaan 3 data IR-TNI.....	91
Tabel 4.20. Hasil Percobaan 3 data VL-NI.....	94
Tabel 4.21. Nilai <i>smoothing parameter</i> dan struktur neuron optimal Percobaan 3 data VL-NI.....	94
Tabel 5.1. Perbandingan kemampuan PNN, OPNN, dan BP data IR-TNI.....	106
Tabel 5.2. Perbandingan kemampuan PNN, OPNN, dan BP data VL-NI.....	114

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Jaringan saraf tiruan (JST) meniru sistem kerja saraf manusia, di mana impuls atau rangsangan mengalir melalui saraf menuju otak agar informasi mengenai rangsangan tersebut dapat diambil dan disimpan sehingga apabila suatu ketika ada rangsangan yang sama, otak dapat mengenal rangsangan tersebut. JST dibuat agar dapat mengenal pola-pola melalui pembelajaran yang telah dilakukan sebelumnya. *Probabilistic Neural Network* (PNN) adalah sebuah algoritma yang bekerja berdasarkan teori Klasifikasi Bayesian dan Estimator *Probability Density Function* (PDF) Parzen. Algoritma ini menghitung tingkat kemiripan (*maximum likelihood*) berdasarkan nilai estimasi PDF antara pola *testing* dengan pola yang telah diketahui sebelumnya. Nilai PDF yang digunakan merupakan nilai estimasi karena sulit untuk menghitung nilai PDF yang sebenarnya. Pola akan terklasifikasi ke dalam kelas yang memiliki tingkat kemiripan paling tinggi.

Dalam algoritma PNN, terdapat sebuah konstanta yang disebut *smoothing parameter* (σ – sigma) yang menentukan kedekatan nilai estimasi PDF dengan nilai PDF sebenarnya. Nilai *smoothing parameter* ini dapat bervariasi untuk setiap permasalahan dan tidak ada suatu aturan baku yang dapat mencari nilai *smoothing parameter* optimal untuk setiap permasalahan. Salah satu teknik optimasi yang banyak digunakan untuk mencari nilai *smoothing parameter* optimal adalah algoritma genetika atau *genetic algorithms* (GA). Penentuan metode dan parameter GA yang digunakan sangat menentukan keberhasilan dari pencarian nilai *smoothing parameter*.

Dalam algoritma PNN, optimasi juga dapat dilakukan pada struktur neuron yang digunakan, selain optimasi pada nilai *smoothing parameter*. Struktur neuron utuh menandakan seluruh pola dalam *database* digunakan sebagai neuron PNN. Struktur ini dapat direduksi dengan memilih neuron-neuron penting yang dapat mewakili pola dari kelasnya. Dengan hanya menggunakan neuron sesedikit mungkin untuk mewakili suatu kelas, diharapkan waktu komputasi untuk mengenali suatu pola dapat berkurang dan hasil rekognisi yang diberikan tidak

berbeda jauh dengan struktur neuron utuh. Optimasi struktur neuron pada PNN dilakukan dengan bantuan algoritma ortogonal atau *orthogonal algorithms* (OA). OA berfungsi untuk mengurutkan neuron menurut tingkat kepentingannya dalam suatu kelas. Neuron dengan tingkat kepentingan yang tinggi memiliki prioritas untuk digunakan terlebih dahulu untuk mewakili kelasnya dalam struktur neuron PNN.

Nilai *smoothing parameter* dan struktur neuron memiliki hubungan yang saling bergantung. Nilai *smoothing parameter* berpengaruh pada OA sehingga nilai yang berbeda dapat memberikan urutan neuron yang berbeda, sedangkan struktur neuron juga mempengaruhi pencarian nilai *smoothing parameter* optimal. Hubungan ini memunculkan permasalahan parameter mana yang harus didahulukan dalam proses optimasi kedua parameter ini. K. Z. Mao, K. C. Tan, dan W. Ser dalam jurnal yang berjudul "*Probabilistic Neural-Network Structure Determination for Pattern Classification*" mengajukan teknik optimasi yang secara garis besar dapat dikatakan menentukan nilai *smoothing parameter* terlebih dahulu, baru kemudian mencari struktur yang optimal terhadapnya. Bagaimanapun teknik dan prosedur yang digunakan, hubungan ini sulit dipisahkan agar dapat dioptimasi masing-masing.

Fokus utama dalam skripsi ini adalah membandingkan 2 teknik optimasi nilai *smoothing parameter* dan struktur neuron yang berbeda. Teknik pertama diadopsi dari jurnal "*Probabilistic Neural-Network Structure Determination for Pattern Classification*" dengan menentukan nilai *smoothing parameter* terlebih dahulu, sedangkan teknik kedua dilakukan dengan membentuk struktur neuron terlebih dahulu sebelum dicari nilai *smoothing parameter* yang optimal terhadapnya. Parameter GA yang digunakan merupakan parameter GA yang digunakan dalam jurnal acuan dan ada pula parameter GA hasil modifikasi. Sebagai pembandingan, dalam skripsi ini juga dibahas mengenai optimasi nilai *smoothing parameter* untuk struktur neuron utuh. Data yang digunakan berupa data wajah frontal yang diambil dengan mode *infrared* dan *visible light*, di mana kedua jenis data ini juga akan dilengkapi dengan metode normalisasi iluminasi (NI) untuk dilakukan perbandingan.

1.2. Tujuan Penelitian

Tujuan dari penelitian dalam skripsi ini adalah sebagai berikut :

1. Mendapatkan nilai *smoothing parameter* optimal untuk struktur neuron PNN tertentu dengan menggunakan algoritma genetika (GA);
2. Mendapatkan struktur neuron optimal untuk nilai *smoothing parameter* tertentu dengan bantuan algoritma orthogonal (OA);
3. Mengetahui pengaruh proses normalisasi iluminasi terhadap hasil percobaan dengan data *infrared* dan *visible light*;
4. Membandingkan hasil yang didapatkan dari PNN teroptimasi dengan hasil yang didapatkan dari PNN dengan struktur neuron utuh;
5. Melakukan perbandingan antara 2 teknik optimasi PNN dan juga 2 parameter GA yang digunakan pada percobaan dalam skripsi ini.

1.3. Pembatasan Masalah

Penelitian dan pembahasan pada skripsi ini dilakukan dengan batasan-batasan sebagai berikut :

1. Pemrograman dan percobaan dalam penelitian ini dilakukan dengan bantuan Matlab R2008a;
2. Data-data yang digunakan pada penelitian ini berupa data wajah yang diambil dengan kamera *infrared* dan kamera *visible light* secara frontal;
3. Data penelitian yang digunakan berasal dari data penelitian orang lain, tidak diambil sendiri oleh penulis;
4. Jenis kromosom yang digunakan pada algoritma genetika berupa kromosom dalam bentuk desimal yang mewakili nilai *smoothing parameter*;
5. Normalisasi iluminasi menggunakan parameter yang dianggap paling optimal dan digunakan sekedar untuk membandingkan pengaruhnya terhadap hasil rekognisi, struktur neuron, dan nilai *smoothing parameter* yang diperoleh;
6. Segala jenis parameter yang digunakan hanya sebagai bahan perbandingan dianggap merupakan parameter optimal.

1.4. Metode Penulisan

Metode penulisan yang digunakan dalam skripsi ini antara lain :

1. Tinjauan pustaka melalui studi literatur dari buku-buku, jurnal, ataupun bentuk literatur lain yang berkaitan dengan bahasan pada skripsi ini;
2. Konsultasi dan diskusi dengan pembimbing skripsi mengenai bahasan skripsi;
3. Penggunaan perangkat keras dan lunak dalam membantu percobaan yang dilakukan.

Seluruh percobaan dalam penelitian ini dilakukan dengan perangkat keras yang sama. Spesifikasi dari perangkat keras yang digunakan dalam penelitian ini adalah :

Processor : Intel (R) Pentium (R) Dual CPU E2160 @ 1.80 GHz 1.80 GHz

Memory (RAM) : 2.5 GB

System type : 32-bit *Operating System*

Operating System : *Windows 7 Ultimate*

Perangkat lunak pendukung penelitian ini adalah Matlab R2008a. Percobaan dilakukan dengan membuat program-program percobaan dalam m-file Matlab. Semua pengolahan data dan perhitungan dilakukan dalam Matlab, tanpa menggunakan bantuan perangkat lunak lain.

1.5. Sistematika Penulisan

Skripsi ini terdiri dari 6 bab, di mana masing-masing bab akan membahas hal-hal sebagai berikut :

- Bab 1 Pendahuluan
Berisikan latar belakang, tujuan penelitian, pembatasan masalah, metode penelitian, dan sistematika penulisan.
- Bab 2 Probabilistic Neural Network, Teknik Optimasi, dan Skema Percobaan
Berisikan cara memperoleh dan mengolah data penelitian, teori-teori mengenai JST algoritma *Probabilistic Neural Network* (PNN), metode

pencarian nilai *smoothing parameter* optimal dengan *Genetic Algorithms*, metode pencarian struktur neuron optimal dengan *Orthogonal Algorithms*, dan skema percobaan yang dilakukan pada penelitian ini

- **Bab 3 Nilai Smoothing Parameter Optimal Dengan Genetic Algorithms**
Berisikan tujuan, prosedur, hasil, dan analisis terkait percobaan mengenai PNN dengan struktur neuron utuh dan nilai *smoothing parameter* teroptimasi dengan data *infrared* dan *visible light*, baik tanpa dan dengan normalisasi iluminasi.
- **Bab 4 Teknik Optimasi Probabilistic Neural Network**
Berisikan tujuan, prosedur, hasil, dan analisis terkait percobaan mengenai PNN dengan struktur neuron dan nilai *smoothing parameter* teroptimasi menggunakan teknik optimasi berorientasi nilai *smoothing parameter* dan struktur neuron.
- **Bab 5 Perbandingan Hasil Percobaan**
Berisikan perbandingan hasil seluruh percobaan yang dilakukan, beserta perbandingannya terhadap *JST backpropagation* yang dijadikan sebagai referensi, berikut analisis terhadap hasil-hasil tersebut.
- **Bab 6 Kesimpulan dan Saran**
Berisikan kesimpulan dan saran yang dapat ditarik dari hasil percobaan dan analisis yang dilakukan pada skripsi.

BAB 2

PROBABILISTIC NEURAL NETWORK, TEKNIK OPTIMASI, DAN SKEMA PERCOBAAN

2.1. Data Penelitian

2.1.1. Principal Component Analysis (PCA)

Principal Component Analysis (PCA) adalah suatu metode statistik yang dapat mengidentifikasi pola suatu data berdimensi tinggi/banyak, dan menyatakannya dengan memperhatikan kesamaan dan ketidaksamaannya. Pola-pola pada data yang berdimensi tinggi kadang sulit ditemukan dan grafisnya sulit digambarkan, oleh sebab itu PCA digunakan sebagai alat yang sangat bermanfaat untuk menganalisis data (Imantaka, 2010).

Konsep utama dari PCA adalah mereduksi dimensi dari suatu set data yang terdiri dari banyak variabel, di mana kemungkinan-kemungkinan variasi sebisa mungkin dipertahankan. Hal ini dicapai dengan mentransformasi set data tersebut menjadi suatu set variabel yang satu sama lain tidak berhubungan, yang disebut *Principal Components* (PCs). PC ini merupakan kumpulan beberapa variasi yang diurutkan dan paling dipilih dari semua variabel asli (Jolliffe, 2002, p. 1).

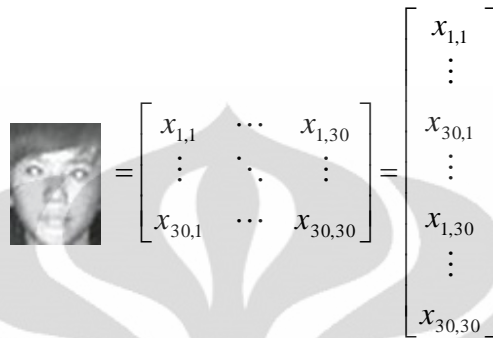
Secara matematis, PCA melakukan transformasi linier dari data X yang berdimensi D ke suatu data baru Y yang berdimensi d dan tidak berkorelasi, di mana $d \leq D$ (Buchala). Keuntungan PCA adalah ketika pola pada data telah ditemukan dan data telah dimampatkan dengan memperkecil jumlah dimensinya, informasi yang terkandung di dalamnya tidak akan mengalami perubahan yang signifikan. Metode ini sering digunakan dalam *image compression* (Imantaka, 2010).

Berikut ini adalah penjelasan mengenai algoritma PCA pada sistem pengenalan wajah. PCA hanya dilakukan pada tahap ekstraksi fitur, yaitu untuk memampatkan dimensi dari input gambar yang jumlahnya besar agar menjadi lebih kecil sehingga mempercepat komputasi (Imantaka, 2010).

1. Menyusun matriks data

Dalam sistem pengenalan wajah, data yang diperoleh merupakan intensitas *gray scale* dari suatu gambar foto yang berukuran (misalkan) 30×30 *pixel*.

Intensitas-intensitas ini dijadikan suatu matriks yang berukuran 30x30. Untuk memudahkan komputasi, matriks 30x30 tersebut diubah menjadi sebuah vektor kolom dengan 900 baris.



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

Gambar 2.1. Foto berukuran 30x30 *pixel* yang dituliskan ke dalam bentuk matriks

Dengan demikian, sebuah data gambar memiliki 900 elemen atau secara matematis kita sebut dimensi. Jika seluruh *database* berisi t gambar dituliskan dalam suatu matriks, matriks tersebut akan berukuran 900 x t , seperti berikut:

$$\text{database} = \begin{bmatrix} x_{1,1} & x_{2,1} & \cdots & x_{t,1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1,900} & x_{2,900} & \cdots & x_{t,900} \end{bmatrix} \quad (2.1)$$

2. Normalisasi nilai *database* dengan mencari nilai *z-score*

Normalisasi *z-score* bertujuan untuk meletakkan nilai rata-rata tiap dimensi di titik nol dan mengatur agar tiap dimensi memiliki nilai variansi yang seragam. Konversi data ini dinyatakan sebagai berikut:

$$z = \frac{X - \mu}{\sigma} \quad (2.2)$$

di mana σ adalah standar deviasi :

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (X_i - \mu)^2}{n-1}} \quad (2.3)$$

Database baru yang memiliki nilai rata-rata yang sama dengan nol dibentuk dari nilai z :

$$database = \begin{bmatrix} x_{1,1} & \cdots & x_{1,900} \\ x_{2,1} & \cdots & x_{2,900} \\ \vdots & \ddots & \vdots \\ x_{t,1} & \cdots & x_{t,900} \end{bmatrix} \Rightarrow Z = \begin{bmatrix} z_{1,1} & \cdots & z_{1,900} \\ z_{2,1} & \cdots & z_{2,900} \\ \vdots & \ddots & \vdots \\ z_{t,1} & \cdots & z_{t,900} \end{bmatrix} \quad (2.4)$$

3. Menghitung matriks kovarians

Matriks kovarians dihitung dengan persamaan :

$$C_Z = \frac{Z^T Z}{(n-1)} \quad (2.5)$$

4. Menghitung vektor eigen dan nilai eigen dari matriks kovarians

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

$$A\mathbf{x} = \lambda\mathbf{x} \text{ atau ekuivalen dengan } (\lambda I - A)\mathbf{x} = \mathbf{0} \quad (2.6)$$

untuk suatu skalar λ . Skalar λ disebut sebagai nilai eigen dari A , dan \mathbf{x} disebut suatu vektor eigen dari A yang berpadanan dengan λ (Anton, 2000). Karena matriks kovarians berbentuk persegi, maka kita dapat mencari semua vektor-vektor eigen dan nilai eigen yang berpasangan dengan vektornya. Nilai dan vektor eigen ini merupakan informasi yang berguna untuk data.

5. Membentuk matriks transformasi

Vektor eigen akan diurutkan dari yang bernilai eigen terbesar hingga terkecil. Matriks transformasi akan dibentuk dari kumpulan vektor eigen yang memiliki nilai eigen terbesar. Nilai eigen sebanding dengan tingkat kepentingan basis tersebut sehingga basis dengan nilai eigen yang kecil dapat dihilangkan karena tidak akan berpengaruh banyak pada informasi penting dalam data. Pada akhirnya matriks data akhir didapat dengan mengalikan matriks data hasil normalisasi z-score dengan matriks transformasi.

$$X = M \times Z \quad (2.7)$$

Di mana M adalah matriks transformasi dan Z adalah matriks data hasil normalisasi z-score. X adalah set data terakhir atau database terbaru yang telah berhasil dimampatkan dimensinya dengan PCA.

2.1.2. Algoritma Kompensasi dan Normalisasi Iluminasi

Setiap Citra 2 dimensi dapat dinyatakan oleh matriks $f(x,y)$, di mana x dan y merupakan indeks *pixel-pixel* secara vertikal dan horizontal, merupakan hasil perkalian antara reflektansi citra $r(x,y)$ dengan iluminasi citra $e(x,y)$ seperti yang terlihat pada persamaan berikut (Chen & et.al, 2006):

$$f(x, y) = r(x, y).e(x, y) \quad (2.8)$$

Pada citra wajah, reflektansi merepresentasikan pola wajah dan terletak pada frekuensi tinggi dalam domain frekuensi. Reflektansi suatu citra hampir tidak pernah berubah sepanjang bentuk fisik objek dalam citra tidak berubah. Sedangkan, iluminasi merepresentasikan tingkat pencahayaan pada setiap *pixel* citra. Tingkat iluminasi pada citra mengalami perubahan yang perlahan sepanjang bidang citra, oleh karena itu variasi iluminasi terletak pada frekuensi rendah dalam domain frekuensi citra. Nilai iluminasi suatu citra pada setiap *pixel* bervariasi bergantung pada kondisi pencahayaan seperti tingkat keterangan, sudut datang cahaya ataupun bayangan benda yang menghalanginya. Untuk

mendapatkan citra yang baik untuk tujuan sistem pengenalan wajah, kita perlu mendapatkan nilai murni reflektansi citra dengan iluminasi yang seragam pada setiap *pixel*. Proses pembuangan variasi iluminasi pada citra sehingga iluminasinya seragam disebut dengan Kompensasi Iluminasi (KI). Kompensasi Iluminasi dalam domain spasial dapat dilakukan dengan cara menyamakan variabel $e(x,y)$ untuk setiap *pixel*. Hal ini sulit dilakukan karena kita harus mengetahui nilai $e(x,y)$ (setiap *pixel*) untuk membuatnya seragam.

2.1.2.1. Transformasi Logaritma Citra

Dalam domain logaritma, persamaan (2.8) bertransformasi menjadi berikut:

$$\log f(x, y) = \log r(x, y) + \log e(x, y) \quad (2.9)$$

Jika dalam domain spasial, Kompensasi Iluminasi dilakukan dengan operasi pembagian , maka dalam domain logaritma, Kompensasi dilakukan dengan operasi pengurangan $e(x,y)$ dengan kompensator $\in(x, y)$. Berikut adalah proses kompensasi dalam persamaan matematika :

$$\begin{aligned} \log f'(x, y) &= \log r(x, y) + \log e'(x, y) \\ &= \log r(x, y) + \log e(x, y) - \in(x, y) \\ &= \log f(x, y) - \in(x, y) \end{aligned} \quad (2.10)$$

di mana

$\log f'(x, y)$	= nilai total <i>pixel</i> citra wajah dalam ruang logaritma
$\log r(x, y)$	= nilai reflektansi <i>pixel</i> citra wajah dalam ruang logaritma
$\log e(x, y)$	= nilai iluminasi <i>pixel</i> citra wajah yang bervariasi dalam ruang logaritma
$\log e'(x, y)$	= nilai iluminasi <i>pixel</i> citra wajah yang seragam dalam ruang logaritma
$\in(x, y)$	= kompensator

Karena citra akan diproses lebih lanjut dalam ruang logaritma, maka citra harus mengalami sedikit penyesuaian saat dilakukan transformasi logaritma seperti pada persamaan (2.11), sehingga citra dalam ruang logaritma tetap berada pada *range* nilai 0-255 yaitu nilai kuantisasi citra digital 8 bit.

$$F(x, y) = \frac{255}{\log(256)} \log(f(x, y) + 1) \quad (2.11)$$

2.1.2.2. Discrete Cosine Transform (DCT)

Discrete Cosine Transform (DCT) merupakan transformasi dari domain spasial ke domain frekuensi (Chen & et.al, 2006). DCT memiliki 4 tipe dan yang digunakan dalam kompensasi iluminasi berikut adalah DCT tipe 2. Transformasi DCT 2D dari gambar $f(x,y)$ adalah :

$$C(u, v) = \alpha(u)\alpha(v) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \times \cos\left[\frac{\pi(2x+1)u}{2M}\right] \cos\left[\frac{\pi(2y+1)v}{2N}\right] \quad (2.12)$$

dan inversnya :

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \alpha(u)\alpha(v)C(u, v) \times \cos\left[\frac{\pi(2x+1)u}{2M}\right] \cos\left[\frac{\pi(2y+1)v}{2N}\right] \quad (2.13)$$

di mana

$$\alpha(u) = \begin{cases} \frac{1}{\sqrt{M}}, & u = 0 \\ \sqrt{\frac{2}{M}}, & u = 1, 2, \dots, M - 1 \end{cases} \quad (2.14)$$

$$\alpha(v) = \begin{cases} \frac{1}{\sqrt{N}}, & v = 0 \\ \sqrt{\frac{2}{N}}, & v = 1, 2, \dots, N - 1 \end{cases} \quad (2.15)$$

2.1.2.3. Kompensasi Iluminasi dan Normalisasi Iluminasi Citra dalam Ruang DCT Logaritma

Tingkat iluminasi pada citra mengalami perubahan yang perlahan sepanjang bidang citra, oleh karena itu variasi iluminasi terletak pada frekuensi rendah dalam domain frekuensi citra (Chen & et.al, 2006). Kompensasi dalam domain DCT dapat dilakukan dengan menghilangkan koefisien frekuensi rendah dalam matriks $C(u,v)$ yaitu matriks koefisien DCT citra. Penghilangan sebanyak n frekuensi rendah serupa dengan operasi pengurangan berikut:

$$\begin{aligned} f'(x,y) &= \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} E(u,v) - \sum_{i=1}^n E(u_i, v_i) \\ &= f(x,y) - \sum_{i=1}^n E(u_i, v_i) \end{aligned} \quad (2.16)$$

di mana :

$$E(u,v) = \alpha(u)\alpha(v)C(u,v) \cos\left[\frac{\pi(2x+1)u}{2M}\right] \cos\left[\frac{\pi(2y+1)v}{2N}\right] \quad (2.17)$$

Jika variasi iluminasi berada pada komponen frekuensi rendah dan $F'(x,y)$ disetarakan dengan nilai total *pixel* citra wajah dalam ruang logaritma seperti pada persamaan (2.10), maka $\sum_{i=1}^n E(u_i, v_i)$ dapat diserupakan sebagai kompensator,

sehingga penghilangan koefisien DCT frekuensi rendah akan sama dengan proses kompensasi variasi iluminasi citra. Penghapusan frekuensi rendah pada gambar dapat dilakukan dengan mengganti koefisien DCT frekuensi rendah menjadi nol. Banyaknya frekuensi rendah yang akan dihilangkan disebut dengan frekuensi *cutoff*.

Koefisien pertama pada matriks DCT $C(0,0)$ disebut komponen DC yang menyatakan tingkat pencahayaan pada seluruh permukaan gambar. Komponen DC ditentukan oleh rumus pada persamaan (2.18).

$$C(0,0) = \log \mu \cdot \sqrt{MN} \quad (2.18)$$

di mana :

M dan N = dimensi citra.

Setiap citra berdimensi berapapun akan memiliki tingkat pencahayaan yang sama jika $\log \mu$ setiap citra bernilai sama. Proses penyetaraan nilai komponen DC untuk semua gambar disebut dengan proses Normalisasi Iluminasi (NI).

Pada penelitian ini, proses KI tidak digunakan karena diasumsikan iluminasi di seluruh permukaan wajah seragam sehingga nilai $e(x,y)$ untuk setiap *pixel* dianggap sudah seragam. Masalah yang tertinggal adalah bagaimana membuat seluruh citra yang digunakan dalam penelitian ini memiliki tingkat iluminasi yang sama atau menyamakan nilai $e(x,y)$ untuk seluruh citra. Untuk mengatasi masalah ini, proses NI digunakan untuk menyamakan tingkat iluminasi pada seluruh citra.





















2.1.3. Pengolahan Data Penelitian

Pada penelitian ini digunakan 2 jenis data wajah orang berbentuk foto yang diambil secara frontal. Faktor yang membedakan kedua jenis data ini adalah mode kamera yang digunakan pada saat pengambilan foto. Jenis data pertama diambil dengan mode *infrared*, sedangkan jenis data kedua diambil dengan mode *visible light*. Data wajah *infrared* merupakan data yang digunakan oleh Stephen Roy Imantaka, mahasiswa Teknik Elektro angkatan 2006, dalam penelitian skripsinya yang berjudul “Sistem Pengenal Wajah Berbasis Ensemble Neural Network untuk Citra Infra Merah”, sedangkan data wajah *visible light* menggunakan data yang diperoleh dari pembimbing skripsi ini, Prof. Dr.Eng. Drs. Benyamin Kusumoputro M.Eng.

Data wajah *infrared* terdiri dari 200 foto yang diambil dari 10 orang dengan masing-masing sebanyak 20 foto wajah frontal. Wajah kesepuluh orang ini diambil dengan kamera mode *infrared* dalam bentuk video dengan beberapa ekspresi. Foto diambil dari rekaman video dengan cara *crop* menggunakan perbandingan ukuran 4:3 untuk mengambil bagian wajah saja. Setelah dilakukan *crop*, foto tersebut disesuaikan ukurannya menjadi 40 x 30 *pixel*. Data wajah

visible light terdiri dari 100 foto yang diambil dari 10 orang dengan masing-masing sebanyak 10 foto wajah frontal. Cara memperoleh foto *visible light* sama dengan langkah-langkah yang dilakukan dalam pengambilan foto *infrared*. Hanya saja, foto *visible light* dibentuk dengan ukuran 30 x 30 *pixel*. Kedua jenis foto disimpan dalam format JPEG (.jpg).

Tabel 2.1. Data *infrared* dan *visible light* penelitian

Kelas	Data <i>infrared</i>		Data <i>visible light</i>	
	Foto	Jumlah (foto)	Foto	Jumlah (foto)
1		20		10
2		20		10
3		20		10
4		20		10
5		20		10
6		20		10
7		20		10
8		20		10
9		20		10
10		20		10
Total		200		100

Data-data wajah berbentuk foto yang telah diambil tidak dapat digunakan langsung sebagai data percobaan, melainkan harus dilakukan beberapa proses pengolahan data. Langkah-langkah pengolahan diterapkan sama untuk kedua jenis data, yaitu :

a. Mengubah foto format warna RGB menjadi *grayscale*

Foto hasil *crop* dan *resize* dari rekaman video mempunyai format warna RGB, di mana terdiri dari 3 dimensi, yaitu dimensi *red* (R), dimensi *green* (G), dan dimensi *blue* (B). Format warna ini perlu diubah menjadi *grayscale* agar hanya tersisa 1 dimensi saja.

b. Menerjemahkan foto menjadi matriks

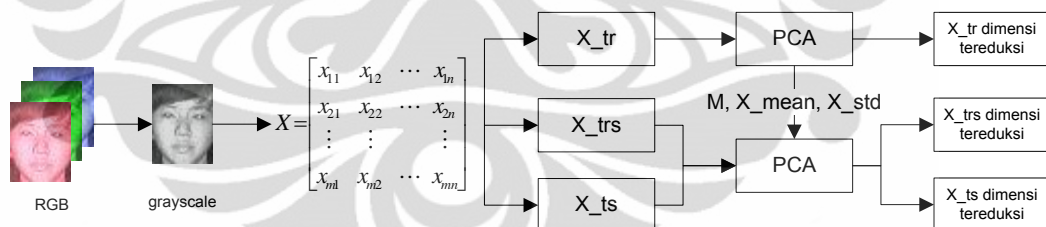
Foto *grayscale* diubah menjadi sebuah matriks berisi angka-angka yang merepresentasikan pola yang terdapat di foto tersebut. Pembentukan matriks dari foto dapat dilakukan dengan menggunakan perintah “*imread*” di program Matlab sehingga akan dihasilkan matriks berukuran 40 x 30 untuk foto *infrared* dan 30 x 30 untuk foto *visible light*. Kemudian, untuk menyederhanakan penyusunan data percobaan, matriks ini perlu disusun ulang tanpa mengubah nilai menjadi matriks baris atau matriks kolom. Dengan penyusunan ini, akan terbentuk matriks berukuran 1 x 1200 untuk data *infrared* yang menandakan data ini memiliki 1200 dimensi dan matriks berukuran 1 x 900 untuk data *visible light* yang menandakan data ini memiliki 900 dimensi.

c. Membagi data untuk implementasi dalam percobaan

Data-data yang telah siap digunakan dalam percobaan akan dibagi menjadi 3 bagian, yaitu data *training* (X_{tr}), data *testing* untuk *training* (X_{trs}), dan data *testing* (X_{ts}). X_{tr} merupakan kumpulan data yang akan digunakan sebagai neuron pada lapisan pola, X_{trs} merupakan kumpulan data yang digunakan sebagai data *testing* pada proses *training* PNN, sedangkan X_{ts} merupakan kumpulan data *testing* untuk validasi kemampuan PNN setelah struktur dan nilai *smoothing parameter* optimal didapatkan. Pembagian menjadi 3 bagian data ini dilakukan dengan proporsi tertentu, misalkan proporsi data yang digunakan $[X_{tr}, X_{trs}, X_{ts}] = [20,30,50]$.

d. Ekstraksi fitur

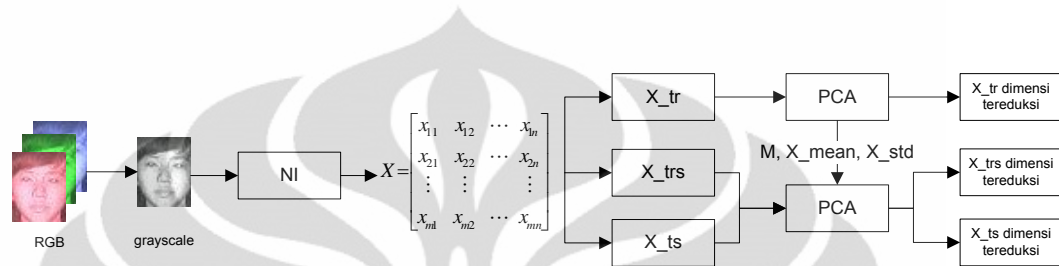
Tahap ekstraksi fitur merupakan proses untuk mereduksi dimensi input yang besar menjadi dimensi yang lebih kecil dengan menggunakan *Principal Component Analysis* (PCA). Data dengan dimensi sebanyak 900 ataupun 1200 terlalu besar untuk diolah karena dapat menyebabkan operasi matematis yang terlalu banyak dan waktu komputasi yang lama. Metode PCA membantu mereduksi dimensi data dengan mengambil dimensi-dimensi yang saling tegak lurus dan mengabaikan dimensi-dimensi lain yang tidak tegak lurus. Banyak dimensi hasil reduksi yang digunakan bergantung pada jenis data dan hasil pengamatan nilai eigen = 0 untuk mengambil dimensi-dimensi yang memiliki nilai eigen besar. Terdapat perbedaan perlakuan ekstraksi fitur antara X_{tr} dengan X_{trs} dan X_{ts} . Metode PCA diterapkan langsung pada X_{tr} sehingga dihasilkan matriks X_{tr} baru dengan dimensi yang telah ditentukan. Dari proses PCA X_{tr} , diambil matriks transformasi, *mean* X_{tr} , dan standar deviasi X_{tr} yang digunakan untuk mereduksi dimensi X_{trs} dan X_{ts} . Hal ini dilakukan dengan tujuan agar ketiga bagian data ini tetap berada dalam basis yang sama.



Gambar 2.2. Langkah pengolahan data tanpa NI

Selain data yang berasal langsung diolah dari foto, penelitian ini juga menggunakan data hasil proses Normalisasi Iluminasi (NI). Data NI didapatkan dengan menambahkan 1 tahapan sebelum foto diterjemahkan ke dalam bentuk matriks. NI dilakukan dengan mengubah komponen DC seluruh foto dengan nilai $\log \mu$ tertentu sehingga seluruh foto mempunyai tingkat iluminasi yang sama. Hasil rekognisi diharapkan menjadi lebih baik setelah menggunakan proses NI karena seluruh foto telah berada dalam tingkat iluminasi yang sama sehingga pengaruh dari faktor-faktor iluminasi dapat diperkecil. Setelah tahap NI,

pengolahan data dilakukan ke tahap penerjemahan foto menjadi matriks hingga ekstraksi fitur. Nilai $\log \mu$ yang digunakan pada penelitian ini sebesar 150 (Pramudia, 2011). Secara keseluruhan, terdapat 4 jenis data yang digunakan dalam penelitian ini, yaitu data *infrared* tanpa normalisasi iluminasi (IR-TNI), data *infrared* dengan normalisasi iluminasi (IR-NI), data *visible light* tanpa normalisasi iluminasi (VL-TNI), dan data *visible light* dengan normalisasi iluminasi (VL-NI).



Gambar 2.3. Langkah pengolahan data dengan NI

Tabel 2.2. Data *infrared* dan *visible light* penelitian setelah normalisasi iluminasi

Kelas	Data <i>infrared</i>		Data <i>visible light</i>	
	IR-TNI	IR-NI	VL-TNI	VL-NI
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

Ringkasan parameter data penelitian :

Data	: <i>Infrared</i> tanpa normalisasi (IR-TNI) <i>Infrared</i> dengan normalisasi (IR-NI) <i>Visible Light</i> tanpa normalisasi (VL-TNI) <i>Visible Light</i> dengan normalisasi (VL-NI)
Proporsi data [X_{tr}, X_{trs}, X_{ts}]	: [20,30,50]; [30,20,50]; [40,10,50]
Dimensi tereduksi setelah PCA	: 30 dimensi
Log μ	: 150

2.2. Jaringan Saraf Tiruan Algoritma Probabilistic Neural Network

2.2.1. Klasifikasi Bayesian dan Estimator PDF Parzen

Klasifikasi Bayesian adalah suatu sistem klasifikasi berdasarkan teori statistik dasar tentang probabilitas dan probabilitas bersyarat. Untuk mengklasifikasikan suatu pola yang belum dikenal, digunakan atribut-atribut pola tersebut untuk mengukur probabilitas pola tergolong dalam kelas tertentu. Jika diberikan C_i adalah daftar kelas-kelas sebagai target klasifikasi, di mana $i = 1, 2, 3, \dots, n$, maka probabilitas sebuah pola termasuk ke dalam sebuah kelas didefinisikan sebagai $P(C_i)$, di mana $0 \leq P(C_i) \leq 1$. (Herry, 2002)

Penggunaan probabilitas bersyarat memungkinkan untuk memasukkan pengetahuan tentang pola yang telah didapatkan sebelumnya untuk meningkatkan kemampuan prediksi keanggotaan pola terhadap kelas-kelas yang ada. Sebagai contoh, jika ingin memprediksi kemungkinan kemunculan kartu As yang dibagi dari sebuah kumpulan kartu setelah 10 kartu terbagi, akan lebih mudah apabila sudah diketahui kartu apa saja yang telah keluar dari 10 kartu yang dibagi sebelumnya. Jika dari 10 kartu tersebut telah terdapat 4 kartu As, maka dapat dipastikan bahwa kartu berikutnya yang akan dibagi bukan merupakan kartu As. Contoh sederhana ini menggambarkan bahwa dengan memasukkan pengetahuan awal ke dalam prediksi, akan memberikan pengaruh yang besar terhadap tingkat keberhasilan prediksi.

Jika terdapat 2 kejadian, X dan Y , maka probabilitas bersyarat $P(X/Y)$ didefinisikan sebagai probabilitas kejadian X terjadi dengan syarat kejadian Y telah terjadi. Berkaitan dengan permasalahan keanggotaan pola terhadap suatu

kelas, maka $P(C_i|X)$ dapat didefinisikan sebagai probabilitas pola X tergolong ke dalam kelas C_i . Aturan Bayes dapat digunakan untuk melakukan klasifikasi sebuah pola ke dalam kelas tertentu. Jika pola X adalah pola yang ingin diklasifikasikan, aturan Bayes akan mengklasifikasikan pola X ke dalam sebuah kelas dengan aturan sebagai berikut :

Pola X diklasifikasi ke dalam kelas C_i bila

$$P(C_i | X) > P(C_j | X) \quad (2.19)$$

untuk i dan $j = 1, 2, \dots, n$ dan $i \neq j$.

Aturan di atas menyatakan bahwa pola X akan diklasifikasi ke dalam kelas yang mempunyai probabilitas bersyarat tertinggi dibandingkan probabilitas pola X diklasifikasikan terhadap kelas-kelas yang lain.

