



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - TE 141599

DESAIN SISTEM PENDETEKSI DAN DIAGNOSIS KEGAGALAN UNTUK PENGUJIAN KINERJA MOTOR INDUKSI MENGUNAKAN METODE *NEURAL NETWORK*

Diandra Tamara Ismira Rendra
NRP 0711164500070

Dosen Pembimbing
Dr. Ir. Joko Susila, MT.
Mohamad Abdul Hady, ST., MT.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2018



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - TE 141599

**DESAIN SISTEM PENDETEKSI DAN DIAGNOSIS
KEGAGALAN UNTUK PENGUJIAN KINERJA MOTOR
INDUKSI MENGGUNAKAN METODE *NEURAL NETWORK***

Diandra Tamara Ismira Rendra
NRP 0711164500070

Dosen Pembimbing
Ir. Joko Susila, MT.
Mohamad Abdul Hady, ST., MT.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2018



FINAL PROJECT - TE 141599

**SYSTEM DESIGN OF FAULT DETECTION AND
DIAGNOSIS FOR INDUCTION MOTOR PERFORMANCE
TEST USING NEURAL NETWORK**

Diandra Tamara Ismira Rendra
NRP 07111645000070

Supervisor
Ir. Joko Susila, MT.
Mohamad Abdul Hady, ST., MT.

DEPARTMENT OF ELECTRICAL ENGINEERING
Faculty of Electrical Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2018

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul :

“Desain Sistem Pendeteksi dan Diagnosis Kegagalan untuk Pengujian Kinerja Motor Induksi Menggunakan Metode *Neural Network*”

adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juli 2018



Diandra Tamara Ismira Rendra

NRP. 07111645000070

Halaman ini sengaja dikosongkan

**DESAIN SISTEM PENDETEKSI DAN DIAGNOSIS
KEGAGALAN UNTUK PENGUJIAN KINERJA MOTOR
INDUKSI MENGGUNAKAN METODE *NEURAL NETWORK***

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan Untuk
Memperoleh Gelar Sarjana Teknik Elektro**

Pada

**Bidang Studi Teknik Sistem Pengaturan
Departemen Teknik Elektro
Institut Teknologi Sepuluh Nopember**

Menyetujui

Dosen Pembimbing I

Dosen Pembimbing II

Ir. Joko Susila, MT.
Nip : 196606061991021001

Mohamad Abdul Hady, ST., MT.
Nip : 198904132015041002



Halaman ini sengaja dikosongkan

DESAIN SISTEM PENDETEKSI DAN DIAGNOSIS KEGAGALAN UNTUK PENGUJIAN KINERJA MOTOR INDUKSI MENGGUNAKAN METODE *NEURAL NETWORK*

Diandra Tamara Ismira Rendra
07111645000070

Dosen Pembimbing I : Ir. Joko Susila, MT.
Dosen Pembimbing II : Mohamad Abdul Hady, ST., MT.

ABSTRAK

Motor merupakan peralatan yang banyak digunakan di industri karena dinilai handal, hemat, dan memiliki kinerja yang tinggi, namun tidak menutup kemungkinan motor dapat mengalami kegagalan. Ketika pengujian awal motor tidak dilakukan, maka keandalan dari motor tidak benar-benar dapat dipastikan. Hal ini akan dapat menimbulkan kerugian ketika kegagalan motor terjadi saat proses di industri sedang berlangsung. Pada Tugas Akhir ini dirancang sebuah sistem untuk melakukan pengujian kinerja awal motor sehingga dapat dilakukan pemastian terhadap keandalan dari sebuah motor induksi. Metode yang digunakan adalah *neural network* untuk mengklasifikasikan serta mempelajari kinerja dari motor. Proses klasifikasi menggunakan *neural network* mampu melakukan klasifikasi terhadap 90 sampel data dengan kesalahan minimal sebesar 0,044.

Kata Kunci : Motor Induksi, *Neural Network*, Kegagalan Motor, Klasifikasi.

Halaman ini sengaja dikosongkan

SYSTEM DESIGN OF FAULT DETECTION AND DIAGNOSIS FOR INDUCTION MOTOR PERFORMANCE TEST USING NEURAL NETWORK

Diandra Tamara Ismira Rendra
07111645000070

Supervisor I : Ir. Joko Susila, MT.
Supervisor II : Mohamad Abdul Hady, ST., MT.

ABSTRACT

Induction motor is one of the most common used in industrial application due to their high performance and low cost. But in fact, motor could experience the fault condition. If the performance test did not run earlier, the reliability of motor could not be ascertained and this will cause amount of loss in industry. This final project intended to design a system to check the performance of an induction motor. Neural network applied to trained and classified the condition and performance of induction motor. By using the neural network method, system could classified 90 samples of datas with amount of minimum error 0.044.

Keywords: Induction Motor, Neural Network, Motor Fault, Classification.

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji syukur kami panjatkan kehadiran Tuhan Yang Maha Esa, karena berkatNya, penulis dapat menyelesaikan laporan Tugas Akhir dengan judul “Desain Sistem Pendeteksi dan Diagnosis Kegagalan untuk Pengujian Kinerja Motor Induksi Menggunakan Metode *Neural network*” guna memenuhi syarat kelulusan pada Bidang Studi Teknik Sistem Pengaturan, Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember (ITS) Surabaya.

Laporan Tugas Akhir ini dapat terselesaikan berkat bantuan, bimbingan, dan dukungan dari berbagai pihak. Penulis ingin mengucapkan terima kasih kepada Ibu dan Adik yang selalu memberi dukungan, semangat, dan doa untuk keberhasilan penulis. Bapak Ir. Joko Susila, MT., dan Bapak Mohamad Abdul Hady, ST., MT selaku dosen pembimbing atas bimbingan dan arahannya. Bapak Imam Arifin, ST., MT. selaku pembimbing laboratorium yang telah memberi banyak masukan selama pengerjaan Tugas Akhir. Dosen Bidang Studi Teknik Sistem Pengaturan dan Jurusan Teknik Elektro atas pendidikan dan ilmunya. Saudara Simeon Dinar, teman-teman AJ104, LJ 2016, dan seluruh pihak yang membantu dalam pengerjaan tugas akhir yang tidak dapat penulis sebutkan satu persatu.

Penulis berharap laporan ini dapat bermanfaat bagi pembaca pada umumnya dan penulis pada khususnya. Laporan ini masih jauh dari sempurna, sehingga penulis mengharapkan kritik dan saran dari pembaca yang bersifat membangun.

Surabaya,

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

PERNYATAAN KEASLIAN TUGAS AKHIR	v
LEMBAR PENGESAHAN	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	3
1.5 Metodologi.....	3
1.6 Sistematika Penulisan	3
1.7 Relevansi	4
BAB II SISTEM PENGUJIAN KINERJA MOTOR INDUKSI	5
2.1 Motor Induksi [5].....	5
2.2 Kegagalan pada Motor Induksi [4]	6
2.2.1 Kerusakan <i>Bearing</i>	7
2.2.2 Kerusakan Stator	7
2.2.3 Kerusakan Rotor	7
2.3 <i>Neural Network</i> [7]	10
2.3.1 <i>Neuron</i>	10
2.3.2 Fungsi Aktivasi	11
2.3.3 Arsitektur <i>Neural network</i>	14
2.3.4 Algoritma <i>Backpropagation</i>	14
BAB III PERANCANGAN SISTEM PENDETEKSI DAN DIAGNOSIS KEGAGALAN MOTOR	21
3.1 Konfigurasi Sistem.....	21
3.1.1 Perancangan Komunikasi LabVIEW dengan PLC	21
3.1.2 Konfigurasi PLC dengan Inverter dan Motor Induksi..	23
3.1.3 Konfigurasi Sensor ACS712 dengan NI MyRIO	25
3.1.4 Konfigurasi <i>Tachogenerator</i> dengan NI MyRIO	25
3.2 Pengondisian Kerusakan Motor	26

