



**JARINGAN SARAF TIRUAN BERBASIS
TRANSFORMASI WAVELET UNTUK
MEMPREDIKSI TINGKAT PENCEMARAN UDARA
SUSPENDED PARTICULATE MATTER DI KOTA
SEMARANG**

Skripsi

disusun sebagai salah satu syarat
untuk memperoleh gelar Sarjana Sains

Program Studi Matematika

UNNES
oleh
UNIVERSITAS NEGERI SEMARANG

Muhammad Imaduddin

4111412069

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI SEMARANG**

2017

PERNYATAAN

Saya menyatakan bahwa yang tertulis dalam skripsi ini benar-benar hasil karya saya sendiri, bukan jiplakan dari karya tulis orang lain, baik sebagian atau seluruhnya. Pendapat atau temuan orang lain yang terdapat dalam skripsi ini dikutip atau dirujuk berdasarkan kode etik ilmiah.



Semarang, 5 Januari 2017



UNNES
UNIVERSITAS NEGERI SEMARANG

Muhammad Imaduddin

4111412069

PENGESAHAN

Skripsi yang berjudul

Jaringan Syaraf Tiruan berbasis Transformasi *Wavelet* untuk Memprediksi Tingkat Pencemaran Udara *Suspended Particulate Matter* di Kota Semarang.

disusun oleh

Muhammad Imaduddin

4111412069

Telah dipertahankan dihadapan sidang Panitia Ujian Skripsi Jurusan Matematika FMIPA Universitas Negeri Semarang pada tanggal 5 Januari 2017



Prof. Dr. Zaemuri S.E., M.Si, Akt.
196412231988031001

Sekretaris

Drs. Arief Agoestanto, M.Si.
196807221993071005

Ketua Penguji

Putriaji Hendikawati S.Si., M.Pd., M.Sc.
198208182006062001

Anggota Penguji/
Pembimbing 1

Dr. Scolastika Mariani, M.Si.
196502101991022001

Anggota Penguji/
Pembimbing 2

Riza Arifudin, S.Pd., M.Cs.
198005252005011001

MOTTO DAN PERSEMBAHAN

MOTTO

- ❖ “Sesungguhnya bersama kesukaran itu ada keringanan. Karena itu bila kau sudah selesai (mengerjakan yang lain). Dan berharaplah kepada Tuhanmu.” (Q.S Al Insyirah: 6-8).
- ❖ Maka nikmat Tuhan kamu manakah yang kamu dustakan? (Q.S Ar-Rahman)
- ❖ Bersyukurlah atas semua kenikmatan Tuhan yang telah berikan dan janganlah kau berputus asa dalam setiap ujian yang Dia berikan.
- ❖ Teruslah berjuang (dengan cara yang khalal) hingga semua yang kau inginkan dapat tercapai.

PERSEMBAHAN

- ❖ Untuk ALLAH SWT, Tuhanku Semesta Alam
- ❖ Untuk kedua orang tua tercinta Ibu Pratiwi Sulistyawati dan Bapak Warsito
- ❖ Untuk Kakak tersayang Muhammad Abdul Aziz Wicaksana
- ❖ Untuk Keluarga besar tercinta
- ❖ Universitas Negeri Semarang (UNNES)
- ❖ Untuk Dosen Jurusan Matematika dan Dosen pembimbing
- ❖ Untuk kekasihku Hapsari Setyo Dyah Utami
- ❖ Untuk teman-teman Universitas Negeri Semarang yang telah memberikan arti sebuah kehidupan dan perjuangan

KATA PENGANTAR

Puji syukur kehadiran ALLAH SWT yang telah memberikan rahmat, nikmat dan karuniaNya serta kemudahan sehingga penulis dapat menyelesaikan skripsi yang berjudul **“Jaringan Syaraf Tiruan berbasis Transformasi Wavelet untuk Memprediksi Tingkat Pencemaran Udara *Suspended Particulate Matter* di Kota Semarang”**.

Penulis menyadari dalam penyusunan skripsi ini penulis telah mendapat banyak bantuan, bimbingan, dan dorongan dari berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Prof. Dr. Fathur Rokhman, M.Hum, Rektor Universitas Negeri Semarang.
2. Prof. Dr. Zaenuri S.E, M.Si,Akt., Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Semarang.
3. Drs, Arief Agoestanto, M.Si., Ketua Jurusan Matematika FMIPA Universitas Negeri Semarang.
4. Drs. Mashuri M.Si., Ketua Prodi Matematika Jurusan Matematika FMIPA Universitas Negeri Semarang dan selaku dosen wali yang telah membimbing dan memberikan masukan selama 4 tahun penulis menjalani perkuliahan.
5. Dr. Scolastika Mariani, M.Si., selaku Dosen Pembimbing I yang telah memberikan bimbingan, pengarahan, nasehat, saran dan dorongan selama penyusunan skripsi ini.
6. Riza Arifudin, S.Pd., M.Cs., selaku Dosen Pembimbing II yang telah memberikan bimbingan, pengarahan, nasehat, saran dan dorongan selama penyusunan skripsi ini.

7. Putriaji Hendikawati, S.Si., M.Pd., M.Sc., selaku Dosen Penguji yang telah memberikan penilaian dan saran dalam perbaikan skripsi ini.
8. Staf Dosen Matematika Universitas Negeri Semarang yang telah membekali penulis dengan berbagai ilmu selama mengikuti perkuliahan sampai akhir penulisan skripsi.
9. Staf Tata Usaha Universitas Negeri Semarang yang telah banyak membantu penulis selama mengikuti perkuliahan dan penulisan skripsi ini.
10. Keluarga Besarku yang selalu mendokan dan menjadi motivasiku dalam menyelesaikan skripsi ini.
11. Kekasihku Hapsari Setyo Dyah Utami, yang telah memberiku semangat dan setia mendampingiku disaat susah ataupun senang.
12. Teman-teman Matematika angkatan 2012, teman-teman KKN Cahaya Manunggal, teman-teman PKL BPS Kabupaten Tegal, teman-teman input data BPS Provinsi Jawa Tengah, teman-teman Kos CB, teman-teman Kos IR 17, teman-teman Kos KMY yang telah memberikan kenangannya.
13. Semua pihak yang tidak dapat penulis sebutkan satu persatu yang telah membantu dalam penyelesaian skripsi ini.

Penulis menyadari bahwa masih banyak keterbatasan pengetahuan dan kemampuan yang penulis miliki. Penulis mengharapkan kritik dan saran yang bisa membangun penelitian-penelitian yang lain. Semoga skripsi ini dapat berguna dan bermanfaat bagi pembaca.

Semarang, 5 Januari 2017

Penulis

ABSTRAK

Imaduddin, Muhammad. 2016. Jaringan Syaraf Tiruan berbasis Transformasi *Wavelet* untuk Memprediksi Tingkat Pencemaran Udara *Suspended Particulate Matter* di Kota Semarang. Skripsi, Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Semarang. Pembimbing Utama Dr. Scolastika Mariani, M.Si. dan Pembimbing Pendamping Riza Arifudin, S.Pd., M.Cs.

Kata kunci: *Suspended Particulate Matter* (SPM), Jaringan Syaraf Tiruan, Transformasi *Wavelet*, Algoritma Nguyen-Widrow, *Software* Matlab.

Penelitian ini mengkaji tentang Jaringan Syaraf Tiruan berbasis Transformasi *Wavelet*, yang merupakan model penggabungan dari model Jaringan Syaraf Tiruan dan Transformasi *Wavelet*. Tujuan dari penelitian ini adalah untuk menemukan model terbaik Jaringan Syaraf Tiruan berbasis Transformasi *Wavelet* pada data pencemaran udara *suspended particulate matter* (SPM) dan meramalkan data pencemaran udara *suspended particulate matter* periode Januari dan Februari 2016. Pengambilan data dilakukan dengan cara mendokumentasikan data sekunder dari Badan Meteorologi Klimatologi dan Geofisika.

Langkah awal dalam penelitian adalah melakukan pendekomposisian data pencemaran udara SPM dengan transformasi *wavelet*, data yang sudah dilakukan pendekomposisian data selanjutnya dilakukan proses normalisasi data dan pembagian data menjadi data latih dan data uji. Data latih dan data uji digunakan untuk proses jaringan syaraf tiruan, dengan memasukkan parameter jaringan antara lain: maksimum *epoch*, *learning rate*, galat *error*, *hidden layer*, dan momentum, didapatkan model terbaik dari proses pelatihan dan pengujian jaringan. Dari proses pelatihan dan pengujian jaringan didapatkan model terbaik dari jaringan syaraf tiruan adalah jaringan syaraf dengan arsitektur jaringan 12-18-1 dan dengan parameter *epoch* = 8, *learning rate* = 0,04, MSE = 0,000480, MAPE = 0,015 residual = 0,935, dan akurasi = 88,24%.

Setelah didapatkan model terbaik, selanjutnya adalah melakukan peramalan data pencemaran udara SPM untuk periode bulan Januari dan Februari 2016, langkah pertama yang dilakukan adalah memasukkan data hasil pendekomposisian dengan metode transformasi *wavelet* dan parameter dari model terbaik pada proses jaringan syaraf tiruan. Keluaran dari proses ini adalah peramalan dekomposisi metode transformasi *wavelet* untuk periode bulan Januari dan Februari 2016 yaitu sebesar -2,5817 dan 67,5394. Setelah didapatkan nilai dekomposisi untuk periode bulan Januari dan Februari 2016, selanjutnya adalah melakukan proses rekonstruksi data, dimana hasil dari proses rekonstruksi data adalah data peramalan pencemaran udara SPM untuk periode bulan Januari dan Februari 2016. Hasil rekonstruksi data diperoleh peramalan data untuk bulan Januari sebesar 64,9577 dan bulan Februari sebesar -70,1211.

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
PERNYATAAN KEASLIAN TULISAN	ii
HALAMAN PENGESAHAN	iii
MOTTO DAN PERSEMBAHAN	iv
PRAKATA	v
ABSTRAK	vii
DAFTAR ISI	viii
DAFTAR TABEL	xiii
DAFTAR GAMBAR	xv
DAFTAR LAMPIRAN	xviii
BAB 1 PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah	6
1.3 Batasan Masalah	6
1.4 Tujuan penelitian	7
1.5 Manfaat Penelitian	7
1.6 Sistematika Penelitian	9

BAB 2 TINJAUAN PUSTAKA

2.1 Jaringan Syaraf Tiruan berbasis Transformasi <i>Wavelet</i>	11
2.2 <i>Wavelet</i>	11
2.2.1 Transformasi <i>Wavelet</i>	13
2.2.2 Transformasi <i>Wavelet</i> Diskrit	13
2.2.3 Dekomposisi <i>Wavelet</i>	18
2.2.4 Dekomposisi <i>Averaging</i> dan <i>Differencing</i>	20
2.2.5 <i>Wavelet</i> Haar	23
2.3 Jaringan Syaraf Tiruan	26
2.3.1 Fungsi Aktivasi	28
2.3.2 Arsitektur Jaringan Syaraf Tiruan	34
2.3.3 Algoritma Pembelajaran	39
2.3.4 Metode <i>Backpropagation</i>	41
2.3.5 Konsep Jaringan Syaraf Tiruan <i>Backpropagation</i>	42
2.3.6 Arsitektur Jaringan <i>Backpropagation</i>	43
2.3.7 Pelatihan dan Pengujian <i>Backpropagation</i>	45
2.3.8 Pembagian Data	52
2.3.9 <i>Backpropagation</i> dalam Prediksi	54
2.3.10 Meningkatkan Hasil Metode <i>Backpropagation</i>	57
2.3.10.1 Pemilihan Bobot dan Bias Awal	57
2.3.10.2 Jumlah Unit Tersembunyi	60
2.3.10.3 Waktu Iterasi	60
2.3.10.4 Pengujian (<i>Testing</i>) pada Metode <i>Backpropagation</i> ..	61

2.3.10.5	Transformasi Normal (Normalisasi)	62
2.3.10.6	Denormalisasi	64
2.3.10.7	Ukuran Akurasi Prediksi	65
2.3.10.7.1	Ukuran Statistik Standar	65
2.3.10.7.2	Ukuran-ukuran Relatif	68
2.3.10.7.3	Akurasi Sistem	70
2.4	<i>Matrix Labory</i> (Matlab)	71
2.4.1	<i>Graphical User Interface</i> (GUI)	71
2.4.2	GUI untuk Mensimulasikan Jaringan Syaraf Tiruan <i>Backpropagation</i>	72
2.4.3	Komponen GUI Matlab	74
2.4.4	<i>Toolbox Backpropagation</i> di Matlab	78
2.5	Pencemaran Udara	79
2.5.1	<i>Suspended Particulate Matter</i> (SPM)	79
2.5.2	Nilai Ambang Batas (NAB) untuk <i>Suspended Particulate Matteri</i> (SPM)	80
2.5.3	Dampak <i>Suspended Particulate Matter</i> (SPM) bagi Kesehatan	80

BAB 3 METODE PENELITIAN

3.1	Studi Pustaka	82
3.2	Perumusan Masalah	83
3.3	Pengumpulan Data	83
3.4	Kerangka Pikir Penelitian	85

3.5	Analisis Perancangan Jaringan Syaraf Tiruan berbasis Transformasi <i>Wavelet</i>	86
3.5.1	<i>Flowchart</i> Proses Jaringan Syaraf Tiruan Berbasis Transformasi <i>Wavelet</i>	87
3.5.2	<i>Flowchart</i> Dekomposisi Data Menggunakan <i>Wavelet</i> Haar .	89
3.5.3	<i>Flowchart</i> Pemodelan Data Menggunakan Jaringan Syaraf Tiruan <i>Backpropagation</i>	91
3.5.4	<i>Flowchart</i> Prediksi Data Menggunakan Jaringan Syaraf Tiruan <i>Backpropagation</i>	93
3.5.5	<i>Flowchart</i> Rekonstruksi data Menggunakan <i>Wavelet</i> Haar ..	95
3.6	Penarikan Kesimpulan	96

BAB 4 HASIL DAN PEMBAHASAN

4.1	Hasil Penelitian	100
4.1.1	Tahap Pengambilan Data	101
4.1.2	Tahap Pengolahan Data	102
4.1.3	Tahap Perancangan Sistem	103
4.1.4	Perancangan Sistem Pelatihan	103
4.1.5	Perancangan Arsitektur Jaringan Syaraf Tiruan	104
4.1.5.1	Bobot dan Bias	104
4.1.5.2	Jumlah <i>Neuron</i> pada Lapisan Tersembunyi	104
4.1.5.3	<i>Error Goal</i> (Kinerja Tujuan)	105
4.1.5.4	<i>Learning Rate</i> (Laju Pembelajaran)	105
4.1.5.5	Maksimum <i>Epoch</i>	106

4.1.6	Perhitungan Jaringan Syaraf Tiruan berbasis Transformasi <i>Wavelet</i>	106
4.1.7	Prediksi Data Tingkat Pencemaran Udara	125
4.1.8	Proses Pengujian	131
4.1.9	Perancangan Desain Sistem	132
4.1.9.1	Desain Tampilan <i>Form</i> Halaman Depan	132
4.1.9.2	Desain Tampilan <i>Form</i> Analisis Data	134
4.1.9.3	Desain Tampilan <i>Form</i> Prediksi Data	138
4.1.10	Simulasi Sistem	140
4.1.11	Simulasi Sistem Transformasi <i>Wavelet</i>	140
4.1.12	Pembahasan Hasil Arsitektur Jaringan Syaraf Tiruan <i>Backpropagation</i>	144
4.1.12.1	Analisis Lapisan Tersembunyi.....	147
4.1.12.2	Analisis Laju Belajar dan Momentum	149
4.1.12.3	Analisis Variasi Jaringan	155
4.1.12.4	Prediksi Tingkat Pencemaran Udara SPM.....	156
4.2	Pembahasan	163
BAB 5 PENUTUP		
5.1	Simpulan	172
5.2	Saran	173
DAFTAR PUSTAKA		175
LAMPIRAN		179

DAFTAR TABEL

Tabel		Halaman
Tabel 4.1	Data Pencemaran Udara <i>Suspended Particulate Matter</i>	101
Tabel 4.2	Hasil Dekomposisi Level 1 Data Pencemaran Udara SPM	107
Tabel 4.3	Hasil Dekomposisi Level 2 Data Pencemaran Udara SPM	107
Tabel 4.4	Hasil Dekomposisi Level 3 Data Pencemaran Udara SPM.....	108
Tabel 4.5	Hasil Dekomposisi Level 4 Data Pencemaran Udara SPM.....	108
Tabel 4.6	Hasil Dekomposisi Level 5 Data Pencemaran Udara SPM.....	108
Tabel 4.7	Hasil Penyusunan Kembali Proses Dekomposisi Data Pencemaran Udara SPM	109
Tabel 4.8	Hasil Normalisasi Data dari Proses Dekomposisi	110
Tabel 4.9	Bobot dari Layar Masukan ke Layar Tersembunyi (vij)	113
Tabel 4.10	Bobot dari Layar Tersembunyi ke Layar Keluaran(wk)	114
Tabel 4.11	Bobot Inisialisasi Nguyen-Widrow Data Pencemaran Udara SPM	115
Tabel 4.12	Suku Perubahan Bobot ke Unit Tersembunyi Data Pencemaran Udara SPM	122

Tabel 4.13	Bobot Baru dari Layar Masukan ke Layar Tersembunyi	
	Data PENCEMARAN UDARA SPM	123
Tabel 4.14	Bobot Baru dari Layar Tersembunyi ke Layar Keluaran	
	Data PENCEMARAN UDARA SPM	124
Tabel 4.15	Data PENCEMARAN UDARA SPM Kota Semarang Tahun	
	2008 – 2015	141
Tabel 4.16	Dekomposisi Transformasi <i>Wavelet</i> Data PENCEMARAN UDARA	
	SPM Kota Semarang Tahun 2008 – 2015	142
Tabel 4.17	Normalisasi Data Hasil Dekomposisi Transformasi <i>Wavelet</i> ..	142
Tabel 4.18	Hasil Pelatihan dengan Variasi Arsitektur Jaringan	147
Tabel 4.19	Hasil Pelatihan dengan Variasi Momentum	149
Tabel 4.20	Hasil Analisis Variasi Jaringan	156
Tabel 4.21	Hasil Prediksi Dekomposisi <i>Wavelet</i> Bulan Januari	157
Tabel 4.22	Hasil Prediksi Dekomposisi <i>Wavelet</i> Bulan Februari	158