Bagaimana menentukan nilai probabilitas bersyarat $P(C_i|X)$ untuk melakukan klasifikasi merupakan hal yang sulit. Pada umumnya, nilai probabilitas bersyarat tersebut tidak diketahui dan harus diestimasi. Oleh karenanya, tingkat keakuratan estimasi yang dilakukan akan menentukan performa dari sistem klasifikasi. Estimasi nilai probabilitas bersyarat dapat dilakukan dengan menggunakan sebuah model distribusi probabilitas dan mengasumsikan bahwa hal tersebut akan memiliki kecenderungan yang sama. Pada hukum Bayes berlaku :

$$P(C_i | X) = \frac{P(X | C_i)P(C_i)}{P(X)} \quad (2.20)$$

di mana

$$P(X) = \sum_{j=1}^n P(X | C_j)P(C_j) \quad (2.21)$$

Komponen $P(X)$ dapat diabaikan tanpa mempengaruhi proses pengambilan keputusan karena komponen ini *independent* terhadap setiap kelas (Sethi & Jain,

1991). Komponen $P(C_i)$ dapat diketahui berdasarkan data pelatihan yang dimiliki, sedangkan komponen $P(X/C_i)$ dapat dicari nilainya dengan menggunakan *probability density function* (PDF) dari pola X terhadap kelas C_i (Sethi & Jain, 1991; Masters, 1995). Bila komponen $P(C_i)$ diasumsikan sama untuk setiap kelas, maka pada akhirnya hanya perlu mencari nilai PDF pola X tertinggi dari seluruh kelas yang ada (Masters, 1995).

Permasalahan kembali muncul dalam menentukan nilai PDF pola X terhadap kelas C_i . Hal ini dapat dipecahkan dengan mengasumsikan bahwa densitas yang ada di setiap kelas mempunyai bentuk yang sama dengan distribusi normal. Distribusi normal digunakan karena sesuai dengan pola data yang ada pada umumnya. Dengan asumsi ini, nilai PDF dihitung dengan menggunakan fungsi estimasi. Salah satu jenis fungsi estimasi adalah fungsi Estimator PDF Parzen. Estimasi PDF Parzen adalah sebuah fungsi yang digunakan untuk melakukan estimasi terhadap nilai PDF. Estimator PDF Parzen merupakan jumlah *multivariate* distribusi Gaussian (normal) yang berpusat pada masing-masing vektor pelatihan yang bisa menghasilkan PDF yang halus dan kontinu (Zaknich, 1998). Perhitungan nilai PDF menggunakan Estimator PDF Parzen untuk n dimensi adalah sebagai berikut (Zaknich, 1998) :

$$p_i(x) = \frac{1}{(2\pi)^{d/2} \sigma^d N_i} \sum_{j=1}^{N_i} \exp \left[-\frac{(x - x_{ij})^T (x - x_{ij})}{2\sigma^2} \right] \quad (2.22)$$

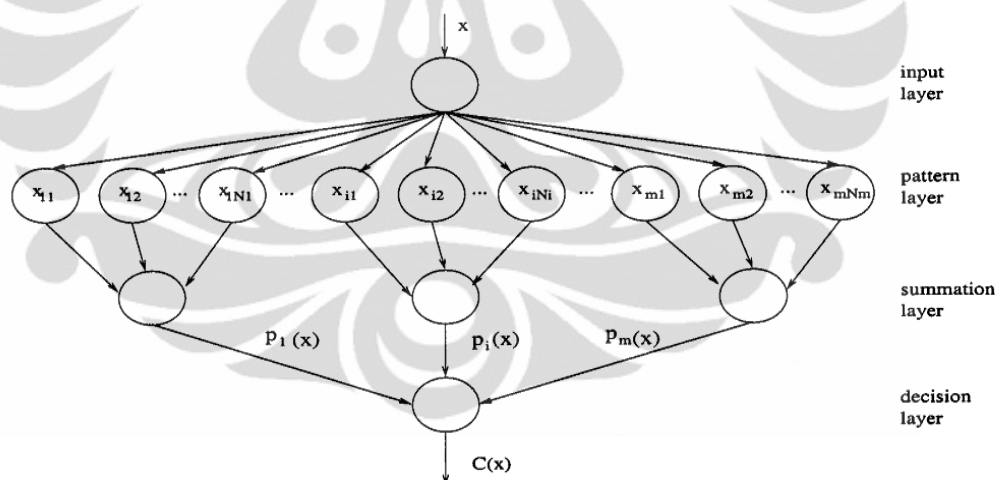
di mana :

- x = data *testing*/pola masukan
- i = indeks kelas
- j = indeks data *training*
- x_{ij} = data *training* ke- j dari kelas i
- $p_i(x)$ = nilai PDF dari kelas i untuk data *testing* x
- d = dimensi data *testing* x
- σ = *smoothing parameter*
- T = *transpose*
- N_i = banyak data *training* dalam kelas i

Dalam penggunaan Estimator PDF Parzen, nilai *smoothing parameter* (σ) akan menentukan seberapa dekat nilai PDF hasil estimasi terhadap nilai PDF sesungguhnya. Semakin sesuai pemilihan nilai *smoothing parameter* terhadap data yang ada, maka akan semakin baik nilai estimasi yang diperoleh.

2.2.2. Probabilistic Neural Network

Probabilistic Neural Network (PNN) adalah jenis jaringan saraf tiruan yang dikembangkan oleh Donald Specht pada tahun 1990 (Fausett, 1994; Masters, 1995; Mao & et.al., 2000). PNN dirancang menggunakan ide dari teori klasik probabilitas, yaitu klasifikasi Bayesian dan Estimator PDF Parzen (Fausett, 1994). PNN telah menjadi salah satu algoritma yang efektif untuk memecahkan banyak permasalahan klasifikasi karena proses pelatihan yang mudah dan adanya fondasi statistik dalam teori estimasi Bayesian (Mao & et.al., 2000).



Gambar 2.4. Struktur PNN

Sumber : (Mao & et.al., 2000)

Arsitektur PNN dapat dilihat pada Gambar 2.4., terdiri dari 4 lapisan (*layer*), yaitu lapisan masukan (*input layer*), lapisan pola (*pattern layer*), lapisan jumlah (*summation layer*), dan lapisan keputusan (*decision layer*). Pada lapisan masukan, tidak dilakukan operasi apapun dan hanya berperan untuk mendistribusikan masukan berupa data *testing* ke neuron-neuron dalam lapisan pola. Lapisan pola

terdiri dari neuron-neuron yang berisi data *training* dan dinotasikan sebagai x_{ij} (neuron/data *training* ke- j dari kelas i), di mana satu neuron merepresentasikan satu data *training* (Fausett, 1994). Setiap neuron pada lapisan ini akan menghasilkan keluaran ($\phi_{ij}(x)$) yang akan diteruskan ke lapisan jumlah. Keluaran neuron x_{ij} merupakan hasil operasi dari fungsi berikut (Mao & et.al., 2000; Tian & al., 2000) :

$$\phi_{ij}(x) = \frac{1}{(2\pi)^{d/2} \sigma^d} \exp \left[-\frac{(x - x_{ij})^T (x - x_{ij})}{2\sigma^2} \right] \quad (2.23)$$

di mana :

- x = data *testing*
- i = indeks kelas
- j = indeks neuron
- x_{ij} = neuron ke- j dari kelas i
- $\phi_{ij}(x)$ = keluaran neuron ke- j dari kelas i
- d = dimensi data *testing* x
- σ = *smoothing parameter*

Pada lapisan jumlah, dihitung *maximum likelihood* data *testing* x ($p_i(x)$) yang diklasifikasikan ke dalam kelas C_i dengan menjumlahkan dan menghitung rata-rata seluruh keluaran neuron-neuron yang berasal dari kelas yang sama pada lapisan pola. Hal ini dilakukan dengan menggunakan fungsi sebagai berikut (Mao & et.al., 2000) :

$$p_i(x) = \frac{1}{N_i} \sum_{j=1}^{N_i} \phi_{ij}(x) \quad i = 1, 2, 3, \dots, m \quad (2.24)$$

di mana :

- $p_i(x)$ = nilai estimasi PDF data *testing* x bila diklasifikasikan ke kelas i
- N_i = banyak neuron kelas i

$\phi_{ij}(x)$ = keluaran neuron ke- j dari kelas i
 m = banyak kelas

Pada lapisan terakhir, yaitu lapisan keputusan, akan ditentukan di kelas manakah data *testing* x diklasifikasikan. Lapisan ini hanya terdiri dari satu neuron yang menerima masukan dari setiap neuron di lapisan jumlah. Cara pengambilan keputusan dalam lapisan ini sesuai dengan teori klasifikasi Bayesian, di mana keputusan diambil berdasarkan hasil perhitungan *maximum likelihood* di lapisan jumlah yang merupakan nilai estimasi PDF dari masing-masing kelas. Data *testing* x akan diklasifikasikan ke dalam kelas yang mempunyai nilai estimasi PDF paling besar.

$$C(x) = \arg \max\{ p_i(x) \} \quad i = 1, 2, 3, \dots, m \quad (2.25)$$

di mana :

$C(x)$ = kelas estimasi data *testing* x

Nilai *smoothing parameter* (σ) merupakan parameter yang penting karena menentukan lebar kurva Gaussian yang terbentuk pada setiap titik sampel (Jatmiko, 2000). Pemilihan nilai σ akan mempengaruhi fungsi probabilitas. Nilai *smoothing parameter* yang terlalu kecil akan membuat setiap kasus sampel sangat berpengaruh sehingga dapat menghilangkan manfaat dari informasi kelompok. Jika nilai σ terlalu besar, maka sampel data akan terlalu kabur di mana informasi detil dari kerapatan akan hilang dan akan mengganggu estimasi dari kerapatan. Pemilihan nilai σ yang sesuai sangat diperlukan karena akan mempengaruhi kinerja dari algoritma PNN dalam proses klasifikasi suatu pola. Nilai σ yang sesuai dapat berbeda-beda, bergantung pada data yang digunakan (Mao & et.al., 2000). Nilai *smoothing parameter* yang umumnya ditentukan dengan menggunakan persamaan heuristik sebagai berikut :

$$\sigma = \frac{d_{\max}}{\sqrt{K}} \quad (2.26)$$

di mana :

d_{max} = jarak Euclidean maksimum antarvektor rata-rata tiap kelas

K = banyak kelas

2.3. Optimasi Struktur Neuron JST PNN

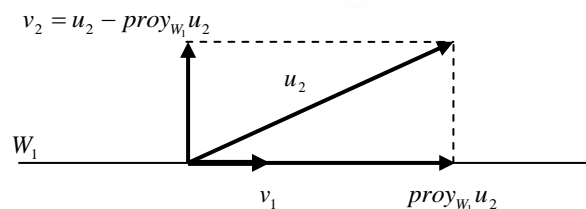
2.3.1. Proses Gram-Schmidt

Proses Gram-Schmidt adalah proses untuk mengubah sembarang basis menjadi basis ortogonal (Anton, 2000). Basis-basis ortogonal dapat diubah menjadi basis-basis ortonormal dengan membagi masing-masing basis ortogonal dengan norma basis tersebut. Bila V adalah sembarang ruang hasil kali dalam tak-nol berdimensi n dan $U = \{u_1, u_2, \dots, u_n\}$ adalah sembarang basis untuk V , maka langkah untuk menghasilkan suatu basis ortogonal $\{v_1, v_2, \dots, v_n\}$ untuk V adalah (Anton, 2000) :

Langkah 1 Anggap $v_1 = u_1$.

Langkah 2 Suatu vektor v_2 yang ortogonal terhadap v_1 diperoleh dengan menghitung komponen u_2 yang ortogonal terhadap ruang W_1 yang terentang oleh v_1 .

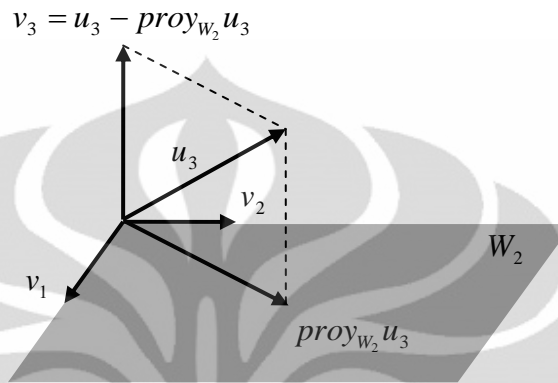
$$\begin{aligned} v_2 &= u_2 - \text{proy}_{W_1} u_2 \\ &= u_2 - \frac{\langle u_2, v_1 \rangle}{\|v_1\|^2} v_1 \end{aligned} \quad (2.27)$$



Gambar 2.5. Ilustrasi pencarian vektor v_2

Langkah 3 Suatu vektor v_3 yang ortogonal terhadap v_1 dan v_2 diperoleh dengan menghitung komponen u_3 yang ortogonal terhadap ruang W_2 yang terentang oleh v_1 dan v_2 .

$$\begin{aligned}
 v_3 &= u_3 - \text{proy}_{W_2} u_3 \\
 &= u_3 - \frac{\langle u_3, v_1 \rangle}{\|v_1\|^2} v_1 - \frac{\langle u_3, v_2 \rangle}{\|v_2\|^2} v_2
 \end{aligned} \tag{2.28}$$



Gambar 2.6. Ilustrasi pencarian vektor v_3

Langkah 4

Suatu vektor v_4 yang ortogonal terhadap v_1 , v_2 , dan v_3 dapat diperoleh dengan menghitung komponen u_4 yang ortogonal terhadap ruang W_3 yang terentang oleh v_1 , v_2 , dan v_3 .

$$\begin{aligned}
 v_4 &= u_4 - \text{proy}_{W_3} u_4 \\
 &= u_4 - \frac{\langle u_4, v_1 \rangle}{\|v_1\|^2} v_1 - \frac{\langle u_4, v_2 \rangle}{\|v_2\|^2} v_2 - \frac{\langle u_4, v_3 \rangle}{\|v_3\|^2} v_3
 \end{aligned} \tag{2.29}$$

...

...

...

Langkah ini diteruskan hingga sebanyak n langkah sehingga didapatkan himpunan vektor-vektor ortogonal $\{v_1, v_2, \dots, v_n\}$.

Keterangan : $\langle a, b \rangle$ = hasil kali dalam (*inner product*) vektor a dan b

Dengan proses Gram-Schmidt dapat pula dihitung vektor-vektor ortonormal dengan membagi vektor-vektor ortogonal yang didapat dengan norma masing-masing vektor.

$$q_i = \frac{v_i}{\|v_i\|} \quad (2.30)$$

2.3.2. Metode Pencarian Struktur Neuron Optimal dengan Orthogonal Algorithms

Orthogonal Algorithms (OA) menggunakan Proses Gram-Schmidt sebagai dasar pengurutan data. Langkah-langkah dari OA adalah sebagai berikut (Mao & et.al., 2000) :

1. Tentukan nilai *smoothing parameter* (σ) yang akan digunakan untuk proses OA. Sebaiknya nilai *smoothing parameter* optimal yang digunakan karena nilai ini mempengaruhi urutan data yang dihasilkan.
2. Untuk mengukur kedekatan neuron ke- j dalam kelas ke- i dengan neuron-neuron di dekatnya, maka perlu dihitung nilai PDF neuron tersebut terhadap seluruh neuron dalam kelas ke- i . Oleh karena itu, seluruh data *training* (X_{tr}) dijadikan neuron pada lapisan pola sekaligus sebagai data yang diujikan ke PNN untuk mendapatkan seluruh nilai PDF dan didapatkan matriks Φ_i yang berisi nilai PDF seluruh data *training* terhadap seluruh neuron, di mana jumlah neuron sama dengan jumlah data *training* yaitu sebanyak N_i .

$$\phi_i = \underbrace{\begin{bmatrix} \phi_{i1}(x_{i1}) & \phi_{i2}(x_{i1}) & \cdots & \phi_{iN_i}(x_{i1}) \\ \phi_{i1}(x_{i2}) & \phi_{i2}(x_{i2}) & \cdots & \phi_{iN_i}(x_{i2}) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{i1}(x_{iN_i}) & \phi_{i2}(x_{iN_i}) & \cdots & \phi_{iN_i}(x_{iN_i}) \end{bmatrix}}_{\text{Sejumlah neuron}} \left. \vphantom{\begin{bmatrix} \phi_{i1}(x_{i1}) & \phi_{i2}(x_{i1}) & \cdots & \phi_{iN_i}(x_{i1}) \\ \phi_{i1}(x_{i2}) & \phi_{i2}(x_{i2}) & \cdots & \phi_{iN_i}(x_{i2}) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{i1}(x_{iN_i}) & \phi_{i2}(x_{iN_i}) & \cdots & \phi_{iN_i}(x_{iN_i}) \end{bmatrix}} \right\} \text{Sejumlah data train}$$

$i = 1, 2, \dots$, (banyak kelas)

3. Transformasi Ortogonal Linear atau Proses Gram-Schmidt diterapkan untuk mendekomposisi matriks Φ_i menjadi persamaan linier dengan basis yang ortogonal.

$$\phi = QR = [Q_1, Q_2, \dots, Q_{N_i}]R \quad (2.31)$$

Dekomposisi QR untuk matriks Φ_i dilakukan untuk setiap kelas dengan langkah sebagai berikut:

a. Mencari nilai Q_l

i. Seluruh neuron dapat dijadikan kandidat Q_l dengan notasi $Q_l^{(\alpha)}$

$$\begin{aligned} Q_l^{(\alpha)} &= \phi_{k_\alpha} \\ \alpha &= 1, 2, \dots, N_i \end{aligned} \quad (2.32)$$

ii. Tingkat kepentingan masing-masing $Q_l^{(\alpha)}$ dapat dihitung dengan rumus berikut.

$$\Gamma_1^{(\alpha)} = [Q_l^{(\alpha)}]^T Q_l^{(\alpha)} \quad (2.33)$$

Semakin besar nilai Γ berarti semakin dekat neuron-neuron lain dengan neuron tersebut.

iii. Q_l terpilih adalah yang memiliki nilai $\Gamma_1^{(\alpha)}$ terbesar.

iv. Neuron terpenting pertama adalah neuron yang bersesuaian dengan $\Gamma_1^{(\alpha)}$ terbesar.

b. Mencari nilai Q_j

i. Asumsi $Q_l - Q_{j-1}$ dalam tiap kelas telah terpilih.

(Contoh: untuk mencari Q_4 maka minimal Q_l sampai Q_3 telah diketahui)

ii. Seluruh neuron yang tersisa dijadikan sebagai kandidat Q_j yang dinotasikan sebagai $Q_j^{(\alpha)}$, yang nilainya merupakan proyeksi ortogonal ϕ_{k_α} terhadap vektor yang tegak lurus terhadap seluruh basis ortogonal yang telah didapat sebelumnya (Q_l sampai Q_{j-1}).

$$Q_j^{(\alpha)} = \phi_{k_\alpha} - \sum_{l=1}^{j-1} r_{l\alpha}^{(\alpha)} Q_l, \quad \alpha = 1, 2, \dots, N_i - j + 1 \quad (2.34)$$

$$r_{l\alpha}^{(\alpha)} = \frac{Q_l^T \phi_{k_\alpha}}{Q_l^T Q_l}, \quad \alpha = 1, 2, \dots, N_i - j + 1, \quad l < j \quad (2.35)$$

di mana ϕ_{k_α} adalah vektor psi yang dijadikan kandidat

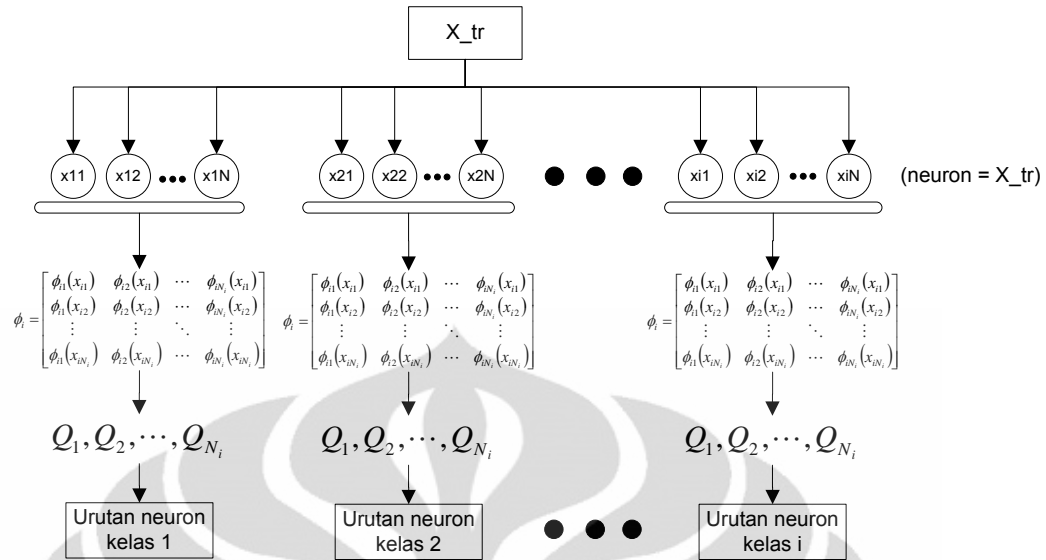
iii. Persamaan (2.34) dapat dijabarkan menjadi berikut:

$$Q_2^{(\alpha)} = \phi_{k_\alpha} - \frac{Q_1^T \cdot \phi_{k_\alpha}}{Q_1^T \cdot Q_1} Q_1 \quad (2.36)$$

$$Q_3^{(\alpha)} = \phi_{k_\alpha} - \frac{Q_1^T \cdot \phi_{k_\alpha}}{Q_1^T \cdot Q_1} Q_1 - \frac{Q_2^T \cdot \phi_{k_\alpha}}{Q_2^T \cdot Q_2} Q_2 \quad (2.37)$$

$$Q_4^{(\alpha)} = \phi_{k_\alpha} - \frac{Q_1^T \cdot \phi_{k_\alpha}}{Q_1^T \cdot Q_1} Q_1 - \frac{Q_2^T \cdot \phi_{k_\alpha}}{Q_2^T \cdot Q_2} Q_2 - \frac{Q_3^T \cdot \phi_{k_\alpha}}{Q_3^T \cdot Q_3} Q_3 \quad (2.38)$$

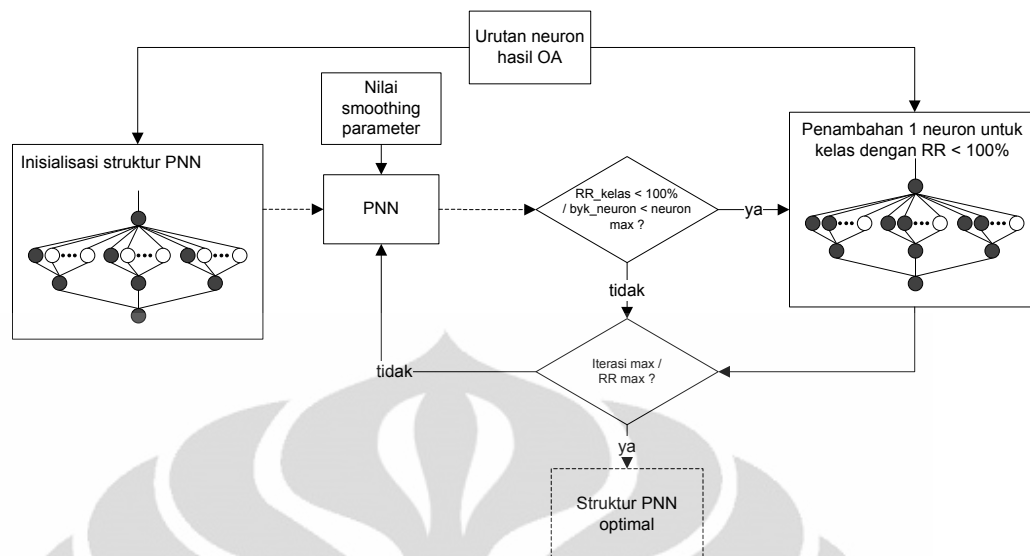
- iv. Tingkat kepentingan masing-masing $Q_j^{(\alpha)}$ dapat dihitung dengan rumus yang sama seperti pada pencarian Q_l .
- v. Q_j dipilih dari kandidat ($Q_j^{(\alpha)}$) yang memiliki nilai $\Gamma_j^{(\alpha)}$ terbesar.
- vi. Neuron terpenting ke- j adalah neuron yang bersesuaian dengan nilai $\Gamma_j^{(\alpha)}$ terbesar.



Gambar 2.7. Orthogonal Algorithms

Proses optimasi struktur neuron diawali dengan penggunaan Orthogonal Algorithms (OA) untuk mencari urutan neuron-neuron yang tersusun dari data *training* (X_{tr}) dalam setiap kelas menurut tingkat kepentingannya. Tingkat kepentingan neuron sebanding dengan tingkat representatif neuron terhadap kelasnya. Dengan kata lain, neuron dengan tingkat kepentingan tertinggi dapat dianggap sebagai neuron yang paling mewakili kelasnya. Neuron-neuron yang akan ditempatkan pada struktur neuron PNN akan diambil secara berurutan dari neuron urutan pertama (neuron terpenting), urutan neuron kedua (neuron terpenting kedua), dan seterusnya. Proses OA dilakukan dengan nilai *smoothing parameter* tertentu, di mana nilai yang berbeda dapat memberikan urutan neuron yang berbeda.

Setelah didapatkan urutan neuron menurut tingkat kepentingannya, mula-mula dibentuk struktur neuron di mana masing-masing kelas hanya tersusun dari 1 neuron. Struktur ini kemudian diuji sehingga diperoleh hasil rekognisi (RR) tiap kelas. Kelas yang mempunyai $RR < 100\%$ akan ditambah 1 neuron pada setiap iterasi secara bertahap hingga diperoleh struktur neuron yang memberikan $RR = 100\%$ atau telah mencapai banyak neuron maksimum tiap kelas. Proses ini dilakukan berulang-ulang hingga mencapai iterasi tertentu. Struktur PNN yang didapatkan dari proses ini dianggap sebagai struktur PNN optimal.



Gambar 2.8. Metode pencarian struktur neuron optimal dengan OA

2.4. Optimasi Nilai Smoothing Parameter JST PNN

2.4.1. Genetic Algorithms

Genetic Algorithms (GA) atau algoritma genetika adalah sebuah teknik pencarian stokhastik yang berdasarkan pada mekanisme seleksi alam dan genetika alam (Gen & Cheng, 1996). Teknik ini dikembangkan oleh Holland dan dilanjutkan oleh Goldberg. GA sering kali digunakan sebagai teknik optimasi untuk mencari solusi yang paling optimal dari suatu kasus. Mekanisme seleksi dan genetika alam didasarkan pada kemampuan individu-individu dalam suatu populasi untuk melakukan perkawinan dan menghasilkan keturunan yang memiliki karakteristik dari kedua orang tuanya sehingga dapat menopang untuk generasi berikutnya.

2.4.1.1. Prosedur Genetic Algorithms

Pada umumnya, GA memiliki 2 operasi utama, yaitu operasi genetik yang meliputi *crossover* (persilangan) dan *mutation* (mutasi) dan operasi evolusi yang meliputi *selection* (seleksi). Operasi genetik merupakan tahap reproduksi untuk menghasilkan kromosom atau individu baru, sedangkan operasi evolusi merupakan tahap seleksi kromosom untuk memilih kromosom orang tua yang kemudian akan digunakan pada tahap reproduksi. Seleksi kromosom orang tua dilakukan dengan memilih kromosom-kromosom dalam populasi berdasarkan

nilai *fitness* kromosom-kromosom tersebut. Dari populasi kromosom orang tua yang terbentuk, dilakukan tahap reproduksi dengan *crossover* atau *mutation* untuk menghasilkan populasi baru. Proses pembentukan 1 populasi baru inilah yang disebut generasi. Generasi terus dilakukan hingga didapatkan konvergensi dengan kromosom yang mempunyai nilai *fitness* yang sesuai dengan solusi yang diinginkan.

Proses yang dibicarakan di atas merupakan proses utama dari GA sehingga dapat dinamakan sebagai mesin GA (*GA engine*). Pada *GA engine*, terdapat 3 tahapan proses, yaitu tahap evaluasi, tahap seleksi, dan tahap reproduksi. Tahapan-tahapan ini diawali dengan pembentukan populasi berisi kromosom-kromosom yang mewakili sejumlah solusi acak dari permasalahan yang ada. Proses transformasi dari bentuk kromosom menjadi bentuk solusi riil dan sebaliknya diperlukan agar solusi-solusi yang mungkin dapat diproses oleh GA dan solusi-solusi optimal yang terpilih dapat diimplementasikan. Proses transformasi ini berlangsung dalam sebuah tahapan yang disebut tahap *coding* (pengkodean). Terdapat 2 jenis tahap *coding*, yaitu tahap *encoding* dan tahap *decoding*. Tahap *encoding* adalah tahap penerjemahan solusi riil menjadi bentuk kromosom yang mewakilinya, sedangkan tahap *decoding* merupakan tahap penerjemahan kembali dari bentuk kromosom menjadi solusi riil.

Prosedur lengkap dari mekanisme GA dari solusi riil dari suatu masalah hingga menjadi solusi optimal adalah sebagai berikut :

1. *Encoding*

Tahap penerjemahan sejumlah solusi riil menjadi kromosom yang mewakilinya sehingga terbentuk populasi

2. *GA Engine*

Tahap terjadinya generasi yang berulang-ulang hingga didapatkan solusi optimal dengan melalui 3 tahapan, yaitu :

- a. *Evaluation*

Tahap pengujian kecocokan (*fitness*) solusi-solusi yang ada dengan solusi yang diinginkan melalui *fitness function*.

- b. *Selection*

Tahap pencarian solusi untuk menjadi kromosom orang tua yang akan digunakan untuk menghasilkan keturunan baru.

c. *Reproduction : crossover dan mutation*

Tahap pembentukan kromosom baru dari kromosom orang tua yang dipilih

3. *Decoding*

Tahap penerjemahan kembali dari kromosom menjadi solusi riil.



Gambar 2.9. Flowchart algoritma genetika

2.4.1.2. Tahap Evaluation (Evaluasi)

Evaluasi merupakan suatu proses penentuan nilai *fitness* (kecocokan) dari setiap kromosom dalam populasi yang dilakukan pada setiap generasi. Setiap kromosom akan diberikan nilai dengan fungsi *fitness* tertentu. Fungsi *fitness* ditentukan dengan memperhatikan kesesuaiannya dengan masalah yang dihadapi. Nilai *fitness* inilah yang kemudian akan digunakan dalam tahap seleksi untuk menentukan kromosom-kromosom untuk membentuk populasi orang tua.

2.4.1.3. Tahap Selection (Seleksi)

Pada tahap seleksi ini, akan dibentuk populasi orang tua beranggotakan kromosom-kromosom terpilih yang akan menjadi cikal bakal pembentukan populasi kromosom baru. Salah satu metode seleksi yang paling sering digunakan adalah metode *Roulette Wheel*. Metode ini dikembangkan oleh Holland dan

bekerja berdasarkan nilai probabilitas yang menyatakan peluang suatu kromosom untuk dipilih menjadi kromosom orang tua. Nilai probabilitas ini sebanding dengan nilai *fitness* yang dimiliki kromosom. Kromosom dengan nilai *fitness* yang baik mempunyai peluang lebih besar untuk dipilih menjadi anggota populasi baru. Prosedur dari metode *Roulette Wheel* adalah sebagai berikut :

- a. Hitung nilai *fitness* $eval(v_k)$ untuk setiap kromosom v_k :

$$eval(v_k) = f(x), \quad k = 1, 2, \dots, pop_size \quad (2.39)$$

- b. Hitung total *fitness* dari sebuah populasi :

$$F = \sum_{k=1}^{pop_size} eval(v_k) \quad (2.40)$$

- c. Hitung probabilitas seleksi p_k untuk setiap kromosom v_k :

$$p_k = \frac{eval(v_k)}{F}, \quad k = 1, 2, \dots, pop_size \quad (2.41)$$

- d. Hitung probabilitas kumulatif q_k untuk setiap kromosom v_k :

$$q_k = \sum_{j=1}^k p_j, \quad k = 1, 2, \dots, pop_size \quad (2.42)$$

- e. Ambil angka r secara acak dalam *range* $[0,1]$.
 f. Untuk $q_{k-1} < r \leq q_k$, maka kromosom ke- k yang dipilih sebagai kromosom orang tua.

2.4.1.4. Tahap Reproduction (Reproduksi)

Reproduksi dalam GA bertujuan untuk menghasilkan kromosom-kromosom baru yang berasal dari kromosom orang tua yang telah dipilih pada tahap seleksi dengan menggunakan operator genetika, yaitu *crossover* (persilangan) dan

mutation (mutasi). Kromosom-kromosom baru untuk merepresentasikan solusi-solusi baru yang diharapkan menjadi solusi yang optimal bagi masalah yang dihadapi. Penetapan metode *crossover* dan *mutation* yang digunakan bergantung pada jenis kromosom yang digunakan dan permasalahan yang dihadapi. Selain 2 operator utama tersebut, ada satu metode lain yang disebut elitis. Metode ini secara langsung menempatkan kromosom dengan nilai *fitness* terbaik tiap generasi ke dalam populasi kromosom anak sehingga kromosom ini selalu diikutsertakan dalam generasi selanjutnya.

Operator *crossover* melakukan persilangan antara 2 kromosom orang tua untuk menghasilkan 2 kromosom baru. Tidak semua kromosom terpilih perlu dilakukan *crossover*. Banyak *crossover* yang dilakukan dikontrol dengan probabilitas tertentu yang disebut *crossover rate* (probabilitas *crossover*) - p_c . Metode sederhana untuk melakukan *crossover* pada kromosom biner adalah dengan memilih titik potong r secara acak dan menghasilkan keturunan dengan mengombinasikan bagian sebelah kiri titik potong salah satu kromosom dengan bagian sebelah kanan titik potong kromosom orang tua lainnya. Penulisan matematis dari *crossover* pada kromosom biner adalah :

Kromosom orang tua : X dan Y

Kromosom anak : X' dan Y'

$$X'_i = \begin{cases} X_i, & \text{untuk } i < r \\ Y_i, & \text{untuk } i \geq r \end{cases} \quad \text{dan} \quad Y'_i = \begin{cases} Y_i, & \text{untuk } i < r \\ X_i, & \text{untuk } i \geq r \end{cases}$$

dengan :

X_i, Y_i = urutan ke- i dari kromosom orang tua

X'_i, Y'_i = urutan ke- i dari kromosom anak



Gambar 2.10. Ilustrasi *crossover*

Operator *mutation* merupakan operator untuk mengubah nilai dari satu atau beberapa gen dalam suatu kromosom. Operasi *crossover* bertujuan untuk memperoleh kromosom-kromosom baru sebagai kandidat solusi pada generasi mendatang dengan *fitness* yang lebih baik menuju solusi optimal yang diinginkan. Akan tetapi, untuk mencapai hal ini, penekanan selektif juga memegang peranan yang penting. Jika dalam proses pemilihan kromosom-kromosom cenderung pada kromosom yang memiliki *fitness* yang tinggi saja, konvergensi prematur (mencapai solusi yang optimal lokal) sangat mudah terjadi. Untuk menghindari konvergensi prematur dan tetap menjaga perbedaan (*diversity*) kromosom-kromosom dalam populasi, selain melakukan penekanan selektif yang lebih efisien, operator mutasi juga dapat digunakan. Proses mutasi dalam sistem biologi berlangsung dengan mengubah isi *allele* gen pada suatu *locus* dengan *allele* yang lain. Proses mutasi ini bersifat acak sehingga tidak selalu menjamin bahwa setelah proses mutasi akan diperoleh kromosom dengan *fitness* yang lebih baik. Tidak setiap gen selalu dimutasi, tetapi operasi mutasi perlu dikontrol dengan probabilitas tertentu yang disebut dengan *mutation rate* (probabilitas mutasi) - p_m .

2.4.2. Metode Pencarian Nilai Smoothing Parameter Optimal dengan Genetic Algorithms

Nilai *smoothing parameter* yang akan diterapkan dalam pada PNN harus ditentukan secara selektif, yaitu menggunakan nilai yang sesuai dengan masalah yang akan ditangani PNN. Semakin sesuai nilai *smoothing parameter* yang digunakan, maka nilai PDF hasil perhitungan akan semakin dekat dengan nilai PDF sebenarnya. Oleh karena itu, perlu dicari nilai *smoothing parameter* yang optimal terhadap permasalahan yang dihadapi (dalam hal ini adalah data wajah). Pencarian nilai *smoothing parameter* pada penelitian ini dilakukan dengan menggunakan *Genetic Algorithms* (GA).

Tidak ada aturan-aturan khusus yang dapat memberikan batasan-batasan yang jelas atas nilai *smoothing parameter*. Pada penelitian ini, nilai *smoothing parameter* diberi batasan (*range*) nilai antara 0-100. Jenis kromosom yang digunakan untuk mewakili nilai *smoothing parameter* adalah kromosom desimal dalam bilangan riil. Oleh karena itu, nilai *smoothing parameter* tidak perlu

dilakukan *encoding* karena jenis kromosom yang digunakan mempunyai bentuk sama dengan nilai *smoothing parameter*. Kromosom dinilai dengan fungsi *fitness* yang didasarkan pada rata-rata hasil rekognisi data *training* (RR_{tr}) dengan data *testing* untuk *training* (RR_{trs}).