3.2.1	Kondisi Kerusakan <i>Bearing</i>	26
3.2.2	Kondisi Kerusakan Rotor	26
3.3	Perancangan <i>Neural Network</i>	27
BAB IV	PENGUJIAN DAN ANALISA DATA	29
4.1	Pengambilan Data Kondisi Motor	29
4.1.1	Pengambilan Data Motor Tanpa Beban.....	29
4.1.2	Pengambilan Data Motor dengan Beban.....	31
4.1.3	Pengambilan Data Motor dengan Kerusakan <i>Bearing</i> .	31
4.2	<i>Training Data</i>	31
4.2.1	<i>Training Data</i> dengan <i>Learning rate</i> 0,1	32
4.2.2	<i>Training Data</i> dengan <i>Learning rate</i> 0,01	39
4.2.3	<i>Training Data</i> dengan <i>Learning rate</i> 0,001	46
4.3	Pengujian Klasifikasi	53
BAB V	KESIMPULAN.....	57
DAFTAR PUSTAKA		59
Lampiran 1.	Data Kerusakan <i>Bearing</i>	61
Lampiran 2.	Pengambilan Data Arus Motor dengan Beban.....	64
Lampiran 3.	Data Kondisi Motor Normal	66
Lampiran 4.	Perubahan Kesalahan pada ek(1).....	69
Lampiran 5.	Perubahan Nilai Bobot wjk(1)	72
Lampiran 6.	Program <i>Neural network</i>	74
RIWAYAT PENULIS		77

DAFTAR GAMBAR

Gambar 2. 1 Rotor Sangkar Tupai	5
Gambar 2. 2 Rotor Gulung	6
Gambar 2. 3 Skema Penyebab Kegagalan pada Motor [4]	8
Gambar 2. 4 Kerusakan pada <i>Bearing</i>	9
Gambar 2. 5 Kerusakan pada Rotor	9
Gambar 2. 6 Kerusakan pada Stator	9
Gambar 2. 7 Model <i>Neuron</i>	10
Gambar 2. 8 Diagram Blok Tahapan Sistem Saraf	11
Gambar 2. 9 Fungsi Threshold	12
Gambar 2.10 Fungsi Sigmoid	12
Gambar 2. 11 <i>Single Layer Feedforward Networks</i>	13
Gambar 2. 12 <i>Multilayer Feedforward Networks</i>	13
Gambar 2. 13 <i>Recurrent Network</i>	13
Gambar 3. 1 Konfigurasi Sistem Pendeteksi dan Diagnosis	22
Gambar 3. 2 <i>Ladder</i> untuk Mengaktifkan <i>Digital Output PLC</i>	22
Gambar 3. 3 Konfigurasi PLC dengan Inverter dan Motor Induksi	23
Gambar 3. 4 Hubungan <i>Sourcing</i> pada Inverter	23
Gambar 3. 5 <i>Control Circuit Terminal</i> pada Inverter	24
Gambar 3. 6 Konfigurasi <i>Tachogenerator</i> dengan NI MyRIO	25
Gambar 3. 7 Konfigurasi Sensor ACS712 dengan NI MyRIO	26
Gambar 3. 8 Struktur <i>Neural network</i>	27
Gambar 4. 1 Perubahan Kesalahan pada ek(1) dengan 4 <i>Hidden Neuron</i>	33
Gambar 4. 2 Perubahan Kesalahan pada ek(1) dengan 10 <i>Hidden Neuron</i>	34
Gambar 4. 3 Perubahan Kesalahan pada ek(1) dengan 14 <i>Hidden Neuron</i>	34
Gambar 4. 4 Perubahan Kesalahan pada ek(1) dengan 20 <i>Hidden Neuron</i>	35
Gambar 4. 5 Perubahan Kesalahan pada ek(2) dengan 4 <i>Hidden Neuron</i>	35
Gambar 4. 6 Perubahan Kesalahan pada ek(2) dengan 10 <i>Hidden Neuron</i>	36
Gambar 4. 7 Perubahan Kesalahan pada ek(2) dengan 14 <i>Hidden Neuron</i>	36

Gambar 4. 8 Grafik Perubahan Kesalahan pada ek(2) dengan 20 <i>Hidden Neuron</i>	37
Gambar 4. 9 Perubahan Nilai Bobot pada wjk(1) dengan 4 <i>Hidden Neuron</i>	37
Gambar 4. 10 Perubahan Nilai Bobot pada wjk(1) dengan 10 <i>Hidden Neuron</i>	38
Gambar 4. 11 Perubahan Nilai Bobot pada wjk(1) dengan 14 <i>Hidden Neuron</i>	38
Gambar 4. 12 Perubahan Nilai Bobot pada wjk(1) dengan 20 <i>Hidden Neuron</i>	39
Gambar 4. 13 Perubahan Kesalahan pada ek(1) dengan 4 <i>Hidden Neuron</i>	40
Gambar 4. 14 Perubahan Kesalahan pada ek(1) dengan 10 <i>Hidden Neuron</i>	41
Gambar 4. 15 Perubahan Kesalahan pada ek(1) dengan 14 <i>Hidden Neuron</i>	41
Gambar 4. 16 Perubahan Kesalahan pada ek(1) dengan 20 <i>Hidden Neuron</i>	42
Gambar 4. 17 Perubahan Kesalahan pada ek(2) dengan 4 <i>Hidden Neuron</i>	42
Gambar 4. 18 Perubahan Kesalahan pada ek(2) dengan 10 <i>Hidden Neuron</i>	43
Gambar 4. 19 Perubahan Kesalahan pada ek(2) dengan 14 <i>Hidden Neuron</i>	43
Gambar 4. 20 Perubahan Kesalahan pada ek(2) dengan 20 <i>Hidden Neuron</i>	44
Gambar 4. 21 Perubahan Nilai Bobot pada wjk(1) dengan 4 <i>Hidden Neuron</i>	44
Gambar 4. 22 Perubahan Nilai Bobot pada wjk(1) dengan 10 <i>Hidden Neuron</i>	45
Gambar 4. 23 Perubahan Nilai Bobot pada wjk(1) dengan 14 <i>Hidden Neuron</i>	45
Gambar 4. 24 Perubahan Nilai Bobot pada wjk(1) dengan 20 <i>Hidden Neuron</i>	46
Gambar 4. 25 Perubahan Kesalahan pada ek(1) dengan 4 <i>Hidden Neuron</i>	47
Gambar 4. 26 Perubahan Kesalahan pada ek(1) dengan 10 <i>Hidden Neuron</i>	48

Gambar 4. 27 Perubahan Kesalahan pada ek(1) dengan 14 <i>Hidden Neuron</i>	48
Gambar 4. 28 Perubahan Kesalahan pada ek(1) dengan 20 <i>Hidden Neuron</i>	49
Gambar 4. 29 Perubahan Kesalahan pada ek(2) dengan 4 <i>Hidden Neuron</i>	49
Gambar 4. 30 Perubahan Kesalahan pada ek(2) dengan 10 <i>Hidden Neuron</i>	50
Gambar 4. 31 Perubahan Kesalahan pada ek(2) dengan 14 <i>Hidden Neuron</i>	50
Gambar 4. 32 Perubahan Kesalahan pada ek(2) dengan 20 <i>Hidden Neuron</i>	51
Gambar 4. 33 Perubahan Nilai Bobot pada wjk(1) dengan 4 <i>Hidden Neuron</i>	51
Gambar 4. 34 Perubahan Nilai Bobot pada wjk(1) dengan 10 <i>Hidden Neuron</i>	52
Gambar 4. 35 Perubahan Nilai Bobot pada wjk(1) dengan 14 <i>Hidden Neuron</i>	52
Gambar 4. 36 Perubahan Nilai Bobot pada wjk(1) dengan 20 <i>Hidden Neuron</i>	53

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 3. 1 Pengalamatan PLC	23
Tabel 3. 2 Pengaturan Frekuensi pada Inverter	24
Tabel 3. 3 Representasi <i>Output layer</i>	27
Tabel 4. 1 Data Motor Tanpa Beban	29
Tabel 4. 2 Data Motor dengan Beban	30
Tabel 4. 3 Data Kerusakan <i>Bearing</i>	30
Tabel 4. 4 Catatan Waktu <i>Training</i> Data dengan <i>Learning rate</i> 0,1	33
Tabel 4. 5 Catatan Waktu <i>Training</i> Data dengan <i>Learning rate</i> 0,01	39
Tabel 4. 6 Catatan Waktu <i>Training</i> Data dengan <i>Learning rate</i> 0,001 ..	46
Tabel 4. 7 Nilai MSE dengan <i>Learning rate</i> 0,1	54
Tabel 4. 8 Nilai MSE dengan <i>Learning rate</i> 0,01	54
Tabel 4. 9 Nilai MSE dengan <i>Learning rate</i> 0,001	55

Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

Latar belakang dilakukannya Tugas Akhir ini berawal dari keinginan untuk mengembangkan sistem pengujian kinerja dari motor. Proses deteksi dan diagnosis kegagalan dari motor masih dilakukan secara manual, sehingga menyulitkan penguji untuk melakukan identifikasi terhadap kondisi dari motor. permasalahan pada Tugas Akhir ini dibatasi dengan jenis motor, struktur *neural network* dan parameter yang digunakan sebagai acuan untuk mendeteksi penyebab kegagalan motor. Tugas Akhir diawali dengan melakukan studi literatur, perancangan sistem, perancangan struktur *neural network*, dan pengujian. Metodologi dan sistematika penulisan dari Tugas akhir juga dipaparkan dalam bab ini.