DAFTAR GAMBAR

Gambar 2.1 Contoh Citra 1 Dimensi	21
Gambar 2.2 Hasil Proses Transformasi Perataan dan Pengurangan dari Gambar 2.1	22
Gambar 2.3 Proses Perataan dan Pengurangan dengan Dekomposisi Penuh (3 level)	22
Gambar 2.4 Hasil Proses Dekomposisi Penuh.....	22
Gambar 2.5 Fungsi <i>Wavelet</i> Haar	24
Gambar 2.6 Grafik Fungsi Undak <i>Biner</i>	29
Gambar 2.7 Grafik Fungsi Undak <i>Biner</i> (<i>threshold</i>)	30
Gambar 2.8 Grafik Fungsi <i>Bipolar</i> (<i>symmetric hard limit</i>)	30
Gambar 2.9 Grafik Fungsi <i>Bipolar</i> dengan <i>Threshold</i>	31
Gambar 2.10 Grafik Fungsi <i>Sigmoid Biner</i> dengan <i>Range</i> (0,1)	32
Gambar 2.11 Grafik Fungsi <i>Sigmoid Bipolar</i> dengan <i>Range</i> (-1,1)	33
Gambar 2.12 Grafik Fungsi Identitas	34
Gambar 2.13 Model Jaringan Syaraf Tiruan Sederhana	35

Gambar 2.14 Jaringan <i>Layer Tunggal</i>	36
Gambar 2.15 Jaringan <i>Layer Jamak</i>	38
Gambar 2.16 Jaringan <i>Recurrent</i>	39
Gambar 2.17 Arsitektur Jaringan <i>Layer Jamak</i>	42
Gambar 2.18 Arsitektur Jaringan <i>Backpropagation</i>	44
Gambar 3.1 Kerangka Pikir	85
Gambar 3.2 <i>Flowchart</i> Proses Jaringan Syaraf Tiruan Berbasis Transformasi <i>Wavelet</i>	87
Gambar 3.3 <i>Flowchart</i> Dekomposisi Data Menggunakan <i>Wavelet Haar</i> ..	89
Gambar 3.4 <i>Flowchart</i> Pemodelan Data Menggunakan Jaringan Syaraf Tiruan <i>Backpropagation</i>	91
Gambar 3.5 <i>Flowchart</i> Prediksi data Menggunakan Jaringan Syaraf Tiruan <i>Backpropagation</i>	93
Gambar 3.6 <i>Flowchart</i> Rekonstruksi Data menggunakan <i>Wavelet Haar</i> ..	95
Gambar 3.7 Artisektur Jaringan Syaraf Tiruan 12-6-1 Data Pencemaran Udara SPM	98
Gambar 4.1 Artisektur Jaringan Syaraf Tiruan 12-6-1 Data Pencemaran Udara SPM	112

Gambar 4.2 Desain Tampilan <i>Form</i> Halaman Depan	132
Gambar 4.3 Desain Tampilan <i>Form</i> Diagram Alir	133
Gambar 4.4 Desain Tampilan <i>Form</i> Analisis Data	135
Gambar 4.5 Desain Tampilan <i>Form</i> Prediksi Tingkat Pencemaran Udara SPM Kota Semarang	139
Gambar 4.6 Grafik Hasil Keluaran JST Data Tingkat Pencemaran Udara SPM Kota Semarang	153
Gambar 4.7 Grafik Prediksi Dekomposisi <i>Wavelet</i> untuk Bulan Januari 2016	160
Gambar 4.8 Grafik Prediksi Dekomposisi <i>Wavelet</i> untuk Bulan Februari 2016	162



DAFTAR LAMPIRAN

Lampiran 1. Data Pencemaran Udara SPM Periode Bulan Januari 2008 – Desember 2015	179
Lampiran 2. Data Pencemaran Udara SPM Periode Bulan Januari 2008 - Desember 2015 Setelah Dilakukan Interpolasai Data	180
Lampiran 3. Data Pelatihan Jaringan Syaraf Tiruan	183
Lampiran 4. Data Pengujian Jaringan Syaraf Tiruan	185
Lampiran 5. Langkah Kerja	187
Lampiran 6. <i>Source Code</i> Halaman Depan	192
Lampiran 7. <i>Source Code</i> Diagram Alir	194
Lampiran 8. <i>Source Code</i> Proses Pelatihan Data SPM	198
Lampiran 9. <i>Source Code</i> Pengujian Data SPM	200
Lampiran 10. <i>Source Code</i> Prediksi Data Tingkat Pencemaran Udara SPM	202

BAB I

PENDAHULUAN

1.1 Latar Belakang

Banyak kota besar di Indonesia yang dilanda oleh permasalahan lingkungan, salah satunya adalah semakin memburuknya kualitas udara. Saat ini polusi udara merupakan bagian yang tidak terpisahkan dari kehidupan banyak kota besar yang ada di Indonesia. Walaupun beberapa kemajuan telah dicapai dalam pengendalian polusi udara dari dua dekade terakhir ini, namun kualitas udara terutama di kota-kota besar negara berkembang seperti Indonesia menjadi lebih buruk.

Polutan yang ada di udara berisiko terhadap kesehatan manusia. Efek terhadap kesehatan manusia dipengaruhi oleh intensitas dan lamanya kontak langsung dengan udara yang tercemar, selain itu juga dipengaruhi oleh status kesehatan penduduk yang mengalami kontak langsung dengan udara yang tercemar. Hal ini dapat dijelaskan bahwa keadaan lingkungan udara yang kurang menguntungkan akan memperburuk kondisi kesehatan seseorang diperburuk lagi (Kusnopranto H, 2000).

Berdasarkan data Dinas Kesehatan Kota Semarang, jenis penyakit yang disebabkan oleh pencemaran udara *Suspended Particulate Matter* (SPM) adalah penyakit Bronkitis akut, Pneumonia, dan penyakit Infeksi Saluran Nafas bagian Atas (ISPA) akut lainnya. Di Kota Semarang jumlah penderita pneumonia untuk usia kurang dari 1 tahun pada tahun 2014 mengalami peningkatan 58 kasus dari 1306 men-

jadi 1364, jumlah penderita pneumonia usia 1 – 4 tahun sebanyak 2880 menurun 240 kasus dibanding dengan tahun 2013, penderita pneumonia berat usia kurang dari 1 tahun sebanyak 12 balita menurun sebanyak 49 dari tahun sebelumnya dan jumlah pneumonia berat umur 1 – 4 tahun sebanyak 39 kasus. Jumlah ini cukup tinggi dan akan terus meningkat mengingat aktivitas kendaraan penyebab pencemaran udara di Kota Semarang semakin meningkat (Dinas Kesehatan Kota Semarang, 2015).

Saat ini, pemantauan kualitas udara di Kota Semarang adalah melalui *Air Quality Monitoring System* (AQMS) yang dilakukan oleh Badan Lingkungan Hidup (BLH). Pemantauan dilakukan dengan memasang jaringan pemantau kualitas udara ambien. Jaringan ini bekerja secara otomatis, dimana polutan akan diukur secara langsung di setiap jamnya kemudian dihitung menjadi nilai rata-rata dalam satu hari dan ditampilkan sebagai nilai yang terukur pada hari itu tanpa adanya prediksi mengenai informasi untuk kondisi esok hari.

Penelitian mengenai prediksi terhadap kualitas udara telah dilakukan oleh beberapa penelitian seperti Novie Fitriani Arifien (2012) dan Budi Warsito (2008). Penelitian yang dilakukan oleh Novie Fitriani Arifien (2012), yaitu memprediksi kadar polutan menggunakan jaringan syaraf tiruan (JST) untuk pemantauan kualitas udara di Kota Surabaya. Dalam penelitian ini, penulis menggunakan parameter Ozon troposfer (O_3) yang merupakan salah satu jenis polutan yang menjadi permasalahan udara di kota-kota besar di dunia, khususnya di daerah Surabaya. Model JST yang digunakan dalam penelitian ini yaitu tipe *multivariate* dan *time series*. Hasil dari kedua model tersebut dibandingkan untuk mendapatkan model terbaik. Parameter

pemilihan model didasarkan pada nilai *root mean square error* (RMSE) dan koefisien determinasi (R^2).

Penelitian Budi Warsito (2008) melakukan penyelesaian masalah *clustering* data menggunakan Jaringan Kohonen dipengaruhi oleh parameter-parameter pelatihan seperti jumlah *cluster* yang akan dibentuk, *learning rate*, maksimum iterasi (*epoch*) sehingga jika proses pelatihan dilakukan beberapa kali dengan data masukan yang sama, akan berpengaruh pada *clustering* data yang dihasilkan. Hal ini disebabkan pada Algoritma Kohonen untuk *clustering* data terdapat pemilihan data secara acak (randomisasi).

Dari penelitian yang telah dilakukan sebelumnya mengenai pencemaran udara, maka dapat diketahui bahwa data pencemaran udara merupakan data yang berupa data runtun waktu, dimana dalam memprediksi tingkat pencemaran udara untuk periode mendatang dibutuhkan data periode sebelumnya. Berdasarkan hal ini, penelitian mengenai data runtun waktu telah dilakukan oleh beberapa penelitian seperti Laeli Ulfiati (2015) dan Dwi Prisita Anggitiningrum (2013). Penelitian yang dilakukan oleh Laeli Ulfiati (2015) yaitu peramalan data saham dengan transformasi *Wavelet* Haar. Pada penelitian ini, peneliti melakukan pengujian data harga saham menggunakan metode *Wavelet* Haar dengan DWT (*Discrete Wavelet Transform*) menggunakan *software* Matlab R2013a dan bagaimana nilai MSEnya. Langkah melakukan peramalan harga saham adalah memilih data saham, mengplot data saham, membagi data saham menjadi data *training* dan data *testing*, kemudian melakukan peramalan.

Penelitian yang dilakukan oleh Dwi Prisita Anggitiningrum (2013) yaitu perbandingan prediksi harga saham dengan menggunakan Jaringan Syaraf Tiruan *Backpropagation* dan Box-Jenkins (ARIMA). Hasil simulasi data menggunakan jaringan syaraf tiruan *backpropagation* diperoleh nilai MSE dari proses *training* sebesar 0,000206 dan proses *testing* sebesar 0,00140. Hasil analisis data menunjukkan bahwa nilai MSE ARIMA lebih kecil dari nilai MSE jaringan syaraf tiruan *backpropagation*. Namun, selisih nilai MSE dari kedua metode tersebut tidak terlalu besar. Sehingga kedua metode tersebut dapat digunakan untuk penelitian prediksi harga saham.

Penggunaan *wavelet* untuk peramalan data *time series* khususnya pada pencemaran udara telah mengalami perkembangan pesat. Transformasi *wavelet* yang dipandang lebih sesuai untuk data *time series* adalah *Discrete Wavelet Transform* (DWT) karena dalam setiap level dekomposisi terdapat koefisien *wavelet* dan skala sebanyak panjang data. Kelebihan ini mereduksi kelemahan pemfilteran dengan DWT (*Discrete Wavelet Transform*) yang dapat dilakukan pada sebarang ukuran sampel. Penentuan level dekomposisi dan koefisien yang digunakan sebagai *input* model menggunakan dekomposisi multi skala. Pengembangan yang dilakukan pada tulisan ini adalah penyempurnaan pada teknik komputasi sehingga level dekomposisi dan banyaknya koefisien pada setiap level dapat terpilih secara otomatis berdasarkan nilai prediksi yang meminimalkan *error*. Keuntungan menggunakan metode *wavelet* adalah secara otomatis memisahkan tren dari data dan menunjukkan komponen musiman datanya.

Transformasi *wavelet* merupakan perbaikan dari transformasi fourier. Pada transformasi fourier hanya dapat menangkap informasi apakah suatu sinyal memiliki frekuensi tertentu atau tidak, tetapi tidak dapat menangkap dimana frekuensi itu terjadi, sedangkan transformasi *wavelet* dapat memberikan informasi tentang kombinasi frekuensi dan skala (waktu). *Wavelet* memiliki prinsip dasar membagi data menjadi beberapa komponen frekuensi yang berbeda-beda, kemudian dilakukan analisis untuk masing-masing komponen menggunakan resolusi yang sesuai dengan skalanya (Sianipar, 2013: 140).

JST merupakan cabang ilmu yang memiliki karakteristik mirip dengan jaringan syaraf biologi. JST dibentuk sebagai generalisasi model matematika dari jaringan syaraf otak manusia, sehingga dapat melakukan proses pembelajaran. JST dapat diaplikasikan dalam beberapa masalah, diantaranya pengenalan pola, pemrosesan sinyal, dan peramalan terhadap apa yang terjadi dimasa yang akan datang berdasarkan generalisasi dari pola yang sudah terjadi sebelumnya (Siang, 2009: 5).

Berdasarkan penelitian Novie Fitriani Arifien (2012) dan Budi Warsito (2008) dalam menyelesaikan masalah pencemaran udara dapat digunakan metode Jaringan Syaraf Tiruan. Dan berdasarkan penelitian Laeli Ulfiati (2015) dan Dwi Prisita Anggitiningrum (2013) dimana pencemaran udara merupakan data yang berbentuk data runtun waktu, penggunaan Jaringan Syaraf Tiruan dan Transformasi *Wavelet* cocok digunakan untuk data runtun waktu.

Berdasarkan uraian tersebut, maka penulis mengambil judul “Jaringan Syaraf Tiruan berbasis Transformasi *Wavelet* untuk Memprediksi Tingkat Pencemaran Udara *Suspended Particulate Matter* di Kota Semarang” melalui penggunaan metode ini, diharapkan dapat memberikan hasil prediksi yang baik sehingga dapat memberikan informasi mengenai kualitas udara di Kota Semarang terutama konsentrasi *Suspended Particulate Matter* dan dapat dijadikan sebagai salah satu upaya untuk meningkatkan sistem pemantauan kualitas udara yang ada di Kota Semarang.

1.2 Rumusan Masalah

Berdasarkan permasalahan yang telah diuraikan pada bagian latar belakang, maka permasalahan yang timbul antara lain.

1. Bagaimana menciptakan suatu sistem yang dapat mengetahui model prediksi tingkat pencemaran udara (SPM) dengan menggunakan metode jaringan syaraf tiruan berbasis transformasi *wavelet*?
2. Bagaimana prediksi tingkat pencemaran udara (SPM) di Kota Semarang untuk periode bulan Januari dan Februari 2016 dengan menggunakan metode jaringan syaraf tiruan berbasis transformasi *wavelet*?

1.2 Batasan Masalah

Berdasarkan permasalahan yang telah diuraikan, batasan masalah dalam penelitian ini yaitu mengkaji faktor pencemaran udara *Suspended Particulate Matter*

(SPM) dari data bulanan mulai tahun 2008 sampai dengan tahun 2015 yang diperoleh dari data *Badan Meteorologi, Klimatologi dan Geofisika* Kota Semarang, Jawa Tengah. Pada tahap selanjutnya adalah menciptakan suatu sistem untuk mengetahui tingkat pencemaran udara berdasarkan data yang ada menggunakan metode jaringan syaraf tiruan berbasis transformasi *wavelet* dengan bantuan *software* Matlab R2014a.

1.3 Tujuan Penelitian

Berdasarkan permasalahan di atas, maka tujuan penelitian ini adalah sebagai berikut.

1. Merancang suatu sistem yang dapat mengetahui model prediksi tingkat pencemaran udara (SPM) dengan menggunakan jaringan syaraf tiruan berbasis transformasi *wavelet*.
2. Diketahui tingkat pencemaran udara (SPM) di Kota Semarang dengan menggunakan jaringan syaraf tiruan berbasis transformasi *wavelet* dengan menggunakan *tools* Matlab untuk periode bulan Januari dan Februari 2016.

1.4 Manfaat Penelitian

Manfaat yang bisa diperoleh dari penelitian ini adalah sebagai berikut.

1. Bagi Ilmu Pengetahuan

Hasil penelitian ini diharapkan dapat menambah pengetahuan tentang tingkat pencemaran udara (SPM) Kota Semarang yang mempengaruhi kejadian Penyakit Infeksi Saluran Pernapasan Atas (ISPA). Dan diharapkan dapat berguna bagi peneliti lainnya untuk melakukan penelitian lebih lanjut.

2. Bagi Peneliti

- a. Peneliti dapat membuat suatu sistem yang dapat mengetahui tingkat pencemaran udara (SPM) dengan jaringan saraf tiruan berbasis transformasi *wavelet*.
- b. Peneliti dapat mengetahui tingkat keakuratan sistem yang digunakan untuk mendeteksi pencemaran udara (SPM) berdasarkan gejala-gejala yang ada.
- c. Melalui penelitian ini diharapkan menjadi bahan proses belajar bagi peneliti, menambah pengalaman serta dapat menambah pengetahuan dan wawasan dalam hal *suspended partikulat matter* (SPM) yang mempengaruhi kejadian Infeksi Saluran Pernapasan Atas di Kota Semarang.

3. Bagi Mahasiswa Matematika

Penelitian ini dapat memberikan inspirasi dan gagasan baru bagi mahasiswa untuk mengembangkan dan merancang aplikasi menggunakan jaringan saraf tiruan berbasis transformasi *wavelet* yang nantinya dapat dimanfaatkan di berbagai bidang ilmu pengetahuan sehingga berguna dalam kehidupan sehari-hari.

4. Bagi Badan Meteorologi, Klimatologi dan Geofisika

Hasil penelitian ini dapat bermanfaat bagi instansi yang berkaitan dengan lingkungan, seperti Badan Meteorologi, Klimatologi dan Geofisika dalam mendiagnosis tingkat pencemaran udara di Kota Semarang serta memberikan informasi kepada pemerintah dan masyarakat pada umumnya untuk menjaga lingkungan dari pencemaran udara yang dapat menimbulkan berbagai gangguan pada kesehatan.

1.5 Sistematika Penelitian

Sistematika penulisan skripsi disusun untuk memudahkan pemahaman tentang struktur dan isi skripsi. Penulisan skripsi ini dibagi menjadi dua bagian, yaitu, bagian awal skripsi, dan bagian pokok skripsi.

1.5.1 Bagian Awal

Dalam penulisan skripsi ini, bagian awal berisi halaman judul, halaman pengesahan, motto dan persembahan, kata pengantar, abstrak, daftar isi, daftar tabel, daftar gambar, dan daftar lampiran.

1.5.2 Bagian Pokok

Bagian pokok dari penulisan skripsi ini adalah isi skripsi yang terdiri atas lima bab yaitu:

(1) **BAB I PENDAHULUAN**

Berisi uraian mengenai hal-hal yang melatar belakangi penulisan, perumusan masalah, batasan masalah, tujuan dan manfaat, serta sistematika penulisan skripsi,

(2) **BAB II LANDASAN TEORI**

Berisi uraian mengenai beberapa penelitian yang telah dilakukan sebelumnya yang berkaitan dengan Pencemaran Udara, penelitian tentang Jaringan Syaraf Tiruan dan penelitian tentang Transformasi *Wavelet*. Landasan Teori berisi uraian teori-teori yang mendukung penelitian ini.