Tahap seleksi kromosom menggunakan metode *Roulette Wheel*, di mana besarnya probabilitas suatu kromosom untuk terpilih sesuai dengan nilai *fitness* yang dimiliki. Metode reproduksi yang digunakan menggunakan ketiga operator reproduksi yang ada, yaitu *crossover*, *mutation*, dan *elitis*. *Crossover rate* ditetapkan sebesar 0.5 dan *crossover* untuk jenis kromosom desimal ini dilakukan dengan persamaan sebagai berikut (Abdolee, Ali, & Rahman) :

$$X' = \frac{1}{4}(3X + Y) \quad (2.43)$$

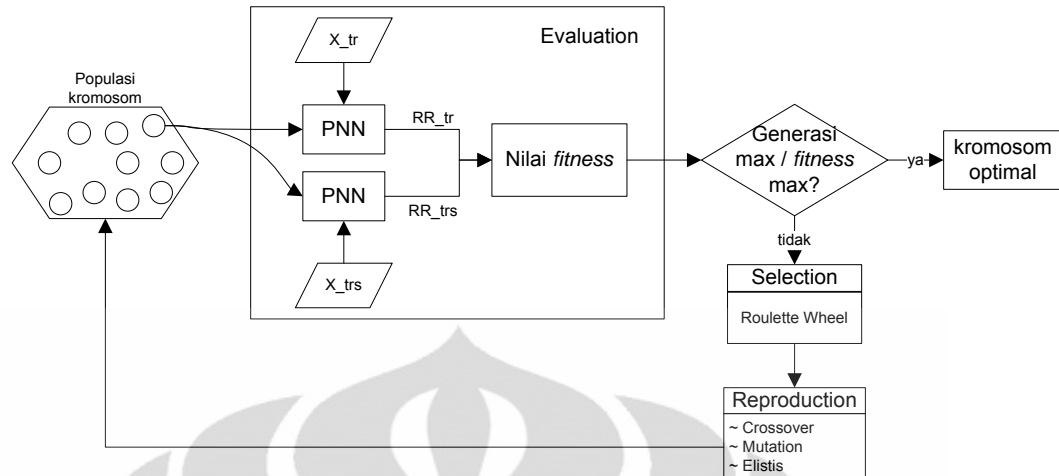
$$Y' = \frac{1}{2}|2Y - X| \quad (2.44)$$

di mana

X dan Y : kromosom orang tua

X' dan Y' : kromosom anak

Mutation rate ditetapkan sebesar 0.2 dan *mutation* dilakukan dengan membentuk kromosom baru secara *random*. Operator ketiga, yaitu *elitis* dilakukan dengan mencari kromosom terbaik dalam setiap generasi dan menempatkan kromosom terbaik tersebut pada urutan pertama dalam populasi anak sehingga akan diperhitungkan kembali pada generasi berikutnya. Hal ini untuk menjamin agar kromosom terbaik yang muncul pada suatu generasi tidak hilang seiring dengan terus bertambahnya generasi. GA akan berhenti ketika dicapai generasi maksimum yang ditetapkan atau didapatkan kromosom yang mempunyai nilai *fitness* maksimum dari fungsi *fitness* yang digunakan.



Gambar 2.11. Metode pencarian nilai *smoothing parameter* dengan GA

Rangkuman parameter GA yang digunakan pada penelitian :

- Kromosom : Nilai *smoothing parameter* (σ)
- Generasi maksimum : 20 generasi
- Jenis kromosom : desimal dengan *range* antara 0-100
- Inisialisasi kromosom : secara *random*
- Banyak kromosom : 10 kromosom dalam 1 populasi
- Metode evaluasi : menghitung hasil rekognisi data *training* (RR_{tr}) dan data *testing* untuk *training* (RR_{trs})
- Fungsi *fitness* : $Fitness = \frac{RR_{tr} + RR_{trs}}{2}$ (2.45)
- Metode seleksi : *Roulette Wheel*
- Metode reproduksi

- *Crossover* :
 - ~ Melakukan perhitungan kromosom anak dari kromosom-kromosom orang tua (2 kromosom orang tua menghasilkan 2 kromosom anak) dengan persamaan :

$$X' = \frac{1}{4}(3X + Y)$$

(2.46)

$$Y' = \frac{1}{2} |2Y - X| \quad (2.47)$$

di mana

X dan Y : kromosom orang tua

X' dan Y' : kromosom anak

- *Mutation* :
~ Inisialisasi ulang nilai *smoothing parameter* baru secara *random*
- *Elitis* :
~ Menempatkan kromosom dengan nilai *fitness* terbaik pada urutan pertama populasi anak

Crossover rate : 0.5

Mutation rate : 0.2

Stopping condition : generasi maksimum tercapai atau

$fitness = \max(fungsi_fitness) = 100$

2.5. Skema Percobaan

Dalam penelitian ini, dilakukan 3 jenis percobaan yang berbeda. Percobaan pertama adalah percobaan yang menggunakan struktur neuron PNN utuh dengan nilai *smoothing parameter* teroptimasi. Percobaan kedua dan ketiga menggunakan struktur neuron dan nilai *smoothing parameter* teroptimasi, namun berbeda pada teknik optimasi yang digunakan. Secara garis besar, dapat dikatakan bahwa optimasi pada percobaan kedua dilakukan dengan menentukan nilai *smoothing parameter* terlebih dahulu, baru kemudian mencari struktur PNN optimal terhadap nilai *smoothing parameter* tersebut sehingga teknik ini dapat disebut teknik optimasi PNN berorientasi nilai *smoothing parameter*. Teknik optimasi pada percobaan ketiga dilakukan dengan mencari nilai *smoothing parameter* yang bersesuaian dengan struktur neuron tertentu sehingga dapat disebut teknik optimasi PNN berorientasi struktur neuron. Penjelasan mengenai skema percobaan-percobaan yang dilakukan :

a. Percobaan 1 : Struktur neuron utuh dan nilai *smoothing parameter* optimal

Percobaan 1 dilakukan dengan struktur neuron utuh yang berarti seluruh data *training* digunakan sebagai neuron pada lapisan pola. Nilai *smoothing parameter* optimal dalam proses *training* PNN dilakukan dengan menggunakan GA. Setelah didapatkan nilai *smoothing parameter* optimal, dilakukan pengetesan menggunakan nilai tersebut untuk mengenali keseluruhan data (X_{tr} , X_{trs} , X_{ts}). Hasil percobaan yang didapatkan antara lain nilai *smoothing parameter* optimal, hasil rekognisi, waktu *training*, dan waktu *testing*. Percobaan dilakukan sebagai 5 kali agar didapatkan parameter hasil percobaan yang dapat merepresentasikan karakteristik percobaan ini.



Gambar 2.12. Skema Percobaan 1

b. Percobaan 2 : Teknik optimasi PNN berorientasi nilai *smoothing parameter*

Percobaan 2 menggunakan teknik optimasi berorientasi nilai *smoothing parameter* yang diusulkan oleh K. Z. Mao, K. C. Tan, dan W. Ser dalam jurnal yang berjudul “*Probabilistic Neural-Network Structure Determination for Pattern Classification*”. Secara garis besar, teknik yang diusulkan ini berisi metode pencarian struktur neuron optimal dengan menggunakan OA dan metode pencarian nilai *smoothing parameter* optimal dengan menggunakan GA. Teknik ini dilakukan dengan mengambil nilai *smoothing parameter* tertentu terlebih dahulu, baru kemudian dicari struktur neuron yang optimal terhadap nilai *smoothing parameter* tersebut. Percobaan 2 terbagi menjadi 2 percobaan yang disebut Percobaan 2A dan Percobaan 2B. Kedua percobaan ini menggunakan teknik optimasi yang sama, namun berbeda pada parameter GA dalam metode pencarian nilai *smoothing parameter*. Percobaan 2A menggunakan parameter yang digunakan pada jurnal acuan di atas, sedangkan Percobaan 2B menggunakan parameter yang dijelaskan pada Subbab 2.4. dan disebut parameter GA modifikasi jurnal. Perbandingan parameter GA yang digunakan pada percobaan ini antara lain :

Tabel 2.3. Perbandingan parameter GA menurut jurnal dan modifikasi jurnal

Parameter GA	Menurut Jurnal	Modifikasi Jurnal
Jenis kromosom	Bit desimal	Nilai desimal
Encoding	Nilai desimal → bit desimal 59.5683 → 5 9 5 6 8 3	-
Fungsi fitness	$\rho_i = \rho_{max} - \frac{\rho_{max} - \rho_{min}}{n_{max} - n_{min}}(n_i - n_{min})$	$fitness = \frac{RR_{tr} + RR_{trs}}{2}$
Crossover	Pertukaran bit desimal $\begin{array}{ c c c } \hline 9 & 6 & 2 & 8 \\ \hline 5 & 8 & 2 & 0 \\ \hline \end{array} \rightarrow \begin{array}{ c c c } \hline 9 & 8 & 2 & 8 \\ \hline 5 & 6 & 2 & 0 \\ \hline \end{array}$	Perhitungan nilai baru $X' = \frac{1}{4}(3X + Y) \quad Y' = \frac{1}{2} 2Y - X $
Mutation	Pembentukan bit baru secara random $\begin{array}{ c c c } \hline 4 & 5 & 1 & 7 \\ \hline \end{array} \rightarrow \begin{array}{ c c c } \hline 4 & 5 & 3 & 7 \\ \hline \end{array}$	Pembentukan nilai baru secara random $45.198 \rightarrow 10.325$
Stopping conditon	Generasi maksimum	Generasi dan fitness maksimum

Keterangan :

ρ = nilai *fitness*

n = banyak neuron aktif

RR = hasil rekognisi

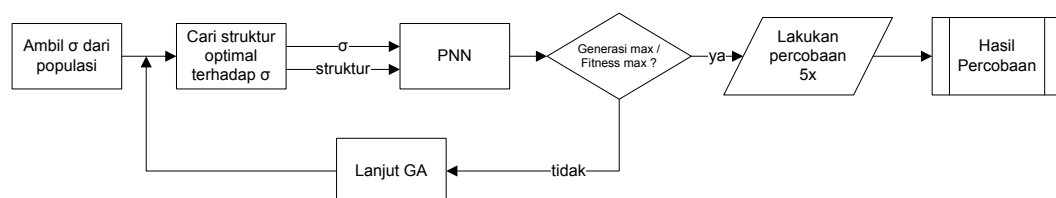
$\rho_{max} = 1$

$\rho_{min} = 0$

n_{max} = sesuai proporsi data

n_{min} = sesuai banyak kelas

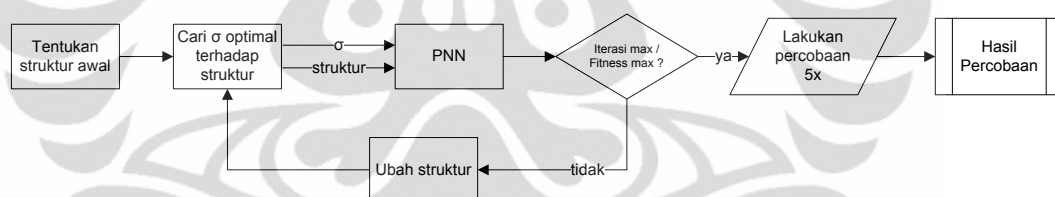
Hasil percobaan yang didapatkan antara lain struktur neuron optimal, nilai *smoothing parameter* optimal, hasil rekognisi, waktu *training*, dan waktu *testing*. Percobaan dilakukan sebagai 5 kali agar didapatkan parameter hasil percobaan yang dapat merepresentasikan karakteristik percobaan ini.



Gambar 2.13. Skema Percobaan 2

c. Percobaan 3 : Teknik optimasi PNN berorientasi struktur neuron

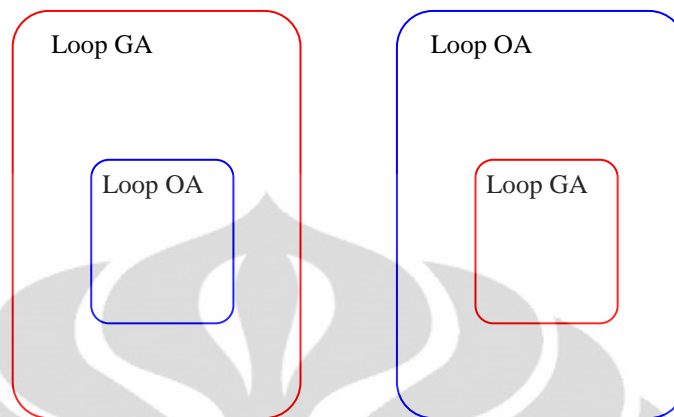
Percobaan 3 juga menggunakan OA sebagai perangkat untuk mencari struktur optimal dan GA untuk mencari nilai *smoothing parameter* optimal. Teknik yang digunakan pada percobaan ini merupakan kebalikan dari teknik pada Percobaan 2. Teknik optimasi pada percobaan ini dilakukan dengan menentukan struktur neuron terlebih dahulu, baru kemudian mencari nilai *smoothing parameter* terbaik untuk struktur tersebut. Struktur neuron optimal dicari sesuai dengan metode pencarian struktur neuron dengan OA, hanya saja proses OA hanya dilakukan sekali pada awal percobaan, di mana hasil urutan neuron dari OA akan selalu dipakai dalam membentuk struktur neuron yang digunakan. Parameter GA yang digunakan pada percobaan ini adalah parameter GA modifikasi jurnal. Hasil percobaan yang didapatkan antara lain struktur neuron optimal, nilai *smoothing parameter* optimal, hasil rekognisi, waktu *training*, dan waktu *testing*. Percobaan dilakukan sebagai 5 kali agar didapatkan parameter hasil percobaan yang dapat merepresentasikan karakteristik percobaan ini.



Gambar 2.14. Skema Percobaan 3

Perbedaan Percobaan 2 dan 3 juga dapat dilihat dari perbedaan letak *loop* metode pencarian optimal dalam prosedur pelaksanaannya. Percobaan 2 dan 3 menggunakan metode pencarian struktur neuron optimal dengan OA dan nilai *smoothing parameter* optimal dengan GA. Setiap metode pencarian ini membentuk sebuah *loop* dalam mencari struktur atau nilai yang optimal. Oleh karena itu, akan terdapat 2 buah *loop* dalam prosedur Percobaan 2 dan 3. Karena Percobaan 2 dilakukan dengan menentukan nilai *smoothing parameter* terlebih dahulu, kemudian dicari struktur yang bersesuaian dengannya, maka *loop* OA akan berada di dalam *loop* GA. Sebaliknya, karena Percobaan 3 dilakukan dengan

penentuan struktur terlebih dahulu, kemudian mencari nilai *smoothing parameter* yang bersesuaian terhadapnya, maka *loop GA* akan berada di dalam *loop OA*.



Gambar 2.15. Perbedaan letak *loop GA* dan *OA* pada Percobaan 2 dan 3

BAB 3

NILAI SMOOTHING PARAMETER OPTIMAL DENGAN GENETIC ALGORITHMS

Bab ini berisikan percobaan pertama yang dilakukan dengan struktur neuron PNN utuh dengan nilai *smoothing parameter* optimal yang dicari dengan menggunakan GA. Parameter GA yang digunakan pada percobaan ini adalah parameter GA modifikasi jurnal. Secara ideal, struktur neuron PNN utuh dan nilai *smoothing parameter* optimal akan memberikan hasil rekognisi (*RR*) terbaik yang dapat diperoleh dengan penggunaan JST algoritma PNN karena akan tersedia cukup banyak pola data yang akan mewakili kelasnya dalam neuron-neuron di lapisan pola. Namun, salah satu kelemahan penggunaan struktur utuh adalah bila terlalu banyak data yang mewakili kelas, hal tersebut dapat membuka peluang adanya data yang justru membuat pola kelas menjadi bergeser dan tidak sesuai. Parameter percobaan ini antara lain :

Data percobaan	: Data <i>infrared</i> tanpa NI (IR-TNI) Data <i>infrared</i> dengan NI (IR-NI) Data <i>visible light</i> tanpa NI (VL-TNI) Data <i>visible light</i> dengan NI (VL-NI)
Proporsi data [X_{tr}, X_{trs}, X_{ts}]	: [20,30,50]; [30,20,50]; [40,10,50]
Metode pendukung	: GA untuk mencari nilai <i>smoothing parameter</i>

3.1. Tujuan Percobaan

Tujuan dari dilakukannya Percobaan 1 : Struktur Neuron Utuh dan Nilai *Smoothing Parameter* Optimal antara lain :

1. Membuktikan metode pencarian nilai *smoothing parameter* dengan menggunakan GA;
2. Menganalisis pengaruh NI pada data IR dan VL;

3.2. Prosedur Percobaan

Prosedur percobaan terdiri dari 2 tahap utama, yaitu tahap pembelajaran JST dan tahap pengetesan JST. Pada tahap pembelajaran JST, data yang digunakan adalah X_{tr} dan X_{trs} , sedangkan pada tahap pengetesan JST, seluruh data akan diikutsertakan. Tahap pembelajaran menggunakan metode pencarian nilai *smoothing parameter* optimal dengan GA. Berikut ini adalah prosedur dalam melakukan Percobaan 1 :

a. Tahap Persiapan Data

Menyiapkan data percobaan berupa foto *infrared*, *visible light* dan mengolahnya hingga didapatkan data percobaan IR-TNI, IR-NI, VL_TNI, dan VL-NI yang masing-masing dibagi dengan proporsi [20,30,50]; [30,20,50]; dan [40,10,50].

b. Tahap Pembelajaran JST PNN

1. Pembentukan struktur neuron utuh.

Menggunakan seluruh data *training* (X_{tr}) sebagai neuron pada lapisan pola.

2. Inisialisasi populasi awal secara *random* berisi 10 kromosom yang mewakili nilai *smoothing parameter*, generasi = 1.

3. Perhitungan nilai *fitness* dengan melakukan perhitungan RR untuk masing-masing kromosom pada struktur utuh, kemudian gunakan RR_{tr} dan RR_{trs} yang didapatkan ke dalam fungsi *fitness*.

4. Bila *stopping condition* GA (generasi maksimum atau nilai *fitness* maksimum) belum terpenuhi, maka proses GA terus dilanjutkan (lanjut ke poin 5). Bila *stopping condition* terpenuhi, maka proses GA berhenti (langsung ke poin 7).

5. Berdasarkan nilai *fitness*, dilakukan pemilihan dan pembentukan populasi orang tua dengan metode *Roulette Wheel*.

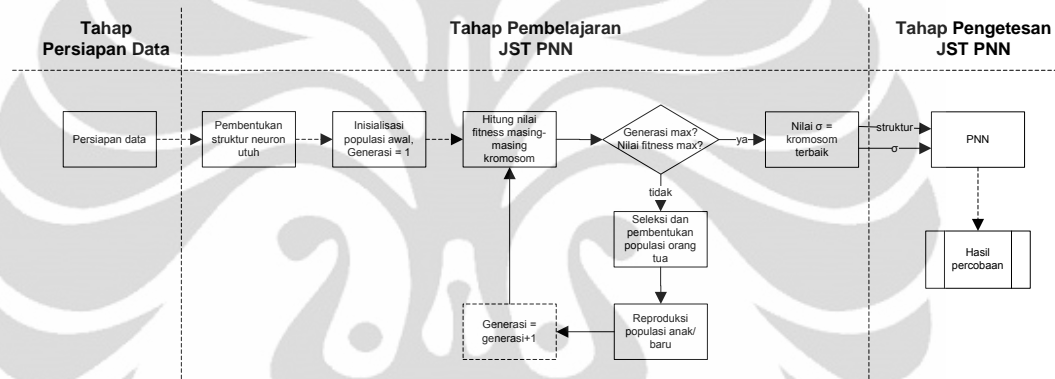
6. Pembentukan populasi anak sebagai populasi kromosom baru dengan melakukan reproduksi (*crossover*, *mutation*, dan *elitis*) terhadap kromosom-kromosom orang tua yang telah dipilih, kemudian ulang kembali dari poin 3. Generasi=generasi+1.

7. Kromosom dengan nilai *fitness* tertinggi merupakan kromosom terbaik (optimal).

c. Tahap Pengetesan JST PNN

1. Ambil nilai *smoothing parameter* berdasarkan kromosom terbaik hasil pembelajaran JST.
2. Lakukan pengetesan pada JST PNN dengan nilai *smoothing parameter* optimal, dan struktur neuron utuh.

Prosedur percobaan ini dilakukan sebanyak 5 kali untuk masing-masing jenis dan proporsi data.



Gambar 3.1. Prosedur Percobaan 1

3.3. Hasil dan Analisis Percobaan

Pengetesan yang dilakukan dengan struktur neuron utuh dan nilai *smoothing parameter* berdasarkan hasil pembelajaran, akan didapatkan hasil rekognisi (RR_{tr} , RR_{trs} , dan RR_{ts}). RR_{tr} dan RR_{trs} hanya digunakan sebagai parameter untuk melihat konvergensi pembelajaran JST PNN, sedangkan RR_{ts} akan digunakan sebagai parameter kemampuan (tingkat pengenalan) JST PNN. Definisi konvergen pada penelitian ini adalah pembelajaran JST PNN mampu memberikan RR_{tr} dan RR_{trs} sebesar 100%. Dengan definisi tersebut, untuk teknik optimasi yang menggunakan parameter GA modifikasi jurnal, pembelajaran berhenti bukan karena mencapai generasi atau iterasi maksimum, melainkan memenuhi syarat *stopping condition* yang lainnya, yaitu nilai *fitness* atau RR mencapai 100% sehingga waktu *training* yang didapatkan lebih cepat. Banyak neuron yang digunakan pada struktur neuron utuh adalah :

- Data *Infrared* (IR)

Total data : 200 foto

Tabel 3.1. Banyak neuron struktur utuh data IR

Proporsi [X_{tr}, X_{trs}, X_{ts}]	Banyak Data <i>Training</i> (X_{tr})	Banyak Neuron pada Struktur
[20,30,50]	40	40
[30,20,50]	60	60
[40,10,50]	80	80

- Data *Visible Light* (VL)

Total data : 100 foto

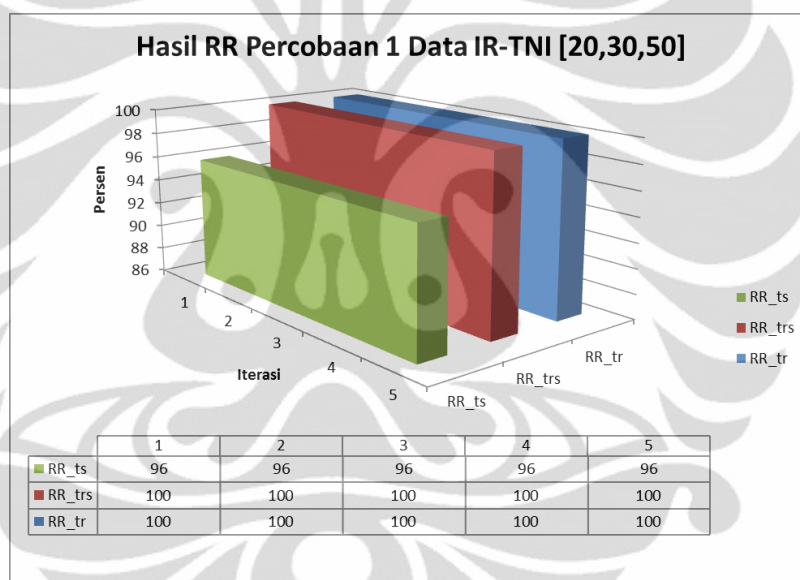
Tabel 3.2. Banyak neuron struktur utuh data VL

Proporsi [X_{tr}, X_{trs}, X_{ts}]	Banyak Data <i>Training</i> (X_{tr})	Banyak Neuron pada Struktur
[20,30,50]	20	20
[30,20,50]	40	40
[40,10,50]	60	60

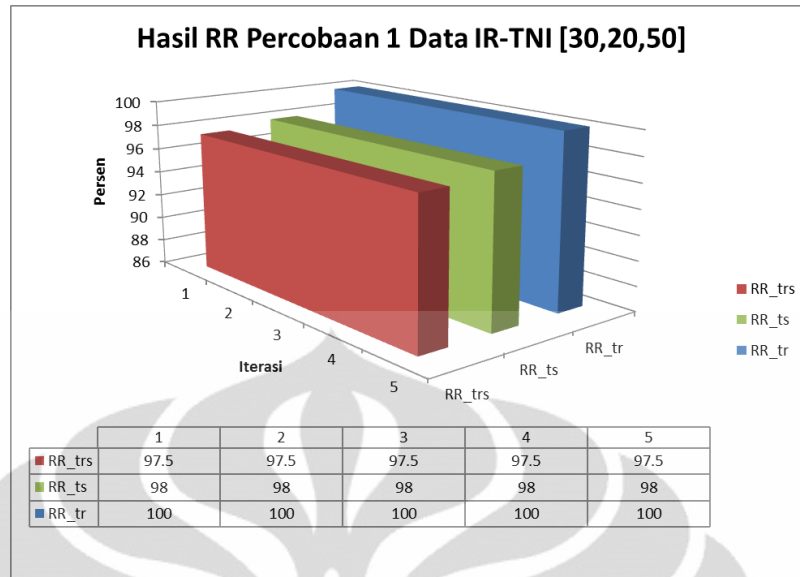
Hasil percobaan yang juga diamati ialah waktu *training*, waktu *testing*, dan nilai *smoothing parameter*. Waktu *training* adalah waktu pada tahap pembelajaran dari awal pembelajaran hingga berhenti karena memenuhi *stopping condition* tertentu. Untuk Percobaan 1, waktu *training* dapat didefinisikan juga sebagai waktu yang dibutuhkan untuk mendapatkan nilai *smoothing parameter* yang dianggap optimal dengan GA. JST PNN dengan pembelajaran konvergen cenderung memiliki waktu pembelajaran yang singkat karena proses pembelajaran tidak berlangsung hingga dicapai generasi atau iterasi maksimum. Waktu *testing* adalah waktu pada tahap pengetesan untuk menghitung RR seluruh data (X_{tr} , X_{trs} , dan X_{ts}). Nilai *smoothing parameter* yang diamati adalah kumpulan nilai *smoothing parameter* hasil pembelajaran.

3.3.1. Hasil dan Analisis Percobaan dengan Data IR-TNI

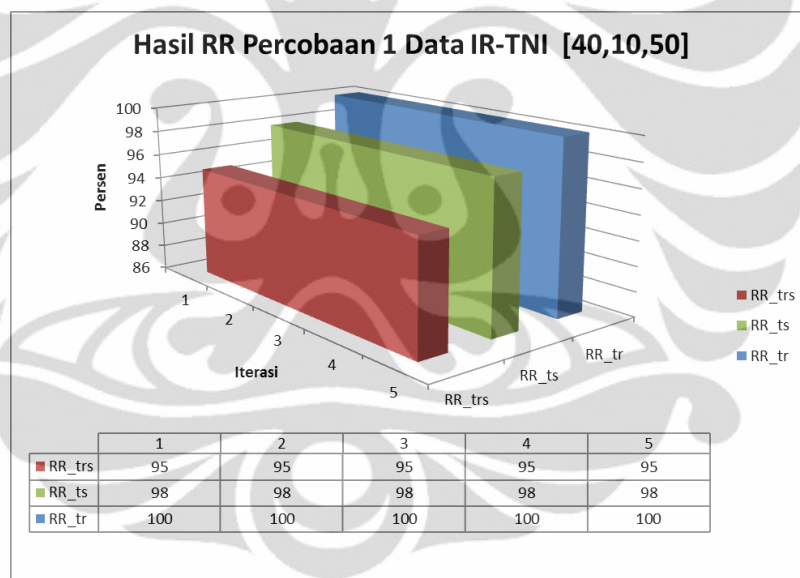
Dari 5 kali percobaan yang dilakukan dengan proporsi data [20,30,50]; [30,20,50]; dan [40,10,50], terlihat bahwa RR_{tr} , RR_{trs} , dan RR_{ts} yang didapatkan memberikan nilai yang konsisten (tidak terjadi fluktuasi). Untuk proporsi data [20,30,50], RR_{tr} dan RR_{trs} yang mencapai 100% untuk setiap perulangan menunjukkan bahwa pembelajaran dengan konfigurasi seperti ini merupakan pembelajaran konvergen. Sedangkan, untuk proporsi [30,20,50] dan [40,10,50], RR_{trs} tidak mencapai 100% sehingga pembelajaran konfigurasi ini tidak konvergen. Akibatnya, waktu *training* untuk proporsi [30,20,50] dan [40,10,50] lebih lama dibandingkan waktu *training* proporsi [20,30,50].



Gambar 3.2. Hasil RR perulangan Percobaan 1 data IR-TNI [20,30,50]



Gambar 3.3. Hasil RR perulangan Percobaan 1 data IR-TNI [30,20,50]



Gambar 3.4. Hasil RR perulangan Percobaan 1 data IR-TNI [40,10,50]

Dapat dilihat pada Tabel 3.3., waktu rata-rata yang dibutuhkan untuk mendapatkan nilai *smoothing parameter* optimal hanya 1.8652 sekon dengan tingkat pengenalan 96% untuk proporsi [20,30,50], sebesar 285.0654 sekon dengan tingkat pengenalan 98% untuk proporsi [30,20,50], dan 361.3316 sekon dengan tingkat pengenalan 98% untuk proporsi [40,10,50]. Dari data tersebut, waktu *training* untuk proporsi [30,20,50] dan [40,10,50] memang membutuhkan waktu yang lebih lama sehingga terbukti bahwa pembelajaran yang konvergen

membutuhkan waktu *training* yang lebih singkat. Waktu *testing* rata-rata tiap proporsi secara berurutan semakin meningkat dengan selisih sekitar 0.036 sekon. Selisih waktu *testing* ini disebabkan karena perbedaan banyak neuron yang digunakan untuk setiap proporsi. Semakin banyak neuron yang digunakan, waktu *testing* cenderung semakin lama dan hasil yang didapatkan telah menunjukkan hubungan ini. Tingkat pengenalan yang dihasilkan juga cenderung meningkat seiring semakin banyak neuron yang digunakan. Dengan tingkat pengenalan yang tergolong baik dari percobaan ini, dapat dikatakan bahwa nilai *smoothing parameter* optimal berhasil dicari dengan GA.

Tabel 3.3. Hasil Percobaan 1 data IR-TNI

Proporsi		<i>RR_tr</i>	<i>RR_trs</i>	<i>RR_ts</i>	Waktu Training	Waktu Testing
[20,30,50]	MIN	100.00	100.00	96.00	0.6338	0.1276
	MAX	100.00	100.00	96.00	6.7072	0.1690
	AVE	100.00	100.00	96.00	1.8652	0.1533
[30,20,50]	MIN	100.00	97.50	98.00	281.6861	0.1796
	MAX	100.00	97.50	98.00	286.8499	0.2047
	AVE	100.00	97.50	98.00	285.0654	0.1941
[40,10,50]	MIN	100.00	95.00	98.00	346.2978	0.2074
	MAX	100.00	95.00	98.00	418.3411	0.2356
	AVE	100.00	95.00	98.00	361.3316	0.2254

Nilai *smoothing parameter* Percobaan 1 data IR-TNI yang memberikan tingkat pengenalan tertinggi adalah sebagai berikut :

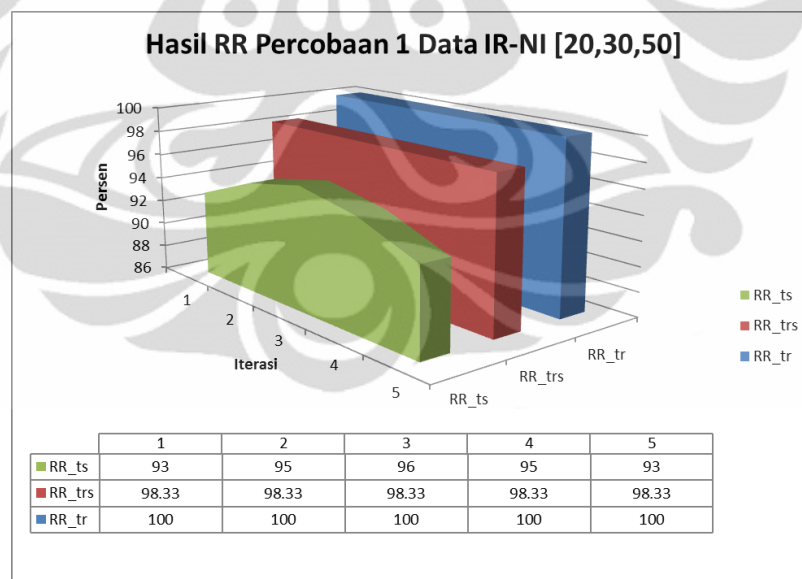
Tabel 3.4. Nilai σ Percobaan 1 data IR-TNI

Proporsi	<i>RR_ts</i> (%)	σ
[20,30,50]	96	1.915
[30,20,50]	98	45.485
[40,10,50]	98	43.202

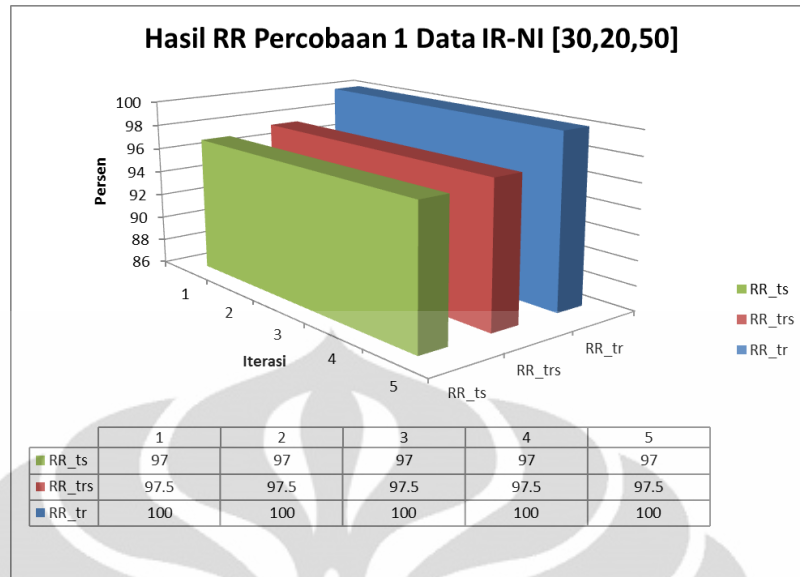
3.3.2. Hasil dan Analisis Percobaan dengan Data IR-NI

Dari 5 kali perulangan yang dilakukan dengan proporsi data [30,20,50]; dan [40,10,50], terlihat bahwa *RR_tr*, *RR_trs*, dan *RR_ts* yang didapatkan memberikan nilai yang konsisten. Hanya saja, untuk proporsi [20,30,50] terjadi fluktuasi *RR_ts*

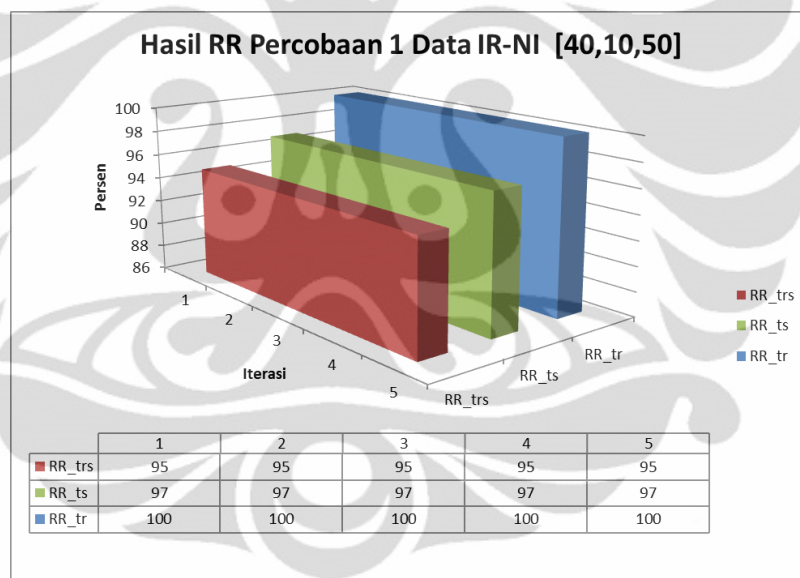
walaupun hasil GA telah mencapai nilai RR_{tr} dan RR_{trs} yang sama. Hal ini menandakan bahwa dengan nilai *smoothing parameter* yang berbeda, dapat dihasilkan RR_{tr} dan RR_{trs} yang sama, namun RR_{ts} yang berbeda. Seperti pada proporsi [20,30,50], terdapat nilai *smoothing parameter* berbeda yang memberikan RR_{tr} dan RR_{trs} yang sama, nilai-nilai yang berbeda ini dianggap sebagai nilai yang optimal oleh GA. Namun, setelah dilakukan pengetesan terhadap X_{ts} ternyata RR_{ts} yang didapatkan berbeda. Nilai *smoothing parameter* yang memberikan RR_{ts} yang lebih rendah tentunya bukan merupakan nilai yang optimal. Artinya, nilai *smoothing parameter* yang didapatkan dari GA belum tentu merupakan nilai yang memberikan tingkat pengenalan tertinggi walaupun telah memenuhi syarat-syarat GA yang ditentukan. Sifat nilai *smoothing parameter* ini menjadi salah satu keterbatasan dari metode GA yang digunakan pada penelitian ini. Pembelajaran pada jenis data IR-NI tidak konvergen karena tidak mampu memberikan $RR_{trs} = 100\%$ untuk seluruh proporsi.



Gambar 3.5. Hasil RR perulangan Percobaan 1 data IR-NI [20,30,50]



Gambar 3.6. Hasil RR perulangan Percobaan 1 data IR-NI [30,20,50]



Gambar 3.7. Hasil RR perulangan Percobaan 1 data IR-NI [40,10,50]

Tabel 3.5. menunjukkan waktu *training* dan waktu *testing* rata-rata yang semakin meningkat seiring dengan bertambahnya neuron yang digunakan. Waktu *training* menunjukkan angka yang besar karena pembelajaran yang tidak konvergen. Tingkat pengenalan yang didapatkan mencapai 94-97% yang termasuk tingkat pengenalan yang baik. Oleh karena itu, dapat dikatakan bahwa nilai *smoothing parameter* berhasil didapatkan walaupun pembelajaran tidak

konvergen. Terlihat pola bahwa semakin banyak neuron yang digunakan, tingkat pengenalan yang didapatkan juga cenderung meningkat.