1.1 Latar Belakang

Keandalan dari sebuah motor merupakan hal yang sangat penting dalam dunia industri, baik itu industri proses maupun manufaktur. Motor merupakan komponen vital yang banyak digunakan pada dunia industri karena dinilai handal, hemat, dan memiliki kinerja yang tinggi [1]. Hal ditunjukkan dengan banyaknya penggunaan motor di industri seperti pada pabrik kertas, batu bara dan lainnya. Walaupun begitu, tidak menutup kemungkinan bahwa sebuah motor dapat mengalami kegagalan saat proses produksi sedang berlangsung [2].

Kegagalan pada motor dipengaruhi oleh dua faktor utama, yaitu faktor internal dan faktor eksternal, yang masing-masing juga dapat dipengaruhi oleh kerusakan mekanik ataupun elektrik [3]. Ditinjau dari konstruksinya, penyebab kegagalan motor terjadi akibat kerusakan dari bearing, rotor, dan stator. Ketika kegagalan motor pada proses industri tidak dideteksi, maka dapat menyebabkan terhentinya proses produksi. Hal ini dapat berimbas pada kerugian besar yang akan dialami oleh industri terkait. Untuk menghindari hal tersebut, pengujian awal pada motor dinilai perlu untuk dilakukan.

Pengujian biasanya dilakukan sebelum motor dikirim dan digunakan pada proses industri. Hal ini berfungsi untuk menjamin kinerja dan kualitas dari motor. Beberapa hal yang perlu diperhatikan saat pengujian berlangsung antara lain kesesuaian perangkat yang digunakan dengan permintaan, perlakuan motor terhadap kebutuhan

permintaan, kesesuaian hubungan masukan dan keluaran dengan permintaan, hingga kesesuaian spesifikasi dari perangkat yang digunakan. Sistem pengujian menampilkan parameter yang terkait dengan identifikasi jenis kegagalan yang terjadi pada motor. Sistem pengujian juga dilengkapi dengan indikator berupa *alarm* untuk memberi sinyal ketika terdapat kegagalan pada motor. Namun hal ini cukup menyulitkan penguji untuk melakukan identifikasi terhadap jenis kegagalan yang terjadi. Oleh karena itu, diperlukan sebuah sistem yang dapat melakukan pengujian awal motor sehingga keandalan dari motor dapat benar-benar dipastikan.

Metode yang dapat digunakan saat pengujian untuk mendeteksi kegagalan motor yaitu dengan melakukan pengukuran langsung pada motor ataupun dengan menggunakan metode analisis secara numerik [4]. Salah satu metode yang dapat digunakan adalah *neural network*. Metode ini dinilai mampu beradaptasi dan mempelajari kinerja dari motor [2]. *Neural network* juga dinilai tidak membutuhkan biaya yang banyak namun dapat secara efektif melakukan deteksi kegagalan motor.

1.2 Perumusan Masalah

Pengujian biasanya dilakukan hanya dengan memantau nilai parameter dari motor, sehingga dapat dirumuskan sebuah permasalahan yaitu ketika terjadi kegagalan, sistem hanya memberikan indikasi berupa *alarm* tanpa memberikan keterangan sebab dari kegagalan yang terjadi pada motor. Hal ini cukup menyulitkan penguji untuk melakukan analisis terhadap kegagalan.

1.3 Batasan Masalah

Tugas Akhir ini membahas mengenai perancangan sistem pendeteksi dan diagnosis kegagalan menggunakan motor satu fasa. Struktur *neural network* yang digunakan yaitu *multilayer network* dengan algoritma *backpropagation* untuk proses revisi bobotnya (*training*). Acuan parameter yang digunakan untuk proses klasifikasi berupa arus dan kecepatan dari motor. Pengambilan data arus menggunakan sensor ACS712 dan kecepatan motor menggunakan tachogenerator dengan motor DC 12 volt.

1.4 Tujuan

Tugas Akhir ini bertujuan untuk merancang sebuah sistem yang mampu mendeteksi dan memberikan diagnosis kegagalan motor menggunakan metode *neural network* dan dapat ditampilkan melalui sebuah HMI, sehingga dapat memudahkan proses analisis dan pengamatan pada kegagalan motor.

1.5 Metodologi

Tugas Akhir ini dilakukan melalui beberapa tahap, diantaranya studi literatur, konfigurasi sistem, pengambilan data motor, perancangan struktur *neural network*, pengujian dan analisis serta penulisan laporan Tugas Akhir. Studi literatur dilakukan untuk mempelajari dan mengumpulkan referensi dari buku dan jurnal. Seluruh perangkat pendukung yang digunakan dalam sistem dikonfigurasi, lalu dilakukan pengambilan data untuk mendapatkan parameter arus dan kecepatan motor saat kondisi normal maupun terjadi kerusakan. Berdasarkan parameter tersebut, dilakukan perancangan terhadap struktur *neural network*. Pengujian secara *offline* dilakukan untuk mendapatkan nilai revisi bobot yang kemudian digunakan untuk proses deteksi dan diagnosis secara *online*.

1.6 Sistematika Penulisan

Penulisan laporan Tugas Akhir ini dapat dijelaskan dengan sistematika sebagai berikut:

BAB I PENDAHULUAN

Bab ini membahas tentang latar belakang, permasalahan, batasan masalah, tujuan Tugas Akhir, metodologi, sistematika penulisan dan relevansi dari Tugas Akhir yang dilakukan.

BAB II SISTEM PENGUJIAN KINERJA MOTOR INDUKSI

Perihal yang mempengaruhi sebuah sistem pengujian kinerja pada motor induksi, seperti penjelasan dasar mengenai motor induksi, jenis kegagalan yang terjadi, dan metode yang dapat digunakan untuk menyelesaikan permasalahan dibahas dalam bab ini.

BAB III PERANCANGAN SISTEM

Bab 3 ini membahas mengenai konfigurasi peralatan yang digunakan, perancangan struktur *neural network*, dan perancangan *Human Machine Interface* (HMI) yang akan diimplementasikan pada sistem.

BAB IV PENGUJIAN DAN ANALISIS

Bab ini membahas mengenai hasil *learning* dari metode *neural network* pada proses deteksi kegagalan dari motor induksi, beserta diagnosis kegagalan yang terjadi.

BAB V KESIMPULAN

Bab 5 membahas tentang kesimpulan berkaitan dengan Tugas Akhir yang telah dilakukan.

1.7 Relevansi

Pengujian biasanya dilakukan hanya dengan memantau nilai parameter dari motor. Ketika terjadi kegagalan, sistem hanya memberikan indikasi saja tanpa memberikan keterangan sebab dari kegagalan yang terjadi pada motor. Pada Tugas Akhir ini dirancang sebuah sistem pengujian kinerja motor yang menggunakan *neural network*. Diharapkan dari hasil Tugas Akhir ini dapat digunakan sebagai acuan untuk penelitian lebih lanjut, dan juga diharapkan dapat memudahkan proses analisis dan pengamatan pada kegagalan motor.

BAB II

SISTEM PENGUJIAN KINERJA MOTOR INDUKSI

Bab ini menjelaskan mengenai konstruksi dan fungsi tiap bagian yang terdapat pada motor induksi seperti rotor, stator, dan *bearing*. Penyebab kegagalan dari motor induksi juga dijelaskan dan agar dapat melakukan deteksi terhadap kegagalan motor, metode *neural network* juga dipaparkan dalam bab ini.

2.1 Motor Induksi [5]

Motor induksi merupakan jenis motor listrik yang dapat mengubah energi listrik menjadi energi mekanik. Dikatakan motor induksi karena tegangan rotor terinduksi pada belitan rotor dan tidak terhubung dengan pengkabelan manapun. Induksi diakibatkan oleh adanya perbedaan relatif antara putaran rotor dengan medan putar yang dihasilkan oleh arus stator. Hal yang membedakan antara motor induksi dengan motor lain adalah motor induksi tidak memerlukan arus searah (DC) untuk menggerakkan motor.