(3) **BAB III METODE PENELITIAN**

Berisi tentang prosedur atau langkah-langkah yang dilakukan dalam penelitian ini meliputi studi pustaka, perumusan masalah, pemecahan masalah, analisis pemecahan masalah, dan penarikan kesimpulan.

(4) **BAB IV HASIL PENELITIAN DAN PEMBAHASAN**

Berisi tentang hasil-hasil penelitian yang telah dilakukan setelah dianalisis dengan rumus yang sesuai dan pembahasan hasil penelitian tersebut.

(5) **BAB V SIMPULAN DAN SARAN**

Berisi tentang simpulan dan saran-saran dari hasil penelitian yang telah dilakuk



BAB II

TINJAUAN PUSTAKA

2.1 Jaringan Syaraf Tiruan Berbasis Transformasi

Wavelet

Menurut Yusuf dkk (2015), Jaringan Syaraf Tiruan berbasis Transformasi *Wavelet* merupakan gabungan (*hybrid model*) dari Transformasi *Wavelet* dan Jaringan Syaraf Tiruan. Dalam pembangunan model jaringan syaraf tiruan berbasis transformasi *wavelet*. Model dari transformasi *wavelet* ini berfungsi untuk menghilangkan *noise* pada data yang akan menghasilkan pola data yang lebih sederhana agar dapat memudahkan jaringan syaraf tiruan dalam mengenali data yang pada akhirnya dapat bertujuan menghasilkan model yang lebih bagus.

2.2 *Wavelet*

Wavelet adalah suatu metode pengolahan sinyal dimana sinyal yang diolah dipecah menjadi beberapa bagian yang menunjuk pada frekuensi yang berbeda-beda. *Wavelet* biasa digunakan untuk mengkaji data atau informasi atau operator dalam komponen-komponen frekuensi yang berlainan, dan kemudian mengkaji setiap komponen dengan suatu resolusi yang sesuai dengan skalanya (Daubechies, 1995).

Fungsi *wavelet* adalah suatu fungsi yang osilasi disekitar nol (seperti fungsi sinus dan cosinus). Fungsi *wavelet* dibedakan atas dua jenis, yaitu *wavelet* ayah (Φ)

dan *wavelet* ibu (Ψ) yang mempunyai $\int_{-\infty}^{\infty} \Phi(x)dx = 1$ dan $\int_{-\infty}^{\infty} \Psi(x)dx = 0$ dengan dilatasi dan translasi, *wavelet* ayah dan ibu melahirkan keluarga *wavelet* yaitu

(2.1)

$$\Phi_{j,k}(x) = (p2^j)^{\frac{1}{2}}\Phi(p2^jx - k)$$

dan

$$\Psi_{j,k}(x) = (p2^j)^{\frac{1}{2}}\Psi(p2^jx - k)$$

Untuk suatu skalar $p > 0$ dan tanpa mengurangi keumuman dapat diambil $p = 1$ sehingga

(2.2)

$$\Phi_{j,k}(x) = 2^{\frac{j}{2}}\Phi(2^jx - k)$$

dan

$$\Psi_{j,k}(x) = 2^{\frac{j}{2}}\Psi(2^jx - k)$$

Alihragam *wavelet* terdiri atas himpunan yang tak-berhingga banyaknya, keluarga *wavelet* yang berbeda akan mempunyai faktor keuntungan-kerugian sendiri antara bagaimana fungsi-fungsi basis teralokasi dalam ruang dan seberapa halus fungsi tersebut. Beberapa jenis *wavelet* mempunyai struktur bagian lebih kecil.

2.2.1 Transformasi *Wavelet*

Transformasi *wavelet* merupakan perbaikan dari transformasi *fourier*. Pada transformasi *fourier* hanya dapat menangkap informasi apakah suatu sinyal memiliki frekuensi tertentu atau tidak, tetapi tidak dapat menangkap dimana frekuensi itu terjadi, sedangkan transformasi *wavelet* dapat memberikan informasi tentang kombinasi frekuensi dan skala (waktu). *Wavelet* memiliki prinsip dasar membagi data menjadi beberapa komponen frekuensi yang berbeda-beda, kemudian dilakukan analisis untuk masing-masing komponen menggunakan resolusi yang sesuai dengan skalanya (Sianipar, 2013:140). Proses ini dinamakan dekomposisi. Transformasi sinyal menjadi koefisien-koefisien *wavelet* diperoleh dengan menapis sinyal menggunakan *highpass filter* dan *lowpass filter*.

2.2.2 Transformasi *Wavelet* Diskrit

Transformasi *wavelet diskrit* terdiri dari pasangan transformasi yang bersifat kebalikan (*reversible*), yaitu transformasi *wavelet* diskrit maju (*forward DWT*) dan transformasi *wavelet* balik (*inverse DWT*). Karena bersifat multiresolusi, maka DWT dapat dilakukan sesuai dengan keinginan pengguna. DWT dapat diimplementasikan oleh sepasang *Quadrature Mirror Filter* (QMF). Dalam hal ini, hasil analisis terhadap data pada skala dan resolusi tertentu akan menghasilkan subband-subband detail data (*subband horizontal*, *subband vertical* dan *subband diagonal*) serta

pendekatan nilai data pada resolusi tersebut. Adapun jenis filter yang digunakan adalah *lowpass filter* dan *highpass filter*.

Dalam analisis sebuah data runtun waktu $\{X_t\}$, *discrete wavelet transform/* transformasi *wavelet* diskrit (DWT) memiliki sifat transformasi ortonormal linear. Dalam analisis komputasi ada $\{W_n: n = 0, \dots, N - 1\}$ menyatakan koefisien DWT, dengan notasi $W = WX$, dengan W adalah vektor kolom dengan panjang $N = 2^J$ dengan elemen ke- n adalah koefisien DWT ke- n pada W_n , dan W adalah sebuah matriks bernilai riil berukuran $N \times N$. Sifat ortonormalitas menyatakan bahwa $X = W^T W$ dan $\|W\|^2 = \|X\|^2$. Oleh karena itu W_n^2 menunjukkan besarnya pengaruh perubahan untuk masing-masing koefisien indeks ke- n .

Transformasi *wavelet* diskrit (DWT) dapat direalisasikan dengan menggunakan teknik filterisasi digital. Secara garis besar proses dalam teknik ini adalah dengan melewati sinyal yang akan dianalisis pada filter dengan frekuensi dan skala yang berbeda. Analisis sinyal dilakukan terhadap hasil filterisasi *highpass filter* dan *lowpass filter* dimana *highpass filter* digunakan untuk menganalisis frekuensi tinggi dan *lowpass filter* digunakan untuk menganalisis frekuensi rendah. Pembagian sinyal menjadi frekuensi tinggi dan frekuensi rendah dalam proses filterisasi *highpass filter* dan *lowpass filter* disebut sebagai dekomposisi (Sianipar, 2013).

Pasangan filter *highpass* dan *lowpass* yang digunakan harus merupakan *quadrature mirror filter* (QMF), yaitu pasangan filter yang memenuhi persamaan berikut

(2.3)

$$h[L - 1 - n] = (-1)^n \cdot g[n]$$

dengan $h[n]$ adalah filter *highpass*, $g[n]$ adalah filter *lowpass* dan L adalah panjang masing-masing filter.

Dari suatu *wavelet* terdapat sebanyak berhingga genap nilai-nilai dengan sifat istimewa. Himpunan nilai-nilai dimaksud disebut filter *wavelet*, yang biasa dituliskan seperti pada persamaan 2.4 dengan sifat seperti dituliskan pada persamaan 2.5 (Percieval dan Walden, 2000).

(2.4)

$$h = [h_0, h_1, \dots, h_{L-1}]$$

yang memenuhi sifat-sifat persamaan (2.5) berikut

(2.5)

$$\sum_{i=1}^{L-1} h_i = 0$$

$$\sum_{i=1}^{L-1} h_i^2 = 1$$

$$\sum_{i=0}^{L-1} h_i h_{i+2n} = 0, n \in Z$$

Sebagai pasangan dari filter *wavelet* terdapat filter skala yang dituliskan sebagai $g = [g_0, g_1, \dots, g_{L-1}]$. Antara h dan g terdapat hubungan seperti dituliskan pada persamaan berikut

(2.6)

$$g_l = (-1)^{l+1} h_{L-1-l}$$

Filter yang diulas pada uraian sebelumnya adalah filter pada tingkat pertama, selanjutnya dinotasikan dengan $h^{(1)}$. Sebelum membangun filter pada tingkatan yang lebih tinggi terlebih dahulu dibentuk *up sampled* dari filter tingkat sebelumnya yaitu dengan cara menyisipkan 0 diantara nilai filter yang tidak sama dengan 0. Jadi *up sampled* dari 2.4 adalah

(2.7)

$$h_{up}^1 = [h_0, 0, h_1, 0, \dots, 0, h_{L-1}, 0, h_{L-1}]$$

Filter pada tingkat 2 dibentuk dengan rumus berikut

(2.8)

$$h^{(2)} = h_{up}^1 * g$$

Dengan $*$ menyatakan operasi konvolusi. Secara umum filter *wavelet* dan filter skala pada tingkat j dibentuk dengan rumus pada persamaan berikut

(2.9)

$$h^{(j)} = h_{up}^{(j-1)} * g$$

$$g^{(j)} = g_{up}^{(j-1)} * g$$

Filter *wavelet* dan filter skala secara bersama-sama dapat membangun matriks transformasi *wavelet* diskrit. Hal ini berakibat setiap realisasi diskrit dari fungsi dalam $L_2(R)$ dengan jeda waktu yang sama dapat didekomposisi (dipecah) ke dalam bagian halus (S) dan bagian-bagian detil (D). Proses untuk memperoleh S dan D disebut transformasi *wavelet* diskrit (*Discrete Wavelet Transform* = DWT). Misalkan $Y = \{Y_t\}_{t=1}^N$ menyatakan realisasi diskrit yang dimaksud dengan $N > L$ dan $N = 2^j$ untuk suatu bilangan asli J. Transformasi *wavelet* diskrit pada tingkat j dapat dituliskan sebagai persamaan 2.10

(2.10)

$$D = H_j Y$$

H_j adalah matriks transformasi tingkat ke-j berukuran $N \times N$ dan D adalah hasil dekomposisi yang disebut juga sebagai matriks koefisien dengan ukuran $N \times 1$. Baris ke-1 sampai dengan baris ke- $\frac{n}{2}$ dari matriks H adalah bentuk periodisasi 2 langkah dari $h^{(1)}$ seperti dituliskan pada persamaan

$$H_1 = \begin{bmatrix} h_1 & h_0 & 0 & 0 & \dots & 0 & 0 & h_{L-1} & h_{L-2} & \dots & h_3 & h_2 \\ h_3 & h_2 & h_1 & h_0 & 0 & \dots & 0 & h_{L-1} & h_{L-2} & \dots & h_4 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \dots & \dots & 0 & \dots & h_{L-1} & h_{L-1} & \dots & h_1 & h_0 \end{bmatrix}$$

Pada matriks H_1 baris ke- $\frac{N}{2} + 1$ sampai dengan baris ke- N adalah bentuk periodisasi 2 langkah filter skala g

$$G_1 = \begin{bmatrix} g_1 & g_0 & 0 & 0 & \dots & 0 & 0 & g_{L-1} & g_{L-2} & \dots & g_3 & g_2 \\ g_3 & g_2 & g_1 & g_0 & 0 & \dots & 0 & 0 & g_{L-1} & g_{L-2} & \dots & g_4 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \dots & \dots & 0 & \dots & g_{L-1} & g_{L-1} & \dots & g_1 & g_0 \end{bmatrix}$$

Pada matriks H_2 sub matriks G_1 akan didekomposisikan (dipecah) menjadi H_2 dan G_2 yang masing-masing merupakan bentuk periodisasi 4 langkah dari filter *wavelet* dan filter skala tingkat 2. Proses dapat dilanjutkan sampai dengan tingkat ke- j , untuk $j \leq J$. Dalam hal ini H_j dan G_j masing-masing merupakan bentuk periodisasi 2^j langkah dari filter *wavelet* dan filter skala tingkat j . lebih lanjut persamaan 2.10 dapat dituliskan kembali seperti pada persamaan 2.11

(2.11)

$$D = [H_1 G_1]^T Y = [H_1 H_2 G_2]^T Y = [H_1 H_2 \dots H_j G_j]^T Y = [D_1 S_1]^T = [D_1 D_2 S_2]^T = \dots \\ = [D_1 D_2 \dots D_j S_j]^T$$

2.2.3 Dekomposisi *Wavelet*

Dalam pembuatan data latihan untuk *wavelet*, langkah-langkah yang harus dilakukan adalah sebagai berikut :

1. Memuat data *input*

Memasukkan data *input* yang telah diolah dan disimpan dalam bentuk mat-file.

2. Menentukan jenis transform *wavelet*, yang digunakan pada metode ini adalah *Discrete Wavelet Transform*. Hal ini disebabkan data *input* berupa diskrit.
3. Menentukan *mother wavelet* yaitu Haar.
4. Menentukan syntax yang akan digunakan pada matlab R2014a.

Proses DWT merupakan proses dekomposisi *wavelet* yang berguna untuk membagi tiap inputan data sesuai dengan *mother wavelet* yang diinginkan.

Tahap awal dari proses dekomposisi adalah menentukan bagian nilai frekuensi rendah dan frekuensi tinggi dari data inputan, yaitu dengan cara memasukkan hasil inputan dalam bagian filter bawah atau *low pass filter* untuk mendapatkan nilai frekuensi tinggi dan kemudian melewatkan inputan ke dalam filter atas atau *high pass filter* untuk mendapatkan nilai hasil frekuensi rendah. Hasil keluaran dari dua bagian ini kemudian mengalami proses *downsampling*.

Fungsi *wavelet* dapat dinyatakan sebagai kombinasi linear dari basis yang dibangun oleh *wavelet* atau dapat dituliskan dalam persamaan 2.12 berikut

(2.12)

$$f(x) = \sum_{k \in \mathbb{Z}} c_{j,k} \Phi_{j,k}(x) + \sum_{j < 1} d_{j,k} \Psi_{j,k}(x)$$

dengan

(2.13)

$$c_{j,k} = \int f(x)\Phi_{j,k}(x)dx$$

(2.14)

$$d_{j,k} = \int f(x)\Psi_{j,k}(x)dx$$

Transformasi pada persamaan 2.12 merupakan transformasi *wavelet* kontinu atau *Continue Wavelet Transform* (CWT) dimana koefisien-koefisien *wavelet* diperoleh melalui proses integrasi sehingga nilai *wavelet* harus terdefinisi pada setiap $x \in R$.

Bentuk transformasi yang lain adalah *Discrete Wavelet Transform* (DWT) dimana nilai-nilai *wavelet* hanya terdefinisi pada titik-titik diskret. Vektor yang memuat nilai-nilai *wavelet* disebut filter *wavelet*. Ada dua jenis filter pada DWT yaitu filter *wavelet* (filter detil) dinotasikan dengan h dan filter skala yang dinotasikan dengan g . panjang suatu filter dinotasikan dengan L . Suatu filter *wavelet* harus memenuhi tiga sifat dasar 2.5.

2.2.4 Dekomposisi *Averaging* dan *Differencing*

Putra (2010) Dekomposisi perataan (*averages*) dan pengurangan (*differencing*) memegang peranan penting untuk memahami transformasi *wavelet*.

Gambar 2.1 merupakan contoh dari dekomposisi perataan dan pengurangan pada citra 1 dimensi dengan dimensi 8.

37	35	28	28	58	18	21	15
----	----	----	----	----	----	----	----

Gambar 2.1. Contoh Citra 1 Dimensi (Putra, 2010).

Perataan dilakukan dengan menghitung nilai rata-rata 2 pasang data dengan rumus

(2.15)

$$p = \frac{x + y}{2}$$

sedangkan pengurangan dilakukan dengan rumus

(2.16)

$$p = \frac{x - y}{2}$$

hasil proses perataan untuk citra di atas adalah

$$\frac{37 + 35}{36} \quad \frac{28 + 28}{28} \quad \frac{58 + 18}{38} \quad \frac{21 + 15}{18}$$

sedangkan hasil proses pengurangannya adalah

$$\frac{37 - 35}{1} \quad \frac{28 - 28}{0} \quad \frac{58 - 18}{20} \quad \frac{21 - 15}{3}$$

Hasil dari proses dekomposisi perataan dan pengurangan terhadap citra asli di atas disajikan pada Gambar 2.2.

36	28	38	18	1	0	20	3
----	----	----	----	---	---	----	---

Gambar 2.2. Hasil Proses Transformasi Perataan dan Pengurangan dari Gambar 2.1 (Putra, 2010).

Proses dekomposisi yang dilakukan di atas 1 kali (1 level) saja. Gambar 2.3 menunjukkan proses transformasi penuh dan berhenti setelah tersisa 1 piksel saja.

37	35	28	28	58	18	21	15
36	28	38	18	1	0	20	3
32	28	4	10				
30	2						

Gambar 2.3. Proses Perataan dan Pengurangan dengan Dekomposisi Penuh (3 level) (Putra, 2010).

Pada setiap level, proses dekomposisi hanya dilakukan pada bagian hasil proses perataan. Hasil proses dekomposisi adalah gabungan dari proses perataan dengan seluruh hasil proses pengurangan. Gambar 2.4 menunjukkan citra hasil dekomposisi penuh di atas.

30	2	4	10	1	0	20	3
----	---	---	----	---	---	----	---

Gambar 2.4. Hasil Proses Dekomposisi Penuh (Putra, 2010).

Pada citra berukuran 2^n maka dibutuhkan sebanyak n level untuk melakukan dekomposisi penuh sehingga dapat dikatakan kompleksitas. Proses rekonstruksi sepenuhnya merupakan kebalikan dari proses dekomposisi sesuai dengan tingkatan pada proses dekomposisi. Proses rekonstruksi sepenuhnya merupakan kebalikan dari proses dekomposisi sesuai dengan tingkatan pada proses dekomposisi (Putra, 2010).