Tabel 3.5. Hasil Percobaan 1 data IR-NI

Proporsi		<i>RR_tr</i>	<i>RR_trs</i>	<i>RR_ts</i>	Waktu <i>Training</i>	Waktu <i>Testing</i>
[20,30,50]	MIN	100.00	98.33	93.00	243.2410	0.1745
	MAX	100.00	98.33	96.00	292.3993	0.2373
	AVE	100.00	98.33	94.40	274.8713	0.1942
[30,20,50]	MIN	100.00	97.50	97.00	337.2995	0.1913
	MAX	100.00	97.50	97.00	389.3470	0.2391
	AVE	100.00	97.50	97.00	355.8378	0.2194
[40,10,50]	MIN	100.00	95.00	97.00	395.2796	0.2532
	MAX	100.00	95.00	97.00	475.7857	0.2822
	AVE	100.00	95.00	97.00	434.1100	0.2702

Nilai *smoothing parameter* Percobaan 1 data IR-NI yang memberikan tingkat pengenalan tertinggi adalah sebagai berikut :

Tabel 3.6. Nilai σ Percobaan 1 data IR-NI

Proporsi	<i>RR_ts</i> (%)	σ
[20,30,50]	96	4.229
[30,20,50]	97	41.276
[40,10,50]	97	41.650

3.3.3. Analisis Pengaruh Proses NI pada Data IR

Dari hasil dan analisis percobaan dengan data IR-TNI dan IR-NI dapat dibuat tabel perbandingan sebagai berikut :

Tabel 3.7. Perbandingan hasil Percobaan 1 data IR-TNI dan IR-NI

		IR-TNI	IR-NI
Konsistensi RR	[20,30,50]	✓	×
	[30,20,50]	✓	✓
	[40,10,50]	✓	✓
Konvergensi Pembelajaran	[20,30,50]	✓	×
	[30,20,50]	×	×
	[40,10,50]	×	×
Tingkat Pengenalan (%)	[20,30,50]	96.00	94.40
	[30,20,50]	98.00	97.00
	[40,10,50]	98.00	97.00
Waktu Training (sekon)	[20,30,50]	1.8652	274.8713
	[30,20,50]	285.0654	355.8378
	[40,10,50]	361.3316	434.1100
Waktu Testing (sekon)	[20,30,50]	0.1533	0.1942
	[30,20,50]	0.1941	0.2194
	[40,10,50]	0.2254	0.2702

Keterangan :

- ✓ : lebih baik atau bernilai baik
- × : lebih buruk atau bernilai buruk

Tabel perbandingan memperlihatkan bahwa NI memberikan pengaruh negatif pada data IR yang menyebabkan tingkat pengenalan, waktu *training*, dan waktu *testing* yang lebih buruk untuk data IR-NI dibandingkan data IR-TNI. Proses NI membuat proporsi [20,30,50] yang pada awalnya konvergen menjadi tidak konvergen. Waktu *training* yang dapat dibandingkan hanya proporsi [30,20,50] dan [40,10,50] yang mana pada data IR-TNI maupun IR-NI tidak konvergen. Waktu *training* antara kedua jenis data ini berbeda sekitar 70 sekon dan waktu *testing* berbeda sekitar 0.04 sekon.

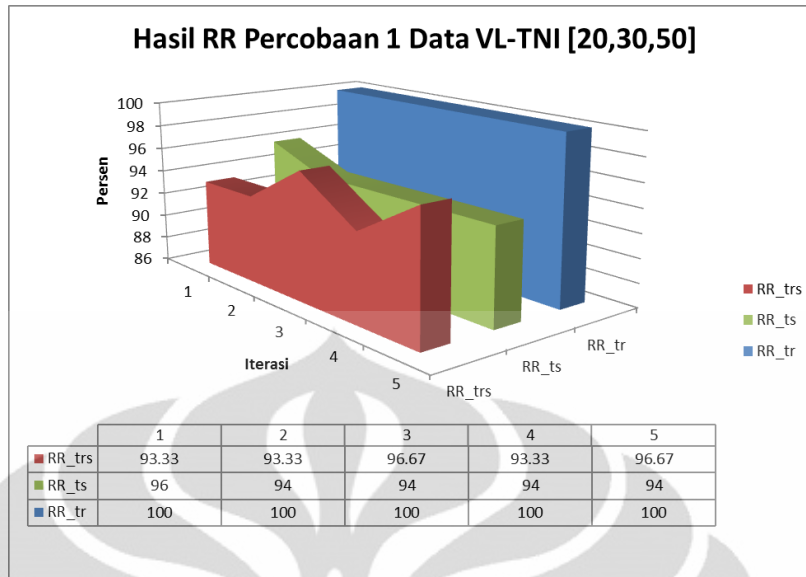
Proses NI juga menyebabkan tingkat pengenalan mengalami penurunan sekitar 1%. Walaupun penurunan tingkat pengenalan yang dibawa NI relatif tidak terlalu signifikan, namun dapat disimpulkan dari percobaan ini bahwa data IR tanpa dilakukan ni (IR-TNI) mempunyai hasil yang lebih baik. Hal ini kemungkinan besar disebabkan karena data IR telah memiliki tingkat iluminasi yang relatif seragam untuk setiap foto. Penggunaan NI pada data IR dapat

mengakibatkan pola-pola wajah yang merupakan informasi penting dalam sistem pengenalan menjadi kabur atau tidak jelas.

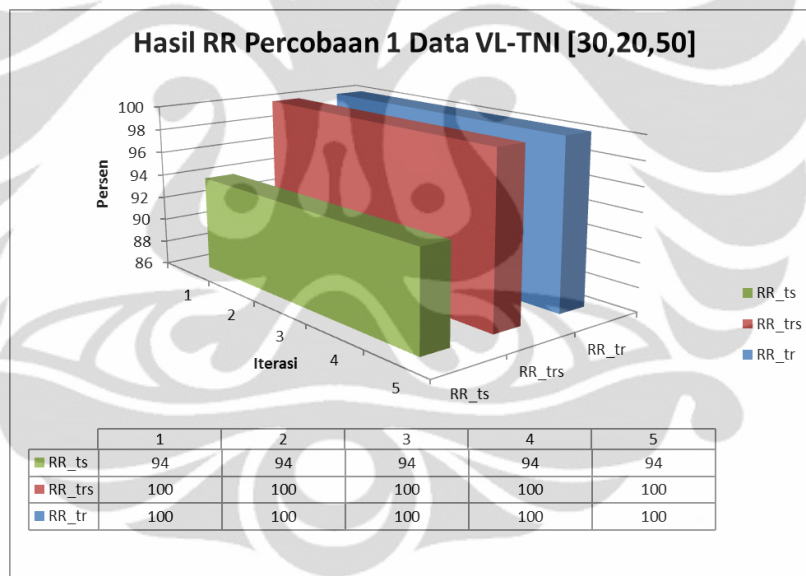
Dari hasil perulangan percobaan ini dengan data IR-TNI dan IR-NI didapatkan nilai *smoothing parameter* pada setiap perulangan tidak mengarah pada suatu nilai tertentu, melainkan mengarah pada nilai yang variatif. Nilai *smoothing parameter* yang berbeda, namun menunjuk pada tingkat pengenalan yang sama, menandakan bahwa nilai *smoothing parameter* optimal untuk konfigurasi tersebut terletak pada suatu *range* dan tidak terletak pada 1 titik saja. Oleh karena itu dapat disimpulkan bahwa nilai *smoothing parameter* optimal baik untuk data IR-TNI maupun IR-NI berada dalam *range* dan tidak berada dalam 1 titik saja. Metode pencarian nilai *smoothing parameter* dengan GA yang digunakan pada percobaan ini belum tentu berhasil memberikan nilai *smoothing parameter* optimal secara global, namun metode ini akan memberikan nilai *smoothing parameter* yang dianggap optimal sesuai dengan parameter GA yang ditentukan.

3.3.4. Hasil dan Analisis Percobaan dengan Data VL-TNI

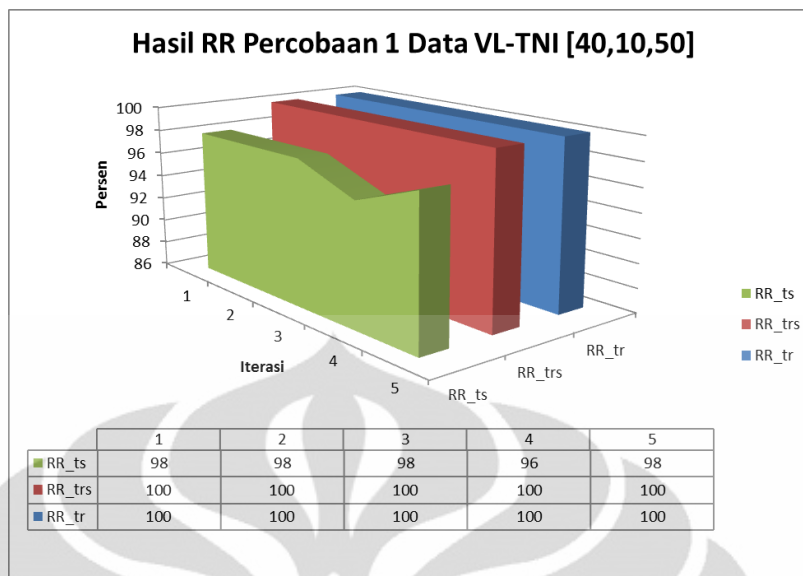
Dari 5 kali percobaan yang dilakukan dengan proporsi data [20,30,50], nilai RR_{trs} dan RR_{ts} yang dihasilkan tidak konsisten, sedangkan [30,20,50] dan [40,10,50] mempunyai RR_{trs} dan RR_{ts} yang lebih konsisten. Pada proporsi [20,30,50] terjadi fluktuasi dengan nilai RR_{trs} bervariasi antara 93.33% dan 96.67%, sedangkan RR_{ts} antara 94% dan 96%. Pada proporsi [40,10,50], terdapat nilai RR_{ts} dari sebuah perulangan yang memberikan hasil yang berbeda walaupun RR_{tr} dan RR_{trs} yang didapatkan sama dengan RR_{tr} dan RR_{trs} perulangan yang lain. Ini menandakan nilai *smoothing parameter* yang dihasilkan dari proporsi ini belum tentu merupakan nilai *smoothing parameter* yang optimal. Pembelajaran proporsi [20,30,50] merupakan pembelajaran tidak konvergen, sedangkan [30,20,50] dan [40,10,50] mempunyai pembelajaran yang konvergen.



Gambar 3.8. Hasil RR perulangan Percobaan 1 data VL-TNI [20,30,50]



Gambar 3.9. Hasil RR perulangan Percobaan 1 data VL-TNI [30,20,50]



Gambar 3.10. Hasil RR Percobaan 1 data VL-TNI [40,10,50]

Waktu *training* rata-rata untuk [20,30,50] mencapai 172.3425 sekon karena pembelajaran tidak konvergen, [30,20,50] mempunyai waktu *training* sangat singkat dengan rata-rata sebesar 0.6585 sekon, dan waktu *training* rata-rata [40,10,50] sebesar 4.1461 sekon. Tampak dari Tabel 3.8., waktu *testing* proporsi [20,30,50] yang didapatkan memiliki selisih antara minimum dan maksimum yang cukup besar, yaitu mencapai sekitar 0.2 sekon. Faktor utama yang dapat menyebabkan hal ini adalah perbedaan kondisi perangkat keras pada setiap perulangan. Namun, bila dilihat dari waktu *testing* rata-rata, masih terlihat pola bahwa semakin banyak neuron yang digunakan, semakin lama waktu *testing* yang diperlukan. Tingkat pengenalan rata-rata berada antara 94-97.6% yang menandakan nilai *smoothing parameter* berhasil didapatkan dengan GA. Semakin banyak neuron yang digunakan, tingkat pengenalan yang dihasilkan cenderung meningkat.

Tabel 3.8. Hasil Percobaan 1 data VL-TNI

Proporsi		<i>RR_tr</i>	<i>RR_trs</i>	<i>RR_ts</i>	Waktu Training	Waktu Testing
[20,30,50]	MIN	100.00	93.33	94.00	155.7295	0.1902
	MAX	100.00	96.67	96.00	190.3004	0.3488
	AVE	100.00	94.67	94.40	172.3425	0.2679
[30,20,50]	MIN	100.00	100.00	94.00	0.5812	0.2554
	MAX	100.00	100.00	94.00	0.7558	0.2954
	AVE	100.00	100.00	94.00	0.6585	0.2703
[40,10,50]	MIN	100.00	100.00	96.00	1.1642	0.2513
	MAX	100.00	100.00	98.00	8.0822	0.5410
	AVE	100.00	100.00	97.60	4.1461	0.3280

Hal menarik terlihat pada hasil rekognisi untuk proporsi data [20,30,50], di mana hasil yang didapatkan adalah sebagai berikut :

Tabel 3.9. Nilai σ dan RR Percobaan 1 data VL-TNI [20,30,50]

σ	<i>RR_tr</i>	<i>RR_trs</i>	<i>RR_ts</i>
3.918	100	93.33	96
27.306	100	93.33	94
30.031	100	93.33	94
1.834	100	96.67	94
2.029	100	96.67	94

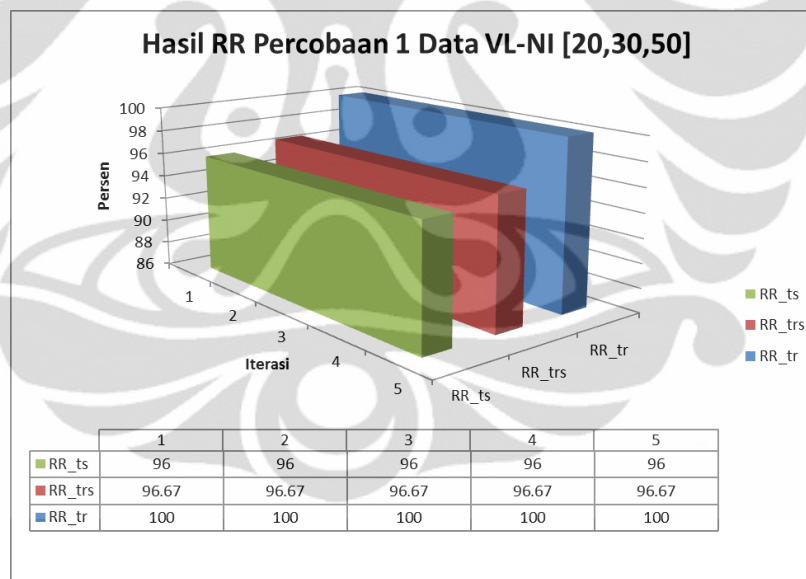
Dari tabel di atas, tingkat pengenalan tertinggi didapatkan dengan pasangan *RR_tr* dan *RR_trs* sebesar 100% dan 93.33%. Hasil ini memberi pernyataan bahwa nilai *smoothing parameter* yang memberikan *RR_tr* dan *RR_trs* terbaik belum tentu menghasilkan tingkat pengenalan yang terbaik, sedangkan pada metode GA pada penelitian ini, digunakan algoritma untuk mencari nilai *smoothing parameter* yang memberikan pasangan *RR_tr* dan *RR_trs* yang terbaik. Keadaan ini memang memungkinkan untuk terjadi sehingga perlu diingat bahwa nilai *smoothing parameter* yang didapatkan belum tentu merupakan nilai yang optimal secara global, namun nilai yang optimal menurut parameter yang digunakan. Nilai *smoothing parameter* Percobaan 1 data VL-TNI yang memberikan tingkat pengenalan tertinggi adalah sebagai berikut :

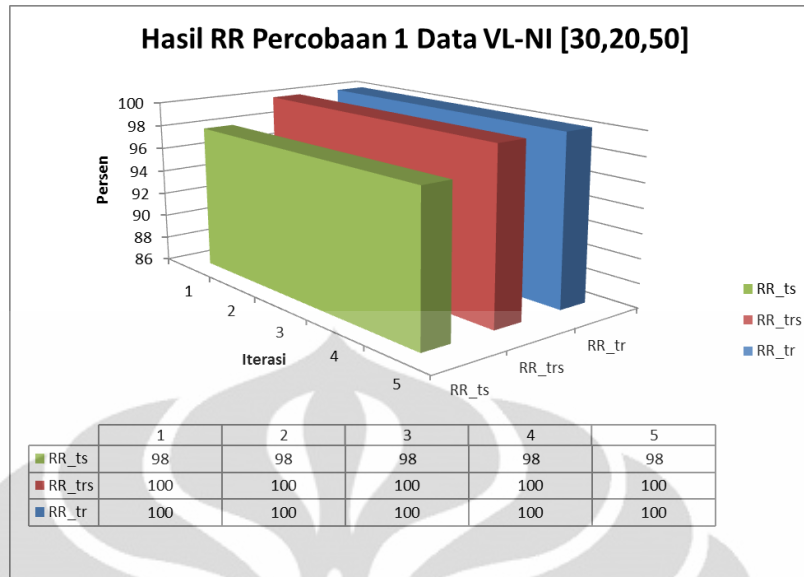
Tabel 3.10. Nilai σ Percobaan 1 data VL-TNI

Proporsi	RR_{ts} (%)	σ
[20,30,50]	96	3.918
[30,20,50]	94	10.633
[40,10,50]	98	2.286

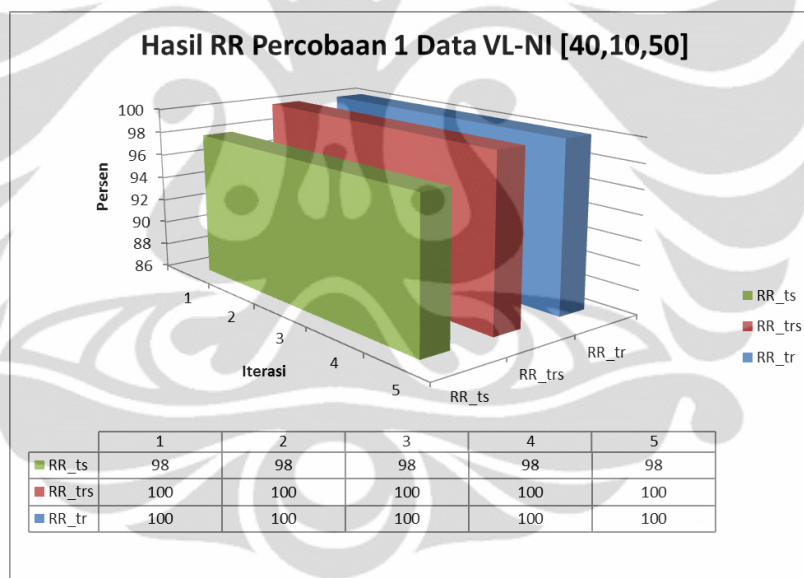
3.3.5. Hasil dan Analisis Percobaan dengan Data VL-NI

Percobaan 1 dengan data VL-NI menghasilkan nilai RR_{tr} , RR_{trs} , dan RR_{ts} yang konsisten untuk setiap perulangannya. Hal ini berlaku untuk semua proporsi yang digunakan, yaitu [20,30,50]; [30,20,50]; dan [40,10,50]. Pembelajaran konvergen dicapai dengan proporsi [30,20,50] dan [40,10,50], sedangkan pembelajaran proporsi [20,30,50] tidak konvergen dengan RR_{trs} maksimum hanya 96.67%.

**Gambar 3.11.** Hasil RR Percobaan 1 data VL-NI [20,30,50]



Gambar 3.12. Hasil RR Percobaan 1 data VL-NI [30,20,50]



Gambar 3.13. Hasil RR Percobaan 1 data VL-NI [40,10,50]

Pembelajaran konvergen pada proporsi [30,20,50] secara rata-rata hanya membutuhkan 0.4992 sekon dan untuk proporsi [40,10,50] hanya dibutuhkan 0.5392 sekon. Waktu *testing* yang dihasilkan dengan jenis data ini menunjukkan adanya fluktuasi yang tidak sebanding dengan banyak neuron yang digunakan. Walaupun perbedaan banyak neuron yang digunakan pada ketiga proporsi ini hanya 10 neuron, seharusnya waktu *testing* rata-rata yang didapatkan semakin besar seiring dengan semakin banyak neuron yang digunakan. Ketidaksesuaian ini

dapat saja terjadi karena adanya program lain yang secara sengaja maupun tidak sengaja membebani kinerja perangkat keras yang digunakan saat pewartuan tersebut dihitung. Tingkat pengenalan yang didapatkan sangat baik dengan *range* antara 96-98% sehingga dapat dikatakan nilai *smoothing parameter* yang didapatkan dari GA telah sesuai. Ada kecenderungan dengan menggunakan neuron yang lebih banyak dapat meningkatkan tingkat pengenalan.

Tabel 3.11. Hasil Percobaan 1 data VL-NI

Proporsi		<i>RR_tr</i>	<i>RR_trs</i>	<i>RR_ts</i>	Waktu Training	Waktu Testing
[20,30,50]	MIN	100.00	96.67	96.00	145.1456	0.1183
	MAX	100.00	96.67	96.00	155.2455	0.2485
	AVE	100.00	96.67	96.00	149.6257	0.1456
[30,20,50]	MIN	100.00	100.00	98.00	0.4539	0.0750
	MAX	100.00	100.00	98.00	0.5633	0.1345
	AVE	100.00	100.00	98.00	0.4992	0.1191
[40,10,50]	MIN	100.00	100.00	98.00	0.4271	0.1343
	MAX	100.00	100.00	98.00	0.6472	0.1739
	AVE	100.00	100.00	98.00	0.5392	0.1522

Nilai *smoothing parameter* Percobaan 1 data VL-TNI yang memberikan tingkat pengenalan tertinggi adalah sebagai berikut :

Tabel 3.12. Nilai σ Percobaan 1 data VL-NI

Proporsi	<i>RR_ts</i> (%)	σ
[20,30,50]	96	1.898
[30,20,50]	98	8.582
[40,10,50]	98	2.133

3.3.6. Analisis Pengaruh Proses NI pada Data VL

Dari hasil dan analisis percobaan dengan data VL-TNI dan VL-NI dapat dibuat tabel perbandingan sebagai berikut :

Tabel 3.13. Perbandingan hasil Percobaan 1 data VL-TNI dan VL-NI

		VL-TNI	VL-NI
Konsistensi RR	[20,30,50]	×	✓
	[30,20,50]	✓	✓
	[40,10,50]	×	✓
Konvergensi Pembelajaran	[20,30,50]	×	×
	[30,20,50]	✓	✓
	[40,10,50]	✓	✓
Tingkat Pengenalan (%)	[20,30,50]	94.40	96.00
	[30,20,50]	94.00	98.00
	[40,10,50]	97.60	98.00
Waktu Training (sekon)	[20,30,50]	172.3425	149.6257
	[30,20,50]	0.6585	0.4992
	[40,10,50]	4.1461	0.5392
Waktu Testing (sekon)	[20,30,50]	0.2679	0.1456
	[30,20,50]	0.2703	0.1191
	[40,10,50]	0.3280	0.1522

Keterangan :

- ✓ : lebih baik atau bernilai baik
- × : lebih buruk atau bernilai buruk

Dari tabel perbandingan di atas dapat disimpulkan bahwa data VL-NI lebih baik dibandingkan data VL-TNI. Proses NI dapat mengubah konsistensi RR untuk setiap perulangan, meningkatkan tingkat pengenalan, mempersingkat waktu *training* dan juga waktu *testing*. Walaupun dalam hal pewaktuan, kurang tepat untuk dilakukan perbandingan langsung mengingat adanya anomali dalam data yang didapatkan, akan tetapi terdapat perbedaan yang cukup signifikan pada waktu *training* dan waktu *testing* kedua jenis data ini. Adanya anomali data relatif tidak akan menyebabkan perbedaan sebesar ini. Oleh karena itu, dapat dikatakan bahwa proses NI pada data VL menyebabkan komputasi yang lebih cepat.

Proses NI pada data VL memberikan hasil yang lebih baik dibandingkan data VL-TNI karena tingkat iluminasi semua foto disamaratakan dengan proses NI. Foto yang dihasilkan dengan mode *visible light* akan mempunyai tingkat iluminasi yang berbeda-beda, tergantung pada pencahayaan yang ada pada saat pengambilan foto. Pola wajah yang terganggu akibat iluminasi akan menjadi lebih jelas setelah tingkat iluminasi disamakan. Terbukti bahwa dengan menyamakan

tingkat iluminasi ini, tingkat pengenalan lebih baik dan waktu komputasi juga lebih singkat.

Sama seperti pada Percobaan 1 dengan data IR-TNI dan IR-NI bahwa nilai *smoothing parameter* yang didapatkan pada percobaan dengan data VL-TNI dan VL-NI juga bervariasi dan tidak terpacu pada suatu nilai tertentu. Ini menandakan nilai *smoothing parameter* optimal yang didapatkan dengan metode ini berada dalam *range* tertentu. Namun, hal yang perlu ditekankan adalah nilai *smoothing parameter* yang didapatkan dari metode pencarian dengan GA ini belum tentu merupakan nilai *smoothing parameter* optimal secara global karena tidak ada yang dapat memprediksikan dengan pasti bagaimana parameter GA yang tepat untuk permasalahan dari data-data yang mempunyai karakteristiknya masing-masing.

3.4. Kesimpulan

Beberapa kesimpulan yang dapat ditarik dari hasil percobaan dan analisis pada Percobaan 1 menggunakan data IR dan VL adalah :

1. Metode GA berhasil diterapkan untuk mencari nilai *smoothing parameter* yang memberikan tingkat pengenalan yang tinggi, baik untuk data IR-TNI, IR-NI, VL-TNI, dan VL-NI, sehingga dapat dianggap nilai tersebut merupakan nilai *smoothing parameter* optimal;
2. Data IR memberikan hasil yang lebih baik untuk data IR-TNI, di mana proses NI pada data IR berpotensi menyebabkan pembelajaran tidak konvergen, menurunkan tingkat pengenalan dan meningkatkan waktu komputasi;
3. Data VL dengan proses NI memberikan hasil yang lebih baik dibandingkan data VL tanpa proses NI. Proses NI pada data VL dapat menghasilkan RR yang konsisten untuk setiap perulangan, meningkatkan tingkat pengenalan, dan mempercepat waktu komputasi;
4. Nilai *smoothing parameter* yang didapatkan dari metode pencarian dengan GA merupakan nilai *smoothing parameter* yang dianggap optimal sesuai dengan parameter yang digunakan sehingga nilai ini belum tentu merupakan nilai yang optimal secara global.

BAB 4

TEKNIK OPTIMASI PROBABILISTIC NEURAL NETWORK

Bab 4 berisikan percobaan-percobaan yang menggunakan teknik optimasi untuk mereduksi neuron aktif pada struktur neuron dan mencari nilai *smoothing parameter* optimal. Terdapat 2 teknik optimasi PNN yang digunakan, yaitu teknik optimasi PNN berorientasi nilai *smoothing parameter* dan teknik optimasi PNN berorientasi struktur neuron. Percobaan 2 menggunakan teknik optimasi PNN berorientasi nilai *smoothing parameter* yang terdiri dari Percobaan 2A dengan parameter GA sesuai dengan jurnal berjudul “*Probabilistic Neural-Network Structure Determination for Pattern Classification*” yang diajukan oleh K. Z. Mao, K. C. Tan, dan W. Ser. dan Percobaan 2B dengan parameter GA yang dijelaskan pada Subbab 2.4. (parameter GA modifikasi jurnal). PNN teroptimasi dengan teknik ini disebut dengan istilah OPNN_SP. Percobaan 3 menggunakan teknik optimasi PNN berorientasi struktur neuron dengan parameter GA modifikasi jurnal. PNN teroptimasi dengan teknik ini disebut dengan istilah OPNN_SN. Data yang digunakan pada percobaan ini mengambil referensi mengenai data pada Percobaan 1. Parameter Percobaan 2 antara lain :

Data percobaan	: Data <i>infrared</i> tanpa NI (IR-TNI) Data <i>visible light</i> dengan NI (VL-NI)
Proporsi data [X_{tr}, X_{trs}, X_{ts}]	: [20,30,50]; [30,20,50]; [40,10,50]
Metode pendukung	: GA untuk mencari nilai <i>smoothing parameter</i> OA untuk mencari struktur neuron optimal

4.1. Tujuan Percobaan

Tujuan dari dilakukannya percobaan mengenai PNN dengan struktur neuron dan nilai *smoothing parameter* teroptimasi antara lain :

1. Membuktikan teknik optimasi PNN dalam mencari struktur neuron dan nilai *smoothing parameter* optimal;
2. Mengetahui sifat-sifat teknik optimasi PNN berorientasi nilai *smoothing parameter* dan juga struktur neuron.

4.2. Teknik Optimasi PNN Berorientasi Nilai Smoothing Parameter

4.2.1. Prosedur Percobaan

Prosedur Percobaan 2 terdiri dari 2 tahap utama, yaitu tahap pembelajaran JST dan tahap pengetesan JST. Pada tahap pembelajaran JST, data yang digunakan adalah X_{tr} dan X_{trs} , sedangkan pada tahap pengetesan JST, seluruh data akan diikutsertakan. Tahap pembelajaran JST menggunakan metode pencarian nilai *smoothing parameter* optimal dengan GA dan metode pencarian struktur neuron optimal dengan OA. Berikut ini adalah prosedur dalam melakukan Percobaan 2 :

a. Tahap Persiapan Data

Menyiapkan data percobaan berupa foto *infrared*, *visible light* dan mengolahnya hingga didapatkan data percobaan IR-TNI dan VL-NI yang masing-masing dibagi dengan proporsi [20,30,50]; [30,20,50]; dan [40,10,50].

b. Tahap Pembelajaran JST PNN

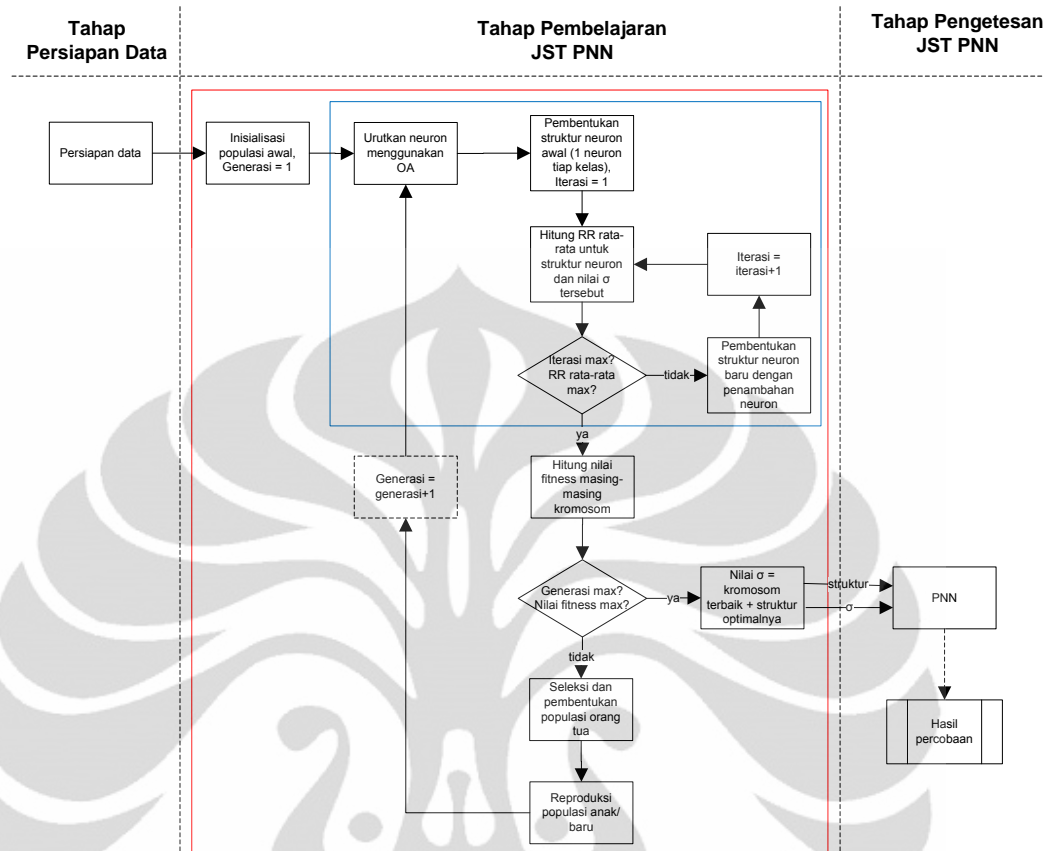
1. Inisialisasi populasi awal secara random berisi 10 kromosom yang mewakili nilai *smoothing parameter*, generasi = 1.
2. Pengurutan neuron dengan OA menggunakan sebuah kromosom dalam populasi.
3. Pembentukan struktur neuron awal yang terdiri dari 1 neuron untuk setiap kelas, iterasi = 1.
4. Hitung RR rata-rata antara RR_{tr} dan RR_{trs} dengan pasangan struktur neuron dan nilai *smoothing parameter* (kromosom) tersebut.
5. Bila *stopping condition* pencarian struktur neuron (iterasi maksimum atau RR rata-rata = 100) belum terpenuhi, proses pencarian struktur neuron dilanjutkan (lanjut ke poin 6). Bila *stopping condition* terpenuhi, maka proses pencarian struktur neuron untuk kromosom tersebut berhenti (langsung ke poin 7).
6. Pembentukan struktur neuron baru dengan menambahkan 1 neuron pada kelas yang mempunyai RR kelas rata-rata < 100% dan banyak neuron pada kelas tersebut < banyak neuron maksimum pada kelas itu, kemudian ulang kembali dari poin 4. Iterasi = iterasi+1.

7. Cari struktur neuron untuk kromosom lainnya hingga seluruh kromosom dalam populasi telah mendapat struktur neuron optimal (ulangi poin 2-6)
 8. Setelah semua kromosom telah mendapatkan pasangan struktur neuron optimalnya, hitung nilai *fitness* dengan melakukan perhitungan RR untuk masing-masing kromosom, kemudian gunakan *RR_tr* dan *RR_trs* yang didapatkan ke dalam fungsi *fitness*.
 9. Bila *stopping condition* GA (generasi maksimum atau nilai *fitness* maksimum) belum terpenuhi, maka proses GA terus dilanjutkan (lanjut ke poin 10). Bila *stopping condition* terpenuhi, maka proses GA berhenti (lanjut ke poin 12).
 10. Berdasarkan nilai *fitness*, dilakukan pemilihan dan pembentukan populasi orang tua dengan metode *Roulette Wheel*.
 11. Pembentukan populasi anak sebagai populasi kromosom baru dengan melakukan reproduksi (*crossover*, *mutation*, dan elitis) terhadap kromosom-kromosom orang tua yang telah dipilih, kemudian ulang kembali dari poin 2. $\text{Generasi} = \text{generasi} + 1$.
 12. Kromosom dan struktur neuron dengan nilai *fitness* tertinggi merupakan kromosom dan struktur neuron terbaik (optimal).
- c. Tahap Pengetesan JST PNN
1. Gunakan pasangan struktur neuron dan nilai *smoothing parameter* (kromosom) dari tahap pembelajaran yang memberikan RR terbaik.
 2. Lakukan pengetesan pada JST PNN dengan nilai *smoothing parameter* optimal, dan struktur neuron tersebut.

Pada tahap pembelajaran JST, poin b.2.-b.7. merupakan metode pencarian struktur neuron optimal dengan OA, sedangkan poin b.1., b.8.-b.12. merupakan metode pencarian nilai *smoothing parameter* optimal dengan GA. Prosedur percobaan ini dilakukan sebanyak 5 kali untuk masing-masing jenis dan proporsi data.

Keterangan :

Pada penjelasan prosedur ini, parameter GA yang digunakan adalah parameter GA modifikasi jurnal. Percobaan 2A memiliki prosedur yang sama, hanya parameter GA yang berbeda, yaitu parameter GA menurut jurnal.