Terdapat dua tipe rotor pada motor induksi, yaitu rotor sangkar tupai (*cage rotor*) dan rotor gulung (*wound rotor*). Rotor sangkar tupai terdiri dari batang konduktor yang tersusun secara seri dan kedua ujungnya dihubungkan oleh cincin hubung singkat. Sesuai dengan namanya, susunan dari batang konduktor berbentuk seperti sangkar tupai. Rotor gulung biasanya terhubung secara Y dan ketiga ujung hubungan tersebut dihubungkan ke cincin slip. Kerugian dari penggunaan rotor gulung adalah biaya yang dibutuhkan untuk melakukan perawatan lebih mahal dari pada jenis rotor sangkar tupai karena penggunaan dari sikat dan cincin slip.



Gambar 2. 1 Rotor Sangkar Tupai



Gambar 2. 2 Rotor Gulung

Selain rotor, bagian penting dari motor induksi adalah stator. Stator merupakan bagian dari motor yang tidak bergerak. Pada stator diberikan tegangan tiga fasa, sehingga menimbulkan aliran arus tiga fasa. Arus yang mengalir tersebut menghasilkan medan magnet yang berputar berlawanan dengan arah jarum jam. Kecepatan putar medan magnet adalah

$$n_{sync} = \frac{120 f_e}{P} \quad (2.1)$$

Dimana f_e adalah frekuensi dalam satuan hertz dan P adalah jumlah kutub pada mesin. Medan magnet putar B_s melewati batang rotor dan menginduksikan tegangan.

2.2 Kegagalan pada Motor Induksi [4]

Motor induksi merupakan komponen yang banyak digunakan pada dunia industri karena dinilai handal dan memiliki kinerja yang tinggi. Namun sama seperti peralatan lainnya di industri, motor dapat mengalami kegagalan. Secara umum, kegagalan dari motor induksi disebabkan oleh dua faktor, yaitu internal dan eksternal. Kedua faktor tersebut juga dipengaruhi oleh faktor mekanik dan elektrik. Skema penyebab kegagalan dari motor dapat dilihat pada Gambar 2. 3.

Mengacu pada sumber studi *Institute of Electrical and Electronics Engineers* (IEEE) dan *Electric Power Research Institute* (EPRI), kegagalan pada motor induksi dikelompokkan berdasarkan Tabel 2. 1 Kemungkinan Kegagalan pada Motor Induksi [6]

Sumber studi	Kegagalan bearing (%)	Kegagalan stator (%)	Kegagalan rotor (%)	Lainnya (%)
IEEE	42	28	8	22
EPRI	41	36	9	14

Kemungkinan terjadinya kerusakan komponen pada motor induksi yang dapat dilihat pada Tabel 2. 1.

2.2.1 Kerusakan Bearing

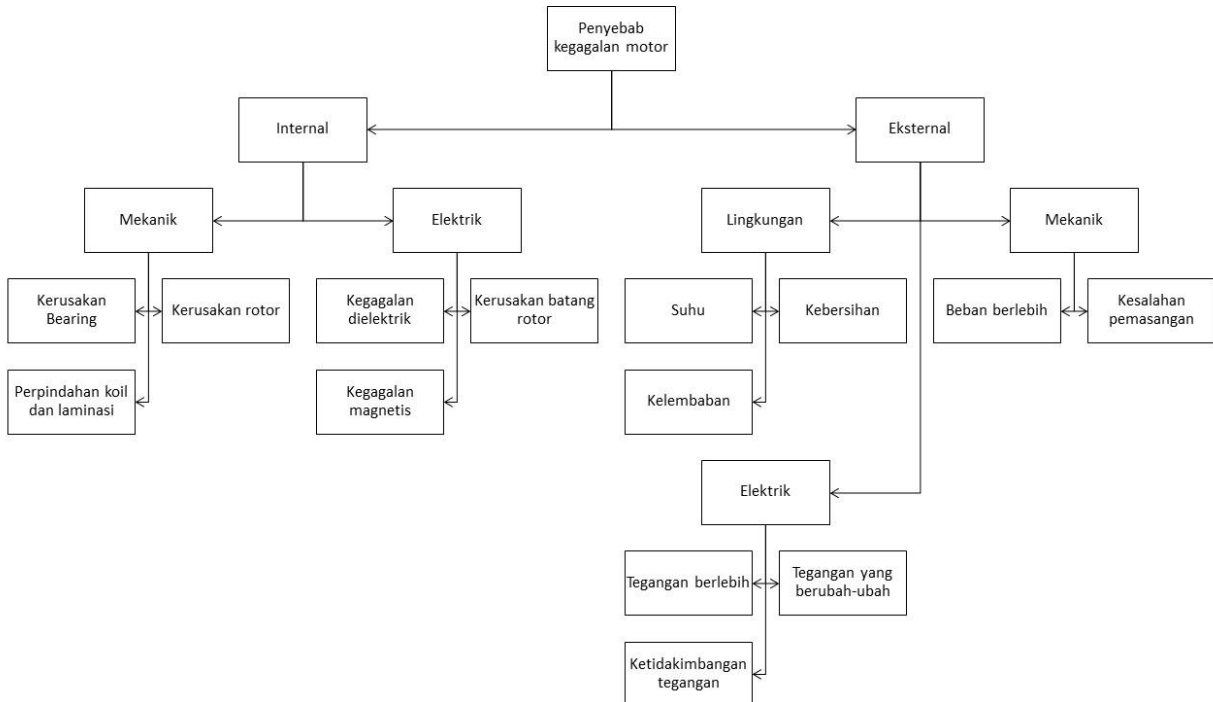
Pada motor induksi, *bearing* berfungsi untuk menyangga batang rotor dan membantu agar rotor dapat berputar dengan bebas. *Bearing* terdiri dari dua cincin yang dipisahkan oleh bola besi (*roller*). Kerusakan pada *bearing* dapat disebabkan oleh beban berlebihan sehingga dapat menyebabkan kerusakan pada lapisan pelumas. Hal ini dapat menyebabkan gaya gesek yang berujung pada peningkatan suhu motor. Korosi juga dapat terjadi pada *bearing* dengan kondisi pelumasan yang tidak baik, maupun kondisi lingkungan yang korosif. Tidak hanya faktor pelumasan, kerusakan *bearing* juga dapat terjadi akibat kesalahan ketika proses fabrikasi seperti retak. Hal ini berdampak pada gaya gesek yang dapat meningkatkan suhu motor. Kondisi *bearing* yang mengalami kerusakan terlihat pada Gambar 2. 4. Kontaminasi material asing juga mempengaruhi kondisi dari *bearing*. Kotoran atau benda asing yang terdapat pada pelumas dapat menyebabkan peningkatan pada getaran motor.

2.2.2 Kerusakan Stator

Stator merupakan bagian dari motor induksi yang tidak berputar yang berfungsi untuk mengalirkan arus, sehingga menghasilkan medan magnet untuk memutar rotor. Kerusakan pada stator dapat disebabkan oleh stres mekanik, elektrik, maupun termal. Kerusakan *bearing* dan posisi poros rotor yang tidak sesuai dapat memicu stres mekanik pada stator. Stres elektrik disebabkan oleh tegangan transien akibat kegagalan fasa dengan fasa atau fasa dengan pentanahan (*grounding*). Hal ini dapat mempengaruhi waktu kerja dari belitan stator. Bentuk Stator yang telah mengalami kerusakan dapat dilihat pada Gambar 2. 6. Stres termal dapat terjadi karena penurunan insulasi dari belitan stator. Hal ini dapat disebabkan oleh arus berlebih akibat beban yang besar maupun suplai tegangan yang tidak stabil.

2.2.3 Kerusakan Rotor

Rotor merupakan komponen motor induksi yang dapat berputar dan terdiri dari lapisan laminasi yang membentuk silinder. Kerusakan pada rotor dapat disebabkan tingkat pemanasan yang terlalu tinggi pada saat operasi transien atau ketika *start-up*, terutama pada motor



Gambar 2. 3 Skema Penyebab Kegagalan pada Motor [4]



Gambar 2. 4 Kerusakan pada *Bearing*



Gambar 2. 5 Kerusakan pada Rotor



Gambar 2. 6 Kerusakan pada Stator

induksi dengan ukuran besar, celah udara yang dapat menyebabkan permasalahan mekanik seperti beban yang tidak seimbang atau poros motor yang tidak sejajar. Kondisi kerusakan rotor dapat dilihat pada Gambar 2. 5.