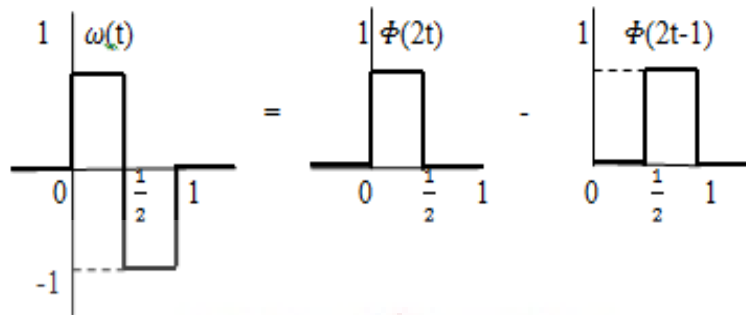
2.2.5 Wavelet Haar

Haar adalah *wavelet* paling tua dan paling sederhana, diperkenalkan oleh Alfred Haar pada tahun 1909. *Wavelet* Haar merupakan keluarga *wavelet* yang paling sederhana dan paling cepat dikomputasi. Haar telah menjadi sumber ide bagi munculnya keluarga *wavelet* lainnya seperti daubechies dan lain sebagainya. Transformasi menggunakan fungsi Haar sebagai fungsi basis merupakan transformasi *wavelet* yang paling sederhana. Fungsi Haar menurut (Mallat, 1998:7) didefinisikan sebagai

(2.17)

$$\psi(x) = \begin{cases} 1, & 0 \leq x < \frac{1}{2} \\ -1, & \frac{1}{2} \leq x < 1 \\ 0, & \text{lainnya} \end{cases}$$

Berikut disajikan fungsi *wavelet* Haar $\varphi(t)$ dapat dilihat pada Gambar 2.5.



Gambar 2.5. Fungsi *Wavelet* Haar

Transformasi *wavelet* Haar merupakan transformasi *wavelet* paling sederhana. Transformasi *wavelet* Haar dilakukan dengan menggunakan penapis lolos rendah (*Low Pass Filter/LPF*) dan penapis lolos tinggi (*High Pass Filter/HPF*) sehingga diperoleh koefisien *wavelet*.

Koefisien transformasi $h_0 = (h_0(0), h_0(1)) = (\frac{1}{2}, \frac{1}{2})$ (tapis *low pass*) dan $h_1 = (h_1(0), h_1(1)) = (\frac{1}{2}, \frac{1}{2})$ (tapis *high pass*) yang dibahas sebelumnya sebenarnya merupakan fungsi basis *wavelet* Haar. Begitu juga dengan dekomposisi perataan dan pengurangan sebenarnya sama dengan melakukan dekomposisi (transformasi) dengan *wavelet* Haar (Putra, 2010).

Untuk *wavelet* Haar itu sendiri tidak menggunakan data pelatihan dan hanya menggunakan data citra uji saja karena pengklasifikasian *wavelet* Haar menggunakan *Low Pass Filter (LPF)* dan *High Pass Filter (HPF)* dalam mengidentifikasi. *Wavelet* Haar merupakan *wavelet* sederhana yang berdasarkan pada fungsi tangga. Fungsi tangga tersebut didefinisikan sebagai berikut

(2.18)

$$\chi_{[a,b)}(t) = \begin{cases} 1, & \text{jika } a \leq t \leq b \\ 0 & \text{lainnya} \end{cases}$$

selanjutnya

(2.19)

$$\Phi_{j,k}(t) = 2^{\frac{j}{2}} \Phi(2^j t - k) \quad j, k \in Z$$

dimana $\{\Phi_{j,k}(t)\}_{j,k \in Z}$ merupakan fungsi skala Haar pada R.

Kemudian diberikan fungsi *wavelet* Haar $\Psi(t)$, merupakan *mother wavelet*

(2.20)

$$\Psi(t) = \chi_{[0, \frac{1}{2})}(t) - \chi_{[\frac{1}{2}, 1)}(t)$$

dan

(2.21)

$$\Psi_{j,k}(t) = 2^{\frac{j}{2}} \Psi(2^j t - k) \quad j, k \in Z$$

Himpunan $\{\Psi_{j,k}(t)\}_{j,k \in Z}$ disebut dengan fungsi keluarga *wavelet* Haar.

2.3 Jaringan Syaraf Tiruan

Jaringan syaraf tiruan bisa dibayangkan seperti otak buatan di dalam cerita-cerita fiksi ilmiah. Otak buatan ini dapat berpikir seperti manusia, dan juga sependai

manusia dalam menyimpulkan sesuatu dari potongan-potongan informasi yang diterima. Khayalan manusia tersebut mendorong para peneliti untuk mewujudkannya. Komputer diusahakan agar bisa berpikir sama seperti cara berpikir manusia. Caranya adalah dengan melakukan peniruan terhadap aktivitas-aktivitas yang terjadi di dalam sebuah jaringan syaraf biologis (Puspitaningrum, 2006).

Menurut (Siang, 2005), JST dibentuk sebagai generalisasi model matematika dari jaringan syaraf biologi, dengan asumsi bahwa :

1. Pemrosesan informasi terjadi pada banyak elemen sederhana (neuron).
2. Sinyal dikirimkan diantara neuron-neuron melalui penghubung-penghubung.
3. Penghubung antar neuron memiliki bobot yang akan memperkuat atau memperlemah sinyal.
4. Untuk menentukan *output*, setiap neuron menggunakan fungsi aktivasi (biasanya bukan fungsi linear) yang dikenakan pada jumlahan *input* yang diterima. Besarnya *output* ini selanjutnya dibandingkan dengan suatu batas ambang.

Beberapa istilah dalam JST yang sering ditemui adalah sebagai berikut.

1. Neuron atau *node* atau unit: sel syaraf tiruan yang merupakan elemen pengolahan jaringan syaraf tiruan. Setiap neuron menerima data *input*, memproses *input* tersebut kemudian mengirimkan hasilnya berupa sebuah *output*.
2. Jaringan: kumpulan neuron yang saling terhubung dan membentuk lapisan.

3. Lapisan tersembunyi (*hidden layer*): lapisan yang tidak secara langsung berinteraksi dengan dunia luar. Lapisan ini memperluas kemampuan jaringan syaraf tiruan dalam menghadapi masalah-masalah yang kompleks.
4. *Input*: sebuah nilai *input* yang akan diproses menjadi nilai *output*.
5. *Output*: solusi dari nilai *input*.
6. Bobot: nilai matematis dari sebuah koneksi antar neuron.
7. Fungsi aktivasi: fungsi yang digunakan untuk meng-*update* nilai-nilai bobot per-iterasi dari semua nilai *input*.
8. Fungsi aktivasi sederhana adalah mengalikan *input* dengan bobotnya dan kemudian menjumlahkannya (disebut penjumlahan sigma) berbentuk linier atau tidak linier dan *sigmoid*.
9. Paradigma pembelajaran: bentuk pembelajaran, *supervised learning*, atau *unsupervised learning*.

JST ditentukan oleh tiga hal, yaitu

1. Pola hubungan antar neuron (disebut arsitektur jaringan).
2. Metode untuk menentukan bobot penghubung (disebut metode *training/learning/algorithm*).
3. Fungsi aktivasi (fungsi transfer).

Neuron adalah unit pemroses informasi yang menjadi dasar dalam pengoperasian jaringan syaraf tiruan (Siang, 2009). *Neuron* terdiri atas tiga elemen pembentuk sebagai berikut.

1. Himpunan unit-unit yang dihubungkan dengan jalur koneksi.

2. Suatu unit penjumlah yang akan menjumlahkan masukan-masukan sinyal yang sudah dikalikan dengan bobotnya.
3. Fungsi aktivasi yang menentukan apakah sinyal dari *input* neuron akan diteruskan ke neuron lain ataukah tidak.

Neuron dalam jaringan syaraf tiruan sering diganti dengan istilah simpul. Setiap simpul tersebut berfungsi untuk menerima atau mengirim sinyal dari atau ke simpul-simpul lainnya. Pengiriman sinyal disampaikan melalui penghubung. Kekuatan hubungan yang terjadi antara setiap simpul yang saling terhubung dikenal dengan nama bobot.

2.3.1 Fungsi Aktivasi

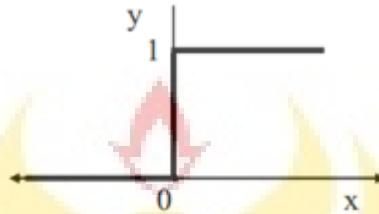
Fungsi aktivasi merupakan salah satu komponen yang terpenting dalam membangun sebuah jaringan syaraf tiruan. Hal ini karena fungsi aktivasi sangat berpengaruh terhadap *output* atau nilai keluaran dari jaringan itu sendiri. Ada beberapa fungsi aktivasi yang biasa digunakan dalam membangun jaringan syaraf tiruan, antara lain:

1. Fungsi undak *biner* (*hard limit*)

Fungsi undak *biner* (*hard limit*) biasanya dipakai pada jaringan berlayar tunggal. Fungsi ini akan mengkonversi masukan yang bernilai kontinu menjadi *output biner* (0 atau 1). Fungsi undak biner dapat dilihat pada Gambar 2.6 Fungsi undak *biner* (*hard limit*) dirumuskan dengan

(2.22)

$$y = \begin{cases} 0, & \text{jika } x \leq 0 \\ 1, & \text{jika } x > 0 \end{cases}$$



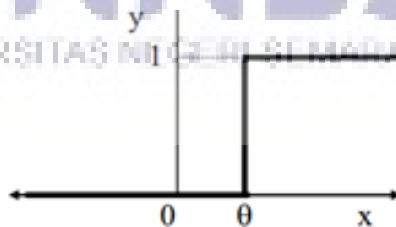
Gambar 2.6. Grafik Fungsi Undak Biner

2. Fungsi undak biner dengan *threshold*

Fungsi undak biner dengan *threshold* pada Gambar 2.7 merupakan fungsi undak biner yang menggunakan nilai ambang (*threshold*). Fungsi undak biner dengan *threshold* dirumuskan dengan

(2.23)

$$y = \begin{cases} 0, & \text{jika } x \leq \theta \\ 1, & \text{jika } x > \theta \end{cases}$$



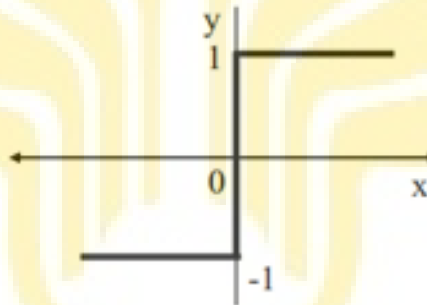
Gambar 2.7. Grafik Fungsi Undak Biner (*threshold*)

3. Fungsi *bipolar* (*symmetric hard limit*)

Fungsi *bipolar* mirip dengan fungsi undak *biner* (*hard limit*), namun *output* yang dihasilkan antara 1, 0, atau -1. Grafik fungsi *bipolar* disajikan pada Gambar 2.8 dengan fungsi

(2.24)

$$y = \begin{cases} 1, & \text{jika } x > 0 \\ 0, & \text{jika } x = 0 \\ -1, & \text{jika } x < 0 \end{cases}$$



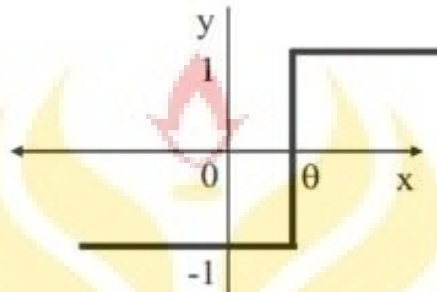
Gambar 2.8. Grafik Fungsi *Bipolar* (*symmetric hard limit*).

4. Fungsi *bipolar* dengan *threshold*

Fungsi *bipolar* dengan *threshold* pada Gambar 2.9 mirip dengan fungsi undak *biner* dengan *threshold*, namun *output* yang dihasilkan antara 1, 0, atau -1. Fungsi ini dirumuskan dengan

(2.25)

$$y = \begin{cases} 1, & \text{jika } x \geq \theta \\ 0, & \text{jika } x = \theta \\ -1, & \text{jika } x < \theta \end{cases}$$



Gambar 2.9. Grafik Fungsi *Bipolar* dengan *Threshold*

5. Fungsi *sigmoid biner*

Fungsi *sigmoid biner* pada Gambar 2.10 merupakan salah satu jenis fungsi yang kontinu, mempunyai nilai *output* diantara 0 sampai 1. Fungsi ini dirumuskan dengan

(2.26)

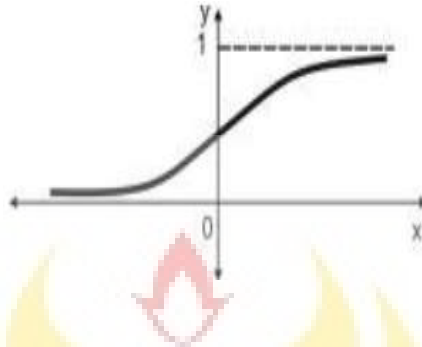
$$y = f(x) = \frac{1}{1 + e^{-x}}$$

dengan turunan

(2.27)

$$f_1'(x) = f_1(x)(1 - f_1(x))$$

Fungsi *sigmoid biner* diilustrasikan pada Gambar 2.10



Gambar 2.10. Grafik Fungsi *Sigmoid Biner* dengan *Range* (0,1)

6. Fungsi *sigmoid bipolar*

Fungsi *sigmoid bipolar* yang disajikan pada Gambar 2.11 merupakan fungsi yang mirip dengan fungsi *sigmoid biner* hanya saja nilai *outputnya* berada pada rentang -1 sampai 1. Fungsi ini dirumuskan dengan

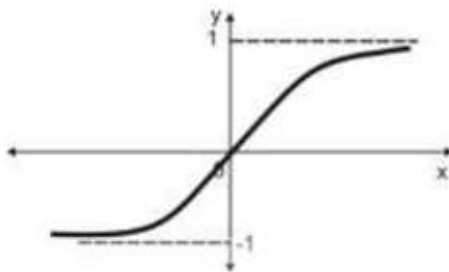
(2.28)

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

dengan turunan

(2.29)

$$f_2'(x) = \frac{1}{2}(1 + f_2(x))(1 - f_2(x))$$



Gambar 2.11. Grafik Fungsi *Sigmoid Bipolar* dengan Range $(-1,1)$

7. Fungsi Tangen Hiperbolik

Fungsi tangen hiperbolik didefinisikan sebagai berikut

(2.30)

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

dengan turunan

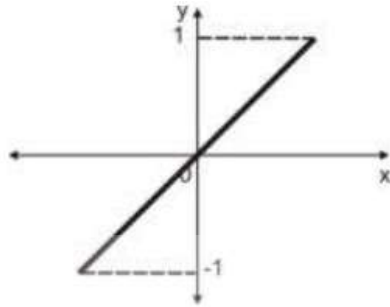
(2.31)

$$\tanh'(x) = (1 + \tanh(x))(1 - \tanh(x))$$

8. Fungsi identitas

Fungsi identitas dirumuskan dengan $y = x$.

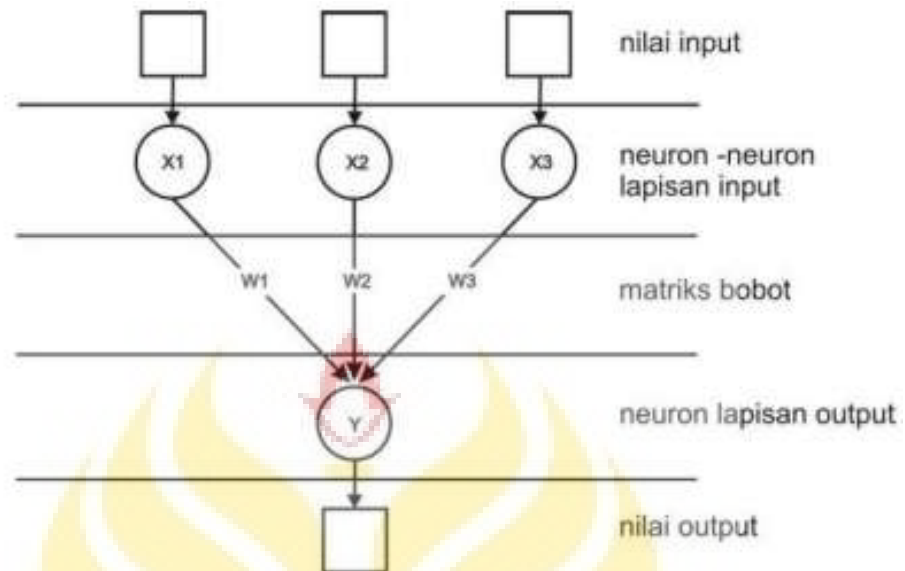
Fungsi identitas diilustrasikan pada Gambar 2.12.



Gambar 2.12. Grafik Fungsi Identitas

2.3.2 Arsitektur Jaringan Syaraf Tiruan

Seperti halnya pada jaringan syaraf biologis, jaringan syaraf tiruan juga tersusun dari unit-unit neuron. Sinyal *input* dari suatu neuron dikirim ke neuron lain melalui penghubung yang memiliki bobot. Pembobotan pada penghubung ini digunakan untuk memodifikasi sinyal (memperkuat atau memperlemah). Selanjutnya untuk menentukan nilai *output* digunakan fungsi aktivasi. Model jaringan syaraf tiruan sederhana disajikan pada Gambar 2.13.



Gambar 2.13. Model Jaringan Syaraf Tiruan Sederhana

Nilai *input* yang diberikan diteruskan ke lapisan neuron *input* (X_1 , X_2 , dan X_3). Ketiga *input* tersebut diberikan pembobotan masing-masing W_1 , W_2 , dan W_3 . Selanjutnya, ketiga neuron lapisan *input* yang terboboti itu dijumlahkan dengan persamaan

(2.32)

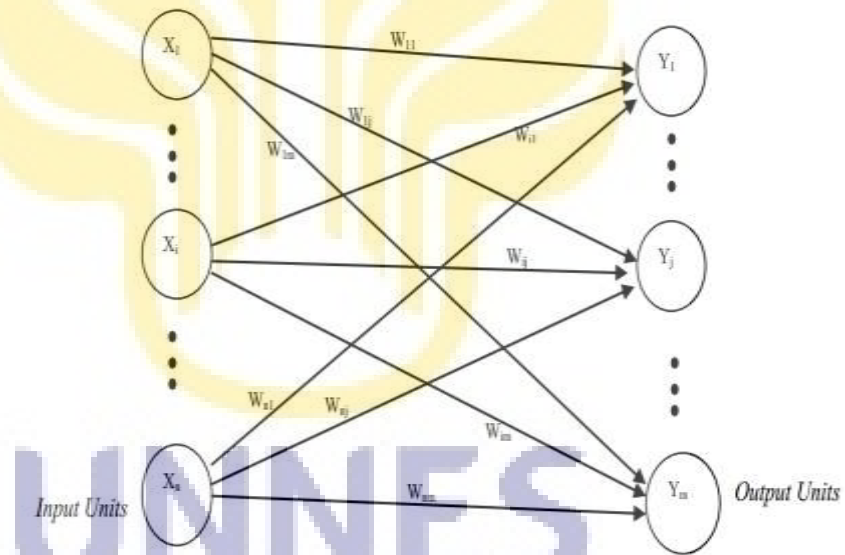
$$net = (X_1 \cdot W_1) + (X_2 \cdot W_2) + (X_3 \cdot W_3)$$

Besarnya nilai yang diterima neuron Y bergantung fungsi aktivasi $Y = f(net)$. Jika nilainya cukup kuat maka akan diteruskan. Nilai fungsi aktivasi (keluaran model jaringan) juga dapat diperiksa sebagai dasar untuk merubah bobot. Terkadang dalam suatu layar diberi unit / neuron tambahan yang nilainya selalu satu. Masukan ini disebut bias.