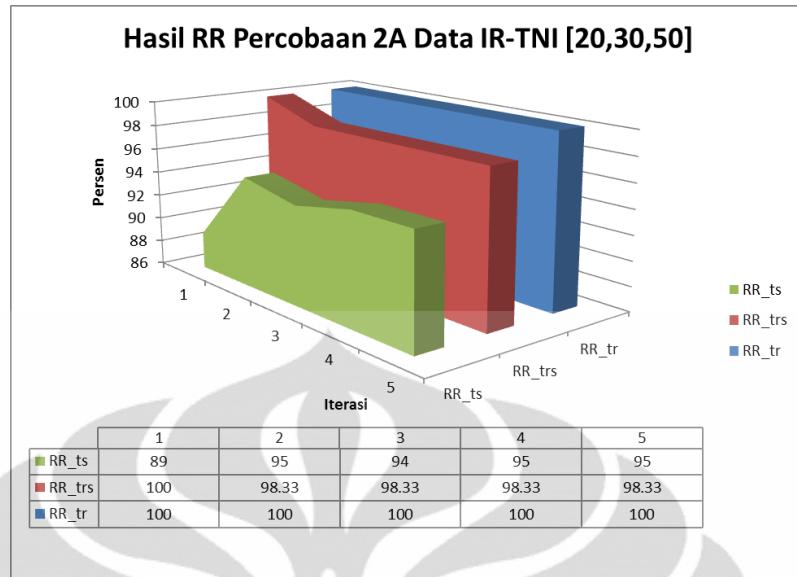


Gambar 4.1. Prosedur Percobaan 2

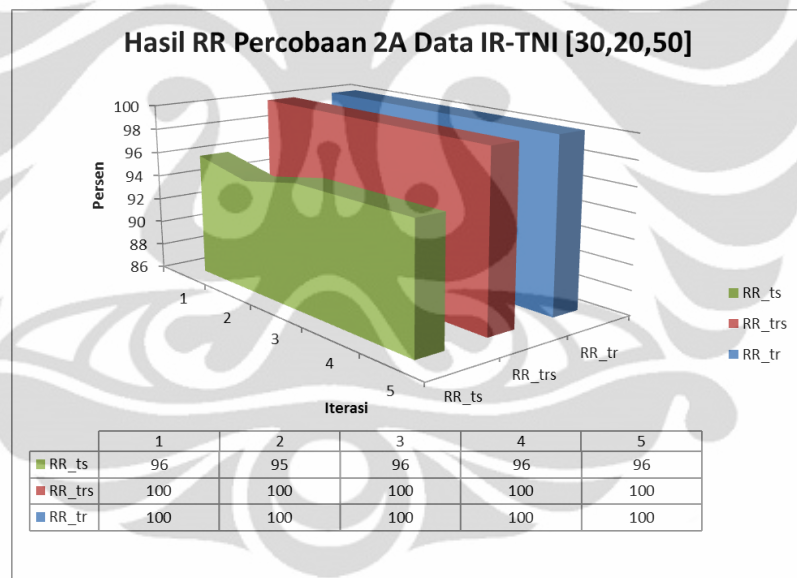
4.2.2. Teknik Optimasi PNN Berorientasi Nilai Smoothing Parameter dengan Parameter GA Menurut Jurnal

4.2.2.1. Hasil dan Analisis Percobaan dengan Data IR-TNI

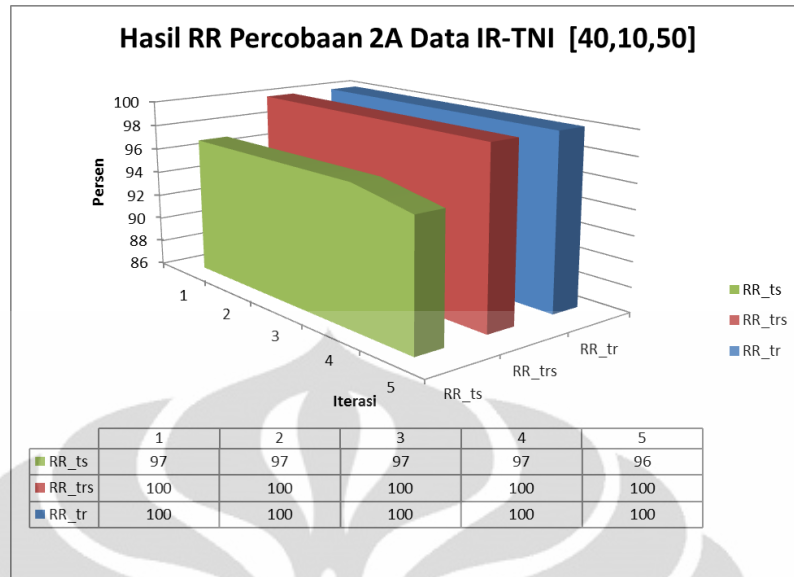
Dari 5 kali perulangan yang dilakukan untuk Percobaan 2A dengan data IR-TNI, terlihat bahwa hasil rekognisi pada seluruh proporsi data tidak konsisten karena pada tiap perulangannya tidak selalu didapatkan hasil rekognisi yang sama. Pembelajaran konvergen didapatkan untuk proporsi [30,20,50] dan [40,10,50], sedangkan proporsi [20,30,50] memiliki pembelajaran yang tidak konvergen karena RR_{trs} tidak mencapai 100%.



Gambar 4.2. Hasil RR Percobaan 2A data IR-TNI [20,30,50]



Gambar 4.3. Hasil RR Percobaan 2A data IR-TNI [30,20,50]



Gambar 4.4. Hasil RR Percobaan 2A data IR-TNI [40,10,50]

Waktu *training* Percobaan 2A dengan data IR-TNI untuk pembelajaran tidak konvergen rata-rata sebesar 243 sekon, sedangkan untuk pembelajaran konvergen berkisar antara 15-18 sekon. Seperti yang dijelaskan sebelumnya bahwa pada parameter GA menurut jurnal, pembelajaran berhenti setelah mencapai generasi maksimum. Dengan alasan ini, pembelajaran konvergen maupun tidak konvergen, keduanya berhenti setelah mencapai generasi maksimum. Perbedaan waktu yang terjadi antara kedua pembelajaran ini disebabkan oleh proses pencarian struktur neuron optimal. Pada proporsi [20,30,50] di mana terjadi pembelajaran tidak konvergen, proses pencarian struktur berhenti setelah mencapai iterasi maksimum, sedangkan pada proporsi [30,20,50] dan [40,10,50], proses pencarian struktur dapat saja berhenti sebelum iterasi maksimum terjadi karena telah didapatkan *RR_tr* dan *RR_trs* 100%. Waktu *testing* untuk ketiga proporsi data hampir sama karena banyak neuron aktif yang didapatkan juga hampir sama, yaitu sekitar 10-12 neuron. Tingkat pengenalan yang didapatkan pun mencapai lebih dari 90% dengan kisaran 93.6-96.8%.

Tabel 4.1. Hasil Percobaan 2A data IR-TNI

Proporsi		<i>RR_tr</i>	<i>RR_trs</i>	<i>RR_ts</i>	Waktu Training	Waktu Testing	Neuron Aktif	
[20,30,50]	MIN	100.00	98.33	89.00	190.8167	0.1059	11	Neuron max = 40
	MAX	100.00	100.00	95.00	278.3533	0.1320	13	
	AVE	100.00	98.66	93.60	243.0792	0.1211	12.6	31.50%
[30,20,50]	MIN	100.00	100.00	95.00	12.4279	0.0993	10	Neuron Max = 60
	MAX	100.00	100.00	96.00	18.8688	0.1220	10	
	AVE	100.00	100.00	95.80	15.7034	0.1104	10	16.67%
[40,10,50]	MIN	100.00	100.00	96.00	17.1308	0.1058	10	Neuron Max = 80
	MAX	100.00	100.00	97.00	20.4177	0.1200	10	
	AVE	100.00	100.00	96.80	18.5914	0.1103	10	12.50%

Tabel 4.2. Nilai *smoothing parameter* dan struktur neuron optimal Percobaan 2A data IR-TNI

Proporsi	σ	Struktur Neuron											<i>RR_ts</i>	
		I	II	III	IV	V	VI	VII	VIII	IX	X	Jumlah		
[20,30,50]	2.582	1	1	2	1	1	1	1	1	1	1	1	11	89.00
	65.524	1	1	4	1	1	1	1	1	1	1	1	13	95.00
	15.057	1	1	4	1	1	1	1	1	1	1	1	13	94.00
	27.408	1	1	4	1	1	1	1	1	1	1	1	13	95.00
	74.502	1	1	4	1	1	1	1	1	1	1	1	13	95.00
[30,20,50]	57.118	1	1	1	1	1	1	1	1	1	1	1	10	96.00
	10.675	1	1	1	1	1	1	1	1	1	1	1	10	95.00
	27.663	1	1	1	1	1	1	1	1	1	1	1	10	96.00
	86.426	1	1	1	1	1	1	1	1	1	1	1	10	96.00
	16.563	1	1	1	1	1	1	1	1	1	1	1	10	96.00
[40,10,50]	74.272	1	1	1	1	1	1	1	1	1	1	1	10	97.00
	63.367	1	1	1	1	1	1	1	1	1	1	1	10	97.00
	85.630	1	1	1	1	1	1	1	1	1	1	1	10	97.00
	87.920	1	1	1	1	1	1	1	1	1	1	1	10	97.00
	24.687	1	1	1	1	1	1	1	1	1	1	1	10	96.00











Tabel 4.2. memperlihatkan struktur neuron, nilai *smoothing parameter*, dan tingkat pengenalan yang didapatkan dari Percobaan 2A data IR-TNI. Nilai *smoothing parameter* yang didapatkan berbeda cukup jauh. Hal ini menunjukkan bahwa nilai *smoothing parameter* optimal tidak berada dalam sebuah titik tertentu atau berada dalam *range* kecil pada nilai tertentu, melainkan berada dalam *range* nilai yang cukup besar. Pada proporsi [20,30,50], struktur neuron yang didapatkan bahkan tidak selalu sama. Faktor yang dapat menyebabkan struktur neuron yang

berbeda ini adalah nilai *smoothing parameter* karena nilai *smoothing parameter* dan struktur neuron saling terkait. Untuk data IR-TNI, struktur neuron yang sama bahkan tidak selalu memberikan tingkat pengenalan yang sama. Perbedaan ini dapat disebabkan oleh hasil pengurutan neuron metode OA yang berbeda untuk nilai *smoothing parameter* yang berbeda. Untuk membuktikan hal ini, akan diambil pasangan struktur neuron dan nilai *smoothing parameter* sebagai sampel :

a. Proporsi [30,20,50]

Nilai *smoothing parameter* (σ) : 57.118











Tabel 4.3. Urutan dan neuron aktif Percobaan 2A data IR-TNI [30,20,50], $\sigma = 57.118$

	I	II	III	IV	V	VI	VII	VIII	IX	X
Struktur neuron	1	1	1	1	1	1	1	1	1	1
Urutan neuron hasil OA	[6,1, 3,5, 2,4]	[5,3, 6,1, 2,4]	[4,2, 5,1, 3,6]	[5,6, 3,1, 4,2]	[3,4, 6,1, 5,2]	[1,6, 3,5, 4,2]	[3,1, 6,5, 4,2]	[4,3, 5,6, 1,2]	[2,5, 6,4, 1,3]	[5,6, 4,1, 2,3]
Neuron aktif	6 	5 	4 	5 	3 	1 	3 	4 	2 	5 

b. Proporsi [30,20,50]

Nilai *smoothing parameter* (σ) : 10.675

Tabel 4.4. Urutan dan neuron aktif Percobaan 2A data IR-TNI [30,20,50], $\sigma = 10.675$

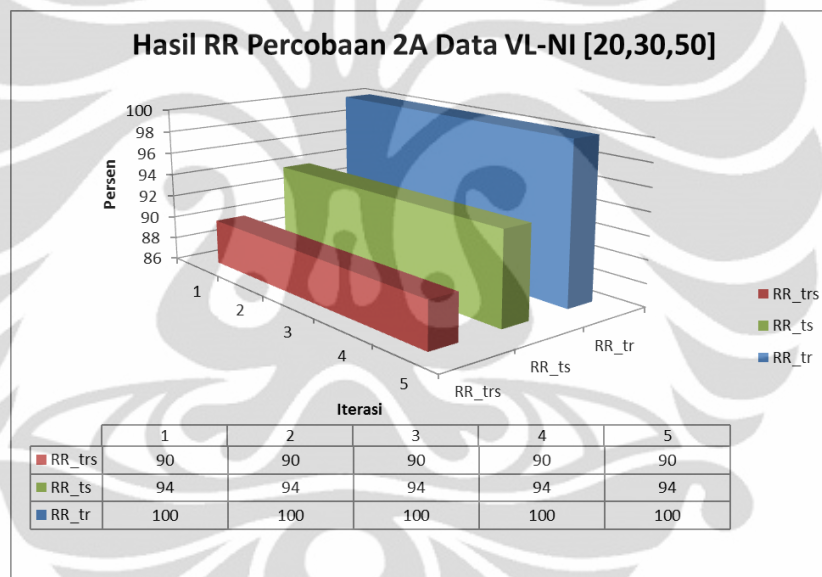
	I	II	III	IV	V	VI	VII	VIII	IX	X
Struktur neuron	1	1	1	1	1	1	1	1	1	1
Urutan neuron hasil OA	[4,3, 1,5, 2,6]	[4,2, 6,3, 1,5]	[6,3, 2,1, 4,5]	[4,6, 3,1, 5,2]	[3,4, 6,1, 5,2]	[1,6, 3,5, 4,2]	[2,1, 6,5, 3,4]	[2,1, 5,6, 4,3]	[2,5, 6,4, 1,3]	[3,6, 1,5, 2,4]
Neuron aktif	4 	4 	6 	4 	3 	1 	2 	2 	2 	3 

Tabel 4.3. dan Tabel 4.4. memperlihatkan bahwa untuk data IR_TNI, struktur neuron yang sama memberikan tingkat pengenalan yang berbeda karena neuron aktif pada struktur tersebut berbeda. Pemilihan neuron aktif yang berbeda disebabkan oleh hasil pengurutan neuron metode OA yang berbeda untuk nilai

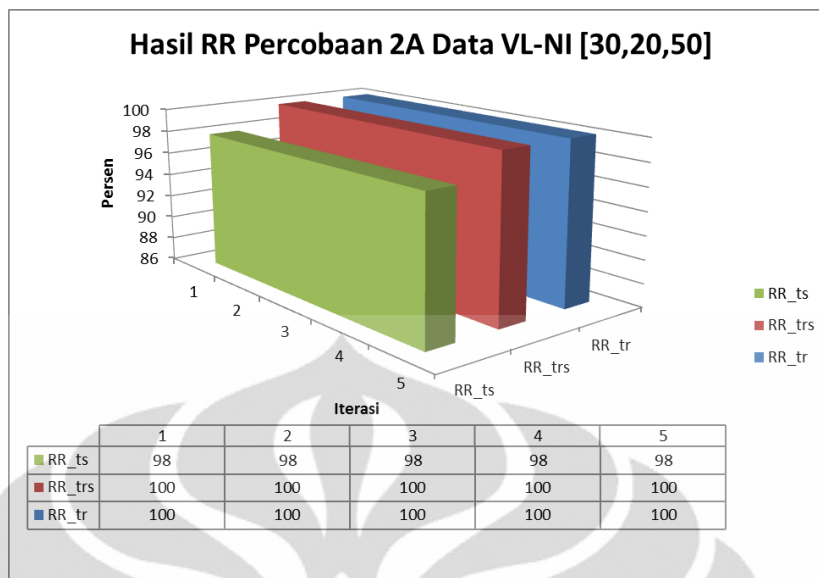
smoothing parameter yang berbeda. Terbukti bahwa struktur neuron yang sama memberikan tingkat pengenalan yang berbeda karena hasil pengurutan neuron yang berbeda.

4.2.2.2. Hasil dan Analisis Percobaan dengan Data VL-NI

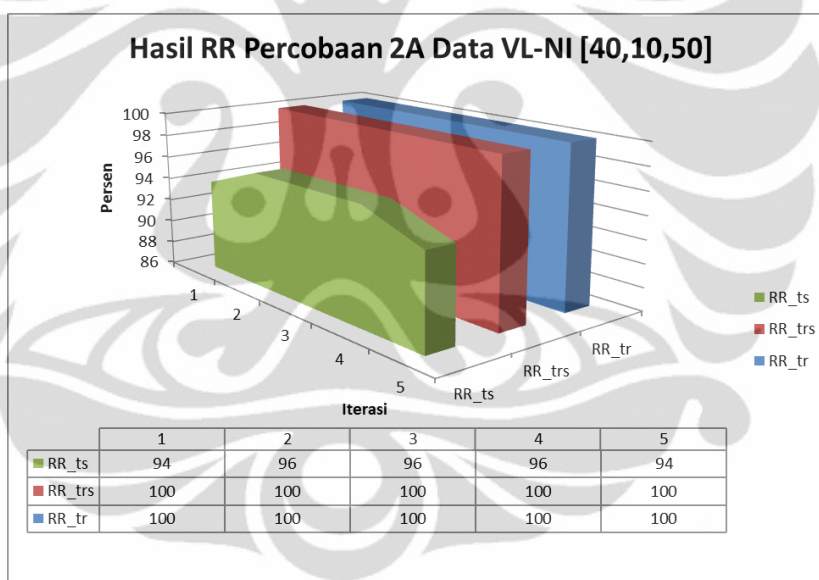
Dari 5 kali perulangan Percobaan 2A dengan data VL-NI, proporsi [20,30,50] dan [30,20,50] memiliki hasil rekognisi yang konsisten, sedangkan proporsi [40,10,50] tidak konsisten. Pembelajaran konvergen didapatkan dari proporsi [30,20,50] dan [40,10,50] dan pembelajaran tidak konvergen didapatkan dari proporsi [20,30,50].



Gambar 4.5. Hasil RR Percobaan 2A data VL-NI [20,30,50]



Gambar 4.6. Hasil RR Percobaan 2A data VL-NI [30,20,50]



Gambar 4.7. Hasil RR Percobaan 2A data VL-NI [40,10,50]

Seperti pada data IR-TNI, pembelajaran tidak konvergen memberikan waktu *training* yang lebih lama dibandingkan pembelajaran konvergen. Hal yang menyebabkan perbedaan waktu ini terdapat pada proses pencarian struktur neuron optimal, di mana pembelajaran tidak konvergen menyebabkan pencarian mencapai iterasi maksimum, sedangkan pembelajaran konvergen tidak perlu mencapai iterasi maksimum. Waktu *training* untuk pembelajaran tidak konvergen data VL-NI sekitar 133 detik dan pembelajaran konvergen sekitar 20-21 detik.

Waktu *testing* untuk ketiga proporsi data dapat dikatakan sama karena banyak neuron aktif yang didapatkan cenderung sama. Tingkat pengenalan yang didapatkan untuk ketiga jenis data berkisar antara 94-98% yang mana tergolong dalam pengenalan yang baik.

Tabel 4.5. Hasil Percobaan 2A data VL-NI

Proporsi		<i>RR_tr</i>	<i>RR_trs</i>	<i>RR_ts</i>	Waktu <i>Training</i>	Waktu <i>Testing</i>	Neuron Aktif	
[20,30,50]	MIN	100.00	90.00	94.00	102.4190	0.0661	13	Neuron max = 20
	MAX	100.00	90.00	94.00	151.7434	0.0832	13	
	AVE	100.00	90.00	94.00	133.0844	0.0752	13	65.00%
[30,20,50]	MIN	100.00	100.00	98.00	17.3428	0.0592	13	Neuron max = 30
	MAX	100.00	100.00	98.00	22.8179	0.0750	14	
	AVE	100.00	100.00	98.00	20.5300	0.0682	13.8	46.00%
[40,10,50]	MIN	100.00	100.00	94.00	18.5516	0.0597	11	Neuron max = 40
	MAX	100.00	100.00	96.00	23.8384	0.0891	12	
	AVE	100.00	100.00	95.20	21.8509	0.0694	11.6	29.00%

Tabel 4.6. Nilai *smoothing parameter* dan struktur neuron optimal Percobaan 2A data VL-NI

Proporsi	σ	Struktur Neuron											<i>RR_ts</i>
		I	II	III	IV	V	VI	VII	VIII	IX	X	Jumlah	
[20,30,50]	43.477	1	2	1	1	1	2	1	1	2	1	13	94.00
	69.985	1	2	1	1	1	2	1	1	2	1	13	94.00
	82.722	1	2	1	1	1	2	1	1	2	1	13	94.00
	38.155	1	2	1	1	1	2	1	1	2	1	13	94.00
	9.373	1	2	1	1	1	2	1	1	2	1	13	94.00
[30,20,50]	83.801	1	2	1	1	1	2	1	1	3	1	14	98.00
	33.038	1	2	1	1	1	2	1	1	3	1	14	98.00
	76.773	1	2	1	1	1	2	1	1	3	1	14	98.00
	88.727	1	2	1	1	1	2	1	1	3	1	14	98.00
	7.485	1	2	1	1	1	2	1	1	2	1	13	98.00
[40,10,50]	64.546	1	3	1	1	1	1	1	1	1	1	12	94.00
	38.268	1	3	1	1	1	1	1	1	1	1	12	96.00
	99.062	1	3	1	1	1	1	1	1	1	1	12	96.00
	22.882	1	2	1	1	1	1	1	1	1	1	11	96.00
	12.358	1	2	1	1	1	1	1	1	1	1	11	94.00











Tabel 4.6. memperlihatkan struktur neuron, nilai *smoothing parameter*, dan tingkat pengenalan yang didapatkan dari Percobaan 2A data VL-NI. Pada proporsi

data yang memberikan tingkat pengenalan yang konsisten, struktur neuron yang didapatkan pada setiap perulangan cenderung sama, namun dengan nilai *smoothing* parameter yang berbeda. Hal ini membuktikan bahwa nilai *smoothing parameter* optimal terletak pada *range* tertentu, bukan hanya terletak pada suatu titik atau sekitarnya. Seperti pada data IR-TNI, struktur yang sama tidak selalu memberikan tingkat pengenalan yang sama. Untuk itu akan diambil pasangan struktur neuron dan nilai *smoothing parameter* untuk melihat kecenderungan sifat yang dimiliki untuk data VL-NI :

a. Proporsi [40,10,50]

Nilai *smoothing parameter* (σ) : 22.882











Tabel 4.7. Urutan dan neuron aktif Percobaan 2A data VL-NI [40,10,50], $\sigma = 22.882$

	I	II	III	IV	V	VI	VII	VIII	IX	X
Struktur neuron	1	2	1	1	1	1	1	1	1	1
Urutan neuron hasil OA	[1,4, 2,3]	[2,4, 1,3]	[2,4, 3,1]	[3,1, 4,2]	[1,4, 3,2]	[2,3, 1,4]	[2,3, 4,1]	[3,4, 2,1]	[2,4, 3,1]	[4,3, 1,2]
Neuron aktif	1 	2,4 	2 	3 	1 	2 	2 	3 	2 	4 

b. Proporsi [40,10,50]

Nilai *smoothing parameter* (σ) : 12.358

Tabel 4.8. Urutan dan neuron aktif Percobaan 2A data VL-NI [40,10,50], $\sigma = 12.358$

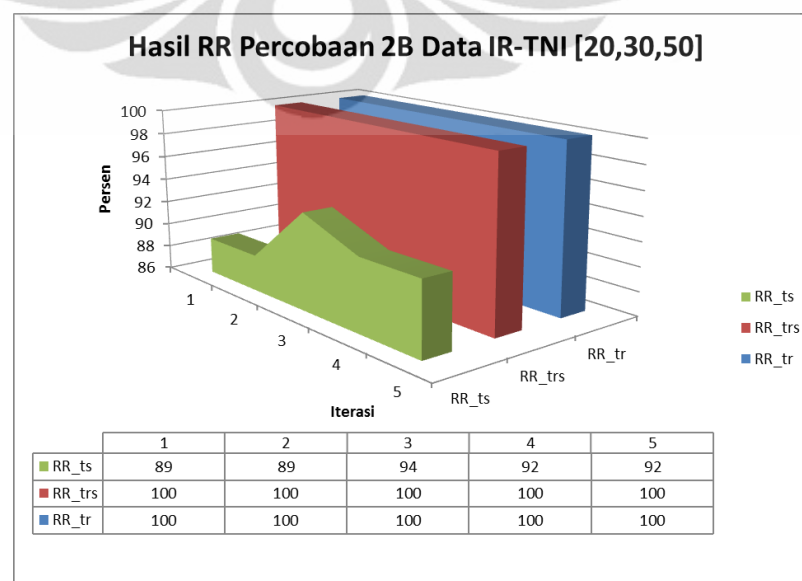
	I	II	III	IV	V	VI	VII	VIII	IX	X
Struktur neuron	1	2	1	1	1	1	1	1	1	1
Urutan neuron hasil OA	[1,4, 2,3]	[2,4, 3,1]	[2,4, 3,1]	[3,1, 4,2]	[1,4, 3,2]	[2,3, 1,4]	[1,3, 4,2]	[3,4, 2,1]	[2,4, 3,1]	[4,3, 1,2]
Neuron aktif	1 	2,4 	2 	3 	1 	2 	1 	3 	2 	4 

Pada data VL-NI, hasil pengurutan neuron metode OA ternyata hampir sama untuk nilai *smoothing parameter* yang berbeda. Sifat ini membuka kesempatan untuk mendapatkan neuron aktif yang sama walaupun menggunakan nilai *smoothing parameter* yang berbeda. Bila didapatkan neuron aktif yang sama pada struktur neuron, tingkat pengenalan hanya akan dibedakan oleh nilai *smoothing parameter* pada perhitungan tingkat kecocokan. Dapat dikatakan bahwa pada data VL-NI, struktur neuron yang sama akan cenderung memiliki neuron aktif terpilih yang sama walaupun nilai *smoothing parameter* yang digunakan untuk mengurutkan neuron berbeda.

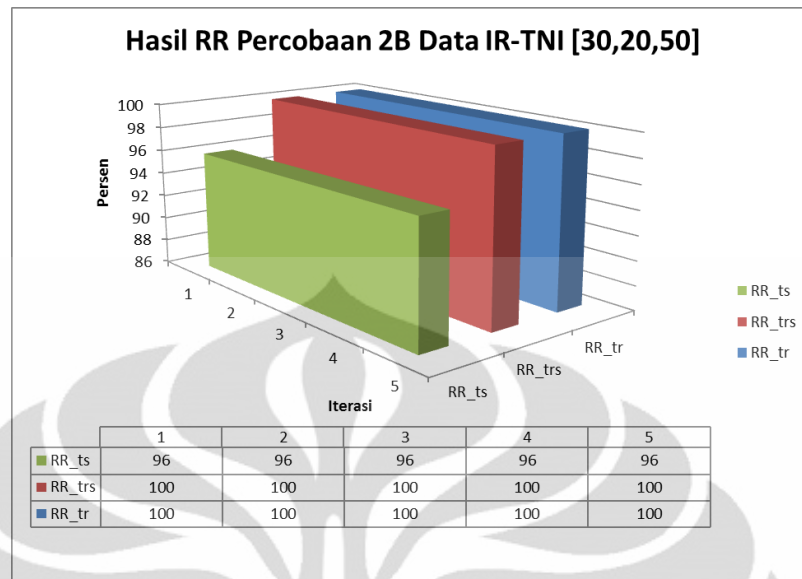
4.2.3. Teknik Optimasi PNN Berorientasi Nilai Smoothing Parameter dengan Parameter GA Modifikasi Jurnal

4.2.3.1. Hasil dan Analisis Percobaan dengan Data IR-TNI

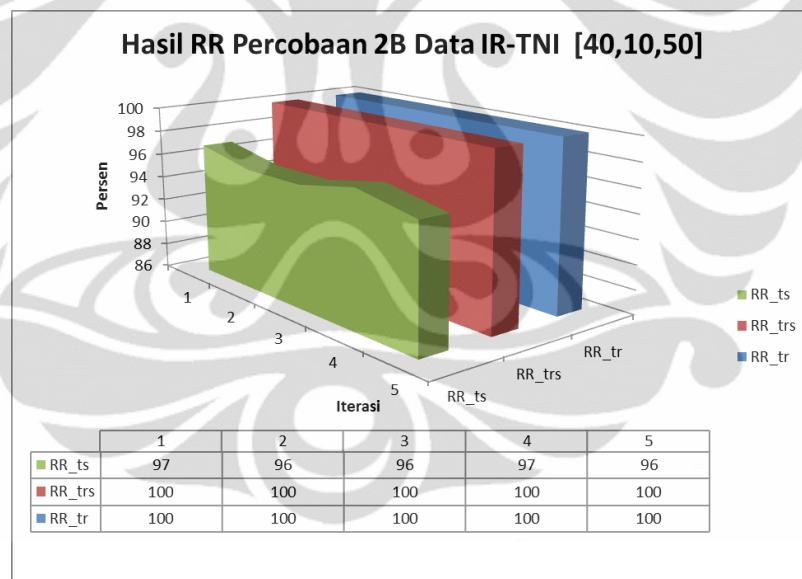
Berdasarkan perulangan yang dilakukan dengan proporsi data [20,30,50]; [30,20,50]; dan [40,10,50] masing-masing sebanyak 5 kali, didapatkan RR_{tr} , RR_{trs} , dan RR_{ts} yang konsisten untuk proporsi [30,20,50], namun tidak konsisten untuk proporsi [20,30,50] dan [40,10,50]. Pembelajaran untuk semua proporsi data IR-TNI merupakan pembelajaran yang konvergen karena mencapai RR_{tr} dan RR_{trs} 100%. Pembelajaran konvergen dengan parameter GA modifikasi jurnal mengakibatkan waktu *training* yang dibutuhkan singkat.



Gambar 4.8. Hasil RR Percobaan 2B data IR-TNI [20,30,50]



Gambar 4.9. Hasil RR Percobaan 2B data IR-TNI [30,20,50]



Gambar 4.10. Hasil RR Percobaan 2B data IR-TNI [40,10,50]

Dapat dilihat pada Tabel 4.9., waktu *training* yang dibutuhkan relatif cukup cepat karena pembelajaran yang konvergen dan waktu *testing* untuk semua proporsi relatif sama karena banyak neuron aktif pada struktur neuron yang didapatkan hampir sama dengan terdiri dari 10-12 neuron. Walaupun seluruh pembelajaran konvergen, namun waktu *training* yang dibutuhkan oleh proporsi [20,30,50] berbeda dengan proporsi lainnya. Faktor yang paling mungkin menyebabkan perbedaan ini adalah nilai *smoothing parameter* optimal. Proporsi

[20,30,50] membutuhkan waktu *training* yang lebih lama menandakan *range* nilai *smoothing parameter* optimalnya lebih sempit sehingga lebih sulit untuk mendapatkan nilai yang sesuai pada proses GA dibandingkan proporsi yang memiliki *range* yang lebar. Tingkat pengenalan dengan struktur neuron dan nilai *smoothing parameter* yang didapatkan juga memperlihatkan hasil yang baik, berkisar antara 91.2-96.4%.

Tabel 4.9. Hasil Percobaan 2B data IR-TNI

Proporsi		<i>RR_tr</i>	<i>RR_trs</i>	<i>RR_ts</i>	Waktu <i>Training</i>	Waktu <i>Testing</i>	Neuron Aktif	
[20,30,50]	MIN	100.00	100.00	89.00	7.3896	0.0815	11	Neuron max = 40
	MAX	100.00	100.00	94.00	41.9623	0.0854	12	
	AVE	100.00	100.00	91.20	21.3755	0.0832	11.2	28.00%
[30,20,50]	MIN	100.00	100.00	96.00	0.4835	0.0756	10	Neuron max = 60
	MAX	100.00	100.00	96.00	1.9801	0.0795	10	
	AVE	100.00	100.00	96.00	0.8211	0.0769	10	16.67%
[40,10,50]	MIN	100.00	100.00	96.00	0.5799	0.0759	10	Neuron max = 80
	MAX	100.00	100.00	97.00	0.5830	0.0777	10	
	AVE	100.00	100.00	96.40	0.5811	0.0764	10	12.50%

Tabel 4.10. Nilai *smoothing parameter* dan struktur neuron optimal Percobaan 2B data IR-TNI

Proporsi	σ	Struktur Neuron											<i>RR_ts</i>
		I	II	III	IV	V	VI	VII	VIII	IX	X	Jumlah	
[20,30,50]	2.633	1	1	2	1	1	1	1	1	1	1	11	89.00
	2.620	1	1	2	1	1	1	1	1	1	1	11	89.00
	4.358	1	1	3	1	1	1	1	1	1	1	12	94.00
	2.318	1	1	2	1	1	1	1	1	1	1	11	92.00
	2.382	1	1	2	1	1	1	1	1	1	1	11	92.00
[30,20,50]	89.883	1	1	1	1	1	1	1	1	1	1	10	96.00
	13.933	1	1	1	1	1	1	1	1	1	1	10	96.00
	81.102	1	1	1	1	1	1	1	1	1	1	10	96.00
	79.011	1	1	1	1	1	1	1	1	1	1	10	96.00
	47.814	1	1	1	1	1	1	1	1	1	1	10	96.00
[40,10,50]	62.115	1	1	1	1	1	1	1	1	1	1	10	97.00
	3.194	1	1	1	1	1	1	1	1	1	1	10	96.00
	31.582	1	1	1	1	1	1	1	1	1	1	10	96.00
	56.965	1	1	1	1	1	1	1	1	1	1	10	97.00
	43.717	1	1	1	1	1	1	1	1	1	1	10	96.00











Tabel 4.10. berisi struktur neuron, nilai *smoothing parameter*, dan tingkat pengenalan yang didapatkan dari Percobaan 2B data IR-TNI yang dilakukan sebanyak 5 kali. Dapat dilihat pada proporsi [20,30,50], struktur neuron, nilai *smoothing parameter*, dan tingkat pengenalan yang didapatkan berbeda-beda walaupun setiap perulangan berhasil mencapai pembelajaran konvergen. Seperti yang telah dipaparkan sebelumnya bahwa nilai *smoothing parameter* optimal proporsi ini berada pada *range* yang sempit sehingga didapatkan nilai yang saling berdekatan. Nilai-nilai *smoothing parameter* ini ternyata memberikan struktur neuron dan tingkat pengenalan yang bervariasi. Oleh karena itu dapat dikatakan bahwa untuk kasus ini, teknik optimasi yang digunakan sulit untuk mendapatkan pasangan struktur neuron dan nilai *smoothing parameter* yang terbaik karena seluruhnya berhasil mencapai konvergen. Hal yang sama juga berlaku bagi proporsi lainnya di mana tingkat pengenalan yang didapatkan tidak selalu sama untuk setiap perulangan walaupun selisihnya tidak besar, bedanya dengan proporsi [20,30,50] terletak pada nilai *smoothing parameter* yang didapatkan mempunyai *range* yang lebih lebar dan struktur neuron yang didapatkan seragam.

Struktur neuron dan nilai *smoothing parameter* yang diambil sebagai sampel adalah sebagai berikut :

a. Proporsi [40,10,50]

Nilai *smoothing parameter* (σ) : 62.115











Tabel 4.11. Urutan dan neuron aktif Percobaan 2B data IR-TNI [40,10,50], $\sigma = 62.115$

	I	II	III	IV	V	VI	VII	VIII	IX	X
Struktur neuron	1	1	1	1	1	1	1	1	1	1
Urutan neuron hasil OA	[6,8, 1,5, 3,7, 2,4]	[4,2, 8,3, 6,1, 7,5]	[4,2, 8,1, 7,5, 3,6]	[5,7, 1,3, 8,6, 2,4]	[3,8, 4,6, 1,5, 2,7]	[2,8, 3,5, 6,7, 1,4]	[3,8, 1,5, 4,6, 7,2]	[6,2, 1,5, 8,4, 7,3]	[2,8, 3,7, 4,6, 1,5]	[3,6, 1,8, 7,5, 2,4]
Neuron aktif	6 	4 	4 	5 	3 	2 	3 	6 	2 	3 

b. Proporsi [40,10,50]

Nilai *smoothing parameter* (σ) : 3.194

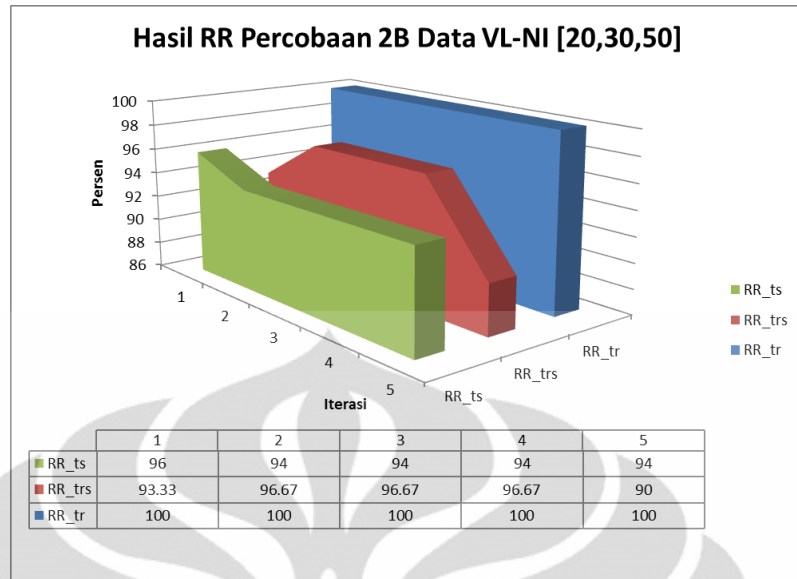
Tabel 4.12. Urutan dan neuron aktif Percobaan 2B data IR-TNI [40,10,50], $\sigma = 3.194$

	I	II	III	IV	V	VI	VII	VIII	IX	X
Struktur neuron	1	1	1	1	1	1	1	1	1	1
Urutan neuron hasil OA	[4,7,3,1,5,8,2,6]	[7,5,3,2,1,4,8,6]	[6,4,1,2,7,8,3,5]	[4,1,6,3,8,7,2,5]	[7,3,1,4,6,2,5,8]	[2,7,5,3,8,1,6,4]	[2,7,1,6,5,8,3,4]	[8,3,4,5,1,6,2,7]	[5,2,8,6,7,1,4,3]	[4,2,5,6,7,8,1,3]
Neuron aktif	4 	7 	6 	4 	7 	2 	2 	8 	5 	4 

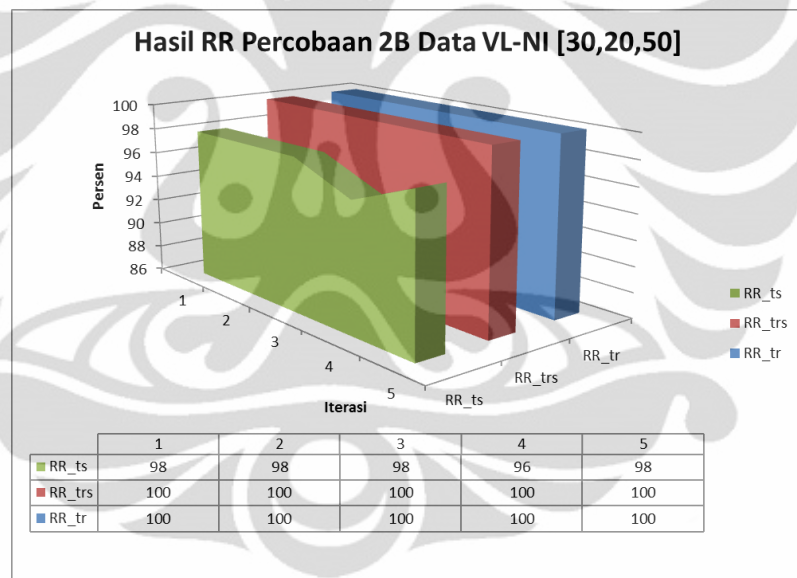
Tabel 4.11. dan Tabel 4.12. memperlihatkan kecenderungan yang dimiliki data IR-TNI terhadap teknik optimasi PNN berorientasi nilai *smoothing parameter* di mana struktur yang sama dapat memiliki neuron-neuron aktif yang berbeda karena pengaruh dari nilai *smoothing parameter* yang digunakan. Akibatnya, tingkat pengenalan yang didapatkan pun dapat berbeda. Hal inilah yang menyebabkan tidak konsistennya hasil yang didapatkan pada percobaan ini.