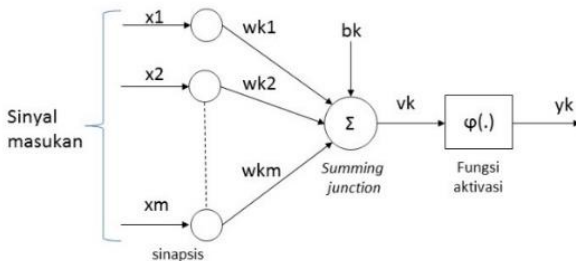
2.3 Neural Network [7]

Otak manusia memiliki kemampuan untuk membuat aturan dan keputusan (*build rules*) melalui pengalaman atau kejadian yang pernah dialami. Sistem saraf manusia terbagi menjadi tiga tahapan utama yaitu reseptor, *neural net*, dan efektor. Reseptor berfungsi untuk mengubah stimulus dari keadaan sekitar menjadi impuls yang membawa informasi ke *neural net*. Efektor berfungsi untuk mengubah impuls yang telah diproses oleh *neural net* menjadi respon keluaran dari sistem. Blok diagram tahapan sistem saraf manusia dapat dilihat pada Gambar 2. 7. Hal ini diadaptasi oleh *neural network* untuk memodelkan kerja otak untuk melakukan tugas atau perintah tertentu.

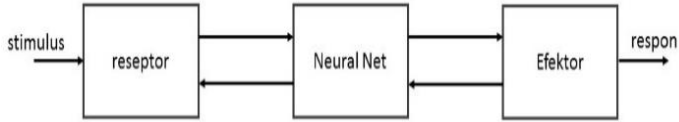
2.3.1 Neuron

Unit pemrosesan informasi dasar yang digunakan pada *neural network* disebut dengan *neuron*. Dalam bentuk gambar, *neuron* dapat direpresentasikan sebagai node yang menghubungkan antara jaringan satu dengan jaringan lainnya. Terdapat 3 elemen dasar yang mendukung suatu model *neuron* seperti pada Gambar 2. 8, yaitu:

1. Sinapsis
Elemen ini ditandai dengan adanya bobot dan sinyal masukan. Nilai dari bobot berada pada rentang nilai negatif hingga positif.
2. *Summing junction*
Nilai masing-masing masukan yang telah diberikan bobot kemudian dijumlahkan pada *summing junction*.
3. Fungsi aktivasi
Batas nilai keluaran dari sebuah *neuron* diperoleh melalui fungsi aktivasi. Elemen ini digunakan untuk membatasi rentang sinyal keluaran *neuron* sehingga mencapai nilai tertentu.



Gambar 2. 7 Model *Neuron*



Gambar 2. 8 Diagram Blok Tahapan Sistem Saraf

Jika dituliskan dalam fungsi matematika, persamaan untuk model *neuron* Gambar 2. 7, diperoleh persamaan 2.2 dan 2.3.

$$v_k = \sum_{j=1}^m W_{kj} X_j + b_k \quad (2.2)$$

$$y_k = \varphi(u_k + b_k) \quad (2.3)$$

Bias b_k merupakan parameter eksternal dari *neuron* yang dapat bernilai positif maupun negatif.

2.3.2 Fungsi Aktivasi

Dilambangkan dengan φ , fungsi aktivasi berfungsi untuk menentukan nilai keluaran *neuron* yang bergantung dari nilai hasil penjumlahan masukan dengan pembobot. Secara umum, terdapat dua jenis dasar dari fungsi aktivasi, yaitu

1. Fungsi *Threshold*

Terdapat dua kondisi yang dihasilkan oleh fungsi aktivasi ini yaitu 1 dan 0, dimana

$$\varphi(v) = \begin{cases} 1, v \geq 0 \\ 0, v < 0 \end{cases} \quad (2.4)$$

$$y(k) = \begin{cases} 1, v_k \geq 0 \\ 0, v_k < 0 \end{cases} \quad (2.5)$$

Nilai fungsi aktivasi akan bernilai 1 jika nilai v lebih besar sama dengan 0, dan akan bernilai 0 jika nilai v lebih kecil daripada 0. Hal ini juga berlaku pada nilai keluaran *neuron*. Pengondisian fungsi *threshold* dapat dilihat pada Gambar 2. 9.

2. Fungsi Sigmoid

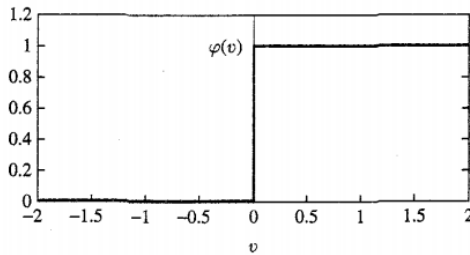
Fungsi aktivasi dengan grafik yang menyerupai bentuk ‘S’ ini merupakan fungsi yang sering digunakan pada *neural network*.

Berbeda dengan fungsi *threshold*, fungsi sigmoid memiliki rentang nilai diantara 0 hingga 1. Bentuk grafik fungsi sigmoid dapat dilihat pada Gambar 2.10. Secara matematis, fungsi sigmoid dapat dituliskan dengan persamaan 2.6.

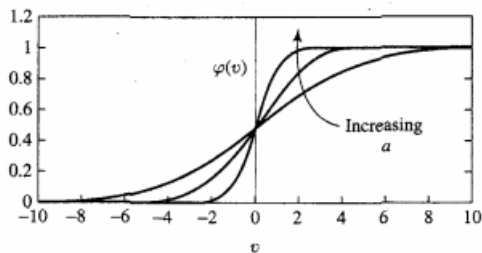
$$\varphi(v) = \frac{1}{1 + e^{(-av)}} \quad (2.6)$$

Jika fungsi sigmoid memiliki rentang nilai 0 hingga 1, terdapat fungsi aktivasi dengan rentang nilai -1 hingga +1 yang disebut dengan fungsi signum. Kondisi fungsi dapat dituliskan secara matematis seperti pada persamaan 2.7.

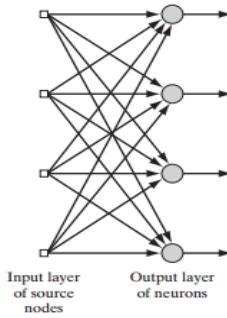
$$\varphi(v) = \begin{cases} 1 & v > 0 \\ 0 & v = 0 \\ -1 & v < 0 \end{cases} \quad (2.7)$$



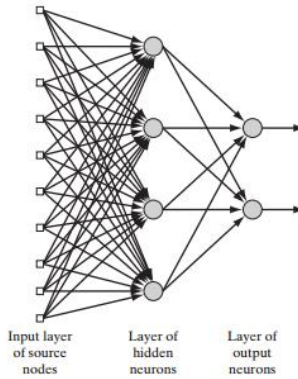
Gambar 2. 9 Fungsi Threshold



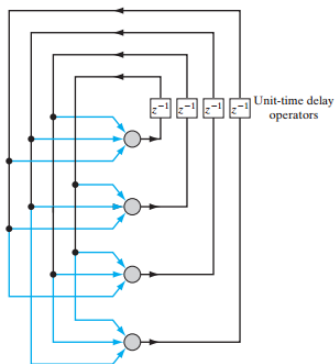
Gambar 2.10 Fungsi Sigmoid



Gambar 2. 11 *Single Layer Feedforward Networks*



Gambar 2. 12 *Multilayer Feedforward Networks*



Gambar 2. 13 *Recurrent Network*

2.3.3 Arsitektur *Neural network*

Neuron pada *neural network* saling terhubung satu dengan lainnya agar dapat digunakan untuk melakukan proses *training*. Secara umum, arsitektur dari *neural network* dibagi menjadi tiga sebagai berikut,

1. *Single layer feedforward network*

Sebuah *network* dikatakan sebagai *single layer* ketika terdapat *node* sumber pada *input layer* yang mengarah langsung ke *output layer* dari *neuron*, namun tidak sebaliknya. Proses penghitungan tidak dilakukan pada *node* sumber (*input layer*). *single layer network* dapat dilihat pada Gambar 2. 11.