Menurut (Siang, 2005), arsitektur jaringan yang sering dipakai dalam jaringan syaraf tiruan antara lain:

1. Jaringan Layar Tunggal (*single layer network*).

Dalam jaringan ini, sekumpulan *input* neuron dihubungkan langsung dengan sekumpulan *output*nya. Dalam beberapa model (missal perceptron), hanya ada sebuah unit neuron *output*. Jaringan Layar Tunggal disajikan pada Gambar 2.14.



Gambar 2.14. Jaringan *Layer* Tunggal

Keterangan:

x_1, x_i, x_n : Nilai *input* ke 1 sampai n

y_1, y_j, y_m : Nilai *output* hasil pembangkitan nilai *input* oleh suatu fungsi aktivasi

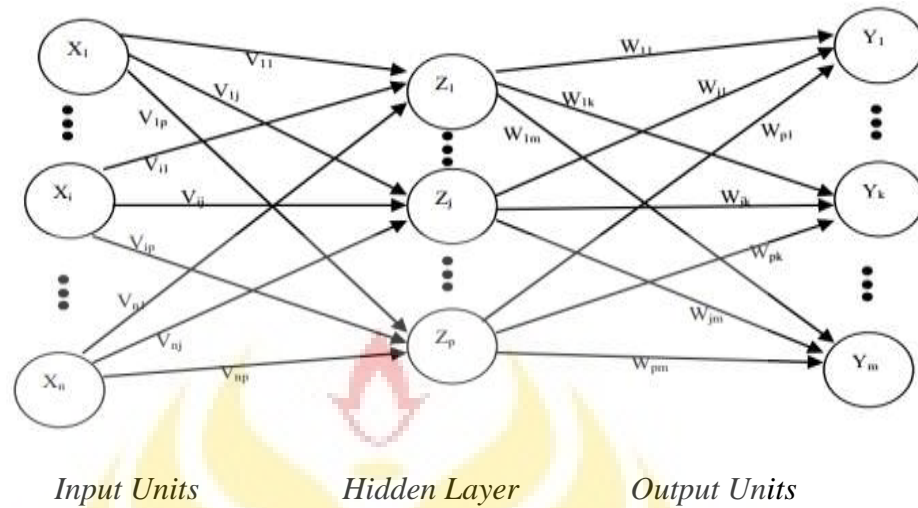
w_i : Bobot atau nilai

Gambar 2.14 menunjukkan arsitektur jaringan dengan n unit *input* (x_1, x_2, x_n) dan m buah unit *output* (y_1, y_2, y_m). Perhatikan bahwa dalam jaringan ini, semua unit *input* dihubungkan dengan semua unit *output*, meskipun dengan bobot yang berbeda-beda. Tidak ada unit *input* yang dihubungkan dengan unit *input* lainnya. Demikian pula dengan unit *output*.

Besaran W_{ji} menyatakan bobot hubungan antara unit ke- i dalam *input* dengan unit ke- j dalam *output*. Bobot-bobot ini saling independen. Selama proses pelatihan, bobot-bobot tersebut akan dimodifikasi untuk meningkatkan keakuratan hasil. Model semacam ini tepat digunakan untuk pengenalan pola karena kesederhanaanya.

2. Jaringan Layar Jamak (*multi layer network*).

Jaringan layar jamak pada Gambar 2.15 merupakan perluasan dari layar tunggal. Dalam jaringan ini, selain unit *input* dan *output*, ada unit-unit lain (sering disebut layar tersembunyi). Dimungkinkan pula ada beberapa layar tersembunyi. Sama seperti pada unit *input* dan *output*, unit-unit dalam satu layar tidak saling berhubungan.



Gambar 2.15. Jaringan Layer Jamak

Keterangan:

x_1, x_i, x_n : Nilai *input* ke 1 sampai n

z_1, z_j, z_p : Nilai dari neuron *hidden layer*

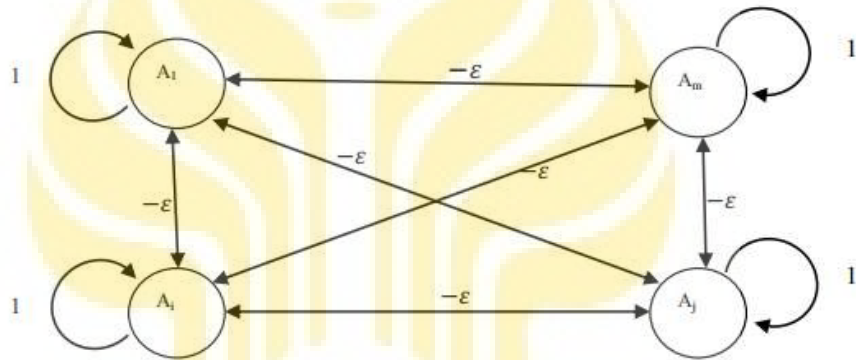
y_1, y_j, y_m : Nilai *output* hasil pembangkitan nilai *input* oleh suatu fungsi aktivasi

w_i, w_j : Bobot atau nilai

Gambar 2.15 adalah jaringan dengan n buah unit *input* (x_1, x_2, x_n), sebuah layer tersembunyi yang terdiri dari p buah unit (z_1, z_2, z_p) dan m buah unit *output* (y_1, y_2, y_m). Jaringan layer jamak dapat menyelesaikan masalah yang lebih kompleks dibandingkan dengan layer tunggal, meskipun kadangkala proses pelatihan lebih kompleks dan lama.

3. Jaringan *Reccurent*

Model jaringan *reccurent* mirip dengan jaringan layar tunggal ataupun ganda. Hanya saja, ada neuron *output* yang memberikan sinyal pada unit *input* (sering disebut *feedback loop*). Dengan kata lain, sinyal mengalir dua arah, yaitu maju dan mundur. Model jaringan *reccurent* disajikan pada Gambar 2.16.



Gambar 2.16. Jaringan *Recurrent*

2.3.3 Algoritma Pembelajaran

Salah satu bagian terpenting dalam konsep jaringan syaraf tiruan adalah terjadinya proses pembelajaran. Tujuan dari proses pembelajaran yaitu melakukan pengaturan terhadap bobot-bobot yang ada pada jaringan syaraf, sehingga diperoleh bobot akhir yang tepat dan sesuai dengan pola data yang dilatih. Selama proses pembelajaran akan terjadi perbaikan bobot-bobot berdasarkan algoritma tertentu.

Menurut Kusumadewi dan Hartati (2010), pada dasarnya ada 2 metode pembelajaran yaitu:

1. Pembelajaran terawasi (*Supervised learning*)

Metode pembelajaran pada jaringan syaraf tiruan disebut terawasi jika *output* yang diharapkan telah diketahui sebelumnya. Satu pola *input* akan diberikan ke satu neuron pada lapisan *input*. Pola ini akan dirambatkan di sepanjang jaringan syaraf hingga sampai ke neuron pada lapisan *output*. Kemudian pola *output* tersebut akan dicocokkan dengan pola *output* targetnya. Apabila terjadi perbedaan antara pola *output* hasil pembelajaran dengan pola target, maka akan muncul *error*. Apabila *error* ini masih cukup besar, maka perlu dilakukan lebih banyak pembelajaran lagi. Contoh model yang menggunakan pembelajaran terawasi antara lain: Perceptron, ADALINE, MADALINE, *Backpropagation*, dll.

2. Pembelajaran tak terawasi (*Unsupervised learning*)

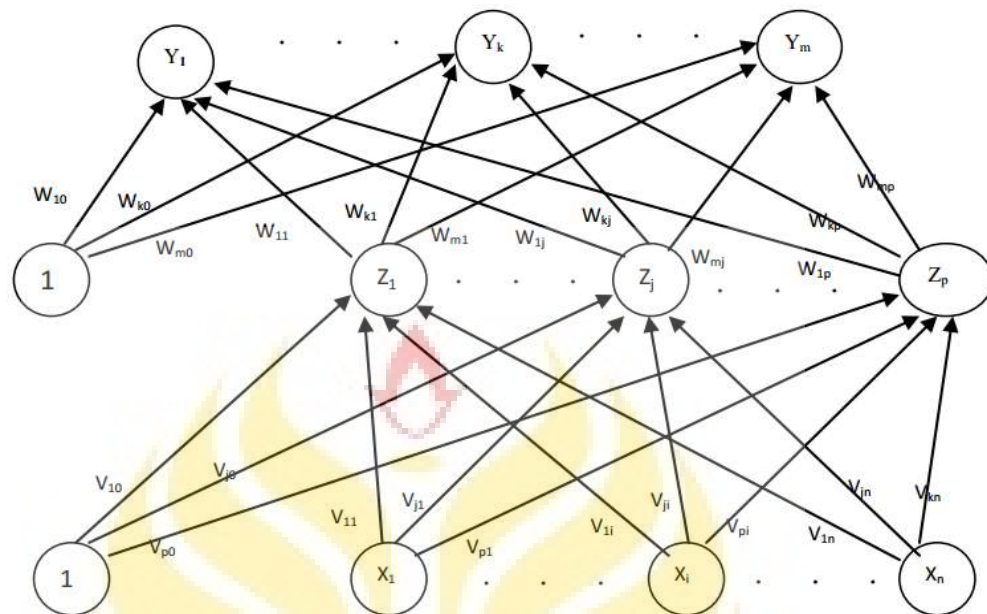
Pembelajaran yang tak terawasi tidak memerlukan target *output*. Pada metode ini, tidak dapat ditentukan hasil yang seperti apakah yang diharapkan selama proses pembelajaran. Selama proses pembelajaran, nilai bobot disusun dalam suatu *range* tertentu tergantung pada nilai *input* yang diberikan. Tujuan dari pembelajaran ini adalah mengelompokkan unit-unit yang hampir sama dalam

suatu area tertentu. Model yang menggunakan pembelajaran ini adalah model jaringan kompetitif.

2.3.4 Metode *Backpropagation*

Backpropagation merupakan salah satu dari metode pelatihan pada jaringan syaraf, dimana ciri dari metode ini adalah meminimalkan *error* pada *output* yang dihasilkan oleh jaringan (Anggraeni Puspita & Euike , 2007). *Backpropagation* melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan mengenali pola yang digunakan selama *training* serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola masukan yang serupa (tapi tidak sama) dengan pola yang dipakai selama pelatihan (Siang, 2009: 97).

Sebagian besar pelatihan untuk jaringan *feedforward* (umpan maju) menggunakan gradien dari fungsi aktivasi untuk menentukan bagaimana mengatur bobot-bobot dalam rangka meminimumkan kinerja. Gradien ini ditentukan dengan menggunakan suatu teknik yang disebut *backpropagation*. *Backpropagation* memiliki beberapa unit yang ada dalam satu atau lebih *layer* tersembunyi (Siang, 2009: 98). Gambar 2.17 adalah arsitektur *backpropagation* dengan n buah masukan (ditambah sebuah bias), sebuah *layer* tersembunyi yang terdiri atas p unit (ditambah sebuah bias), serta m buah keluaran. Arsitektur jaringan ini disebut jaringan *layer* jamak.



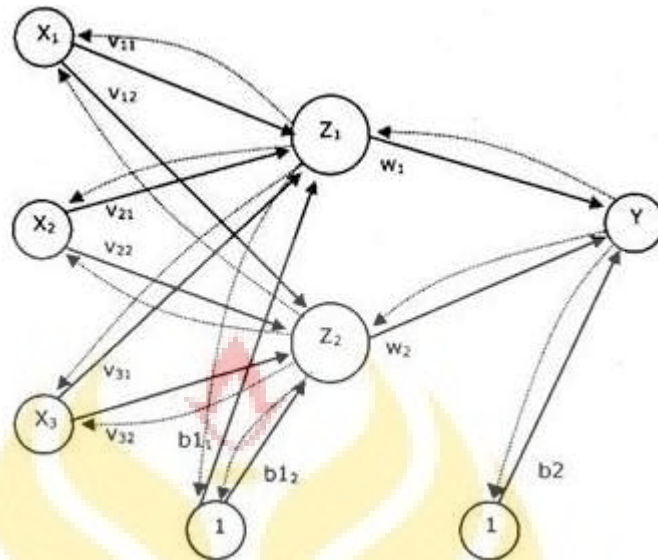
Gambar 2.17. Arsitektur Jaringan *Layer* Jamak

2.3.5 Konsep Jaringan Syaraf Tiruan *Backpropagation*

Backpropagation merupakan algoritma pembelajaran yang terawasi dan biasanya digunakan oleh perceptron dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyinya. Algoritma *backpropagation* menggunakan *output error* untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward*). Untuk mendapatkan *error* ini, tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu. Pada saat perambatan maju, neuron-neuron diaktifkan dengan menggunakan fungsi aktivasi yang dapat dideferensiasikan, seperti *sigmoid*.

2.3.6 Arsitektur Jaringan *Backpropagation*

Arsitektur jaringan *backpropagation* seperti terlihat pada Gambar 2.18 terdiri atas 3 (tiga) unit (neuron) pada lapisan *input*, yaitu x_1, x_2 , dan x_3 ; 1 lapisan tersembunyi dengan 2 neuron, yaitu z_1 dan z_2 , serta 1 unit pada lapisan *output*, yaitu y bobot yang menghubungkan x_1, x_2 , dan x_3 (V_{ji} bobot yang menghubungkan neuron *input* ke- i ke neuron ke- j pada lapisan tersembunyi). Perlu diingat bahwa untuk pemakaian *Toolbox neural network* pada Matlab, bobot V_{ij} memiliki pengertian yang sebaliknya (V_{ij} bobot yang menghubungkan neuron ke- j pada suatu lapisan ke neuron ke- i pada lapisan sesudahnya). Misal V_{12} adalah bobot yang menghubungkan neuron ke-2 pada lapisan *input*, ke neuron ke-1 pada lapisan tersembunyi. Kembali pada gambar b_{1_1} , dan b_{1_2} adalah bobot bias yang menuju ke neuron pertama dan kedua pada lapisan tersembunyi. Bobot yang menghubungkan z_1 dan z_2 dengan neuron pada lapisan *output*, adalah w_1 dan w_2 , bobot bias b_2 menghubungkan lapisan tersembunyi dengan lapisan *output*.



Gambar 2.18. Arsitektur Jaringan *Backpropagation*

Simbol-simbol yang digunakan pada algoritma pelatihan untuk jaringan *backpropagation* adalah sebagai berikut :

X vektor masukan pelatihan :

$$X = (X_1, \dots, X_i, \dots, X_n).$$

Y vektor keluaran target :

$$Y = (Y_1, \dots, Y_k, \dots, Y_m).$$

δ_k informasi tentang kesalahan pada unit y_k yang disebarkan kembali keunit tersembunyi.

Z unit tersembunyi

$$Z = (Z_1, \dots, Z_p, \dots, Z_n)$$

α = *Learning Rate*.

w_{nj} bobot koneksi antara unit layar tersembunyi j dengan unit *output* ke n .

v_{ij} bobot garis dari unit masukan ke j dengan unit tersembunyi ke i .

2.3.7 Pelatihan dan Pengujian *Backpropagation*

Pada pelatihan *backpropagation* meliputi 3 fase, yaitu:

Fase I : Propagasi maju

Selama propagasi maju, sinyal masukan ($= x_i$) dipropagasikan ke layar tersembunyi menggunakan fungsi aktivasi yang ditentukan. Keluaran dari setiap unit layar tersembunyi ($= z_j$) tersebut selanjutnya dipropagasikan maju lagi ke layar tersembunyi di atasnya menggunakan fungsi aktivasi yang ditentukan. Demikian seterusnya hingga menghasilkan keluaran jaringan ($= y_k$). Selanjutnya keluaran tersebut dibandingkan dengan target yang harus dicapai ($= t_k$). Selisih $t_k - y_k$ adalah kesalahan yang terjadi. Jika kesalahan ini lebih kecil dari batas toleransi yang ditentukan maka iterasi dihentikan. Namun, jika kesalahan masih lebih besar dari batas toleransinya, maka bobot pada setiap garis akan dimodifikasikan untuk mengurangi kesalahan yang terjadi.

Fase II : Propagasi mundur

Berdasarkan kesalahan $t_k - y_k$, dihitung faktor $\delta_k (k = 1, 2, \dots, m)$ yang dipakai untuk mendistribusikan kesalahan di unit y_k ke semua unit tersembunyi yang terhubung langsung dengan y_k . δ_k juga dipakai untuk mengubah bobot garis yang berhubungan langsung dengan unit keluaran. Dengan cara yang sama, dihitung faktor δ_j di setiap unit dilayar tersembunyi sebagai dasar perubahan bobot semua garis yang berasal dari unit tersembunyi sebagai dasar perubahan bobot semua garis yang berasal dari unit tersembunyi di layar di bawahnya. Demikian seterusnya hingga faktor δ di unit yang tersembunyi yang berhubungan dengan unit masukan dihitung.

Fase III : Setelah semua faktor δ dihitung, bobot semua baris dimodifikasi bersamaan. Perubahan bobot suatu garis didasarkan atas faktor δ neuron di layar atasnya.

Ketiga fase tersebut diulang-ulang terus hingga kondisi penghentian (iterasi) dipenuhi. Iterasi akan dihentikan jika jumlah iterasi yang dilakukan sudah melebihi jumlah maksimum iterasi yang ditetapkan.

Algoritma pelatihan untuk jaringan dengan satu layar tersembunyi (dengan fungsi aktivasi *sigmoid biner*) terbagi dalam 7 langkah, sebagai berikut:

Langkah 1 : Inisialisasi bobot (pada langkah ini, nilai bobot pada tiap-tiap lapisan diinisialisasikan dengan nilai *random* yang cukup kecil)

Langkah 2 : Tetapkan: Maksimum Epoch, Target *error*, dan *Learning Rate*

Langkah 3 : Inisialisasi: epoch = 0, MSE = 1

Langkah 4 : Jika kondisi penghentian belum terpenuhi maka lakukan langkah berikut:

Langkah 5 : Untuk setiap pasangan data pelatihan, lakukan *fase feed forward* :

- tiap unit masukan menerima sinyal dan meneruskannya ke unit tersembunyi di atasnya.
- Hitung semua keluaran di unit tersembunyi z_j ($k = 1, 2, 3, \dots, p$)

Fase I : Propagasi maju

Langkah 6 : Hitung semua keluaran di unit tersembunyi z_j ($j = 1, 2, \dots, p$).