4.2.3.2. Hasil dan Analisis Percobaan dengan Data VL-NI

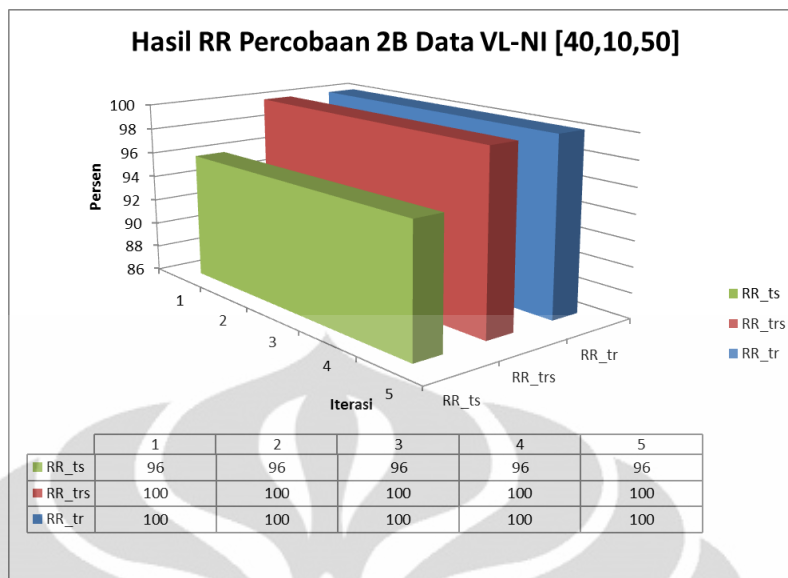
Dari hasil 5 kali perulangan, hasil rekognisi proporsi [20,30,50] dan [30,20,50] mempunyai nilai yang tidak konsisten, sedangkan proporsi [40,10,50] mempunyai nilai yang konsisten. Faktor yang dapat mempengaruhi konsistensi hasil rekognisi ini tentunya adalah proporsi data, struktur neuron, dan nilai *smoothing parameter* yang didapatkan. Pembelajaran untuk Percobaan 2B dengan data VL-NI merupakan pembelajaran tidak konvergen untuk proporsi [20,30,50] karena *RR_trs* maksimum yang didapatkan hanya 96.67%, namun merupakan pembelajaran konvergen untuk proporsi [30,20,50] dan [40,10,50].



Gambar 4.11. Hasil RR Percobaan 2B data VL-NI [20,30,50]



Gambar 4.12. Hasil RR Percobaan 2B data VL-NI [30,20,50]



Gambar 4.13. Hasil RR Percobaan 2B data VL-NI [40,10,50]

Tabel 4.13. menunjukkan hasil Percobaan 2 dengan data VL-NI. Waktu *training* untuk proporsi [20,30,50] cukup lama karena pembelajaran tidak konvergen, sedangkan untuk proporsi [30,20,50] dan [40,10,50] sangat singkat dengan rata-rata 0.9253 sekon dan 0.8508 sekon karena memiliki pembelajaran konvergen. Waktu *testing* untuk ketiga proporsi ini relatif hampir sama karena banyak neuron aktif yang tidak berbeda jauh. Secara total hanya digunakan sekitar 13 neuron untuk mewakili seluruh kelas yang ada. Tingkat pengenalan yang didapatkan pun tergolong baik dengan berkisar antara 94-97%.

Tabel 4.13. Hasil Percobaan 2B data VL-NI

Proporsi		RR_tr	RR_trs	RR_ts	Waktu Training	Waktu Testing	Neuron Aktif	
[20,30,50]	MIN	100.00	90.00	94.00	97.5952	0.0613	13	Neuron max = 20
	MAX	100.00	96.67	96.00	112.6489	0.0766	14	
	AVE	100.00	94.67	94.40	103.1578	0.0681	13.2	
[30,20,50]	MIN	100.00	100.00	96.00	0.7726	0.0570	13	Neuron max = 30
	MAX	100.00	100.00	98.00	1.3199	0.0632	15	
	AVE	100.00	100.00	97.60	0.9253	0.0608	14	
[40,10,50]	MIN	100.00	100.00	96.00	0.7882	0.0551	11	Neuron max = 40
	MAX	100.00	100.00	96.00	0.9329	0.0611	12	
	AVE	100.00	100.00	96.00	0.8508	0.0581	11.8	

Tabel 4.14. Nilai *smoothing parameter* dan struktur neuron optimal Percobaan 2B data VL-NI











Proporsi	σ	Struktur Neuron										RR _{ts}	
		I	II	III	IV	V	VI	VII	VIII	IX	X		Jumlah
[20,30,50]	0.716	1	2	1	2	1	2	1	1	2	1	14	96.00
	3.337	1	2	1	1	1	2	1	1	2	1	13	94.00
	3.235	1	2	1	1	1	2	1	1	2	1	13	94.00
	2.861	1	2	1	1	1	2	1	1	2	1	13	94.00
	47.550	1	2	1	1	1	2	1	1	2	1	13	94.00
[30,20,50]	75.500	1	2	1	1	1	2	1	1	3	1	14	98.00
	11.366	1	2	1	1	1	2	1	1	2	1	13	98.00
	59.863	1	2	1	1	1	2	1	1	3	1	14	98.00
	2.978	1	3	1	1	1	2	1	1	3	1	15	96.00
	26.000	1	2	1	1	1	2	1	1	3	1	14	98.00
[40,10,50]	23.325	1	2	1	1	1	1	1	1	1	1	11	96.00
	57.537	1	3	1	1	1	1	1	1	1	1	12	96.00
	62.743	1	3	1	1	1	1	1	1	1	1	12	96.00
	42.864	1	3	1	1	1	1	1	1	1	1	12	96.00
	90.144	1	3	1	1	1	1	1	1	1	1	12	96.00

Struktur neuron, nilai *smoothing parameter*, dan tingkat pengenalan yang didapatkan dari Percobaan 2B dengan data VL-NI ditunjukkan pada Tabel 4.7.. Struktur neuron dan nilai *smoothing* yang didapatkan tidak selalu sama untuk setiap perulangannya. Akibatnya, tingkat pengenalan yang didapatkan pun bervariasi. Dapat dilihat dalam tabel bahwa untuk data VL-NI, struktur neuron yang sama cenderung memberikan tingkat pengenalan yang sama walaupun nilai *smoothing parameter* berbeda. Sifat ini juga menyebabkan tingkat pengenalan yang didapatkan relatif lebih konsisten dibandingkan data IR-TNI. Struktur neuron dan nilai *smoothing parameter* yang diambil sebagai sampel adalah sebagai berikut :

- a. Proporsi [30,20,50]

Nilai *smoothing parameter* (σ) : 75.500











Tabel 4.15. Urutan dan neuron aktif Percobaan 2B data VL-NI [30,20,50], $\sigma = 75.500$

	I	II	III	IV	V	VI	VII	VIII	IX	X
Struktur neuron	1	2	1	1	1	2	1	1	3	1
Urutan neuron hasil OA	[1,2,3]	[2,3,1]	[2,3,1]	[3,1,2]	[2,3,1]	[1,2,3]	[2,3,1]	[3,2,1]	[2,3,1]	[1,3,2]
Neuron aktif	1 	2,3 	2 	3 	2 	1,2 	2 	3 	2,3,1 	1 

b. Proporsi [30,20,50]

Nilai *smoothing parameter* (σ) : 26.000

Tabel 4.16. Urutan dan neuron aktif Percobaan 2B data VL-NI [30,20,50], $\sigma = 26.000$

	I	II	III	IV	V	VI	VII	VIII	IX	X
Struktur neuron	1	2	1	1	1	2	1	1	3	1
Urutan neuron hasil OA	[1,2,3]	[2,3,1]	[2,3,1]	[3,1,2]	[2,3,1]	[1,2,3]	[2,3,1]	[3,2,1]	[2,3,1]	[1,3,2]
Neuron aktif	1 	2,3 	2 	3 	2 	1,2 	2 	3 	2,3,1 	1 

Alasan kedua pasangan ini diambil sebagai sampel adalah keduanya memiliki struktur neuron yang sama, namun nilai *smoothing parameter* yang berbeda jauh. *Smoothing parameter* yang satu bernilai 75.500, sedangkan yang lainnya bernilai 26.000. Pada data IR-TNI, nilai *smoothing parameter* yang berbeda jauh akan memiliki hasil pengurutan dan neuron aktif yang berbeda. Tampak pada Tabel 4.15. dan Tabel 4.16. bahwa nilai *smoothing parameter* yang berbeda jauh tersebut memberikan urutan neuron dan neuron aktif yang sama. Oleh karena itu, tingkat pengenalan tinggal dipengaruhi oleh nilai *smoothing parameter* pada

perhitungan tingkat kecocokan pada algoritma PNN. Dengan latar belakang ini, tingkat pengenalan data VL-NI cenderung sama untuk struktur neuron yang sama.

4.2.4. Perbandingan Teknik Optimasi PNN Berorientasi Nilai Smoothing Parameter dengan Parameter GA Menurut Jurnal dan Modifikasi Jurnal

Teknik optimasi PNN berorientasi nilai *smoothing parameter* dengan parameter GA menurut jurnal (Percobaan 2A) dan dengan parameter GA modifikasi jurnal (Percobaan 2B) memiliki kemampuan pengenalan yang hampir sama. Hal ini dapat dilihat dari tingkat pengenalan dan struktur neuron yang didapatkan dari kedua teknik optimasi ini hampir sama. Struktur neuron yang relatif sama menandakan banyak neuron aktif dari kedua teknik ini juga relatif sama sehingga waktu *testing* kedua teknik inipun hampir sama. Walaupun waktu *testing* data IR-TNI terdapat selisih perhitungan, hal ini disebabkan karena perbedaan kondisi perangkat keras pada saat dilakukan perhitungan pewaktuan. Kedua teknik optimasi ini memiliki kelemahan dalam hal konsistensi, di mana struktur neuron dan tingkat pengenalan yang diberikan untuk setiap perulangan berbeda-beda, namun keunggulan dari inkonsistensi ini adalah peluang yang lebih besar untuk mendapatkan tingkat pengenalan tertinggi secara global.

Perbedaan mendasar dari kedua teknik optimasi dengan parameter GA berbeda ini terletak pada waktu *training*. Parameter GA menurut jurnal mengharuskan pembelajaran berhenti pada generasi maksimum, sedangkan parameter GA modifikasi jurnal memungkinkan pembelajaran berhenti sebelum mencapai generasi maksimum, yaitu setelah berhasil mencapai nilai *fitness* maksimum. Hasil dari perbedaan stopping condition antara kedua parameter ini dapat dilihat pada Tabel 4.17., di mana parameter GA modifikasi jurnal memiliki waktu *training* yang lebih cepat untuk data IR-TNI maupun VL-NI. Nilai *fitness* maksimum dapat ditambahkan pada syarat stopping condition parameter GA modifikasi jurnal karena nilai *fitness* parameter GA ini berbasiskan hasil rekognisi *RR_tr* dan *RR_trs* yang dapat diketahui nilai *fitness* maksimumnya. Sedangkan, nilai *fitness* parameter GA menurut jurnal berbasiskan banyak neuron aktif pada struktur neuron yang mana tidak diketahui berapa banyak neuron yang optimal.

Tabel 4.17. Perbandingan waktu *training*, konsistensi, dan konvergensi Percobaan 2A dan 2B

Data	Proporsi	Menurut Jurnal			Modifikasi Jurnal		
		Waktu <i>Training</i>	Konsis- tensi	Konver- gensi	Waktu <i>Training</i>	Konsis- tensi	Konver- gensi
IR-TNI	[20,30,50]	243.0792	×	×	21.3755	×	✓
	[30,20,50]	15.7034	×	✓	0.8211	✓	✓
	[40,10,50]	18.59138	×	✓	0.5811	×	✓
VL-NI	[20,30,50]	133.0844	✓	×	103.1578	×	×
	[30,20,50]	20.53002	✓	✓	0.9253	×	✓
	[40,10,50]	21.85086	×	✓	0.8508	✓	✓

Percobaan 2A dan 2B juga memberikan gambaran mengenai perbedaan sifat penerapannya untuk data IR-TNI dan VL-NI. Dari hasil percobaan yang dilakukan pada data IR-TNI dan VL-NI dapat dilihat bahwa untuk struktur neuron yang sama dan nilai *smoothing parameter* yang berbeda, data IR-TNI cenderung memiliki tingkat pengenalan yang berbeda, sedangkan data VL-NI cenderung memiliki tingkat pengenalan yang sama. Hal ini disebabkan karena pada data IR-TNI, didapatkan hasil urutan neuron yang berbeda untuk nilai *smoothing parameter* yang berbeda, sedangkan pada data VL-NI, didapatkan hasil urutan neuron yang relatif hampir sama untuk nilai *smoothing parameter* yang berbeda. Perbedaan urutan pada data IR-TNI disebabkan karena foto-foto *infrared* mempunyai nilai-nilai yang berdekatan antarfoto, sedangkan foto-foto *visible light* mempunyai nilai-nilai yang cukup bervariasi antarfoto. Faktor ini yang menyebabkan urutan neuron yang didapatkan dapat berbeda untuk nilai *smoothing parameter* yang berbeda pada data IR-TNI.

4.3. Teknik Optimasi PNN Berorientasi Struktur Neuron

4.3.1. Prosedur Percobaan

Prosedur Percobaan 3 terdiri dari 2 tahap utama, yaitu tahap pembelajaran JST dan tahap pengetesan JST. Pada tahap pembelajaran JST, data yang digunakan adalah X_{tr} dan X_{trs} , sedangkan pada tahap pengetesan JST, seluruh data akan diikutsertakan. Tahap pembelajaran JST menggunakan metode pencarian nilai

smoothing parameter optimal dengan GA dan metode pencarian struktur neuron optimal dengan OA. Berikut ini adalah prosedur dalam melakukan Percobaan 3 :

a. Tahap Persiapan Data

Menyiapkan data percobaan berupa foto *infrared*, *visible light* dan mengolahnya hingga didapatkan data percobaan IR-TNI dan VL-NI yang masing-masing dibagi dengan proporsi [20,30,50]; [30,20,50]; dan [40,10,50].

b. Tahap Pembelajaran JST PNN

1. Perhitungan nilai *smoothing parameter* awal sesuai dengan persamaan :

$$\sigma_{awal} = \frac{d_{max}}{\sqrt{K}} \quad (4.1)$$

di mana :

d_{max} = jarak Euclidean maksimum antarvektor rata-rata tiap kelas

K = banyak kelas

2. Pengurutan neuron dengan OA menggunakan nilai *smoothing parameter* awal.
3. Pembentukan struktur neuron awal yang terdiri dari 1 neuron untuk setiap kelas, iterasi = 1.
4. Inisialisasi populasi awal secara *random* berisi 10 kromosom yang mewakili nilai *smoothing parameter*, generasi = 1.
5. Dengan struktur neuron tersebut dan setiap kromosom, hitung nilai *fitness* dengan cara melakukan perhitungan RR, kemudian gunakan RR_{tr} dan RR_{trs} yang didapatkan ke dalam fungsi *fitness*.
6. Bila *stopping condition* GA (generasi maksimum atau nilai *fitness* maksimum) belum terpenuhi, maka proses GA terus dilanjutkan (lanjut ke poin 7). Bila *stopping condition* terpenuhi, maka proses GA berhenti (lanjut ke poin 9).
7. Berdasarkan nilai *fitness*, dilakukan pemilihan dan pembentukan populasi orang tua dengan metode *Roulette Wheel*.
8. Pembentukan populasi anak sebagai populasi kromosom baru dengan melakukan reproduksi (*crossover*, *mutation*, dan *elitis*) terhadap

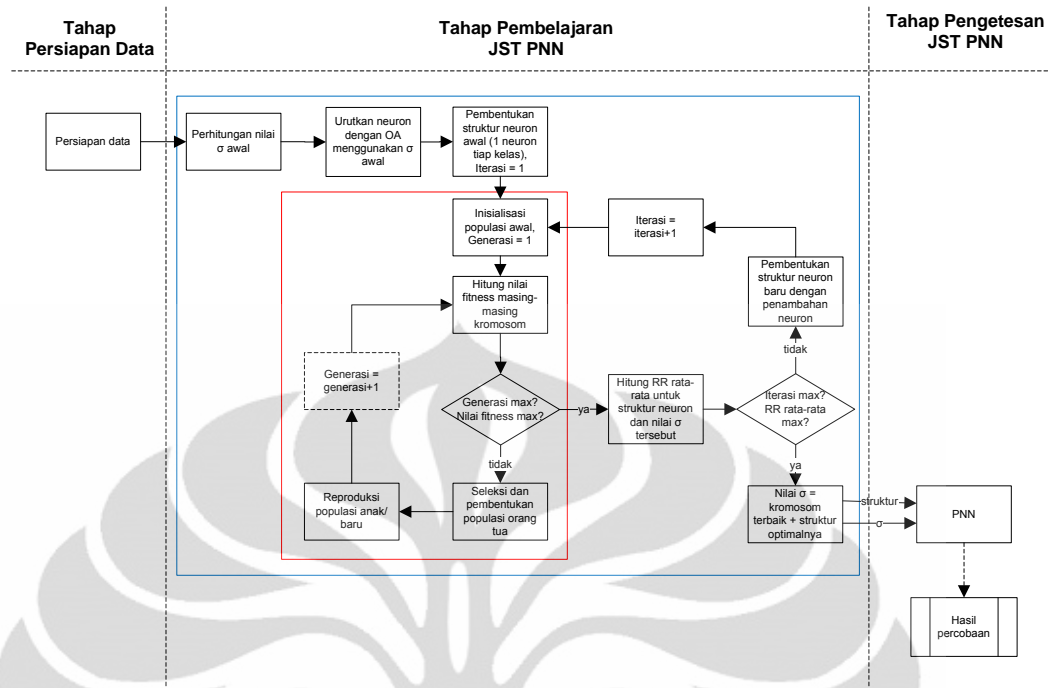
kromosom-kromosom orang tua yang telah dipilih, kemudian ulang kembali dari poin 5. $\text{Generasi} = \text{generasi} + 1$.

9. Hitung RR rata-rata antara RR_{tr} dan RR_{trs} dengan pasangan struktur neuron dan nilai *smoothing parameter* (kromosom) tersebut.
10. Bila *stopping condition* pencarian struktur neuron (iterasi maksimum atau RR rata-rata = 100) belum terpenuhi, proses pencarian struktur neuron dilanjutkan (lanjut ke poin 11). Bila *stopping condition* terpenuhi, maka proses pencarian struktur neuron untuk kromosom tersebut berhenti (langsung ke poin 12).
11. Pembentukan struktur neuron baru dengan menambahkan 1 neuron pada kelas yang mempunyai RR kelas rata-rata $< 100\%$ dan banyak neuron pada kelas tersebut $<$ banyak neuron maksimum pada kelas itu, kemudian ulang kembali dari poin 4. $\text{Iterasi} = \text{iterasi} + 1$.
12. Kromosom dan struktur neuron yang memiliki nilai *fitness* tertinggi merupakan kromosom dan struktur neuron terbaik (optimal).

c. Tahap Pengetesan JST PNN

1. Gunakan pasangan struktur neuron dan nilai *smoothing parameter* (kromosom) dari tahap pembelajaran yang memberikan RR terbaik.
2. Lakukan pengetesan pada JST PNN dengan nilai *smoothing parameter* optimal, dan struktur neuron tersebut.

Pada tahap pembelajaran JST, poin b.4.-b.8. merupakan metode pencarian nilai *smoothing parameter* optimal dengan GA, sedangkan poin b.1.-b.3., b.9.-b.12. merupakan metode pencarian struktur neuron optimal dengan OA. Prosedur percobaan ini dilakukan sebanyak 5 kali untuk masing-masing jenis dan proporsi data.

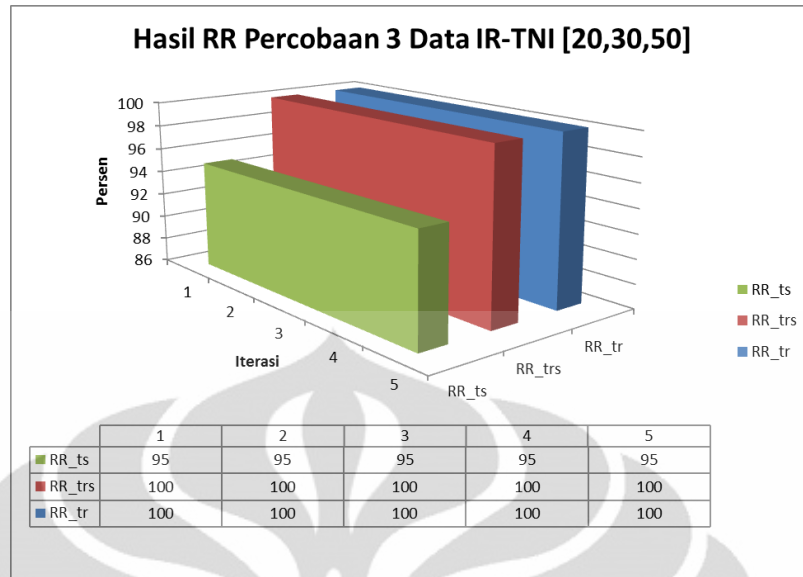


Gambar 4.14. Prosedur Percobaan 3

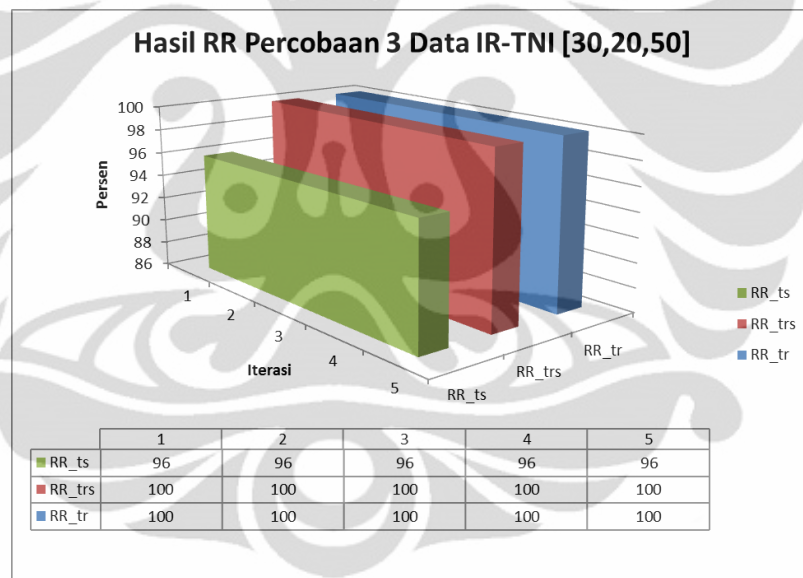
4.3.2. Hasil dan Analisis Percobaan

4.3.2.1. Hasil dan Analisis Percobaan dengan Data IR-TNI

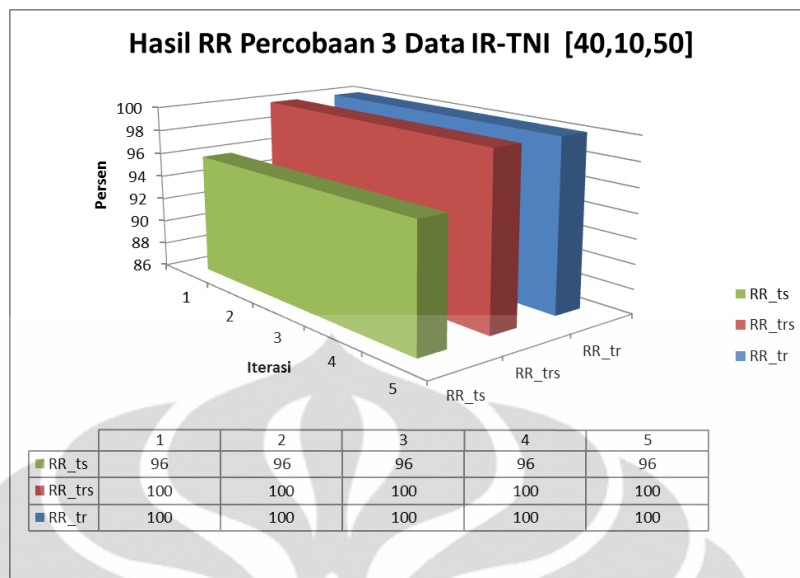
Hasil rekognisi yang didapatkan dari 5 kali perulangan Percobaan 3 dengan data IR-TNI menunjukkan nilai yang konsisten untuk semua proporsi data. Setiap proporsi data juga memiliki pembelajaran yang konvergen dengan RR_{tr} dan RR_{trs} yang mencapai 100%. Karena pembelajaran konvergen ini, waktu *training* yang didapatkan relatif lebih cepat.



Gambar 4.15. Hasil RR Percobaan 3 data IR-TNI [20,30,50]



Gambar 4.16. Hasil RR Percobaan 3 data IR-TNI [30,20,50]



Gambar 4.17. Hasil RR Percobaan 3 data IR-TNI [40,10,50]

Dapat dilihat pada

Tabel 4.18., waktu *training* untuk proporsi [20,30,50] lebih lama dari proporsi lainnya walaupun seluruh pembelajaran ini merupakan pembelajaran konvergen. Faktor yang paling mungkin menyebabkan perbedaan ini adalah nilai *smoothing parameter* optimal. Proporsi [20,30,50] membutuhkan waktu *training* yang lebih lama menandakan *range* nilai *smoothing parameter* optimalnya lebih sempit sehingga lebih sulit untuk mendapatkan nilai yang sesuai pada proses GA dibandingkan proporsi yang memiliki *range* yang lebar. Waktu *testing* untuk ketiga proporsi tidak berbeda jauh karena banyak neuron yang digunakan pada struktur neuron juga tidak berbeda jauh. Proporsi [20,30,50] menggunakan 12 neuron dari maksimum 40 neuron untuk mewakili seluruh kelas, proporsi [30,20,50] dan [40,10,50] cukup diwakili oleh 10 neuron dari maksimum 60 dan 80 neuron.

Tabel 4.18. Hasil Percobaan 3 data IR-TNI

Proporsi		RR_tr	RR_trs	RR_ts	Waktu Training	Waktu Testing	Neuron Aktif
[20,30,50]	MIN	100.00	100.00	95.00	13.2806	0.0811	12
	MAX	100.00	100.00	95.00	19.2869	0.0979	12
	AVE	100.00	100.00	95.00	14.6189	0.0851	12
[30,20,50]	MIN	100.00	100.00	96.00	0.3786	0.0751	10
	MAX	100.00	100.00	96.00	0.3808	0.0758	10

	AVE	100.00	100.00	96.00	0.3793	0.0754	10	16.67%
[40,10,50]	MIN	100.00	100.00	96.00	0.3874	0.0753	10	Neuron max = 80
	MAX	100.00	100.00	96.00	0.3900	0.0764	10	
	AVE	100.00	100.00	96.00	0.3887	0.0759	10	12.50%

Tabel 4.19. Nilai *smoothing parameter* dan struktur neuron optimal Percobaan 3 data IR-TNI

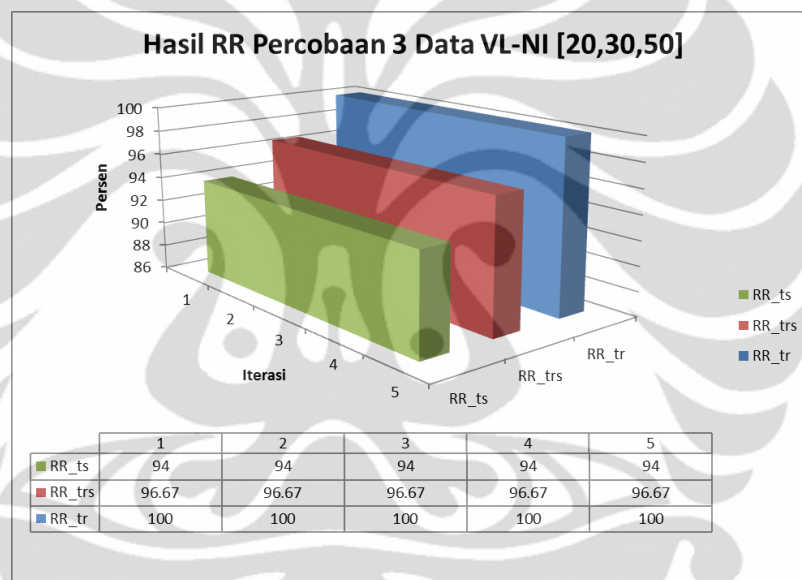
Proporsi	σ	Struktur Neuron										RR _{ts}		
		I	II	III	IV	V	VI	VII	VIII	IX	X		Jumlah	
[20,30,50]	3.801	1	1	3	1	1	1	1	1	1	1	1	12	95.00
	2.556	1	1	3	1	1	1	1	1	1	1	1	12	95.00
	4.391	1	1	3	1	1	1	1	1	1	1	1	12	95.00
	3.414	1	1	3	1	1	1	1	1	1	1	1	12	95.00
	4.615	1	1	3	1	1	1	1	1	1	1	1	12	95.00
[30,20,50]	42.094	1	1	1	1	1	1	1	1	1	1	1	10	96.00
	14.063	1	1	1	1	1	1	1	1	1	1	1	10	96.00
	35.212	1	1	1	1	1	1	1	1	1	1	1	10	96.00
	23.555	1	1	1	1	1	1	1	1	1	1	1	10	96.00
	19.920	1	1	1	1	1	1	1	1	1	1	1	10	96.00
[40,10,50]	20.168	1	1	1	1	1	1	1	1	1	1	1	10	96.00
	98.512	1	1	1	1	1	1	1	1	1	1	1	10	96.00
	63.361	1	1	1	1	1	1	1	1	1	1	1	10	96.00
	7.714	1	1	1	1	1	1	1	1	1	1	1	10	96.00
	62.092	1	1	1	1	1	1	1	1	1	1	1	10	96.00

Tabel 4.19. berisi struktur neuron, nilai *smoothing parameter*, dan tingkat pengenalan yang didapatkan dari Percobaan 3 data IR-TNI yang dilakukan sebanyak 5 kali. Terlihat dari tabel tersebut, struktur neuron dan tingkat pengenalan yang didapatkan sama, hanya nilai *smoothing parameter* yang berbeda untuk setiap perulangan. Pengurutan neuron dengan metode OA hanya dilakukan sekali pada Percobaan 3 sehingga hasil pengurutan neuron dengan nilai *smoothing parameter* awal akan digunakan untuk menentukan neuron aktif hingga percobaan selesai. Oleh karena itu, bila didapatkan struktur neuron yang sama dalam setiap perulangan, akan didapatkan neuron aktif yang sama pula. Dengan struktur neuron yang sama, selanjutnya, hasil rekognisi akan bergantung pada kesesuaian nilai *smoothing parameter* terhadap struktur neuron tersebut. Hal ini yang menjamin teknik optimasi berorientasi struktur neuron memberikan nilai yang relatif konsisten. Dilihat dari tingkat pengenalan, struktur neuron dan nilai *smoothing*

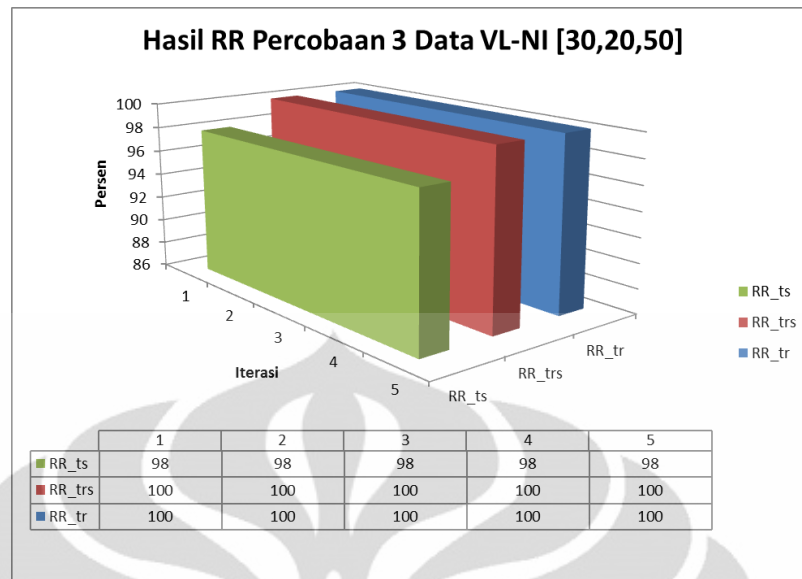
parameter yang didapatkan telah memberikan tingkat pengenalan yang cukup tinggi antara 95-96% pada semua proporsi data.

4.3.2.2. Hasil dan Analisis Percobaan dengan Data VL-NI

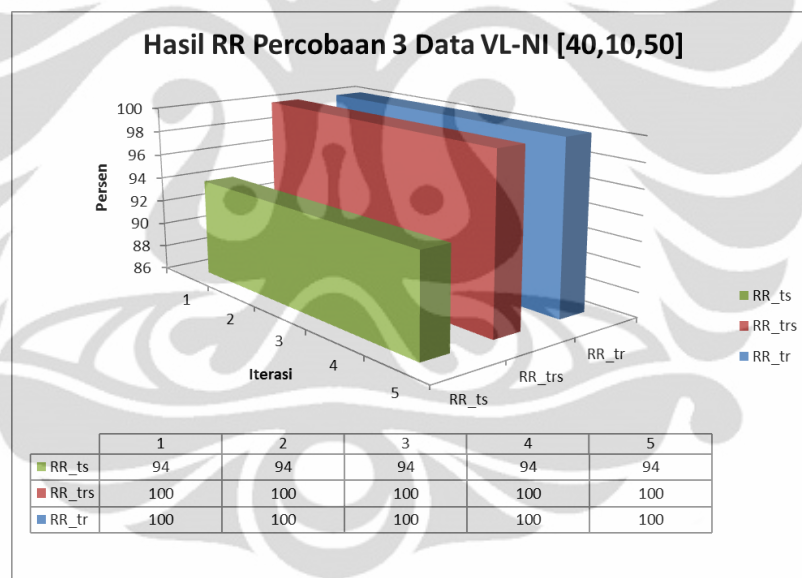
Percobaan 3 dengan data VL-NI yang dilakukan perulangan sebanyak 5 kali memberikan hasil rekognisi yang konsisten pada setiap perulangannya. Data VL-NI dengan proporsi [20,30,50] gagal mencapai $RR_{trs} = 100\%$ yang menandakan pembelajaran proporsi data ini tidak konvergen. Proporsi [30,20,50] dan [40,10,50] memiliki pembelajaran konvergen karena RR_{tr} dan RR_{trs} mencapai 100%.



Gambar 4.18. Hasil RR Percobaan 3 data VL-NI [20,30,50]



Gambar 4.19. Hasil RR Percobaan 3 data VL-NI [30,20,50]



Gambar 4.20. Hasil RR Percobaan 3 data VL-NI [40,10,50]

Waktu *training* rata-rata proporsi [20,30,50] sebesar 71.7029 sekon karena pembelajaran yang tidak konvergen. Waktu *training* proporsi [30,20,50] dan [40,10,50] tergolong cepat dengan rata-rata 3.8025 dan 3.5910 sekon. Waktu *testing* ketiga proporsi tidak berbeda jauh karena banyak neuron yang digunakan hanya berbeda sekitar 2 neuron. Banyak neuron yang digunakan untuk mewakili pola-pola kelas sebanyak 11-13 neuron. Proporsi [20,30,50] membutuhkan neuron yang lebih banyak, tetapi tidak mampu mencapai konvergen karena terbatasnya

data yang dapat dijadikan neuron (maksimal hanya 2 neuron tiap kelas) sehingga sulit mendapatkan neuron yang dapat mewakili kelas secara keseluruhan. Walaupun demikian, tingkat pengenalan yang diberikan pun tidak buruk karena masih berada pada tingkat pengenalan sekitar 94-98%.