2. *Multilayer feedforward network*

Hal utama yang membedakan antara *multilayer* dengan *single layer network* yaitu terdapat *hidden layer* pada *multilayer network*. *Hidden layer* merupakan *layer* yang tidak dapat dilihat secara langsung, baik itu melalui masukan maupun keluaran. Fungsi dari *hidden layer* adalah sebagai penengah antara masukan eksternal dengan *network* keluaran. Semakin banyak penggunaan *hidden layer*, maka *network* akan mampu melakukan ekstraksi pada orde lebih tinggi pada bagian masukan. Ketika tiap *node* pada tiap *layer* terhubung dengan semua *node* yang terdapat pada *forward layer* terdekat, maka *network* tersebut dapat disebut dengan *fully connected*. Jika tidak semua terhubung, *network* tersebut dikatakan sebagai *partially connected*. Gambar 2. 12 menunjukkan bentuk dari *multi layer feedforward networks*.

3. *Recurrent network*

Berbeda dengan sebelumnya, *reccurent network* memiliki *feedback loop*. Sinyal keluaran dari *neuron* digunakan kembali sebagai masukan dari *neuron* lainnya. *Feedback loop* memberi dampak mendalam pada kemampuan *learning* dari sebuah *network*. Gambar 2. 13 menunjukkan bentuk dari *recurrent network*.

2.3.4 Algoritma *Backpropagation*

Metode yang sering digunakan untuk proses *training* pada *multilayer network* adalah algoritma *backpropagation*. Proses *trainingnya* dibagi menjadi dua fase, yaitu fase *forward* dan *backward*.

1. Fase *forward*

Pada fase ini, nilai masukan diberikan pembobot yang kemudian disebarkan melalui setiap *layer* hingga mencapai keluarannya. Nilai keluaran dibatasi dengan fungsi aktivasi yang terdapat pada tiap *layer*. Jika dituliskan dalam bentuk matematis, diperoleh persamaan 2.8,

$$y_k = \varphi\left(\sum y_{hj}w_{jk}\right) \quad (2.8)$$

Dimana j merupakan jumlah *neuron* pada *hidden layer* dan w_{jk} merupakan bobot antara *hidden layer* dengan *output layer*. Nilai keluaran *hidden layer* (y_{hj}) diperoleh melalui persamaan 2.9

$$y_{hj} = \varphi\left(\sum x_iw_{ij}\right) \quad (2.9)$$

Dimana x_i merupakan nilai dari *input layer* dan w_{ij} adalah bobot antara *input layer* dengan *hidden layer*.

2. Fase *backward*

Nilai keluaran dari fase *forward* dibandingkan dengan nilai target sehingga diperoleh nilai sinyal *kesalahan*, kemudian kembali disebarkan melalui tiap *layer* dengan arah yang berkebalikan dengan fase *forward*. Dari hasil penghitungan balik ini, diperoleh nilai revisi bobot terbaik.

Untuk mendapatkan nilai *kesalahan*, digunakan persamaan 2.10.

$$e_k = t_k - y_k \quad (2.10)$$

Dimana t merupakan nilai target. Nilai perbaikan bobot diperoleh menggunakan nilai *kesalahan* yang diturunkan parsial terhadap bobot antara *hidden layer* dan *output layer*. Nilai tersebut diperoleh menggunakan penghitungan *chain rule* seperti pada persamaan 2.11.

$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial E}{\partial e_k} \frac{\partial e_k}{\partial y_k} \frac{\partial y_k}{\partial w_{jk}} \quad (2.11)$$

Nilai *kesalahan* diturunkan parsial terhadap sinyal kesalahan, sehingga diperoleh persamaan 2.12.

$$\begin{aligned}
\frac{\partial E}{\partial e_k} &= \frac{\partial \frac{1}{2} e_k^2}{\partial e_k} \\
&= \frac{1}{2} 2e_k \\
&= e_k
\end{aligned}
\tag{2.12}$$

Sinyal kesalahan diturunkan parsial terhadap keluaran *output layer*, diperoleh persamaan 2.13.

$$\begin{aligned}
\frac{\partial e_k}{\partial y_k} &= \frac{\partial t_k - y_k}{\partial y_k} \\
&= -1
\end{aligned}
\tag{2.13}$$

Keluaran *output layer* diturunkan terhadap bobot antara *hidden layer* dengan *output layer*, diperoleh persamaan 2.14.

$$\begin{aligned}
\frac{\partial y_k}{\partial w_{jk}} &= \frac{\partial \phi(y_{hj} w_{jk})}{\partial w_{jk}} \\
&= \phi'(y_{hj})
\end{aligned}
\tag{2.14}$$

Sehingga, hasil dari nilai *kesalahan* yang diturunkan terhadap bobot *jk* seperti pada persamaan 2.15.

$$\frac{\partial E}{\partial w_{jk}} = -e_k \phi'(y_{hj})
\tag{2.15}$$

Untuk mendapatkan hasil dari turunan parsial sinyal *kesalahan* terhadap bobot antara *input layer* dengan *hidden layer*, digunakan *chain rule* seperti pada persamaan 2.16.

$$\frac{\partial e_k}{\partial w_{ij}} = \frac{\partial e_k}{\partial y_k} \frac{\partial y_k}{\partial y_{hj}} \frac{\partial y_{hj}}{\partial w_{ij}}
\tag{2.16}$$

Keluaran dari *output layer* diturunkan parsial terhadap keluaran dari *hidden layer*, diperoleh persamaan 2.17.

$$\begin{aligned}\frac{\partial y_k}{\partial y_{hj}} &= \frac{\partial \varphi(y_{hj} w_{jk})}{\partial y_{hj}} \\ &= w_{jk}\end{aligned}\quad (2.17)$$

Selanjutnya nilai keluaran *hidden layer* diturunkan parsial terhadap bobot antara *input layer* dengan *hidden layer*, diperoleh persamaan 2.18,

$$\begin{aligned}\frac{\partial y_{hj}}{\partial w_{ij}} &= \frac{\partial \varphi(x_i w_{ij})}{\partial w_{ij}} \\ &= x_i\end{aligned}\quad (2.18)$$

Sehingga hasil dari turunan parsial sinyal kesalahan terhadap bobot antara *input layer* dengan *hidden layer* dapat didefinisikan dengan persamaan 2.19.

$$\begin{aligned}\frac{\partial e_k}{\partial w_{ij}} &= \frac{\partial e_k}{\partial y_k} \frac{\partial y_k}{\partial y_{hj}} \frac{\partial y_{hj}}{\partial w_{ij}} \\ &= -(w_{jk})(x_i)\end{aligned}\quad (2.19)$$

Turunan parsial sinyal kesalahan terhadap keluaran *hidden layer* diperoleh menggunakan *chain rule* seperti pada persamaan 2.20.

$$\frac{\partial e_k}{\partial y_{hj}} = \frac{\partial e_k}{\partial y_k} \frac{\partial y_k}{\partial y_{hj}} \quad (2.20)$$

Menggunakan persamaan 2.13 dan 2.17 diperoleh persamaan 2.21.

$$\begin{aligned}\frac{\partial e_k}{\partial y_{hj}} &= \frac{\partial e_k}{\partial y_k} \frac{\partial y_k}{\partial y_{hj}} \\ &= -w_{jk}\end{aligned}\quad (2.21)$$

Untuk mendapatkan hasil dari turunan parsial kesalahan terhadap keluaran *output layer*, digunakan *chain rule* seperti pada persamaan 2.22

$$\frac{\partial E}{\partial y_k} = \frac{\partial E}{\partial e_k} \frac{\partial e_k}{\partial y_k} \quad (2.22)$$

Menggunakan persamaan 2.12 dan 2.13, diperoleh persamaan 2.23.

$$\frac{\partial E}{\partial y_k} = -e_k \quad (2.23)$$

Untuk mendapatkan hasil dari turunan parsial kesalahan terhadap keluaran *hidden layer*, digunakan *chain rule* seperti pada persamaan 2.24.

$$\frac{\partial E}{\partial y_{hj}} = \frac{\partial E}{\partial e_k} \frac{\partial e_k}{\partial y_k} \frac{\partial y_k}{\partial y_{hj}} \quad (2.24)$$

Menggunakan persamaan 2.12, 2.13 dan 2.17, diperoleh persamaan 2.25

$$\frac{\partial E}{\partial y_{hj}} = -e_k w_{jk} \quad (2.25)$$

Delta rule dari perbaikan bobot antara *output layer* dengan *hidden layer* dituliskan pada persamaan 2.26

$$\Delta w_{jk} = -\eta \frac{\partial E}{\partial w_{jk}} \quad (2.26)$$

Dimana η merupakan *learning rate* dari algoritma *backpropagation*. Dari persamaan (2.15) dan (2.26), maka diperoleh nilai *delta rule* seperti pada persamaan 2.27.

$$\Delta w_{jk} = \eta e_k \phi'(y_{hj}) \quad (2.27)$$

Sehingga perbaikan bobot dari w_{jk} diperoleh persamaan 2.28.

$$\begin{aligned} w_{jk}(k+1) &= w_{jk} + \eta \Delta w_{jk} \\ &= w_{jk} + \eta e_k \phi'(y_{hj}) \end{aligned} \quad (2.28)$$

Delta rule dari perbaikan bobot antara *output layer* dengan *hidden layer* dituliskan dalam persamaan 2.29.