Untuk mendapatkan nilai lapisan keluaran, perhitungan dilakukan dari lapisan ke lapisan. Keluaran pada unit lapisan tersembunyi adalah

(2.33)

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

- Gunakan fungsi aktivasi untuk menghitung sinyal *output*nya

(2.34)

$$Z_j = f(z_{in_j}) = \frac{1}{1 + e^{-z_{in_j}}}$$

- Hitung semua keluaran jaringan di unit Y_k ($k = 1, 2, 3, \dots, m$)

(2.35)

$$y_{in_k} = b_k + \sum_{i=1}^p z_j w_{jk}$$

- Gunakan fungsi aktivasi untuk menghitung sinyal *output* nya

(2.36)

$$Y_k = \frac{1}{1 + e^{-z_{in_k}}}$$

Fase II : Propagasi mundur

Langkah 7 : Hitung perubahan pada bobot

Untuk menghitung perubahan bobot, vektor *output* pada tiap-tiap lapisan dibandingkan dengan nilai *output* yang diharapkan (target). Tiap-tiap unit *output* ($y_k, k = 1, 2, 3, \dots, m$) menerima target pola yang berhubungan dengan pola *input* pembelajaran, hitung inferensi *error*nya

(2.37)

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

δ_k merupakan unit kesalahan yang dipakai dalam perubahan bobot layar di bawahnya.

Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai w_{kj}) dengan laju pembelajaran α

(2.38)

$$\varphi_{ik} = \delta_k z_j$$

(2.39)

$$\beta_{ik} = \delta_k$$

(2.40)

$$\Delta w_{jk} = \alpha \varphi_{ik}$$

Hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai b_k)

(2.41)

$$\Delta b_k = \alpha \beta_k$$

Langkah ini juga dilakukan sebanyak jumlah lapisan tersembunyi, yaitu menghitung informasi *error* dari suatu lapisan tersembunyi ke lapisan tersembunyi sebelumnya.

Tiap-tiap unit tersembunyi ($Z_j, j = 1, 2, 3, \dots, p$) menjumlahkan delta *input* (dari *input* unit-unit yang berada pada lapisan di atasnya)

(2.42)

$$\delta_{in_j} = \sum_{i=1}^m \delta_k w_{jk}$$

Kalikan nilai ini dengan turunan dari fungsi aktivasinya untuk menghitung informasi *error*

(2.43)

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

(2.44)

$$\varphi_{1_{ij}} = \delta_{1_j} x_i$$

(2.45)

$$\beta_{1_j} = \delta_{1_j}$$

Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai v_{ij}) dengan laju pembelajaran α

(2.46)

$$\Delta v_{ij} = \alpha \phi_{1_{ij}}$$

Hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai b_{1_j})

(2.47)

$$\Delta b_{1_j} = \alpha \beta_{1_j}$$

Fase III : Perubahan bobot

Langkah 8 : Hitung semua perubahan nilai bobot.

Perubahan bobot yang menuju ke unit keluaran. Tiap-tiap unit $output(Y_k, k = 1, 2, 3, \dots, m)$ memperbaiki bias dan bobotnya ($j=0, 1, 2, \dots, p$)

(2.48)

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj}$$

(2.49)

$$b_{2_k}(\text{baru}) = b_{2_k}(\text{lama}) + \Delta b_{2_k}$$

Tiap-tiap unit tersembunyi (Z_j , $j = 1, 2, 3, \dots, p$) memperbaiki bias dan bobotnya ($i = 0, 1, 2, \dots, n$)

(2.50)

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

(2.51)

$$b_{1j} \text{ (baru)} = b_{1j} \text{ (lama)} + \Delta_j b_{1j}$$

Setelah tahap pelatihan selesai, tahap selanjutnya yaitu tahap pengujian jaringan. Pada tahap pengujian, langkah yang dilakukan hanya sampai pada fase I yaitu propagasi maju saja, tidak ada fase II (propagasi mundur) apalagi fase III (perubahan bobot). Pada tahap pengujian ini, jaringan diharapkan dapat mengenali pola berdasarkan data baru yang diberikan.

2.3.8 Pembagian Data

Tujuan utama penggunaan *backpropagation* adalah mendapatkan keseimbangan antara pengenalan pola pelatihan secara benar dan respon yang baik untuk pola lain yang sejenis (disebut data pengujian). Umumnya, data dibagi menjadi dua bagian saling asing, yaitu pola data yang dipakai sebagai pelatihan dan data yang dipakai sebagai pengujian. Beberapa komposisi data pelatihan dan pengujian yang sering digunakan adalah sebagai berikut (Wibowo, 2007: 23):

1. 80% untuk data pelatihan dan 20% untuk data pengujian.
2. 70% untuk data pelatihan dan 30% untuk data pengujian.
3. $\frac{2}{3}$ untuk data pelatihan dan $\frac{1}{3}$ untuk data pengujian.
4. 50% untuk data pelatihan dan 50% untuk data pengujian.
5. 60% untuk data pelatihan dan 40% untuk data pengujian.

Aspek pembagian data harus ditekankan agar jaringan mendapat data pelatihan yang secukupnya dan data pengujian dapat menguji prestasi pelatihan yang dilakukan berdasarkan nilai MAPE data pelatihan dan pengujian. Bilangan data yang kurang untuk proses pelatihan akan menyebabkan jaringan mungkin tidak dapat mempelajari taburan data dengan baik. Sebaliknya, data yang terlalu banyak untuk proses pelatihan akan melambatkan proses pemusatan (konvergensi). Masalah *overtraining* (data pelatihan yang berlebihan) akan menyebabkan jaringan cenderung untuk menghafal data yang dimasukkan daripada menggeneralisasi.

Perubahan bobot dilakukan berdasarkan pola pelatihan. Akan tetapi selama pelatihan, kesalahan yang terjadi dihitung berdasarkan semua data (pelatihan dan pengujian). Selama kesalahan ini menurun, pelatihan terus dijalankan. Akan tetapi jika kesalahannya sudah meningkat, pelatihan tidak ada gunanya untuk diteruskan lagi. Jaringan sudah mulai mengambil sifat yang hanya dimiliki secara spesifik oleh data pelatihan (tapi tidak dimiliki oleh data pengujian) dan sudah mulai kehilangan data pelatihan (tapi tidak dimiliki oleh data pengujian) dan sudah mulai kehilangan kemampuan melakukan generalisasi.

2.3.9 Backpropagation dalam Prediksi

Salah satu bidang dimana *backpropagation* dapat diaplikasikan dengan baik adalah bidang prediksi (*forecasting*). Secara umum, masalah prediksi dapat dinyatakan sebagai berikut :

1. Diketahui sejumlah data runtun waktu (*time series*) x_1, x_2, \dots, x_n . Masalahnya adalah memperkirakan berapa harga x_{n+1} berdasarkan x_1, x_2, \dots, x_n .
2. Dengan *backpropagation*, *record* data dipakai sebagai data pelatihan untuk mencari bobot yang optimal. Sehingga perlu menetapkan besarnya periode dimana data berfluktuasi. Periode ini ditentukan secara intuitif. Misalkan pada data besarnya debit air sungai dengan data bulanan, periode data dapat diambil selama satu tahun karena pergantian musim terjadi selama satu tahun.
3. Jumlah data dalam satu periode ini dipakai sebagai jumlah masukan dalam *backpropagation*. Sebagai targetnya diambil data bulan pertama setelah periode berakhir. Pada data bulanan dengan periode satu tahun, maka masukan *backpropagation* yang dipakai terdiri dari 12 masukan. Keluaran adalah 1 unit.
4. Bagian tersulit adalah menentukan jumlah layer (dan unitnya). Tidak ada teori yang dengan pasti dapat dipakai. Tetapi secara praktis dicoba jaringan yang kecil terlebih dahulu (misal terdiri dari 1 layer tersembunyi dengan beberapa unit saja). Jika gagal (kesalahan tidak turun dalam epoch yang besar), maka jaringan diperbesar dengan menambahkan unit tersembunyi atau bahkan menambah layer tersembunyi.

Langkah-langkah membangun struktur jaringan untuk prediksi sebagai berikut:

1. *Preprocessing* / normalisasi

Sebelum digunakan untuk proses pelatihan, perlu dilakukan penskalaan terhadap harga-harga *input* dan target sedemikian hingga data-data *input* dan target tersebut masuk dalam suatu *range* tertentu yang disebut *preprocessing* atau normalisasi data.

2. Perancangan struktur jaringan yang optimum

Langkah selanjutnya adalah penentuan jumlah lapisan masukan, lapisan tersembunyi, dan jumlah lapisan keluaran yang akan digunakan dalam jaringan. Penggunaan jaringan dengan dua atau lebih lapisan tersembunyi dalam masalah prediksi kebanyakan tidak akan memberikan pengaruh yang sangat besar terhadap prestasi jaringan untuk melakukan prediksi. Selain itu akan melambatkan proses pelatihan yang disebabkan bertambahnya unit.

3. Pemilihan parameter-parameter jaringan

Pemilihan parameter-parameter jaringan yang meliputi fungsi aktivasi, algoritma pelatihan, banyaknya jumlah iterasi, kinerja tujuan, dan laju pembelajaran (*learning rate*) mempunyai peranan yang penting untuk struktur jaringan yang akan dibangun dan digunakan dalam prediksi. Hasil keputusan yang kurang memuaskan dapat diperbaiki dengan penggunaan laju pembelajaran untuk mendapatkan nilai bobot yang paling *optimum*.

4. *Postprocessing*/denormalisasi

Selanjutnya dilakukan pelatihan jaringan menggunakan data pelatihan yang sudah ditentukan. Kemudian, harga-harga ternormalisasi dari *output* jaringan harus dikembalikan (denormalisasi) ke harga aslinya untuk mendapatkan nilai *output* pada *range* yang sebenarnya. Setelah tahap pelatihan selesai, jaringan diujikan menggunakan data pengujian, dimana data tersebut berbeda dengan data pelatihan dan dinormalisasi terlebih dahulu. Pada tahap pengujian ini, jaringan diharapkan dapat mengenali pola berdasarkan data baru yang diberikan. Hasil *output* pengujian juga dikembalikan (denormalisasi) ke harga aslinya, seperti halnya pada tahap pelatihan.

5. Pemilihan jaringan optimum

Untuk mendapatkan jaringan yang optimum, hasil *output* pada tahap pelatihan dan pengujian, masing-masing dibandingkan dengan target atau data aslinya untuk mengetahui seberapa besar perbedaan atau kesalahan yang terjadi.

Kesalahan tersebut dirumuskan sebagai berikut

(2.52)

$$MAPE = \frac{1}{n} \left(\sum_{t=1}^n \left| \frac{y_t - \hat{y}_t}{y_t} \right| \times 100\% \right)$$

Dengan n = banyaknya data

y_t = nilai aktual pada waktu t

\hat{y}_t = nilai ramalan pada waktu t

(Makridakris, 1999: 59)

Jaringan dengan nilai MAPE terendah pada tahap pengujian dan mempunyai nilai MAPE yang tidak beda jauh dengan pelatihan, dipilih sebagai jaringan yang optimum untuk digunakan dalam prediksi.

2.3.10 Meningkatkan Hasil Metode *Backpropagation*

Masalah utama yang dihadapi dalam *Backpropagation* adalah lamanya iterasi yang harus dilakukan. *Backpropagation* tidak dapat memberikan kepastian tentang berapa *epoch* yang harus dilalui untuk mencapai kondisi yang diinginkan. Untuk meningkatkan hasil yang diperoleh dengan metode *backpropagation* dapat dilakukan dengan analisis bobot dan bias awal, jumlah unit tersembunyi, waktu iterasi dan penambahan mom.

2.3.10.1 Pemilihan Bobot dan Bias Awal

Penentuan nilai bobot awal dalam algoritma pembelajaran mempengaruhi kecil atau besarnya *error* yang juga menentukan jumlah iterasi yang harus dilakukan. Selain dengan cara *random* dalam inisialisasi bobot awal terdapat juga inisialisasi bobot Nguyen-Widrow. Metode Nguyen-Widrow digunakan untuk meningkatkan proses konvergensi *backpropagation* yang menghasilkan bobot awal dan nilai bias untuk setiap layer pada arsitektur *backpropagation*.

Bobot dalam *backpropagation* tidak boleh diberi nilai yang sama. Penyebabnya adalah karena jika bobot sama jaringan sama jaringan tidak akan terlatih dengan benar. Jaringan mungkin saja gagal untuk belajar terhadap

serangkaian contoh-contoh pelatihan. Misalnya dengan kondisi tetap atau bahkan *error* semakin besar dengan diteruskannya proses pelatihan. Untuk mengatasi hal ini maka inisialisasi bobot dibuat secara acak.

Bobot yang menghasilkan nilai turunan aktivasi yang kecil sedapat mungkin dihindari karena akan menyebabkan perubahan bobotnya menjadi sangat kecil. Demikian pula nilai bobot awal tidak boleh terlalu besar karena nilai turunan fungsi aktivasinya menjadi sangat kecil juga. Dalam standar *backpropagation*, bobot dan bias diisi dengan bilangan acak kecil. Untuk inisialisasi bobot awal secara random maka nilai yang digunakan adalah antara -0,5 sampai 0,5 atau -1 sampai 1.

Laurene Fauset dalam bukunya (*Fundamentals of Neural Network*, halaman 297) mengatakan bahwa inisialisasi Nguyen-Widrow terbukti mempercepat proses pembelajaran (iterasi). Inisialisasi bobot Nguyen-Widrow dari *input* layer ke *hidden* layer, memperhatikan jumlah *input node* dan *hidden node* dalam menentukan nilai bobot awalnya dengan pendekatan transformasi *Fourier*. Sedangkan untuk bobot dari *hidden* layer ke *input* layer dilakukan dengan *random* dengan nilai dari -0,5 sampai 0,5.

Pemilihan bobot dan bias akan mempengaruhi apakah jaringan mencapai *error* minimum global atau *error* minimum lokal, dan bagaimana mengatasinya dengan cepat. Dalam hal ini, sangat penting menghindari pemilihan bobot dan bias yang membuat fungsi aktivasi atau turunannya mendekati nol. Nilai inisialisasi bobot tidak dapat terlalu besar atau sinyal input terhadap unit *hidden* atau unit *output* akan

jatuh di wilayah dimana turunan fungsi sigmoid mempunyai nilai yang sangat kecil. Dengan kata lain, jika inisialisasi bobot terlalu kecil, input jaringan terhadap unit *hidden* dan unit *output* akan mendekati nol dan menyebabkan proses pembelajaran yang sangat lambat.

Menurut Kusumadewi dan Hartati (2010), metode Nguyen-widrow akan menginisialisasi bobot-bobot lapisan dengan nilai antara -0,5 sampai 0,5. Sedangkan bobot-bobot dari lapisan *input* ke lapisan *hidden* dirancang sedemikian rupa sehingga dapat meningkatkan kemampuan lapisan tersembunyi dalam melakukan proses pembelajaran. Metode Nguyen-widrow secara sederhana dapat diimplementasikan dengan prosedur sebagai berikut.

Tetapkan n : n = jumlah neuron (unit) pada lapisan *input*.

P = jumlah neuron (unit) pada lapisan tersembunyi.

β = faktor penskalaan ($=0,7(p)^{\frac{1}{n}}$)

Kerjakan untuk setiap unit pada lapisan tersembunyi ($j=1,2,\dots,p$):

1. Inisialisasi bobot-bobot dari lapisan *input* ke lapisan tersembunyi :

V_{ij} bilangan *random* antara -0,5 sampai 0,5 (atau antara $(-\gamma)$ sampai γ).

2. Hitung $\|V_j\|$

(2.53)

$$\|V_j\| = \sqrt{(v_{1j})^2 + (v_{2j})^2 + \dots + (v_{nj})^2}$$

3. Inisialisasi ulang bobot-bobot

(2.54)

$$V_j = \frac{\beta v_{ij}}{\|V_j\|}$$

4. Set bias :

b_j = bilangan *random* antara $-\beta$ sampai β . Analisa Nguyen-widrow didasarkan atas fungsi aktivasi tangen hiperbolik.

2.3.10.2 Jumlah Unit Tersembunyi

Berdasarkan hasil teoritis, *Backpropagation* dengan sebuah *hidden layer* sudah cukup untuk mampu mengenali sembarang pasangan antara masukan dan target dengan tingkat kelitian yang ditentukan. Akan tetapi penambahan jumlah *hidden layer* kadangkala membuat pelatihan lebih mudah. Jika jaringan memiliki lebih dari *hidden layer*, maka algoritma pelatihan yang dijabarkan sebelumnya perlu direvisi. Dalam propagasi maju, keluaran harus dihitung untuk setiap *layer*, dimulai dari *hidden layer* paling bawah (terdekat dengan unit masukan). Sebaliknya dalam propagasi mundur, faktor δ perlu dihitung untuk tiap *hidden layer*, dimulai dari lapisan keluaran (Hajar, 2005).

2.3.10.3 Waktu Iterasi

Tujuan utama penggunaan *backpropagation* adalah mendapatkan keseimbangan antara pengenalan pola pelatihan secara benar dan respon yang baik

untuk pola lain yang sejenis. Jaringan dapat dilatih terus menerus hingga semua pola pelatihan dikenali dengan benar. Akan tetapi hal itu tidak menjamin jaringan akan mampu mengenali pola pengujian dengan tepat. Jadi tidaklah bermanfaat untuk meneruskan iterasi hingga semua kesalahan pola pelatihan = 0.

Umumnya data dibagi menjadi dua bagian, yaitu pola data pelatihan dan data pengujian. Perubahan bobot dilakukan berdasarkan pola pelatihan. Akan tetapi selama pelatihan (misalkan setiap 10 *epoch*), kesalahan yang terjadi dihitung berdasarkan semua data (pelatihan dan pengujian). Selama kesalahan ini menurun, pelatihan terus dijalankan. Akan tetapi jika kesalahannya sudah meningkat, pelatihan tidak ada gunanya diteruskan. Jaringan sudah mulai mengambil sifat yang hanya dimiliki secara spesifik oleh data pelatihan (tapi tidak dimiliki oleh data pengujian) dan sudah mulai kehilangan kemampuan melakukan generalisasi.

2.3.10.4 Pengujian (*Testing*) pada Metode *Backpropagation*

Dalam proses *testing* ini diberikan *input* data yang disimpan dalam disk (*file testing*). JST yang telah dilatih akan mengambil data tersebut dan memberikan *output* yang merupakan “Hasil Prediksi JST”. JST memberikan *output* berdasarkan bobot yang disimpan dalam proses pelatihan.

Pada akhir *testing* dilakukan perbandingan antara hasil prediksi (*output* JST) dan hasil asli (kondisi nyata yang terjadi). Hal ini adalah untuk menguji tingkat keberhasilan JST dalam melakukan prediksi.