Tabel 4.20. Hasil Percobaan 3 data VL-NI

Proporsi		<i>RR_tr</i>	<i>RR_trs</i>	<i>RR_ts</i>	Waktu <i>Training</i>	Waktu <i>Testing</i>	Neuron Aktif	
[20,30,50]	MIN	100.00	96.67	94.00	71.6348	0.0603	13	Neuron max = 20
	MAX	100.00	96.67	94.00	71.8219	0.0619	13	
	AVE	100.00	96.67	94.00	71.7029	0.0613	13	65.00%
[30,20,50]	MIN	100.00	100.00	98.00	3.5703	0.0566	13	Neuron max = 30
	MAX	100.00	100.00	98.00	4.4604	0.0606	13	
	AVE	100.00	100.00	98.00	3.8025	0.0582	13	43.33%
[40,10,50]	MIN	100.00	100.00	94.00	3.5590	0.0552	11	Neuron max = 40
	MAX	100.00	100.00	94.00	3.6725	0.0569	11	
	AVE	100.00	100.00	94.00	3.5910	0.0559	11	27.50%

Tabel 4.21. Nilai *smoothing parameter* dan struktur neuron optimal Percobaan 3 data VL-NI

Proporsi	σ	Struktur Neuron										<i>RR_ts</i>	
		I	II	III	IV	V	VI	VII	VIII	IX	X		Jumlah
[20,30,50] $\sigma_{awal} = 18.247$	0.834	1	2	1	1	1	2	1	1	2	1	13	94.00
	1.876	1	2	1	1	1	2	1	1	2	1	13	94.00
	1.608	1	2	1	1	1	2	1	1	2	1	13	94.00
	3.093	1	2	1	1	1	2	1	1	2	1	13	94.00
	3.191	1	2	1	1	1	2	1	1	2	1	13	94.00
[30,20,50] $\sigma_{awal} = 17.105$	10.542	1	2	1	1	1	2	1	1	2	1	13	98.00
	3.185	1	2	1	1	1	2	1	1	2	1	13	98.00
	3.407	1	2	1	1	1	2	1	1	2	1	13	98.00
	12.021	1	2	1	1	1	2	1	1	2	1	13	98.00
	4.458	1	2	1	1	1	2	1	1	2	1	13	98.00
[40,10,50] $\sigma_{awal} = 16.527$	15.499	1	2	1	1	1	1	1	1	1	1	11	94.00
	24.640	1	2	1	1	1	1	1	1	1	1	11	94.00
	14.245	1	2	1	1	1	1	1	1	1	1	11	94.00
	25.282	1	2	1	1	1	1	1	1	1	1	11	94.00
	24.389	1	2	1	1	1	1	1	1	1	1	11	94.00

Tabel 4.21. berisi struktur neuron, nilai *smoothing parameter*, dan tingkat pengenalan yang didapatkan dari Percobaan 3 data VL-NI yang dilakukan

sebanyak 5 kali. Untuk struktur neuron yang sama dan nilai *smoothing parameter* yang berbeda, didapatkan tingkat pengenalan sama pada setiap perulangan yang dilakukan (konsisten) karena dengan struktur neuron yang sama akan didapatkan susunan neuron aktif yang sama. Hal ini terjadi karena pemilihan neuron aktif didasarkan pada pengurutan neuron yang dilakukan hanya sekali pada awal percobaan. Namun, sifat ini hanya berlaku untuk proporsi yang sama. Proporsi yang berbeda akan memiliki hasil pengurutan neuron yang berbeda karena nilai *smoothing parameter* awal dan banyak neuron yang akan diurutkan juga berbeda. Oleh karena struktur neuron dan neuron aktif pada lapisan pola PNN telah sama, faktor yang menentukan tingkat pengenalan adalah kesesuaian nilai *smoothing parameter* terhadap struktur tersebut pada perhitungan tingkat kecocokan.

4.3.3. Analisis Teknik Optimasi PNN Berorientasi Struktur Neuron

Nilai *smoothing parameter* awal untuk teknik optimasi berorientasi struktur neuron tidak harus menggunakan nilai yang dihitung berdasarkan persamaan (4.1). Persamaan tersebut digunakan untuk menghitung nilai *smoothing parameter* dalam percobaan ini karena tidak diketahuinya nilai *smoothing parameter* yang tepat. Oleh karena itu, sebagai referensi awal, maka digunakanlah nilai sesuai persamaan (4.1). Bila telah diketahui nilai *smoothing parameter* yang bernilai baik, nilai *smoothing parameter* awal untuk proses pengurutan neuron dapat menggunakan nilai tersebut. Mengenai apakah hasil rekognisi yang diberikan akan berbeda dengan hasil percobaan ini, perlu dilakukan percobaan lebih lanjut, namun percobaan tersebut tidak dilakukan pada penelitian ini.

Dari hasil penerapan teknik optimasi berorientasi struktur neuron pada data IR-TNI dan VL-NI, dapat disimpulkan bahwa struktur neuron dan hasil rekognisi yang didapatkan cenderung sama untuk setiap perulangannya. Hal ini tidak terlepas dari proses pengurutan neuron yang hanya dilakukan 1 kali di awal percobaan. Implikasi dari pengurutan neuron hanya sekali adalah pada struktur neuron yang sama dalam proporsi data yang sama, akan didapatkan neuron-neuron aktif yang sama pula sehingga hasil rekognisi akan bergantung pada kesesuaian nilai *smoothing parameter* terhadap struktur tersebut. Nilai *smoothing parameter* yang sesuai dengan struktur neuron akan memberikan hasil rekognisi

yang baik, sebaliknya bila nilai *smoothing parameter* tidak sesuai, maka didapatkan hasil rekognisi yang baik. Diperolehnya struktur neuron dan nilai *smoothing parameter* yang memberikan hasil rekognisi yang konsisten ini merupakan kelebihan yang dimiliki oleh teknik optimasi berorientasi struktur neuron.

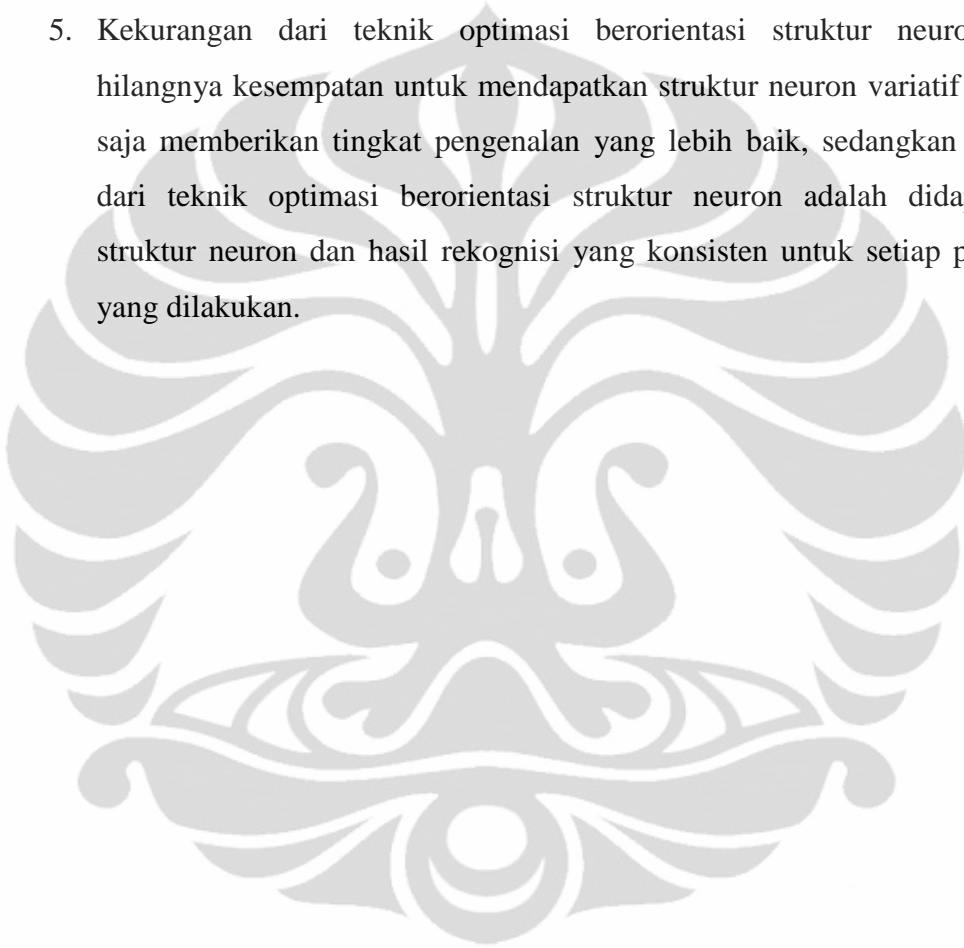
Teknik optimasi ini juga memiliki kekurangan dari sifat konsisten yang dimilikinya, yaitu tidak mampu mendapatkan tingkat pengenalan tertinggi yang mungkin dapat diperoleh dengan struktur neuron yang berbeda dari struktur yang didapatkan. Dengan kata lain, sifat konsistensi yang dimiliki teknik optimasi ini menghilangkan kesempatan untuk mendapatkan struktur neuron lebih variatif yang mungkin dapat memberikan tingkat pengenalan yang lebih baik. Seperti yang terjadi pada percobaan dengan data VL-NI proporsi [40,10,50] yang memiliki tingkat pengenalan yang lebih rendah dari [30,20,50]. Bila struktur neuron yang digunakan berbeda dari struktur neuron yang didapatkan pada percobaan tersebut, bisa saja didapatkan tingkat yang pengenalan yang sama.

4.4. Kesimpulan

Beberapa kesimpulan yang dapat ditarik dari hasil percobaan dan analisis pada Percobaan 2 dan 3 adalah :

1. Teknik optimasi berorientasi nilai *smoothing parameter*, baik dengan parameter GA menurut jurnal maupun modifikasi jurnal, berhasil mendapatkan pasangan struktur neuron dan nilai *smoothing parameter* yang memberikan tingkat pengenalan yang tinggi (>90%), baik untuk data IR-TNI maupun VL-NI;
2. Kekurangan dari teknik optimasi berorientasi nilai *smoothing parameter* adalah tidak selalu memberikan hasil rekognisi (RR_{tr} , RR_{trs} , RR_{ts}) yang konsisten untuk setiap perulangan, sedangkan kelebihan dari teknik optimasi berorientasi nilai *smoothing parameter* adalah terbukanya peluang untuk mendapatkan pasangan struktur neuron dan nilai *smoothing parameter* yang memberikan tingkat pengenalan (RR_{ts}) tertinggi dari inkonsistensi nilai yang dimilikinya;

3. Perbedaan mendasar antara teknik optimasi berorientasi *smoothing parameter* menggunakan parameter GA menurut jurnal dan parameter GA modifikasi jurnal terletak pada waktu *training*;
4. Teknik optimasi berorientasi struktur neuron berhasil mendapatkan pasangan struktur neuron dan nilai *smoothing parameter* yang memberikan tingkat pengenalan yang tinggi (>90%), baik untuk data IR-TNI maupun VL-NI;
5. Kekurangan dari teknik optimasi berorientasi struktur neuron adalah hilangnya kesempatan untuk mendapatkan struktur neuron variatif yang bisa saja memberikan tingkat pengenalan yang lebih baik, sedangkan kelebihan dari teknik optimasi berorientasi struktur neuron adalah didapatkannya struktur neuron dan hasil rekognisi yang konsisten untuk setiap perulangan yang dilakukan.



BAB 5

PERBANDINGAN HASIL PERCOBAAN

Bab ini akan membahas mengenai perbandingan kemampuan JST PNN dengan struktur utuh (PNN Full) dan PNN teroptimasi (OPNN) yang dilakukan pada penelitian ini. PNN teroptimasi hasil percobaan-percobaan yang telah dilakukan diberikan istilah sebagai berikut :

- OPNN_SP_Jurnal : PNN teroptimasi dengan teknik optimasi berorientasi nilai *smoothing parameter* dengan parameter GA menurut jurnal
- OPNN_SP : PNN teroptimasi dengan teknik optimasi berorientasi nilai *smoothing parameter* dengan parameter dengan parameter GA modifikasi jurnal
- OPNN_SN : PNN teroptimasi dengan teknik optimasi berorientasi struktur neuron

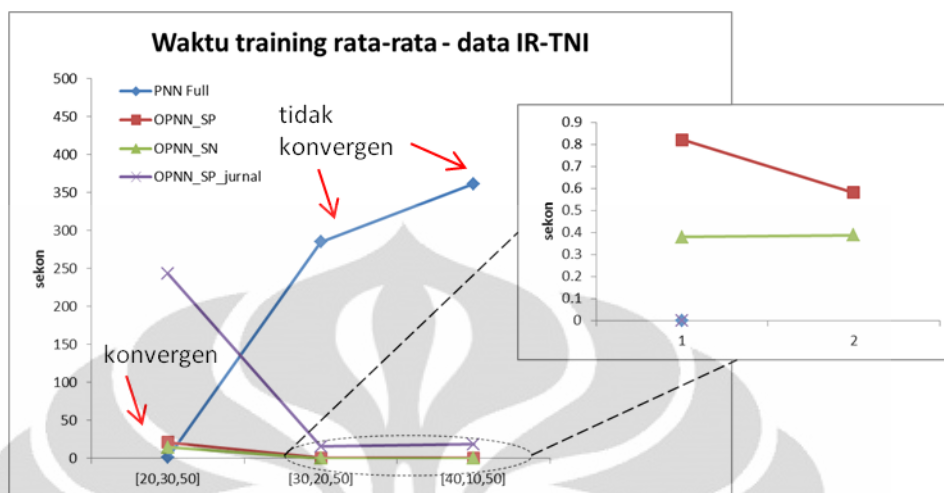
Hasil percobaan yang dibandingkan adalah hasil percobaan dengan data IR-TNI dan VL-NI yang telah dilakukan sebelumnya. Aspek-aspek yang akan dibandingkan antara lain waktu *training*, waktu *testing*, banyak neuron aktif, RR_{tr} , RR_{trs} , dan RR_{ts} . Selain itu, akan ditambahkan hasil percobaan dengan algoritma Backpropagation (BP) sebagai referensi. Parameter BP yang digunakan adalah :

- Laju pembelajaran (α) : 0.2
- Perbandingan data [X_{tr} , X_{ts}] : [50,50]
- Banyak neuron masukan : 30 neuron
- Banyak neuron tersembunyi : 15 neuron
- Fungsi error : Kuadratis
- Stopping condition : error total ≤ 0.01 atau epoh ≥ 10000

5.1. Perbandingan PNN, OPNN_SP_Jurnal, OPNN_SP, dan OPNN_SN

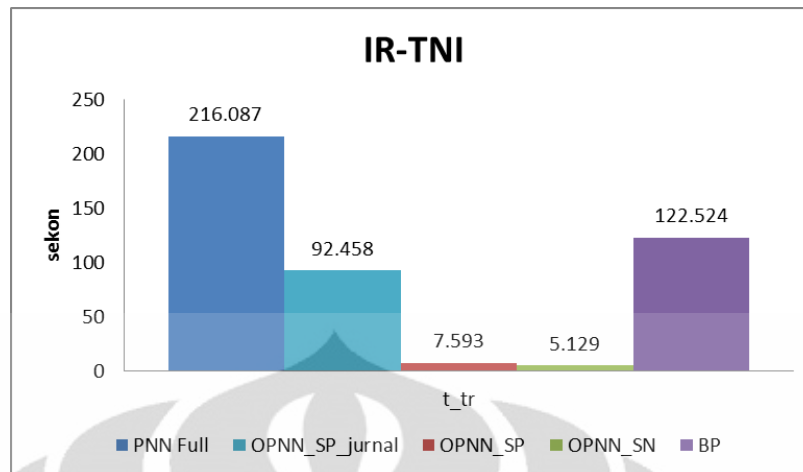
5.1.1. Perbandingan Hasil Percobaan Data IR-TNI

Berikut adalah pembahasan mengenai perbandingan hasil percobaan data IR-TNI berdasarkan aspek-aspek perbandingan :

a. Waktu *training*

Gambar 5.1. Waktu *training* rata-rata PNN dan OPNN data IR-TNI

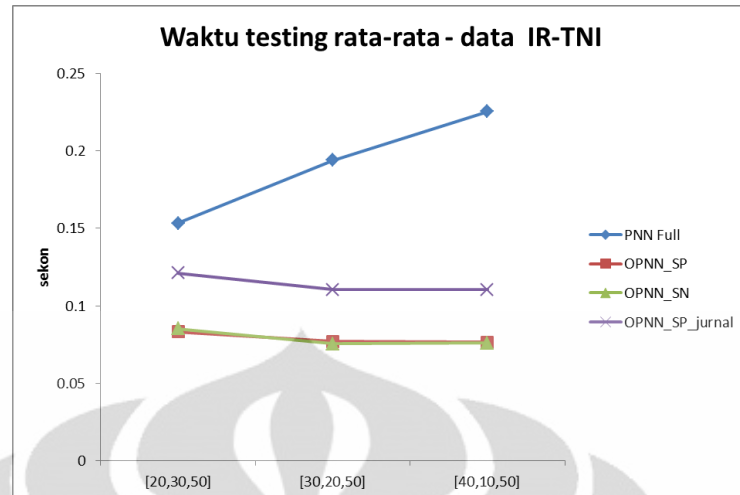
Gambar 5.1. memperlihatkan adanya perbedaan kecenderungan waktu *training* untuk pembelajaran yang konvergen dan tidak konvergen pada PNN Full. Apabila PNN_Full memiliki pembelajaran konvergen, waktu *training* yang dibutuhkan lebih singkat dibandingkan OPNN_SP dan OPNN_SN yang memiliki pembelajaran konvergen, sedangkan OPNN_SP_Jurnal yang sama-sama memiliki pembelajaran tidak konvergen memiliki waktu *training* paling lama (terlihat pada proporsi [20,30,50]). Lamanya waktu *training* pada OPNN_SP_Jurnal disebabkan karena tidak tercapainya rekognisi 100% untuk setiap kelas pada saat mencari struktur optimal dan syarat stopping condition yang mengharuskan tercapainya generasi maksimum pada proses GA. Pembelajaran PNN Full yang tidak konvergen jelas membutuhkan waktu *training* yang cukup lama dibandingkan seluruh OPNN yang mampu memberikan pembelajaran yang konvergen. OPNN_SP_Jurnal tetap memiliki waktu *training* yang lebih lama dibandingkan kedua OPNN lainnya karena *stopping condition* GA yang mengharuskan mencapai generasi maksimum. Walaupun tidak berselisih jauh, OPNN_SN memiliki waktu *training* yang relatif lebih singkat dibandingkan OPNN_SP. Gambar 2.14



Gambar 5.2. Waktu *training* rata-rata PNN, OPNN, dan BP data IR-TNI

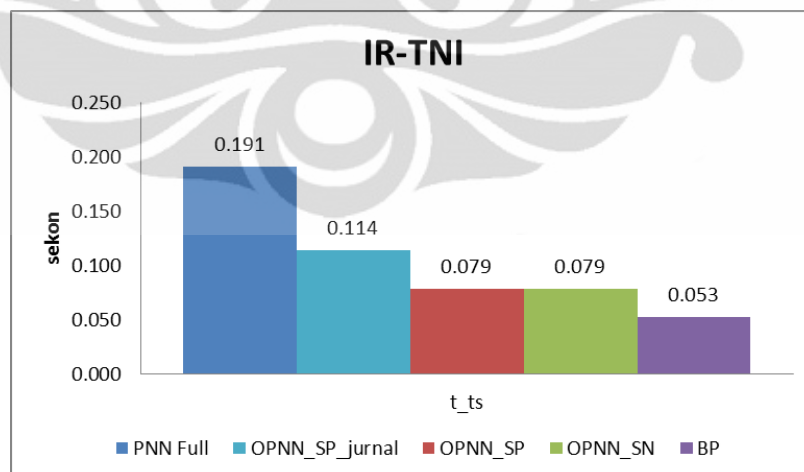
Waktu *training* proporsi [20,30,50]; [30,20,50]; dan [40,10,50] dirata-ratakan agar didapatkan gambaran waktu *training* umum untuk pembagian data [50,50]. Waktu *training* rata-rata untuk setiap JST PNN kemudian akan dibandingkan dengan waktu *training* BP sebagai referensi. Dari perbandingan ini dapat disimpulkan bahwa OPNN_SP dan OPNN_SN memiliki waktu *training* yang jauh lebih cepat dibandingkan BP, OPNN_SP_Jurnal memiliki waktu *training* lebih rendah dari BP sekitar 30 detik, sedangkan PNN Full memiliki waktu *training* yang lebih lama sekitar 1.7 kali waktu *training* BP. Dapat disimpulkan bahwa teknik optimasi PNN mampu mereduksi waktu *training* dan teknik optimasi dengan parameter modifikasi jurnal memberikan waktu *training* jauh lebih singkat dibandingkan teknik optimasi dengan parameter menurut jurnal.

b. Waktu *testing*



Gambar 5.3. Waktu *testing* rata-rata PNN dan OPNN data IR-TNI

Waktu *testing* untuk OPNN lebih singkat dibandingkan waktu *testing* PNN Full. Waktu *testing* mampu direduksi sekitar 40% dengan teknik optimasi PNN. OPNN_SP_Jurnal seharusnya memiliki waktu *testing* yang sama dengan OPNN_SP dan OPNN_SN karena banyak neuron aktif yang relatif sama, namun ternyata waktu *testing* OPNN_SP_Jurnal yang tampak pada Gambar 5.3. lebih lama. Hal ini dapat terjadi karena perbedaan kondisi perangkat keras pada saat perhitungan pewaktuan dilakukan.

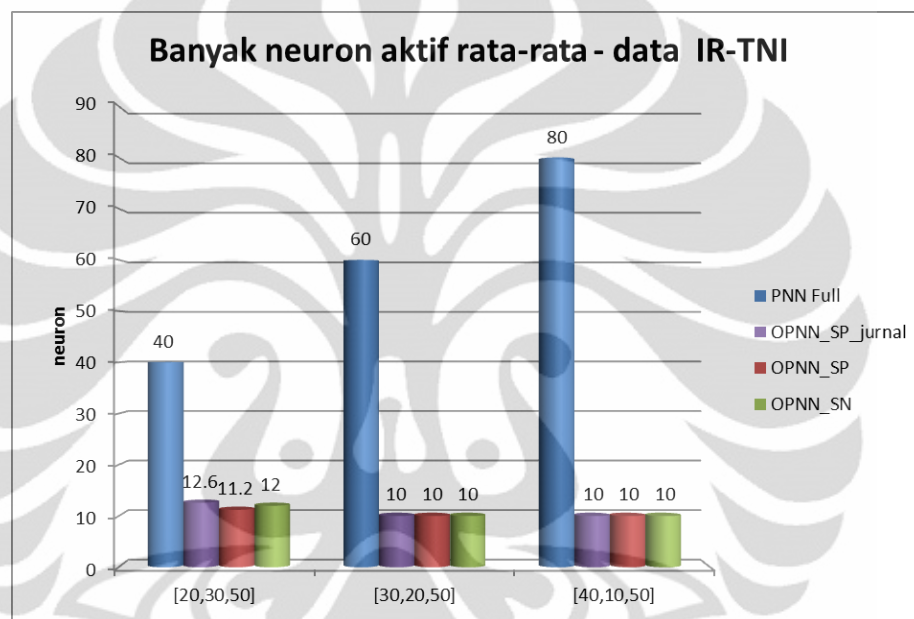


Gambar 5.4. Waktu *testing* rata-rata PNN, OPNN, dan BP data IR-TNI

Bila waktu *testing* untuk ketiga proporsi dirata-rata dan dibandingkan dengan waktu *testing* BP, maka waktu *testing* seluruh JST PNN lebih lama. BP memiliki

waktu *testing* yang lebih singkat sekitar 30% dibandingkan waktu *testing* JST OPNN atau 4 kali lebih cepat dibandingkan waktu *testing* PNN Full. Dari Gambar 5.4. terlihat bahwa teknik optimasi PNN mampu mereduksi waktu *testing* PNN_Full walaupun tidak mampu memberikan nilai yang lebih kecil dari waktu *testing* BP.

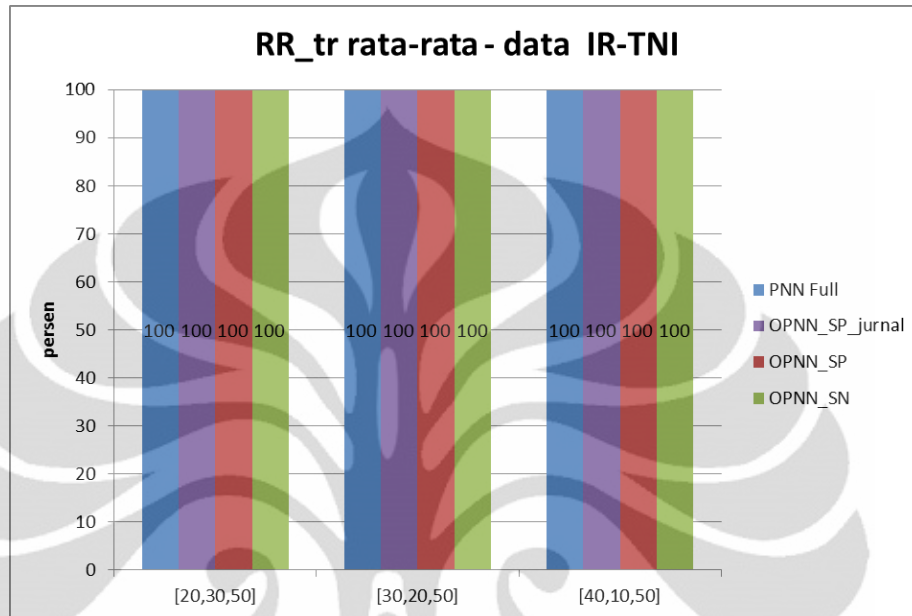
c. Banyak neuron aktif



Gambar 5.5. Banyak neuron aktif rata-rata data IR-TNI

Banyak neuron aktif untuk PNN Full meningkat seiring dengan semakin besarnya proporsi X_{tr} , sedangkan untuk OPNN, cenderung konstan. Setiap kenaikan 10% proporsi X_{tr} data IR-TNI, banyak neuron aktif bertambah 20 neuron, sedangkan pada OPNN, banyak neuron aktif tetap berkisar antara 10-12 neuron. Perbedaan banyak neuron aktif ini menunjukkan struktur neuron pada OPNN jauh lebih sederhana dan hanya menggunakan sedikit neuron untuk mewakili kelasnya. Penggunaan 10 neuron aktif yang didapatkan dari OPNN untuk proporsi [30,20,50] dan [40,10,50] merupakan struktur neuron paling sederhana dengan setiap kelas hanya diwakili oleh 1 neuron. Walaupun optimasi PNN dilakukan dengan teknik dan parameter yang berbeda, ternyata banyak neuron aktif yang didapatkan cenderung sama.

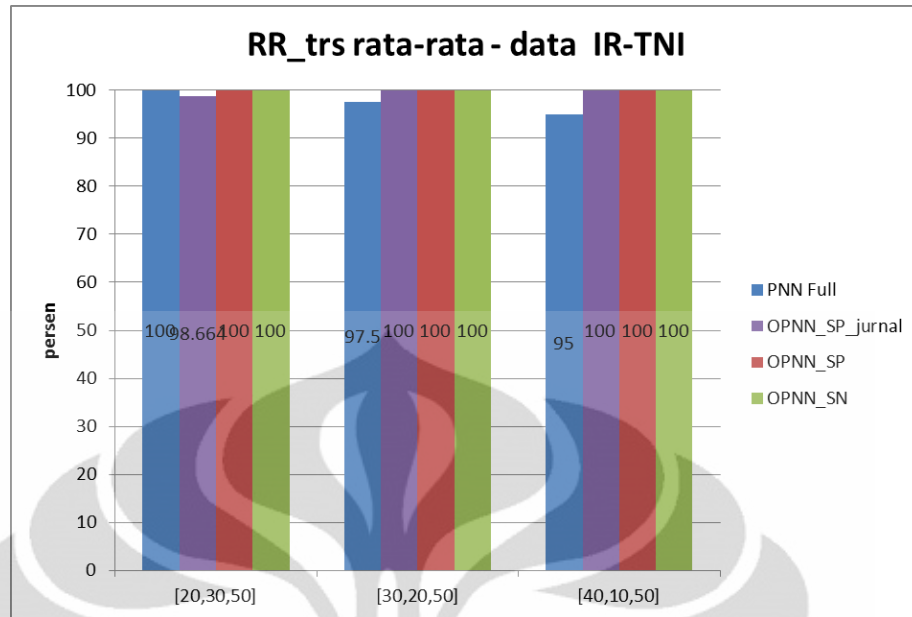
d. Hasil rekognisi data *training* (RR_{tr})



Gambar 5.6. RR_{tr} rata-rata PNN dan OPNN data IR-TNI

Untuk data *training* (X_{tr}), hasil rekognisi yang diberikan PNN Full dan OPNN sebesar 100%. Neuron-neuron aktif pada struktur neuron mampu mengenal X_{tr} dengan sangat baik. Secara logis hal ini memang tidak aneh karena neuron-neuron aktif memang berasal dari X_{tr} dan dilatih untuk mendapatkan RR_{tr} yang baik dengan mencari nilai *smoothing parameter* dan struktur neuron yang sesuai.

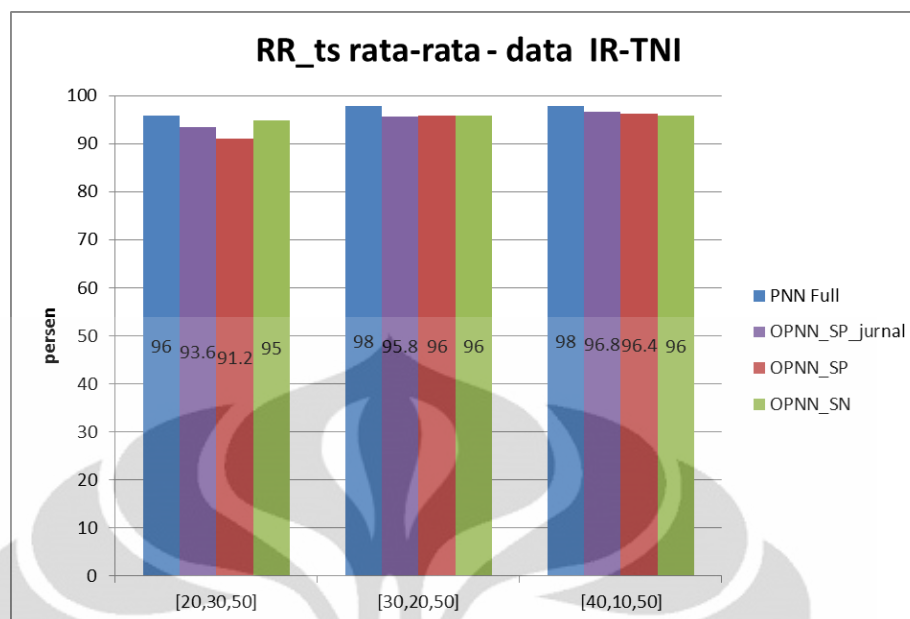
e. Hasil rekognisi data *testing* untuk *training* (RR_{trs})



Gambar 5.7. RR_{trrs} rata-rata PNN dan OPNN data IR-TNI

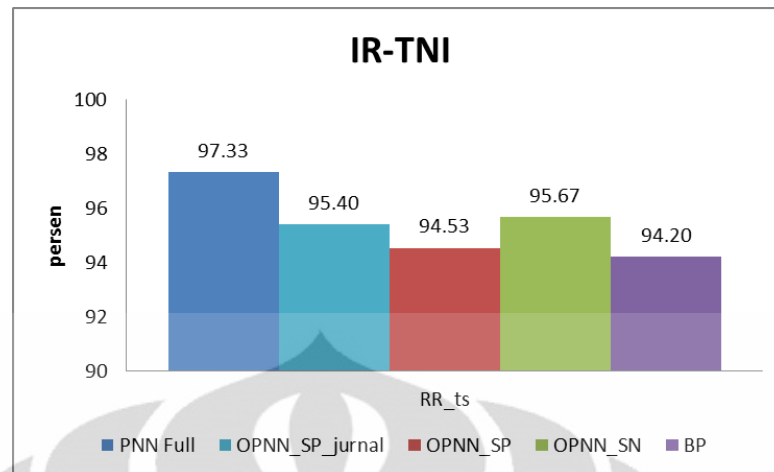
JST OPNN_SP dan OPNN_SN mampu mengenal X_{trrs} dengan sangat baik dengan rekognisi 100% untuk seluruh proporsi data. Hal yang berbeda terjadi pada JST PNN Full di mana RR_{trrs} untuk proporsi [30,20,50] dan [40,10,50] tidak mencapai 100%, begitu juga yang terjadi pada OPNN_SP_Jurnal untuk proporsi [20,30,50]. Tidak tercapainya RR_{trrs} 100% yang menyebabkan pembelajaran tidak konvergen. Ketidakonvergenan pada PNN_Full membuktikan bahwa penggunaan seluruh data atau neuron dengan banyak pola untuk mewakili kelas tidak selalu memberikan rekognisi data yang lebih baik. Dilihat secara umum, OPNN memberikan RR_{trrs} yang lebih baik dibandingkan PNN Full.

f. Hasil rekognisi data *testing* (RR_{ts})



Gambar 5.8. *RR_{ts}* rata-rata PNN dan OPNN data IR-TNI

Gambar 5.8. menunjukkan bahwa PNN Full memiliki *RR_{ts}* yang lebih tinggi di antara JST PNN lainnya. Namun, *RR_{ts}* yang diberikan OPNN pun secara umum hanya mengalami penurunan kecil sekitar 2-3%. Terlihat bahwa *RR_{ts}* yang diberikan oleh OPNN hampir sama, hanya berbeda untuk proporsi [20,30,50]. Proporsi [20,30,50] memang sebuah proporsi yang kurang tepat sehingga hasil rekognisi yang didapatkan bisa bervariasi. Penurunan *RR_{ts}* pada OPNN merupakan hal yang wajar karena penggunaan struktur neuron yang lebih sederhana memang lebih rentan menurunkan *RR_{ts}*. Hal penting yang didapatkan adalah walaupun menggunakan struktur yang jauh lebih sederhana pada OPNN, *RR_{ts}* yang diberikan masih tergolong baik karena penurunan yang terjadi tidak begitu besar.



Gambar 5.9. RR_{ts} rata-rata PNN, OPNN, dan BP data IR-TNI

RR_{ts} untuk seluruh proporsi data dihitung rata-ratanya agar dapat dibandingkan dengan RR_{ts} JST BP. Dapat dilihat pada Gambar 5.9. bahwa RR_{ts} JST PNN Full lebih baik daripada RR_{ts} JST BP untuk data IR-TNI dengan selisih sekitar 3%. RR_{ts} untuk OPNN_SP dan BP dapat dikatakan sama karena hanya memiliki selisih sekitar 0.3%, sedangkan OPNN_SP_Jurnal dan OPNN_SN memiliki RR_{ts} yang lebih tinggi sekitar 1% dari OPNN_SP dan BP. Walaupun RR_{ts} OPNN terjadi penurunan dari PNN Full, namun OPNN masih mampu menyamai kemampuan JST BP. Dapat disimpulkan untuk data IR-TNI, PNN Full memberikan RR_{ts} yang paling baik, disusul OPNN_SN dan OPNN_SP_Jurnal, kemudian OPNN_SP dan BP.