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} \quad (2.29)$$

Nilai turunan parsial antara nilai *kesalahan* dengan bobot antara *input layer* dengan *output layer* diperoleh menggunakan *chain rule* seperti pada persamaan 2.30.

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial e_k} \frac{\partial e_k}{\partial y_k} \frac{\partial y_k}{\partial y_{hj}} \frac{\partial y_{hj}}{\partial w_{ij}} \quad (2.30)$$

Turunan parsial dari keluaran *hidden layer* terhadap bobot antara *input layer* dengan *hidden layer*, diperoleh persamaan 2.31.

$$\begin{aligned} \frac{\partial y_{hj}}{\partial w_{ij}} &= \frac{\partial \varphi \left(\sum x_i w_{ij} \right)}{\partial w_{ij}} \\ &= x_i \end{aligned} \quad (2.31)$$

Menggunakan persamaan 2.12, 2.13, 2.17, dan 2.18, diperoleh persamaan 2.32.

$$\frac{\partial E}{\partial w_{ij}} = -e_k w_{jk} x_i \quad (2.32)$$

Dari persamaan 2.32, diperoleh *delta rule* seperti pada persamaan 2.33.

$$\Delta w_{ij} = \eta e_k \varphi'(y_{hj}) x_i \quad (2.33)$$

Sehingga perbaikan bobot dari w_{ij} diperoleh persamaan 2.34.

$$\begin{aligned} w_{ij}(k+1) &= w_{ij} + \eta \Delta w_{ij} \\ &= w_{ij} + \eta e_k \varphi'(y_{hj}) x_i \end{aligned} \quad (2.34)$$

Halaman ini sengaja dikosongkan

BAB III

PERANCANGAN SISTEM PENDETEKSI DAN DIAGNOSIS KEGAGALAN MOTOR

Bab ini membahas mengenai konfigurasi sistem secara keseluruhan, pengondisian kerusakan motor, dan perancangan *neural network*. Konfigurasi sistem meliputi motor satu fasa, inverter, PLC, NI MyRIO, sensor ACS712, dan tachogenerator. Pengondisian kerusakan yang digunakan pada Tugas Akhir ini yaitu kerusakan bearing dan kondisi pembebanan pada motor. Perancangan *neural network* bertujuan agar sistem dapat melakukan klasifikasi terhadap kondisi dari motor.

3.1 Konfigurasi Sistem

Sistem yang dirancang pada Tugas Akhir ini berfungsi untuk melakukan deteksi dan diagnosis terhadap kegagalan pada motor induksi. Peralatan yang mendukung sistem ini antara lain PLC, inverter, motor satu fasa, sensor arus ACS712, motor DC 12V sebagai *tachogenerator*, dan NI MyRIO. Konfigurasi dari sistem secara keseluruhan dapat dilihat pada Gambar 3. 1.

3.1.1 Perancangan Komunikasi LabVIEW dengan PLC

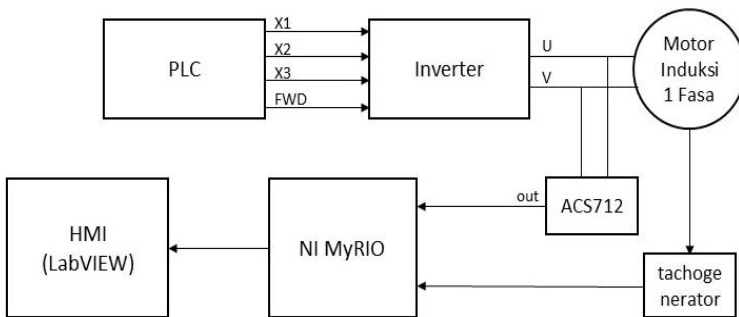
Penggunaan PLC pada sistem ini adalah sebagai pengatur nilai frekuensi yang akan masuk ke motor melalui inverter. PLC memanfaatkan *port* digital *output* sebagai masukan dari inverter. *Port* digital *output* diaktifkan melalui memori yang dideklarasikan pada LabVIEW. Untuk dapat mengaktifkan *port* digital *output* tersebut, dilakukan perancangan komunikasi antara PLC dan LabVIEW dengan menggunakan OPC.

Object Linking and Embedding (OLE) for Process Control (OPC) merupakan sebuah media yang digunakan untuk menjembatani komunikasi antara perangkat satu dengan lainnya. OPC banyak dikembangkan di dunia industri sebagai standar komunikasi antar perangkat dengan manufaktur yang berbeda, sehingga memungkinkan pengguna untuk menghubungkan berbagai perangkat industri dengan merk atau manufaktur yang berbeda guna memenuhi permintaan industri.

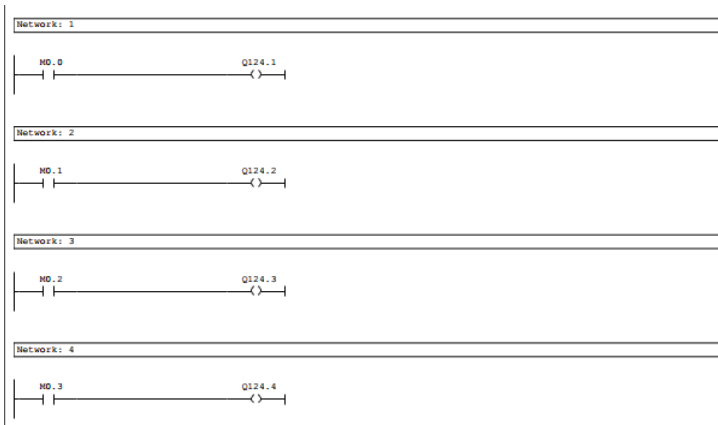
Protokol komunikasi yang digunakan pada OPC harus disesuaikan dengan protokol yang terdapat pada PLC. Siemens S7-

300 menggunakan protokol komunikasi *multi-point interface* (MPI), sehingga pada OPC diatur dengan protocol S7 Simatic Net. Pengalamatan yang digunakan pada OPC juga disesuaikan pada pengalamatan yang digunakan pada PLC sesuai pada Tabel 3. 1.

Pengalamatan masukan dibuat sebagai memori agar perintah dapat diberikan melalui LabVIEW. Pengaturan alamat dan *tagname* pada OPC harus ditransfer ke OPC *Server* yang akan diakses. Hal ini bertujuan agar variabel tersebut dapat diakses pada LabVIEW. Pemrograman *ladder* yang digunakan pada PLC seperti pada Gambar 3. 2.



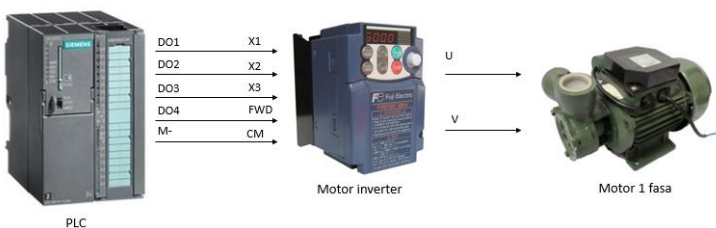
Gambar 3. 1 Konfigurasi Sistem Pendeteksi dan Diagnosis



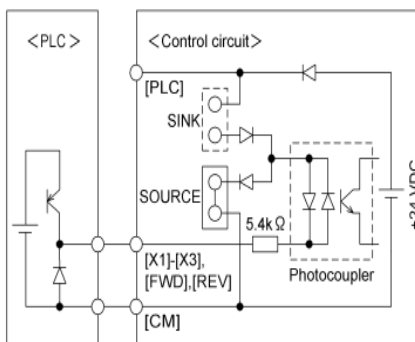
Gambar 3. 2 Ladder untuk Mengaktifkan Digital Output PLC

Tabel 3. 1 Pengalaman PLC

No.	Pengalaman Masukan	Pengalaman Keluaran
1	M0.0	Q124.1
2	M0.1	Q124.2
3	M0.2	Q124.3
4	M0.3	Q124.4