2.3.10.5 Transformasi Normal (Normalisasi)

Pada proses perkiraan dengan menggunakan *backpropagation*, sebelum dilakukan proses pelatihan, data *input* dan target *output* harus dinormalisasi terlebih dahulu. Normalisasi adalah penskalaan terhadap nilai-nilai masuk ke dalam suatu range dari fungsi aktivasi yang digunakan dalam jaringan.

Jika menggunakan fungsi aktivasi sigmoid biner, data harus ditransformasikan terlebih dahulu karena *range* keluaran fungsi aktivasi sigmoid adalah $[0,1]$. Data bisa ditransformasikan ke interval $[0,1]$. Tapi akan lebih baik jika ditransformasikan ke interval yang lebih kecil, misal pada interval $[0,1, 0,9]$, karena fungsi sigmoid merupakan fungsi asimtotik yang nilainya tidak pernah mencapai 0 ataupun 1 (Siang, 2004).

Menurut Hidayat dan Suprpto (2012) data-data yang ada dilakukan normalisasi dengan membagi nilai data tersebut dengan nilai *range* data (Nilai data maksimum – nilai data *minimum*). Tujuan dari normalisasi yaitu :

1. Untuk menghilangkan kerangkapan data.
2. Untuk mengurangi kompleksitas.
3. Untuk mempermudah pemodifikasian data.

Untuk mentransformasikan data digunakan rumus pada persamaan (2.54)

(2.55)

$$x' = \frac{0,8(x-a)}{b-a} + 0,1$$

Dengan : x = Data awal

a = Nilai minimal data awal

b = Nilai maksimal data awal

Kemudian untuk mendapatkan nilai sebenarnya dari *output* perlu dilakukan proses denormalisasi.

Proses normalisasi data juga dapat dilakukan dengan bantuan mean dan standar deviasi. Pada MATLAB proses normalisasi data dengan bantuan mean dan standar deviasi menggunakan perintah *prestd* yang akan membawa data ke dalam bentuk normal. Berikut adalah perintah proses normalisasi data pada MATLAB:

`[pn, meanp, stdp, tn, meant, stdt] = prestd (P,T)`

dengan

P : matriks input

T : matriks target

Fungsi pada matlab akan menghasilkan:

Pn : matriks input yang ternormalisasi (mean = 0, deviasi standar 1)

tn : matriks target yang ternormaliasasi (mean = 0, deviasi standar = 1)

meanp : mean pada matriks input asli (p)

stdp : deviasi standar pada matriks input asli (p)

meant : mean pada matriks target asli (t)

stdt : deviasi standar pada matriks target asli (t)

2.3.10.6 Denormalisasi

Menurut Hidayat dkk (2012) denormalisasi dapat memberikan atau mengembalikan data, sehingga didapatkan predicted sales dari data training.

Menurut Indra (2014) adapun rumus denormalisasi dalam range [0, 1] adalah

(2.56)

$$X_i = y(b - a) + a$$

Keterangan :

X_i = Nilai data normal

y = data keluaran

a = data dengan nilai minimum

b = data dengan nilai maximum

Data akan di denormalisasi dengan fungsi *poststd* pada matlab, dengan perintah sebagai berikut:

[P,T] = poststd (pn, meanp, stdp, tn, meant, stdt)

2.3.10.7 Ukuran Akurasi Prediksi

Validasi metode prediksi terutama dengan menggunakan metode-metode di atas tidak dapat lepas dari indikator dalam pengukuran akurasi prediksi. Bagaimanapun juga terdapat sejumlah indikator dalam pengukuran kesesuaian suatu metode prediksi. Dalam banyak hal, ketepatan (akurasi) menunjukkan seberapa jauh model prediksi mampu memproses data yang diberikan (Makridakis dkk, 1999: 39).

2.3.10.7.1 Ukuran Statistik Standar

Jika X_i merupakan data aktual untuk periode i dan F_i adalah hasil prediksi untuk periode yang sama, maka kesalahan didefinisikan sebagai

(2.57)

$$e_i = X_i - F_i$$

Jika terdapat nilai aktual dan ramalan untuk n periode waktu, maka akan terdapat n buah kesalahan. Terdapat beberapa ukuran statistik standar (Makridakis dkk, 1999: 40).

a. *Mean Error* (ME)

Mean Error (ME) adalah rata-rata kesalahan meramal dengan menghitung kesalahan dengan banyaknya data

(2.58)

$$ME = \sum_{i=1}^n \frac{e_i}{n}$$

Keterangan:

ME = rata-rata kesalahan

 e_i = kesalahan prediksi

n = banyaknya data

b. *Mean Absolute Error* (MAE)

Mean Absolute Error (MAE) adalah rata-rata absolute dari kesalahan meramal, tanpa menghiraukan tanda positif atau negatif

(2.59)

$$MAE = \sum_{i=1}^n \frac{|e_i|}{n}$$

Keterangan:

MAE = rata-rata absolut kesalahan

 $|e_i|$ = nilai absolut dari kesalahan prediksi

n = banyaknya data

c. *Sum Of Squared Error* (SSE)

Sum Of Squared Error (SSE) adalah penjumlahan dari kuadrat kesalahan

(2.60)

$$SSE = \sum_{i=1}^n e_i^2$$

d. *Mean Squared Error* (MSE)

Mean Square Error (MSE) adalah fungsi kinerja yang sering digunakan untuk *backpropagation*. Fungsi ini mengambil rata-rata kuadrat *error* yang terjadi antara *output* jaringan dan target.

MSE dihitung sebagai berikut (Sinuhaji, 2009) :

1. Hitung keluaran Jaringan Syaraf Tiruan untuk masukan pertama aktivasi prediksi.
2. Hitung selisih antara nilai target dengan nilai keluaran prediksi.
3. Kuadratkan setiap selisih tersebut.
4. Jumlahkan semua kuadrat selisih dari tiap-tiap data pembelajaran dalam satu epoch.
5. Bagi hasil penjumlahan tersebut dengan jumlah data pembelajaran.

UNNES
UNIVERSITAS NEGERI SEMARANG

Menurut Hansun (2013) kriteria MSE menyatakan besarnya kesalahan

rata-rata kuadrat dari suatu metode prediksi dengan rumus perhitungan

(2.61)

$$MSE = \frac{\sum_{t=1}^n e_t^2}{n}$$

(2.62)

$$e_t = y_t - \hat{y}_t$$

Keterangan

n = jumlah data pembelajaran

y_t = nilai target pada waktu t

\hat{y}_t = nilai prediksi pada waktu t

Tujuan optimalisasi statistik seringkali dilakukan untuk memilih suatu model agar nilai MSE minimal, tetapi ukuran ini mempunyai dua kesalahan. Pertama ukuran ini menunjukkan pencocokan (*fitting*) suatu model terhadap data historis. Pencocokan seperti ini tidak selalu mengimplikasikan prediksi yang baik. Suatu model yang terlalu cocok (*over fitting*) dengan deret data berarti sama dengan memasukkan unsur random sebagai sebagian proses bangkitan, adalah sama buruknya dengan dengan tidak berhasil mengenai pola non acak dalam data. Kekurangan kedua dalam MSE sebagai ukuran ketepatan model adalah berhubungan dengan kenyataan bahwa metode berbeda akan menggunakan prosedur yang berbeda pula dalam fase pencocokan.

2.3.10.7.2 Ukuran-ukuran Relatif

Keterbatasan MSE sebagai suatu ukuran ketepatan prediksi membuat diusulkannya ukuran-ukuran alternatif, yang diantaranya menyangkut kesalahan persentase (Makridakis dkk, 199: 42). Terdapat tiga ukuran yang sering digunakan.

a. *Percentage Error* (PE)

Percentage Error (PE) adalah persentase dari kesalahan prediksi

(2.63)

$$PE_t = \left(\frac{X_t - F_t}{X_t} \right) * 100$$

Keterangan:

PE_t = persentase kesalahan pada periode ke-t

X_t = nilai data periode ke-t

F_t = nilai ramalan periode ke-t

b. *Mean Percentage Error* (MPE)

Mean Percentage Error (MPE) adalah rata-rata dari persentase kesalahan hasil prediksi

(2.64)

$$MPE = \sum_{t=1}^n \frac{PE_t}{n}$$

Keterangan:

MPE = rata-rata persentase kesalahan

PE_t = persentase kesalahan pada periode ke-t

n = banyaknya data

c. *Mean Absolute Percentage Error* (MAPE)

Mean Absolute Percentage Error (MAPE) merupakan ukuran ketepatan relatif yang digunakan untuk mengetahui nilai rata-rata kesalahan persentase absolute dari suatu prediksi

(2.65)

$$MAPE = \sum_{t=1}^n \frac{|PE_t|}{n}$$

Keterangan:

MAPE = rata – rata absolut persentase kesalahan

$|PE_t|$ = nilai absolut dari persentase kesalahan prediksi

n = banyaknya data

2.3.10.7.3 Akurasi Sistem

Akurasi sistem digunakan untuk membandingkan keakuratan hasil prediksi dengan data aktual. Untuk menghitung akurasi sistem menggunakan persamaan 2.66 (Hermawan, 2006)

(2.66)

$$\text{Akurasi} = \left(1 - \frac{|y_t - y_a|}{y_a} \right) * 100\%$$

Keterangan :

y_t = nilai aktual

y_a = nilai prediksi

2.4 Matrix Labory (Matlab)

Matlab merupakan perangkat lunak yang cocok dipakai sebagai alat komputasi yang melibatkan penggunaan matriks dan vektor. Fungsi-fungsi dalam *toolbox* Matlab dibuat untuk memudahkan perhitungan tersebut. Sebagai contoh, matlab dapat dengan mudah dipakai untuk menyelesaikan permasalahan sistem persamaan linier, program linear dengan simplek, hingga sistem yang kompleks seperti prediksi runtun waktu (*time series*), pengolahan citra , dll (Siang, 2005).

2.4.1 Graphical User Interface (GUI)

Graphical user interface (GUI) adalah sebuah alat penghubung antara pemakai dengan objek grafis yang akan dibangun. Komponen GUI terdiri atas menu tombol, *teks*, *silinders*, *frame*, *axes*. Jika GUI dirancang dengan baik, maka pemakai akan mengetahui dengan jelas bahwa komponen yang dibuatnya berfungsi dengan baik.

Dengan menyediakan suatu alat penghubung antara pemakai dengan aplikasi yang dibuat, GUI memungkinkan pemakai untuk mengoperasikan aplikasi tanpa mengetahui perintah yang diperlukan untuk menjalankan perintah yang ditulis pada *command line*. Karena alasan ini, aplikasi yang menyediakan GUI lebih mudah untuk dipelajari dan digunakan dibandingkan dengan suatu aplikasi yang menggunakan *command line*.

Dengan Matlab diberikan fasilitas untuk pengaturan suatu susunan GUI agar tampilan GUI tersebut menjadi lebih baik. Fasilitas-fasilitas tersebut adalah:

1. *Layout Editor* : menambahkan dan menyusun objek di jendela figur.
2. *Aligment Tool* : menyejajarkan suatu objek dengan objek lain.
3. *Property Inspector* : memeriksa dan mengatur komponen GUI.
4. *Object Browser* : mengamati / memeriksa suatu daftar komponen GUI.
5. *Menu Editor* : menciptakan menu jendela dan menu konteks.

2.4.2 GUI untuk Mensimulasikan Jaringan Syaraf Tiruan *Backpropagation*

Dalam tugas akhir ini metoda jaringan saraf tiruan *backpropagation* yang digunakan adalah batch gradient descent (traingd). Dalam metoda tersebut parameter yang harus diset untuk pembelajaran ini adalah fungsi pelatihan train menjadi traind. Tidak seperti algoritma pembelajaran lainnya dimana setiap bobot memiliki fungsi pelatihan sendiri-sendiri, tetapi dalam algoritma pembelajaran ini untuk pelatihan hanya menggunakan 1 fungsi pelatihan saja. Dalam batch gradient descent ada 7 (tujuh) parameter yang harus diset untuk pelatihan ini, yaitu:

1. Maksimum epoch (Max_epoch)

Epoch dapat diartikan sebagai interval suatu waktu, dalam bidang ilmu komputer epoch bisa diartikan dengan suatu jangka waktu dalam waktu tertentu. Maksimum epoch adalah jumlah epoch maksimum yang boleh

dilakukan selama proses pelatihan. Iterasi akan dihentikan apabila nilai epoch melebihi maksimum epoch.

Perintah di MATLAB : `net.trainParam.epochs = MaxEpoch`

2. Kinerja Tujuan

Kinerja tujuan adalah target nilai fungsi kinerja. Iterasi akan dihentikan apabila nilai fungsi kinerja kurang dari atau sama dengan kinerja tujuan.

Perintah di MATLAB : `net.trainParam.goal = TargetError`

3. Laju pembelajaran (*Learning rate*)

Semakin besar nilai learning rate akan berimplikasi pada semakin besarnya langkah pembelajaran. Jika learning rate diset terlalu besar, maka algoritma akan menjadi tidak stabil. Sebaliknya, jika learning rate diset terlalu kecil, maka algoritma akan konvergen dalam jangka waktu yang sangat lama.

Perintah di MATLAB : `net.trainParam.lr = LearningRate`

4. Maksimum kegagalan

Maksimum kegagalan diperlukan apabila pada algoritma disertai dengan validitas (*optional*). Maksimum kegagalan adalah ketidakvalitan terbesar yang diperbolehkan. Apabila gradient pada iterasi ke-k lebih besar pada gradient iterasi ke-(k-1), maka kegagalan akan bertambah 1 (satu). Iterasi akan dihentikan apabila jumlah kegagalan lebih dari maksimum kegagalan.

Perintah di MATLAB : `net.trainParam.max_fail = MaxFaile`

5. Gradien minimum

Adalah akar dari jumlah kuadrat semua gradient (bobot input, bobot lapisan, bobot bias) terkecil diperbolehkan. Iterasi akan dihentikan apabila nilai akar jumlah kuadrat semua gradient ini kurang dari gradient minimum.

Perintah di MATLAB : `net.trainParam.min_grad = MinGradien`

6. Show

Menunjukkan beberapa jumlah epoch berselang yang akan ditunjukkan kemajuannya.

Perintah di MATLAB : `net.trainParam.show = EpochShow`

7. Maksimum waktu

Menunjukkan waktu maksimum yang diijinkan untuk melakukan pelatihan. Iterasi akan dihentikan apabila waktu pelatihan melebihi waktu maksimum.

Perintah di MATLAB : `net.trainParam.time = MaxTime`

2.4.3 Komponen GUI Matlab

Untuk membuat sebuah *user interface* Matlab dengan fasilitas GUI harus mulai dengan membuat sebuah desain *figure*. Untuk membuat sebuah desain *figure* dapat memanfaatkan *uicontrol* (control *user interface*). Beberapa *uicontrol* yang ada pada Matlab, antara lain:

1) *Push Button*

Push Button merupakan jenis control berupa tombol tekan yang akan menghasilkan tindakan jika diklik, misalnya *OK*, *Cancel*, Hitung, Hapus, dan sebagainya. Untuk menampilkan tulisan pada *push button*, pengaturannya dapat melalui *property inspector* atau menggunakan klik kanan dan pilih *property inspector*. Selanjutnya, isilah *tab string* dengan table yang diinginkan.

2) *Toggle Button*

Toggle Button menghasilkan efek yang hamper sama dengan *push button*. Perbedaannya adalah saat *push button* ditekan, maka tombol akan kembali pada posisi semula jika tombol *mouse* dilepas, sedangkan pada *toggle button*, tombol tidak akan kembali pada posisi semula, kecuali jika menekannya kembali.

3) *Radio Button*

Radio Button digunakan untuk memilih atau menandai satu pilihan dari beberapa pilihan yang ada. Misalnya, sewaktu akan dibuat aplikasi konversi suhu. Suhu awal dalam derajat Celcius *diinputkan* dan selanjutnya pilihan untuk mengonversi suhu Celcius ke Reamur, Fahrenheit, atau Kelvin.

4) *Checkboxes*

Kontrol *checkboxes* berguna jika terdapat beberapa pilihan mandiri atau tidak bergantung dengan pilihan-pilihan lainnya. Contoh aplikasi

penggunaan *checkboxes* adalah saat pemilihan hobi. Karena hobi bisa lebih dari satu, maka *checkboxes* dapat diklik lebih dari satu kali.

5) *Edit Text*

Kontrol *edit text* merupakan sebuah tempat yang memungkinkan untuk memasukkan atau memodifikasi *text*. Kemudian, *edit text* bermanfaat pula untuk menginputkan suatu dari dari *keyboard*. Sebagai contoh, suatu aplikasi untuk menentukan luas dan keliling sebuah lingkaran. *Input* dan *outputnya* disajikan dari *edit text*.

6) *Static Text*

Kontrol *static text* akan menghasilkan teks bersifat statis (tetap), sehingga pemakai tidak dapat melakukan perubahan padanya. Pada *static text*, teks dapat diatur dengan beberapa fasilitas, antara lain jenis dan ukuran *font*, warna justifikasi (*left, center, right*), dan lain-lain. Semuanya juga dapat dimodifikasi melalui *property inspector*.

7) *Slider*

Slider berguna jika *inputan* nilai yang diinginkan tidak menggunakan *keyboard*, tetapi hanya dengan cara menggeser *slider* secara vertical maupun horizontal ke nilai yang kita inginkan. Dengan menggunakan *slider*, pemasukan nilai data dapat dilakukan secara lebih fleksibel karena nilai *max*, nilai *min*, serta *sliderstep* dapat diatur sendiri.

8) *Frames*

Frames merupakan kotak tertutup yang dapat digunakan untuk mengelompokkan kontrol-kontrol yang berhubungan. Tidak seperti kontrol lainnya, *frames* tidak memiliki rutin *callback*.

9) *Listboxes*

Kontrol *listboxes* menampilkan semua daftar item yang terdapat pada *string property* dan item yang ada dapat dipilih satu atau lebih. *Value property* berisi indeks yang dihubungkan dengan daftar item yang dapat dipilih. Jika item yang dipilih lebih dari satu, maka nilai yang dikirimkan merupakan sebuah vektor. Indeks-indeks item sebuah *listbox* merupakan bilangan bulat, dimana item pertama diberi indeks 0, item kedua diberi indeks 1, dan seterusnya.

10) *Popup Menu*

Popup menu berguna menampilkan daftar pilihan yang didefinisikan pada *string property* ketika mengklik tanda panah pada aplikasi dan memiliki fungsi yang sama seperti *radio button*. Ketika tidak dibuka, *popup menu* hanya menampilkan satu item yang menjadi pilihan pertama pada *string property*. *Popup menu* sangat bermanfaat ketika sebuah pilihan tanpa jarak diberikan, tidak seperti *radio button*.

11) *Axes*

Axes berguna untuk menampilkan sebuah grafik atau gambar (*image*). *Axes* sebenarnya tidak masuk dalam *uicontrol*, tetapi *axes* dapat

deprogram agar pemakai dapat berinteraksi dengan *axes* dan objek grafik yang ditampilkan melalui *axes*.