Hasil perbandingan untuk data IR-TNI dirangkum dalam sebuah tabel agar dapat dibandingkan secara keseluruhan :

Tabel 5.1. Perbandingan kemampuan PNN, OPNN, dan BP data IR-TNI

	DATA IR-TNI				
	PNN Full	OPNN_SP_Jurnal	OPNN_SP	OPNN_SN	BP
Waktu training	*	***	****	*****	**
Waktu testing	*	**	***	***	****
Banyak neuron aktif	*	***	***	***	-

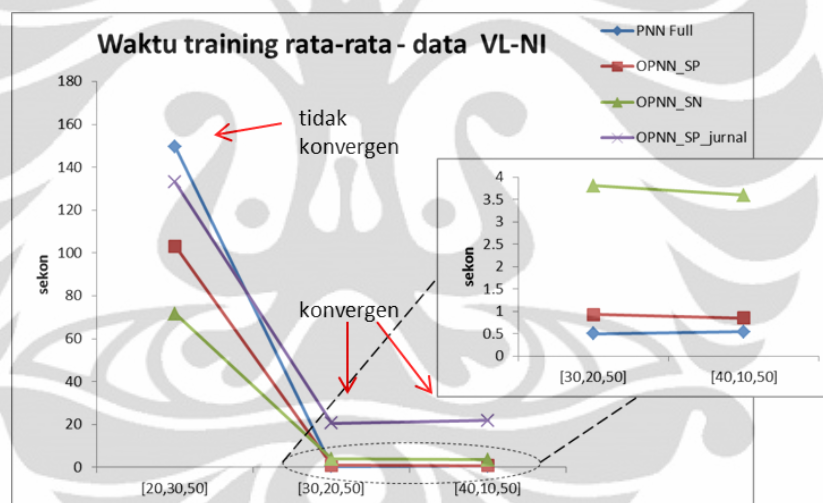
<i>RR_tr</i>	****	****	****	****	****
<i>RR_trs</i>	**	***	****	****	-
<i>RR_ts</i>	****	***	**	***	**

Keterangan : Semakin banyak *, semakin baik

5.1.2. Perbandingan Hasil Percobaan Data VL-NI

Berikut adalah pembahasan mengenai perbandingan hasil percobaan data VL-NI berdasarkan aspek-aspek perbandingan :

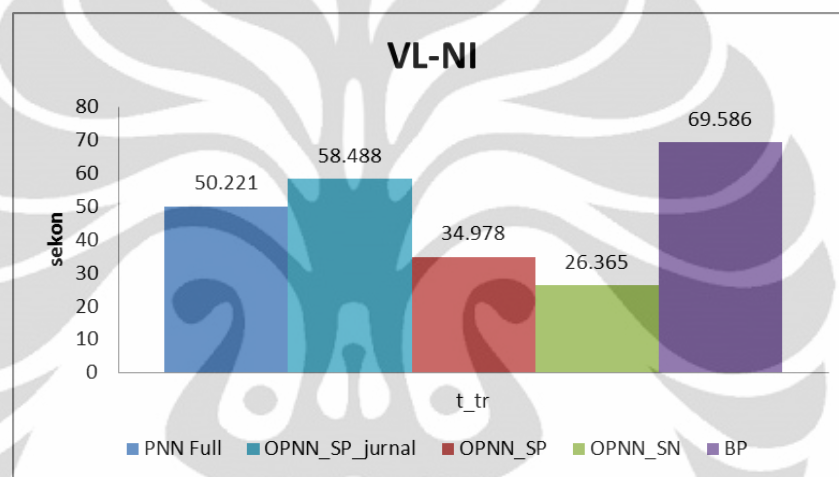
a. Waktu *training*



Gambar 5.10. Waktu *training* rata-rata PNN dan OPNN data VL-NI

Perbandingan waktu *training* untuk JST PNN dapat dibagi menjadi 2, yaitu waktu *training* untuk pembelajaran konvergen dan waktu *training* untuk pembelajaran tidak konvergen. Seluruh JST PNN dengan data VL-NI memiliki pembelajaran tidak konvergen untuk proporsi [20,30,50] dan pembelajaran konvergen untuk kedua proporsi lainnya. Untuk pembelajaran tidak konvergen, OPNN memiliki waktu *training* lebih cepat dibandingkan PNN Full. Urutan OPNN dari waktu *training* tercepat untuk pembelajaran konvergen adalah OPNN_SN, OPNN_SP, dan OPNN_SP_Jurnal. Hal yang berbeda berlaku pada pembelajaran konvergen. Untuk pembelajaran konvergen, PNN Full memiliki

waktu *training* yang lebih cepat dibandingkan OPNN. Waktu *training* OPNN_SP_Jurnal lebih lama dibandingkan OPNN_SP dan OPNN_SN dengan selisih sekitar 15-20 sekon. OPNN_SP memiliki waktu *training* hampir sama dengan PNN Full, sedangkan waktu *training* OPNN_SN lebih lama sekitar 3 sekon dari OPNN_SP. OPNN_SP_Jurnal memiliki waktu *training* cukup besar disebabkan oleh parameter GA menurut jurnal yang mengharuskan pembelajaran berhenti setelah mencapai *geenrasi* maksimum, sedangkan OPNN lain memiliki syarat pemberhentian lain, yaitu *fitness* maksimum.

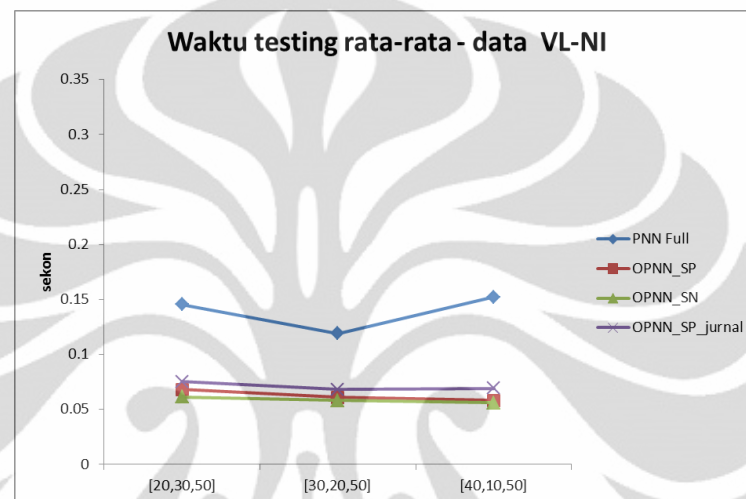


Gambar 5.11. Waktu *training* rata-rata PNN, OPNN, dan BP data VL-NI

Bila waktu *training* rata-rata untuk proporsi [50,50] dihitung dan dibandingkan dengan waktu *training* BP, diperoleh perbandingan seperti yang ditunjukkan Gambar 5.11. Dapat dilihat bahwa secara umum, waktu *training* JST PNN lebih cepat dibandingkan waktu *training* BP. OPNN_SN memiliki waktu *training* tercepat dengan 26.365 sekon, sedangkan BP memiliki waktu *training* terlama dengan 69.586 sekon. OPNN_SP, PNN Full, dan OPNN_SP_Jurnal berada diantaranya dengan 34.978, 50.221, dan 58.488 sekon. Secara rata-rata, waktu *training* OPNN_SP_Jurnal lebih lama dibandingkan PNN Full sehingga tujuan untuk mereduksi waktu *training* dalam optimasi PNN tidak terpenuhi untuk OPNN_SP_Jurnal. Sedangkan, untuk OPNN_SP dan OPNN_SN, tujuan optimasi PNN untuk mereduksi waktu *training* berhasil dilakukan karena mampu mereduksi waktu *training* PNN Full hingga mencapai 50% atau 30%. Hal ini menunjukkan perbedaan parameter GA menurut jurnal dan modifikasi jurnal, di

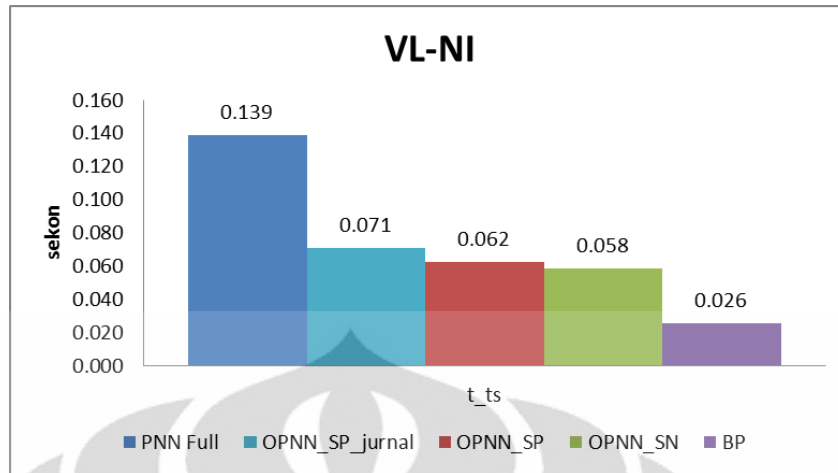
mana parameter GA menurut jurnal tidak mampu sepenuhnya mengakomodir tujuan optimasi PNN. Parameter GA menurut jurnal mampu memberikan waktu *training* yang lebih cepat pada pembelajaran tidak konvergen, tetapi memiliki waktu *training* yang lebih lama bila dilihat secara keseluruhan.

b. Waktu *testing*



Gambar 5.12. Waktu *testing* rata-rata PNN dan OPNN data VL-NI

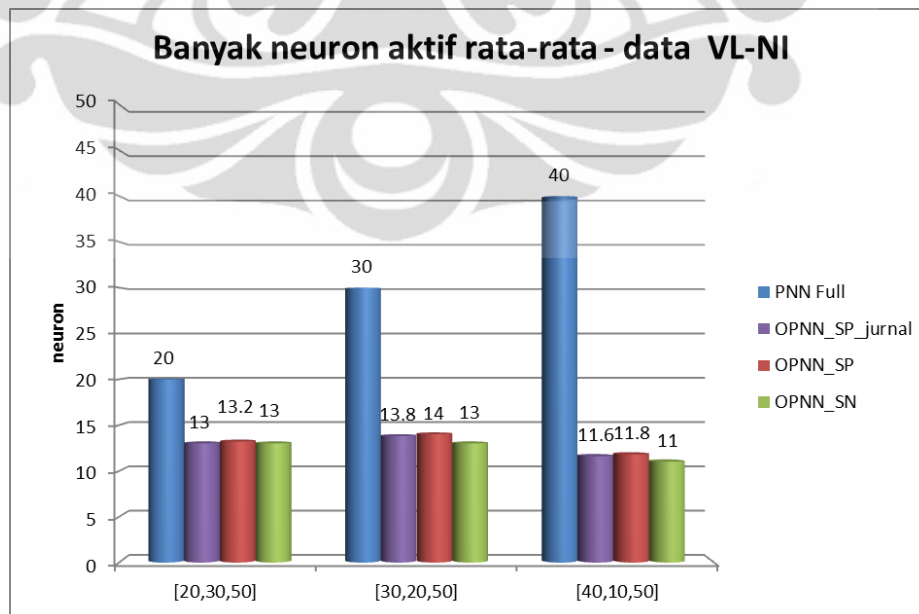
Gambar 5.12. memperlihatkan waktu *testing* OPNN lebih cepat dibandingkan PNN Full. Teknik optimasi mampu mereduksi waktu *testing* PNN Full menjadi setengahnya. Waktu *testing* untuk OPNN_SP, OPNN_SN, dan OPNN_SP_Jurnal dapat dikatakan sama karena memiliki selisih yang kecil dan banyak neuron aktif pada ketiga OPNN ini hampir sama.



Gambar 5.13. Waktu *testing* rata-rata PNN, OPNN, dan BP data VL-NI

Dibandingkan waktu *testing* BP, waktu *testing* seluruh JST PNN lebih lama. Waktu *training* PNN Full merupakan waktu *testing* yang terlalu lama hingga mencapai 5 kali waktu *testing* BP. Waktu *testing* OPNN_SN, OPNN_SP, dan OPNN_SP_Jurnal lebih cepat dari PNN Full. Ketiga OPNN ini memiliki waktu *testing* yang dapat dikatakan sama, yaitu sekitar 2.2 kali waktu *testing* BP.

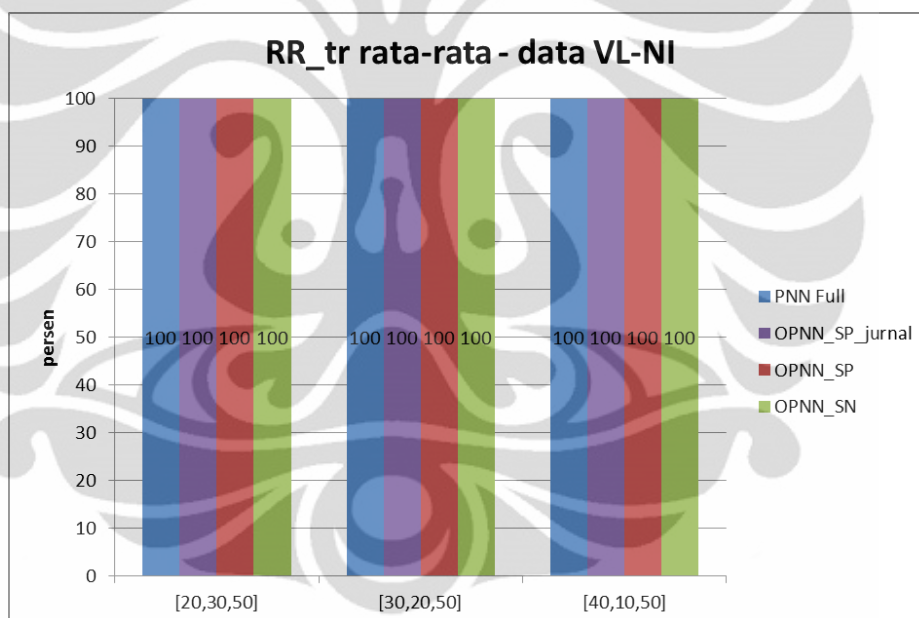
c. Banyak neuron aktif



Gambar 5.14. Banyak neuron aktif rata-rata data VL-NI

Banyak neuron aktif PNN Full meningkat seiring dengan semakin besarnya proporsi X_{tr} . Setiap kenaikan 10% proporsi X_{tr} VL-NI, neuron aktif bertambah 10 neuron, sedangkan neuron aktif OPNN cenderung tetap dengan 11-13 neuron. Oleh sebab itu, dapat disimpulkan bahwa teknik optimasi mampu mereduksi penggunaan neuron hingga hanya digunakan sekitar 65% dari neuron PNN Full untuk proporsi [20,30,50], 46% untuk proporsi [30,20,50], dan 29% untuk proporsi [40,10,50]. Salah satu penyebab munculnya kecenderungan penurunan banyak neuron aktif pada proporsi [40,10,50] adalah ditemukannya data pada proporsi tersebut yang telah mampu mewakili kelasnya, di mana data tersebut tidak terdapat pada proporsi sebelumnya.

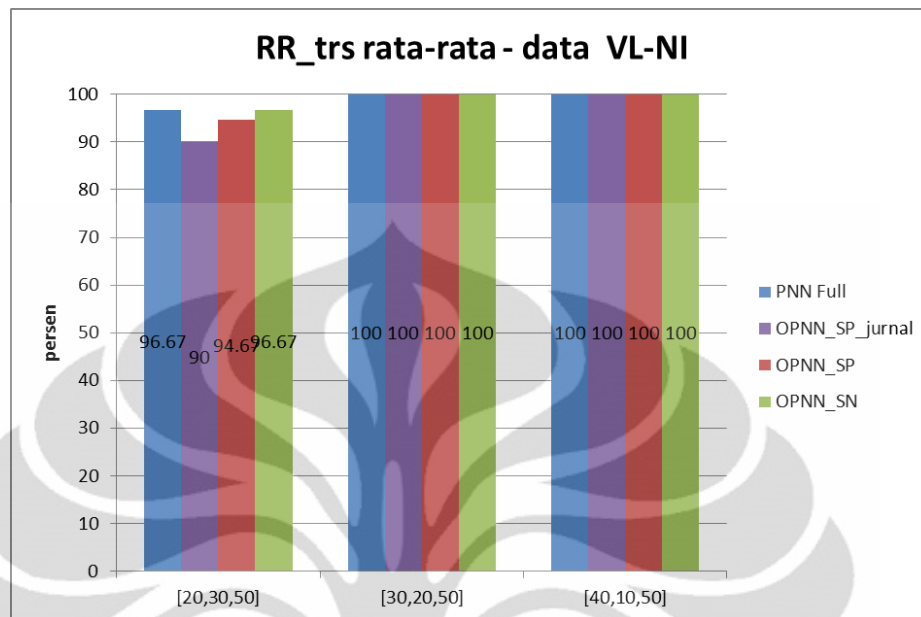
d. Hasil rekognisi data *training* (RR_{tr})



Gambar 5.15. RR_{tr} rata-rata PNN dan OPNN data VL-NI

Seluruh JST PNN mampu memberikan RR_{ts} sebesar 100%. Hal ini merupakan hal yang wajar karena neuron-neuron aktif juga berasal dari X_{tr} sehingga tidak aneh bila neuron-neuron tersebut dapat mengenali dirinya sendiri, apalagi dalam pembelajaran PNN memang diarahkan untuk mendapatkan RR_{tr} terbaik melalui fungsi *fitness*.

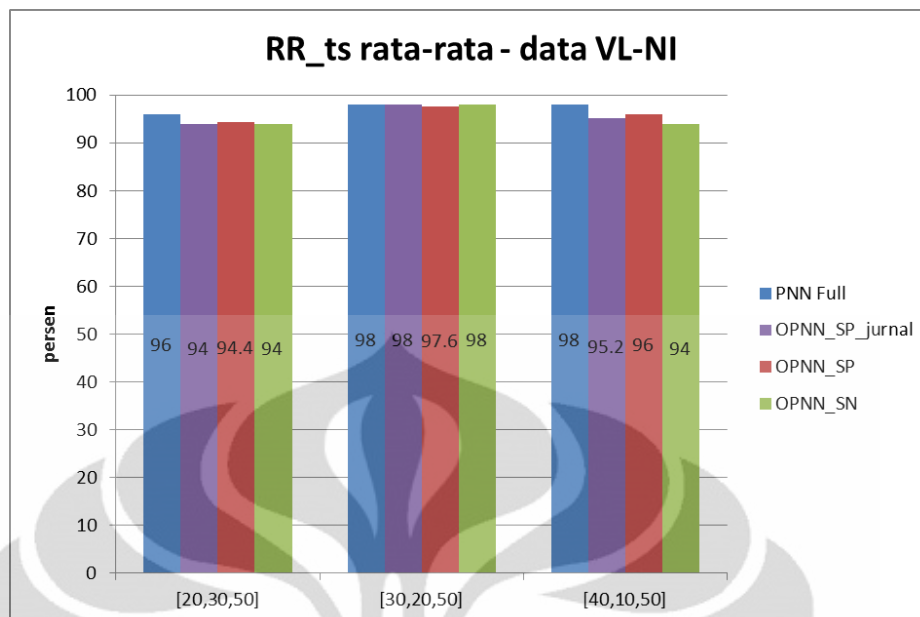
e. Hasil rekognisi data *testing* untuk *training* (RR_{trs})



Gambar 5.16. RR_{trs} rata-rata PNN dan OPNN data VL-NI

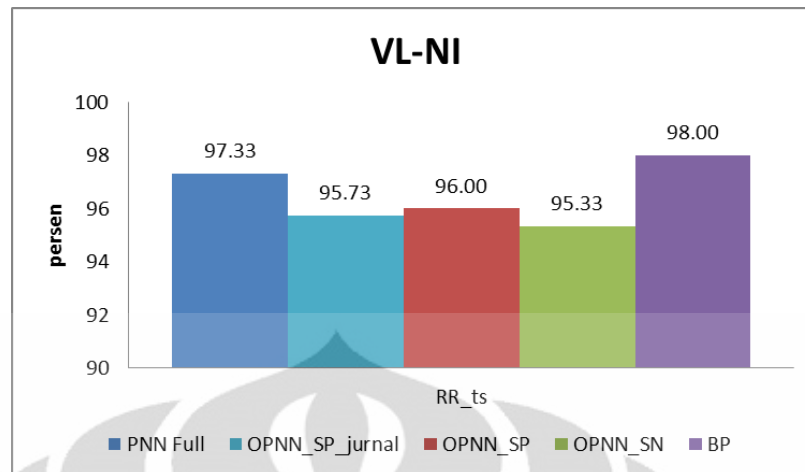
JST PNN tidak mampu mencapai RR_{trs} 100% untuk proporsi [20,30,50]. Hal ini yang menyebabkan pembelajaran untuk proporsi tersebut tidak konvergen. Alasan paling mungkin yang melatarbelakangi tidak berhasilnya mencapai RR_{trs} 100% untuk proporsi ini adalah kekurangan neuron untuk mewakili kelas. Proporsi [20,30,50] yang hanya mampu memberikan maksimum 2 neuron pada masing-masing kelas tidak mampu mengenali semua pola kelas yang ada. Untuk proporsi lainnya, JST PNN mampu mengenali 100%.

f. Hasil rekognisi data *testing* (RR_{ts})



Gambar 5.17. RR_{ts} rata-rata PNN dan OPNN data VL-NI

Secara umum, JST OPNN memberikan RR_{ts} yang lebih rendah dari PNN Full dengan selisih sekitar 2-3%. Terjadinya penurunan RR_{ts} pada OPNN dapat diterima mengingat penggunaan neuron yang lebih sedikit untuk mewakili pola kelas. Ketiga OPNN memberikan RR_{ts} yang hampir sama walaupun pada proporsi [40,10,50], terdapat selisih sekitar 1%. Hal penting yang didapatkan adalah walaupun menggunakan struktur yang jauh lebih sederhana pada OPNN, RR_{ts} yang diberikan masih tergolong baik karena penurunan yang terjadi tidak begitu besar. Oleh karena itu, tujuan PNN untuk menjaga tingkat pengenalan tetap berada pada nilai yang tinggi berhasil dicapai.



Gambar 5.18. RR_{ts} rata-rata PNN, OPNN, dan BP data VL-NI

Gambar 5.18. menunjukkan hasil perhitungan RR_{ts} rata-rata semua JST PNN untuk proporsi [50,50], kemudian dilakukan perbandingan dengan RR_{ts} BP . Dari gambar tersebut dapat dilihat bahwa BP dan PNN Full memiliki kemampuan mengenali data VL-NI yang hampir sama, sedangkan ketiga OPNN memiliki kemampuan yang lebih rendah sekitar 2% dibandingkan BP dan PNN Full. RR_{ts} yang diberikan oleh ketiga OPNN dapat dikatakan sama. OPNN mampu memberikan RR_{ts} yang baik walaupun tidak mampu menyamai kemampuan BP untuk mengenali data VL-NI.

Hasil perbandingan untuk data VL-NI dirangkum dalam sebuah tabel agar dapat dibandingkan secara keseluruhan :

Tabel 5.2. Perbandingan kemampuan PNN, OPNN, dan BP data VL-NI

	DATA VL-NI				
	PNN Full	OPNN_SP_Jurnal	OPNN_SP	OPNN_SN	BP
Waktu training	***	**	*****	*****	*
Waktu testing	*	***	***	***	*****
Banyak neuron aktif	*	**	**	**	-
RR_{tr}	*****	*****	*****	*****	*****

<i>RR_trs</i>	***	**	***	***	-
<i>RR_ts</i>	****	***	***	***	****

Keterangan : Semakin banyak *, semakin baik

5.1.3. Analisis Perbandingan

Hasil perbandingan kemampuan JST PNN untuk mengenal data IR-TNI dan VL-NI menunjukkan bahwa pada dasarnya seluruh JST PNN pada percobaan ini telah mampu mengenal data IR-TNI dan VL-NI dengan tingkat pengenalan antara 94-98%. Waktu *training* yang dibutuhkan untuk pembelajaran tidak konvergen pun masih dalam hitungan menit, yaitu maksimum sekitar 6-7 menit. Waktu *testing* maksimum untuk JST PNN terbilang cepat dengan waktu kurang lebih 0.25 sekon. Perhitungan waktu ini memang tergantung pada beberapa faktor seperti banyak data yang diperhitungkan, spesifikasi perangkat keras yang dilakukan, dan beban program lain yang mempengaruhi kinerja perangkat keras.

Perbandingan kemampuan JST PNN terhadap referensi BP dapat dikatakan bahwa secara relatif JST PNN unggul dalam hal waktu *training*, BP unggul dalam hal waktu *testing*, sedangkan mengenai tingkat pengenalan tergantung pada data yang diuji. Untuk data IR-TNI, BP yang mempunyai tingkat pengenalan 94.2% lebih buruk dibandingkan PNN yang mampu mengenal antara 94.53-97.33%. Untuk data VL-NI, BP yang mempunyai tingkat pengenalan 98% unggul sedikit dari PNN yang mampu mengenal antara 95.33-97.33%.

Teknik optimasi PNN berorientasi nilai *smoothing parameter*, maupun teknik optimasi PNN berorientasi struktur neuron berhasil untuk mereduksi waktu komputasi dan banyak neuron aktif yang digunakan pada struktur neuron PNN Full. Walaupun tingkat pengenalan untuk OPNN mengalami penurunan, namun OPNN tetap dapat memberikan tingkat pengenalan yang baik dan penurunan yang terjadi pun secara rata-rata hanya sekitar 2-3% dari tingkat pengenalan PNN Full. Oleh sebab itu, kemampuan PNN Full dan OPNN tidak berbeda jauh, namun OPNN memiliki keunggulan dalam hal waktu komputasi dibandingkan PNN Full.

Reduksi waktu *training* yang didapatkan kurang tepat bila dinyatakan dalam persentase karena reduksi waktu *training* bergantung pada jenis data dan jenis pembelajaran. Untuk kasus di mana seluruh JST PNN sama-sama mempunyai pembelajaran konvergen, waktu *training* PNN Full lebih cepat dibandingkan OPNN karena dengan banyaknya neuron yang digunakan berarti lebih banyak pola-pola kelas yang dimiliki PNN sehingga lebih mudah untuk mendapatkan rekognisi yang baik, sedangkan OPNN membutuhkan waktu lebih lama dalam mencari konfigurasi struktur neuron yang tepat untuk mengenali pola kelas. Untuk kasus di mana seluruh JST PNN sama-sama mempunyai pembelajaran tidak konvergen, waktu *training* OPNN lebih cepat dibandingkan PNN Full karena pengaruh dari banyak neuron aktif yang menyebabkan akumulasi waktu perhitungan. JST dengan neuron yang lebih banyak akan mempunyai waktu *training* yang lebih lama.

Dari segi waktu *testing*, OPNN mampu mereduksi waktu *testing* hingga 60% dari waktu *testing* PNN Full. Teknik optimasi pada OPNN akan mencari struktur paling sederhana yang telah mampu mewakili pola untuk setiap kelasnya, di mana hasil struktur yang didapatkan dapat berbeda untuk setiap jenis dan proporsi data atau permasalahan yang dihadapi. Untuk JST PNN, banyak neuron aktif mempengaruhi waktu *testing* JST tersebut. Semakin banyak neuron aktif, waktu *testing* yang dihasilkan semakin besar. OPNN memiliki waktu *testing* yang lebih cepat dari PNN Full karena teknik optimasi mampu mereduksi banyak neuron aktif untuk mewakili kelasnya sehingga tidak perlu menggunakan seluruh data *training* sebagai neuron.

Berkaitan dengan teknik optimasi PNN, teknik dengan menggunakan parameter GA modifikasi jurnal memiliki keunggulan dalam hal waktu *training* dibandingkan parameter GA menurut jurnal. Parameter GA menurut jurnal mengharuskan pembelajaran berhenti pada generasi maksimum, sedangkan parameter GA modifikasi jurnal memungkinkan pembelajaran berhenti ketika dicapai nilai *fitness* maksimum selain mencapai generasi maksimum. Dapat dilihat pada Gambar 5.1. dan Gambar 5.10. bahwa OPNN_SP_Jurnal selalu memiliki waktu *training* yang lebih lama dibandingkan OPNN_SP dan OPNN_SN, baik untuk pembelajaran konvergen maupun tidak konvergen. Waktu *training* untuk

OPNN dengan parameter GA modifikasi jurnal cenderung sama walaupun secara rata-rata dapat dilihat pada Gambar 5.2, dan Gambar 5.11. bahwa OPNN_SN memiliki waktu *training* relatif lebih cepat dari OPNN_SP. OPNN dengan parameter GA menurut dan modifikasi jurnal memberikan banyak neuron aktif yang hampir sama sehingga waktu *testing* ketiga OPNN ini juga hampir sama. Untuk tingkat pengenalan yang diberikan, memang ada perbedaan di antara ketiga OPNN ini, namun perbedaan yang tercipta $< 1\%$ sehingga dapat dikatakan kemampuan pengenalan ketiga OPNN secara rata-rata adalah sama. OPNN_SP_Jurnal dan OPNN_SP memberikan hasil rekognisi yang tidak konsisten untuk setiap perulangan yang dilakukan, sedangkan OPNN_SN selalu memberikan hasil rekognisi yang konsisten.

Berdasarkan perbandingan yang telah dilakukan, dapat dikatakan bahwa parameter GA modifikasi jurnal lebih baik dari parameter GA menurut jurnal karena memberikan waktu *training* yang lebih cepat, sedangkan faktor-faktor lainnya dapat dianggap sama. Teknik optimasi mana yang lebih baik antara teknik optimasi berorientasi nilai *smoothing parameter* atau teknik optimasi berorientasi struktur neuron sulit ditentukan bila dilihat dari angka-angka dalam hasil yang didapatkan. Hanya saja, bila mengacu pada aplikasi teknik optimasi ini perlu diperhatikan konsistensi yang diberikan pada setiap perulangannya. Bila teknik tersebut tidak konsisten dalam setiap perulangannya, maka pasangan struktur neuron dan nilai *smoothing parameter* yang didapatkan bisa saja memberikan hasil rekognisi yang rendah atau justru tinggi. Terdapat ketidakpastian dalam setiap perulangan yang mana tidak diketahui pada perulangan mana yang memberikan pasangan struktur neuron dan nilai *smoothing parameter* terbaik. Sedangkan, bila digunakan teknik yang konsisten pada setiap perulangannya, maka dapat dipastikan bahwa pasangan struktur neuron dan nilai *smoothing parameter* yang didapatkan bisa dianggap memberikan hasil rekognisi yang optimal. Dengan alasan ini dan hasil percobaan yang didapatkan, maka dapat dikatakan bahwa OPNN_SN lebih baik dari OPNN_SP.

5.2. Kesimpulan

Beberapa kesimpulan yang dapat ditarik dari hasil perbandingan dan analisis pada bab ini adalah :

1. Teknik optimasi PNN berorientasi nilai *smoothing parameter*, maupun teknik optimasi PNN berorientasi struktur neuron berhasil untuk mereduksi waktu komputasi dan banyak neuron aktif yang digunakan pada struktur neuron PNN Full walaupun tingkat pengenalan untuk OPNN mengalami penurunan;
2. OPNN_SP lebih baik dibandingkan OPNN_SP_Jurnal karena mempunyai waktu *training* yang lebih cepat, sedangkan faktor-faktor lain antara kedua OPNN ini dapat dikatakan sama;
3. OPNN_SP dan OPNN_SN memiliki waktu komputasi dan hasil rekognisi yang tidak berbeda jauh, hanya saja kedua OPNN ini berbeda dalam hal konsisten hasil rekognisi yang diberikan. Bila mengacu pada konsistensi, dapat dikatakan bahwa OPNN_SN lebih baik daripada OPNN_SP.

BAB 6

KESIMPULAN DAN SARAN

6.1. Kesimpulan

Kesimpulan yang dapat diambil dari percobaan-percobaan yang dilakukan, perbandingan, serta analisis pada skripsi ini :

1. Data IR memberikan hasil yang lebih baik untuk data IR-TNI, di mana proses NI pada data IR berpotensi menyebabkan pembelajaran tidak konvergen, menurunkan tingkat pengenalan dan meningkatkan waktu komputasi. Data VL dengan proses NI memberikan hasil yang lebih baik dibandingkan data VL tanpa proses NI. Proses NI pada data VL dapat menghasilkan RR yang konsisten untuk setiap perulangan, meningkatkan tingkat pengenalan, dan mempercepat waktu komputasi;
2. Nilai *smoothing parameter* yang didapatkan dari metode pencarian dengan GA yang digunakan merupakan nilai *smoothing parameter* yang dianggap optimal sesuai dengan parameter yang digunakan sehingga nilai ini belum tentu merupakan nilai yang optimal secara global;
3. Teknik optimasi berorientasi nilai *smoothing parameter*, baik dengan parameter GA menurut jurnal maupun modifikasi jurnal, dan teknik optimasi berorientasi struktur neuron berhasil untuk mereduksi waktu komputasi dan banyak neuron aktif yang digunakan pada struktur neuron PNN Full walaupun tingkat pengenalan untuk OPNN mengalami penurunan;
4. Kekurangan dari teknik optimasi berorientasi nilai *smoothing parameter* adalah tidak selalu memberikan hasil rekognisi (RR_{tr} , RR_{trs} , RR_{ts}) yang konsisten untuk setiap perulangan, sedangkan kelebihan dari teknik optimasi berorientasi nilai *smoothing parameter* adalah terbukanya peluang untuk mendapatkan pasangan struktur neuron dan nilai *smoothing parameter* yang memberikan tingkat pengenalan (RR_{ts}) tertinggi dari inkonsistensi nilai yang dimilikinya;
5. Kekurangan dari teknik optimasi berorientasi struktur neuron adalah hilangnya kesempatan untuk mendapatkan struktur neuron variatif yang bisa saja memberikan tingkat pengenalan yang lebih baik, sedangkan kelebihan dari

teknik optimasi berorientasi struktur neuron adalah didapatkannya struktur neuron dan hasil rekognisi yang konsisten untuk setiap perulangan yang dilakukan;

6. OPNN_SP lebih baik dibandingkan OPNN_SP_Jurnal karena mempunyai waktu *training* yang lebih cepat, sedangkan faktor-faktor lain antara kedua OPNN ini dapat dikatakan sama;
7. OPNN_SP dan OPNN_SN memiliki waktu komputasi dan hasil rekognisi yang tidak berbeda jauh, hanya saja kedua OPNN ini berbeda dalam hal konsisten hasil rekognisi yang diberikan. Bila mengacu pada konsistensi, dapat dikatakan bahwa OPNN_SN lebih baik daripada OPNN_SP.

6.2. Saran

Saran yang dapat diberikan agar dapat melakukan percobaan mengenai optimasi PNN :

1. Melakukan percobaan dengan data yang lebih bervariasi untuk dapat menemukan sifat-sifat lain yang dimiliki oleh OPNN_SP dan OPNN_SN;
2. Penggunaan parameter GA yang lain agar didapatkan parameter yang paling sesuai untuk mendapatkan tingkat pengenalan tertinggi. Parameter GA yang dapat disesuaikan antara lain jenis kromsoma dan fungsi *fitness*;
3. Melihat sifat bertolak belakang yang dimiliki teknik optimasi PNN berorientasi nilai *smoothing parameter* dan teknik optimasi PNN berorientasi struktur neuron, prosedur dari kedua teknik ini dapat dilebur menjadi teknik optimasi baru yang mempunyai kelebihan-kelebihan yang dimiliki oleh 2 teknik optimasi tersebut.

DAFTAR PUSTAKA

- Abdolee, R., Ali, M. T., & Rahman, T. A. (n.d.). Decimal Genetics Algorithms for Null Steering and Sidelobe Cancellation in Switch Beam Smart Antenna System. *International Journal of Computer Science and Security*, 1(3).
- Anton, H. (2000). *Dasar-Dasar Aljabar Linear* (7th ed., Vol. 1). Batam: Interaksara.
- Buchala, S. (n.d.). *Principal Component Analysis of Gender, Ethnicity, Age, and Identity of Face Images*. Retrieved from <http://info.scopus.com>
- Chen, W., & et.al. (2006). Illumination Compensation and Normalization for Robust Face Recognition Using Discrete Cosine Transform in Logarithm Domain. *IEEE Transactions on System, Man, and Cybernetics - Part B : Cybernetics*, 36(2).
- Fausett, L. (1994). *Fundamentals of Neural Networks Architecture, Algorithms, and Applications*. Prentice Hall.
- Gen, M., & Cheng, R. (1996). *Genetic Algorithms & Engineering Design*. New York: John Wiley & Sons Inc.
- Herry. (2002). *Kinerja PNN-Teroptimasi Berbasis Algoritma Genetika dalam Pengenalan Aroma 2 Campuran*. Skripsi. Fakultas Ilmu Komputer Universitas Indonesia.
- Imantaka, S. R. (2010). *Sistem Pengenal Wajah Berbasis Ensemble Neural Network untuk Citra Infra Merah*. Skripsi. Fakultas Teknik Universitas Indonesia.
- Jatmiko, W. (2000). *Karakteristik Sistem Penciuman Elektronik Dalam Mengenal Aroma dan Komposisi Aroma*. Tesis Magister UI.
- Jolliffe, I. (2002). *Principal Component Analysis* (2nd ed.). Springer Series in Statistics.
- Mao, K. Z., & et.al. (2000, July). Probabilistic Neural-Network Structure Determination for Pattern Classification. *IEEE Transaction on Neural Networks*, 11(4).

- Masters, T. (1995). *Advanced Algorithms for Neural Networks : A C++ Sourcebook*. John Wiley & Sons.
- Pramudia, H. B. (2011). *Perbandingan Performa Jaringan Syaraf Tiruan Backpropagation dengan Learning Vector Quantization dalam mengenali Wajah Hasil Kompensasi dan Normalisasi Iluminasi Menggunakan Metode Discrete Cosine Transform dalam Ruang Logaritma*. Seminar. Fakultas Teknik Universitas Indonesia.
- Sethi, I., & Jain, A. (1991). *Artificial Neural Networks and Statistical Pattern Recognition Old & New Connections*. Elsevier Science Publishers B.V.
- Tian, B., & al., e. (2000, May). Temporal Updating Scheme for Probabilistic Neural Networks with Application to Satellite Cloud Classification. *IEEE Transaction on Neural Networks*, 11(4).
- Zaknich, A. (1998). *Artificial Neural Networks An Introductory Course*. Retrieved from http://www.maths.uwa.edu.au/~rkealley/ann_all/ann_all.html

LAMPIRAN

A. Lampiran 1 : Data Penelitian

Foto *infrared* :

Kelas I :



Kelas II :



Kelas III :



Kelas IV :



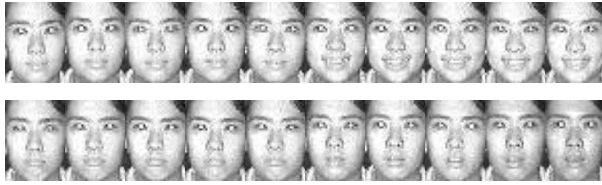
Kelas V :



Kelas VI :



Kelas VII :



Kelas VIII :



Kelas IX :



Kelas X :

Foto *visible light* :

Kelas I :



Kelas II :



Kelas III :



Kelas IV :



Kelas V :



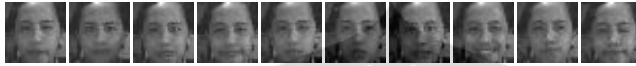
Kelas VI :



Kelas VII :



Kelas VIII :



Kelas IX :



Kelas X :