2.4.4 Toolbox *Backpropagation* di Matlab

Software Matlab mempunyai beberapa perintah yang dapat digunakan untuk membuat jaringan saraf tiruan *backpropagation*. Perintah-perintah tersebut adalah (Suharjito, 2008):

- 1) `newff` : digunakan untuk membentuk jaringan saraf dengan *backpropagation*
- 2) `train` : digunakan untuk melakukan pelatihan jaringan;
- 3) `sim` : digunakan untuk melakukan *output* dari jaringan yang sudah dilatih;
- 4) `minmax` : digunakan untuk mencari nilai minimum dan maksimum data *input*;
- 5) `trainParam` : digunakan untuk menentukan parameter pelatihan jaringan seperti *epoch*, *goal*, *learning rate*, dan momentum *coefisient* (mc);
- 6) `tansig`, `logsig` : parameter fungsi aktivasi dari suatu *layer* jaringan: `tansig` adalah fungsi aktivasi *bipolar sigmoid* dan `logsig` fungsi aktivasi *binary sigmoid*.

2.5 Pencemaran Udara

Berdasarkan PP No. 41 Tahun 1999, pencemaran udara diartikan sebagai masuknya atau dimasukkannya zat, energi, dan/atau komponen lain ke dalam ambien oleh kegiatan manusia, sehingga mutu udara ambien turun sampai ke tingkat tertentu yang menyebabkan udara ambien tidak dapat memenuhi fungsinya (Priyanto, 2007). Sumber pencemaran dapat merupakan kegiatan yang bersifat alami dan aktivitas manusia. Sumber pencemaran alami adalah letusan gunung berapi, kebakaran hutan, dan lain sebagainya, sedangkan pencemaran udara aktivitas manusia secara kuantitatif sering lebih besar seperti transportasi, industry, pertambangan, dari sampah baik akibat dekomposisi ataupun pembakaran dan rumah tangga (Soedomo, 2001).

Sumber pencemaran dapat merupakan kegiatan yang bersifat alami (natural) dan aktivitas manusia (kegiatan antropogenik). Sumber pencemaran alami adalah letusan gunung berapi, kebakaran hutan, dan lain sebagainya, sedangkan pencemaran udara aktivitas manusia secara kuantitatif sering lebih besar seperti transportasi, industry, pertambangan, dari sampah baik akibat dekomposisi ataupun pembakaran dan rumah tangga (Soedomo, 2001).

2.5.1 *Suspended Particulate Matter* (SPM)

Debu yang terdapat dalam udara terbagi dua yaitu *Deposite Particulate Matter* (DPM) yaitu partikel debu yang berada sementara di udara, segera mengendap akibat gaya Tarik bumi dan *Suspended Particulate Matter* (SPM) yaitu debu yang

tetap berada di udara dan tidak mudah mengendap. DPM dan SPM sering juga disebut debu tota (TSP). (Anonim dalam Skripsi Siregar, Sandra, 2011). *Suspended Particulate matter* (SPM) merupakan campuran yang sangat rumit dari berbagai senyawa organik dan anorganik yang tersebar di udara dengan diameter yang sangat kecil, mulai dari < 1 mikron sampai dengan maksimal 500 mikron. Partikulat debu tersebut akan berada di udara dalam waktu yang relatif lama dalam keadaan melayang-layang di udara dan masuk kedalam tubuh manusia melalui saluran pernafasan. Selain dapat berpengaruh negative terhadap kesehatan, partikel debu juga dapat mengganggu daya tembus pandang mata dan juga mengadakan berbagai reaksi kimia di udara. Partikel debu SPM pada umumnya mengandung berbagai senyawa. Partikel-partikel ini berada di atmosfer dalam berbagai ukuran dengan berbagai sifat fisik dan kimianya kimia yang berbeda, dengan berbagai ukuran dan bentuk yang berbeda pula, tergantung dari mana sumber emisinya.

2.5.2 Nilai Ambang Batas (NAB) untuk *Suspended Particulate Matter* (SPM)

Berdasarkan Peraturan Pemerintah No. 41 tahun 1999, nilai ambang batas SPM adalah $230 \mu\text{g}/\text{Nm}^3$ untuk standard harian atau 24 jam dan untuk standard tahunan adalah $90 \mu\text{g}/\text{Nm}^3$.

2.5.3 Dampak *Suspended Particulate Matter* (SPM) bagi Kesehatan

Partikulat dapat terdeposit pada bagian sistem pernafasan manusia sangat tergantung pada ukuran partikel. Partikulat dengan ukuran $\geq 100 \mu\text{m}$ terdeposit pada

bagian hidung dan disebut *inhalable particle*, partikulat dengan ukuran $> 4-10 \mu m$ terdeposit pada bagian toraks dan disebut *thoracic particle*, partikulat dengan ukuran $\leq 4 \mu m$ terdeposit pada bagian paru-paru, disebut *respirable particle*. (Lestari & Fatma, 2009).



UNNES
UNIVERSITAS NEGERI SEMARANG

BAB V

SIMPULAN DAN SARAN

5.1 Simpulan

Berdasarkan hasil penelitian Jaringan Syaraf Tiruan berbasis Transformasi *Wavelet* untuk Prediksi Tingkat Pencemaran Udara *Suspended Particulate Matter* di Kota Semarang diperoleh kesimpulan sebagai berikut:

1. Untuk menciptakan suatu sistem yang dapat mengetahui model prediksi tingkat pencemaran udara (SPM) dengan menggunakan metode Jaringan Syaraf Tiruan berbasis Transformasi *Wavelet*, langkah yang dilakukan adalah menyiapkan data pencemaran udara (SPM) yang selanjutnya dilakukan proses dekomposisi transformasi *wavelet*. Setelah didapatkan data baru hasil dari proses dekomposisi *wavelet*, langkah selanjutnya adalah menormalisasi data, membagi data menjadi data latih dan data uji, dan memasukkan parameter-parameter dari model jaringan syaraf terbaik pada pelatihan dan pengujian data. Dari proses pelatihan dan pengujian data menggunakan jaringan syaraf tiruan, didapatkan model terbaik yaitu arsitektur jaringan 12-18-1 dengan parameter jaringan maksimum *epoch* = 1000, *learning rate* = 0,04, galat *error* = 0,001, *hidden layer* 18, dan *momentum* = 0,9. Dari model jaringan terbaik, tahap selanjutnya

yaitu memprediksi data dekomposisi *wavelet* untuk bulan Januari dan bulan Februari 2016 dengan melakukan 10 kali *running* untuk memperoleh MAPE terbaik. Data yang digunakan untuk prediksi dekomposisi *wavelet* bulan Januari 2016 adalah data prediksi dekomposisi *wavelet* bulan Desember 2015 pada saat dilakukan proses pengujian data menggunakan Jaringan Syaraf Tiruan dan dengan cara yang serupa dilakukan prediksi dekomposisi *wavelet* untuk bulan Februari 2016 dengan menggunakan data hasil prediksi dekomposisi *wavelet* pada bulan Januari dengan MAPE terbaik. Setelah diperoleh data prediksi dekomposisi *wavelet*, maka tahap terakhir adalah proses rekonstruksi data untuk mendapatkan data prediksi pencemaran udara SPM beberapa periode mendatang.

2. Dari penelitian yang telah dilakukan, prediksi pencemaran udara SPM di Kota Semarang dengan menggunakan jaringan syaraf tiruan berbasis transformasi *wavelet*, didapatkan prediksi tingkat pencemaran udara untuk periode bulan Januari dan Februari 2016, yaitu sebesar 64,9577 dan -70,121.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan dan keterbatasan-keterbatasan yang diperoleh dalam penelitian ini, maka peneliti memberikan beberapa saran sebagai berikut

1. Penelitian ini masih perlu dilakukan pengembangan, maka dari itu perlu dilakukan penelitian lanjutan seperti mengkombinasikan metode-metode yang sudah diteliti dengan metode-metode yang lain.

2. Dalam pembuatan sistem prediksi tingkat pencemaran udara *Suspended Particulate Matter* ini dengan metode Jaringan Syaraf Tiruan berbasis Transformasi *Wavelet*, ini sangat bergantung dengan data tingkat pencemaran udara, ini akan menghasilkan dekomposisi *wavelet* dimana hasil dekomposisi *wavelet* dilakukan rekonstruksi ke data awal yang mungkin akan menghasilkan nilai negatif pada data tingkat pencemaran udara.
3. Dalam penelitian ini, peneliti belum bisa menerapkan proses dekomposisi *wavelet* menggunakan Matlab secara langsung dalam program, peneliti melakukan proses dekomposisi *wavelet* menggunakan *excel* dan hasilnya baru ditampilkan pada Matlab. Selain itu pada proses normalisasi data hasil dari dekomposisi *wavelet* peneliti masih melakukan proses normalisasi menggunakan *excel* dan pembagian data menjadi data uji dan data latih juga menggunakan *excel* yang hasilnya ditampilkan pada Matlab. Oleh karena itu, peneliti berharap perlu dilakukannya pengembangan aplikasi agar sistem prediksi tingkat pencemaran udara (SPM) dapat berjalan dengan lebih mudah.
4. Penelitian ini masih perlu dilakukan pengembangan dengan menggunakan jenis *wavelet* yang lain, seperti : Coiflet, Symlets, Discrete Meyer, Daubechies, dan Morlet.

DAFTAR PUSTAKA

- Agnes, M.P., Siang, J.J. 2005. *Kiat Jitu Menyusun Skripsi Jurusan Informatika/Komputer*. Yogyakarta: ANDI.
- Amin, S. 2012. *Sistem Deteksi Dini Hama Wereng Batang Coklat Menggunakan Jaringan Syaraf Tiruan Backpropagation*. Skripsi. Semarang: FMIPA Universitas Negeri Semarang.
- Anugerah P.S.W. 2007. *Perbandingan Jaringan Syaraf Tiruan Backpropagation dan Metode Deret Berkala Box Jenkins (ARIMA) Sebagai Metode Peramalan Curah Hujan*. Skripsi: Universitas Negeri Semarang.
- Anggitiningrum, D.P. 2013. *Perbandingan Prediksi Harga Saham Dengan Menggunakan Jaringan Syaraf Tiruan Backpropagation dan Box-Jenkins (ARIMA)*. Universitas Negeri Semarang.
- Arifien, N.F. 2013. *Prediksi Kadar Polutan Menggunakan Jaringan Syaraf Tiruan (JST) Untuk Pemantauan Kualitas Udara Di Kota Surabaya*. Institut Teknologi Surabaya.
- Budiasih, L.K. *Metode Levenberg Marquardt Untuk Masalah Kuadrat Terkecil Nonlinear*. Prosiding Universitas Sanata Dharma Yogyakarta.
- Burrus, C.S., R. Gopinah, & H. Guo. 1998. *Introduction to Wavelet and Wavelets Transformation*. New Jersey: Prentice Hall International Inc.
- Daubechies, I. 1992. *Ten Lectures on Wavelets*, Society for Industrial and Applied Mathematics. SIAM.
- Diana. 2015. *Simulasi dan Prediksi Pasang Surut Air Laut Menggunakan Wavelet Neural Network*. Skripsi: Universitas Maritim Raja Ali Haji.
- Dinkes Kota Semarang. *Profil Kesehatan Kota Semarang 2014*. Tersedia di: http://www.depkes.go.id/resources/download/profil/PROFIL_KAB_KOTA_2014/3374_Jateng_Kota_Semarang_2014.pdf. [diakses 04-11-2016].
- Handayanto, R.T. 2015. *Peramalan dengan Jaringan Syaraf Tiruan*. Tersedia di: <https://rahmadya.com/?s=Peramalan+dengan+Jaringan+Syaraf+Tiruan>. [diakses 02-11-2016].

- Hansun, S. 2013. *Penerapan WEMA dalam peramalan data IHSG*. Tangerang: Ultimatics.
- Hasan, M.R.. *Prediksi Harga Saham Menggunakan Kombinasi Wavelet Transform dan Artificial Neural Networks*. Universitas Telkom.
- Hermawan, A. 2006. *Jaringan Syaraf Tiruan: Teori dan Aplikasi*. Yogyakarta: ANDI.
- Hidayat, R., and Suprpto. 2012. *Meminimalisasi Nilai Error Peramalan dengan Algoritma Extreme Learning Machine*. *UNAND Jurnal Optimasi Sistem Industri.*, 11(1): 187 – 192. Tersedia di <http://josi.ft.unand.ac.id/index.php/josi/article/view/69> [diakses 06-11-2016].
- Indra, F, C, S., 2014., *Jaringan Syaraf Tiruan Memprediksi Ketersediaan Bahan Bakar Solar dengan Menggunakan Metode Backpropagation.*, pelita informatika budi darma Volume : VIII, Nomor 1 Desember 2014.
- Indrabayu, Harun, H., Pallu, M.S., dan Achmad, A. 2011. *Prediksi Curah Hujan di Wilayah Makassar Menggunakan Metode Wavelet-Neural Network*. *Jurnal Ilmiah “Elektrikal Enjiniring” UNHAS*, Volume 09/ No.02/Mei-Agustus/2011.
- Kusumadewi, S. 2003. *Artificial Intelligent Edisi 2*. Yogyakarta: Graha Ilmu.
- Kusumadewi, S. & Hartati, S. 2010. *Neuro Fuzzy: Integrasi Sistem Fuzzy & Jaringan Syaraf Edisi 2*. Yogyakarta: Graha Ilmu.
- Kusumadewi, S. 2004. *Membangun Jaringan Syaraf Tiruan dengan Matlab dan Exel Link*. Yogyakarta: Graha Ilmu.
- Lestari, Fatma. 2009. *Bahaya Kimia : Sampling dan Pengukuran Kontaminan Kimia di Udara*. Jakarta: Penerbit Buku Kedokteran EGC.
- Makridakis, S., Wheelwright, S. C., & Mc Gee, V. E. 1999. *Metode dan Aplikasi Peramalan* (2nd ed). Translated by Untung Sus Andriyanto dan Abdul Basith. Jakarta: Erlangga.
- Mirawanti, Y., Maslim R.S., Agus W., & Dewi F.E., 2010. *Neural Network*. Jurusan Statistika (Komputasi Statistik) Institut Teknologi Sepuluh November Surabaya.

- Mallat, S. 1998. *A Wavelet Tour of Signal Processing* Second edition. Academic Press.
- Nasapi, M. *Peramalan Permintaan Susu Pasteurisasi Menggunakan Metode Jaringan Syaraf Tiruan dan Time Series*. Skripsi: Universitas Brawijaya.
- Percival DB dan Walden AT. 2000. *Wavelet Methods for Time Series Analysis*. Cambridge University Press. Cambridge. United Kingdom
- Priyanto, H. (2007). *Hasil Pemeriksaan Atas Program Langit Biru pada Kementerian Lingkungan Hidup dan Instansi Terkait*. Jakarta: 2007
- Puspita, A. & Euike. 2007. *Penggunaan Jaringan Syaraf Tiruan Metode Backpropagation untuk Memprediksi Bibir Sumbing*. Seminar Nasional Teknologi.
- Putra, D. 2010. *Pengolahan Citra Digital*. Yogyakarta: ANDI.
- Puspitaningrum, D. 2006. *Pengantar Jaringan Syaraf Tiruan*. Yogyakarta: ANDI.
- Rufiyanti, D.F. 2015. *Implementasi Jaringan Syaraf Tiruan Backpropagation Dengan Input Model Arima Untuk Peramalan Harga Saham*. Universitas Negeri Semarang.
- Sandra, S. 2011. *Hubungan Tingkat Konsentrasi Sulfur Dioksida (SO₂), Total Suspended Particulate Matter (TSP) dan Lingkungan Fisik dengan Kejadian Infeksi Saluran Pernafasan (ISPA) Pada Penduduk di Kotamadya Jakarta Timur Tahun 2008-2010*. Skripsi: Universitas Indonesia.
- Siang, J.J. 2004. *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan MATLAB*. Yogyakarta: ANDI.
- Siang, J.J. 2005. *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan MATLAB*. Yogyakarta: ANDI.
- Siang, J.J. 2009. *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan MATLAB*. Yogyakarta: ANDI.
- Sianipar, R. H. 2013. *Pemrograman MATLAB Dalam Contoh dan Penerapan*. Bandung: Informatika.

- Sinuhaji, F. 2009. *Jaringan Syaraf Tiruan untuk Prediksi Keputusan Medis pada Penyakit Asma*. Tersedia di: [http:// repository.usu.ac.id/bitstream/123456789/14082/1/09E01147.pdf](http://repository.usu.ac.id/bitstream/123456789/14082/1/09E01147.pdf). [diakses 20-10-2016].
- Suharjito. 2008. *Perbandingan Penggunaan Fungsi Aktivasi Logsig dan Tansig dalam Aplikasi Jaringan Backpropagaion untuk Prediksi Harga CPO*. Prosiding Seminar Nasional Sains dan Teknologi-II 2008. Lampung: Universitas Lampung.
- Suharto, I., B. Girisuta, dan A. Miryanti. 2004. *Perekayasa Metodologi Penelitian*. Yogyakarta: Andi.
- Soedomo, M. 2001. *Pencemaran Udara*. Bandung: Institut Teknologi Bandung.
- Ulfiati, L. 2015. *Peramalan Data Saham Dengan Transformasi Wavelet Haar*. Skripsi: Universitas Negeri Semarang.
- Warsito, B. 2008. Clustering Data Pencemaran Udara Sektor Industri Di Jawa Tengah Dengan Kohonen *Neural Network*. *Universitas Diponegoro Jurnal PRESIPITASI*. Tersedia di <http://ejournal.undip.ac.id/index.php/presipitasi/article/view/10967> [diakses 10-11-2016].
- Wayahdi, M.R. 2014. *Analisis Algoritma Nguyen Widrow Pada Proses Prediksi Curah Hujan Kota Medan Menggunakan Metode Backpropagation Neural Network*. *Seminar Nasional Informatika 2014*. Teknik Informatika, STMIK Potensi Utama.
- Wibowo, A.P.S. 2007. *Perbandingan Jaringan Syaraf Tiruan Backpropagation dan Metode Deret Berkala Box-Jenkins (ARIMA) sebagai Metode Peramalan Curah Hujan*. Skripsi. Semarang: FMIPA Universitas Negeri Semarang.
- Yusuf, E.A., Suprayogi, I., Lilis, Y.H. 2015. Model Hidrolgi Runtun Waktu untuk Peramalan Debit Sungai Menggunakan Metode Gabungan Transformasi Wavelet-Artificial Neural Network. *Universitas Riau Jom FTEKNIK*. Tersedia di <http://jom.unri.ac.id/index.php/JOMFTEKNIK/article/view/6280> [diakses 16-11-2016].