Gambar 3. 3 Konfigurasi PLC dengan Inverter dan Motor Induksi



Gambar 3. 4 Hubungan Sourcing pada Inverter

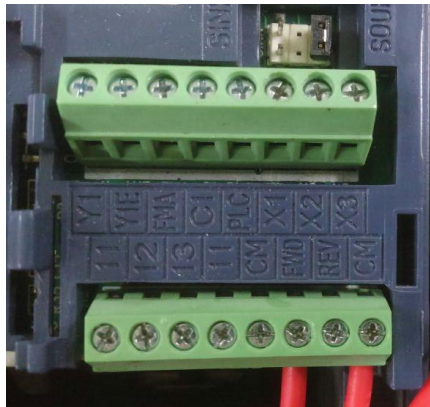
3.1.2 Konfigurasi PLC dengan Inverter dan Motor Induksi

Port digital output pada PLC terhubung dengan inverter melalui *port* X1, X2, X3 yang merupakan *port digital input* dari inverter untuk mengatur nilai frekuensi yang masuk ke motor. *Port* ini berada pada control circuit terminal dari inverter. Gambar 3. 3 menampilkan konfigurasi PLC dengan inverter dan motor. Inverter juga memiliki *port digital input* FWD yang berfungsi untuk mengatur arah putaran

dari motor seperti pada Gambar 3. 5. Pengaturan *port* untuk mengaktifkan frekuensi tertentu dilakukan berdasarkan Tabel 3. 2.

Tabel 3. 2 Pengaturan Frekuensi pada Inverter

Terminal [X3]	Terminal [X2]	Terminal [X1]	Frekuensi yang Dihasilkan (Hz)
nonaktif	nonaktif	nonaktif	0
nonaktif	nonaktif	aktif	10
nonaktif	Aktif	nonaktif	20
nonaktif	Aktif	aktif	25
aktif	Nonaktif	nonaktif	30
aktif	Nonaktif	aktif	35
aktif	Aktif	nonaktif	40
aktif	Aktif	aktif	50



Gambar 3. 5 Control Circuit Terminal pada Inverter

PLC terhubung dengan inverter secara *sourcing*, dimana PLC digunakan sebagai sumber yang memberikan tegangan ke inverter untuk mengaktifkan tiap *port* inverter. Pengaturan *sourcing* pada inverter dilakukan seperti pada Gambar 3. 4. Parameter pada inverter diatur agar dapat bekerja sesuai dengan kebutuhan. Motor yang digunakan berupa motor pompa air satu fasa dengan tegangan rating sebesar 220 volt, arus rating sebesar 1 ampere, dan keluaran motor yaitu 125 watt.

3.1.3 Konfigurasi Sensor ACS712 dengan NI MyRIO

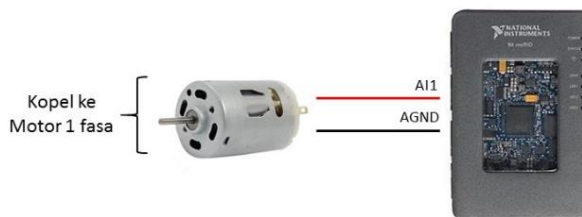
Pengukuran arus motor dilakukan menggunakan sensor ACS712 yang dipasang seri terhadap sumber tegangan dan beban. Sensor ACS712 memiliki nilai ketelitian tiap perubahan 1A adalah sebesar 0,185mV. Keluaran dari sensor ACS712 berupa tegangan yang selanjutnya digunakan sebagai masukan *port analog input* dari NI MyRIO. Sensor ACS712 memerlukan sumber tegangan 5V agar dapat beroperasi. Konfigurasi Sensor ACS712 dengan NI MyRIO dibuat seperti pada Gambar 3. 7. Untuk mengubah nilai tegangan keluaran ACS712 menjadi nilai arus, dilakukan proses linierisasi menggunakan persamaan 3.1.

$$arus = \frac{|V_{out} - 2,5|}{0,185} \quad (3.1)$$

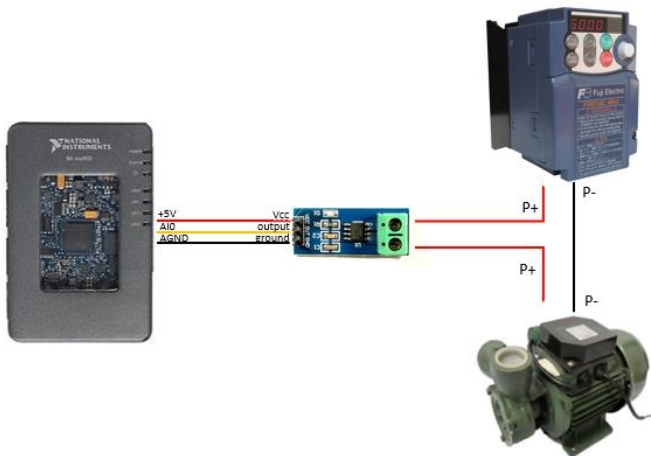
Dimana nilai 2,5 merupakan tegangan keluaran sensor ketika dalam kondisi arus yang terbaca adalah 0 atau arus minimal, sedangkan nilai 0,185 merupakan nilai perubahan tegangan keluaran ketika arus berubah 1 A.

3.1.4 Konfigurasi Tachogenerator dengan NI MyRIO

Data kecepatan motor diperoleh melalui tegangan yang dihasilkan oleh *tachogenerator* kemudian dikonversikan menjadi kecepatan dan dinyatakan dalam satuan radian per menit (rpm). Tegangan keluaran *tachogenerator* digunakan sebagai masukan analog pada NI MyRIO. Konfigurasi *tachogenerator* dengan NI MyRIO dapat dilihat pada Gambar 3. 6. Motor *tachogenerator* yang digunakan berupa motor DC 12 Volt. Motor yang digunakan sebagai *tachogenerator* haruslah memiliki nilai rpm yang lebih besar daripada motor utama. Hal ini perlu diperhatikan agar tidak terjadi pembebanan pada motor *tachogenerator*.



Gambar 3. 6 Konfigurasi Tachogenerator dengan NI MyRIO



Gambar 3. 7 Konfigurasi Sensor ACS712 dengan NI MyRIO

3.2 Pengondisian Kerusakan Motor

Motor pompa air satu fasa dikondisikan dengan kerusakan yang berbeda. Hal ini bertujuan untuk memeriksa kesesuaian hasil deteksi program dengan kondisi aktual dari motor. Kerusakan yang dikondisikan berupa kerusakan *bearing* dan kerusakan rotor yang diimplementasikan dalam bentuk pembebanan pada motor.

3.2.1 Kondisi Kerusakan *Bearing*

Terhambatnya putaran *bearing* dapat menandakan bahwa *bearing* tidak dapat bekerja dengan baik. Pengondisian kerusakan *bearing* pada motor pompa air satu fasa dilakukan dengan memberikan kondisi cacat pada bagian pemisah antara bola *bearing* dan dengan mengurangi pelumas. Cacat pada pemisah antara bola *bearing* menyebabkan putaran *bearing* menjadi lebih berat sehingga tercipta pembebanan pada motor.

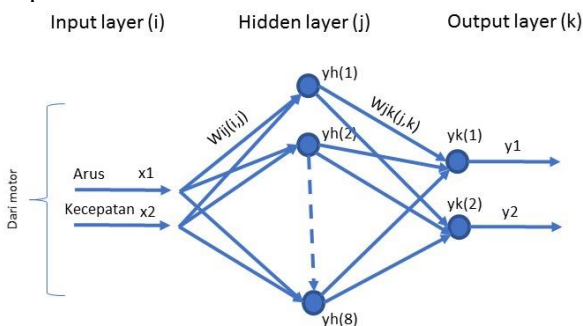
3.2.2 Kondisi Kerusakan Rotor

Tidak hanya diakibatkan oleh masa penggunaan, kesalahan saat proses manufaktur juga dapat menyebabkan kerusakan pada rotor. Efek pembebanan pada motor dapat terjadi ketika rotor tidak dapat bekerja dengan baik. Oleh karena itu, pada Tugas Akhir ini kondisi kerusakan motor digantikan dengan pembebanan mekanis. Media air

digunakan untuk mendapatkan kondisi pembebanan. Keran air digunakan untuk mengatur skala pembebanan motor. Semakin kecil bukaan keran air, maka beban yang dihasilkan akan semakin besar. Pembebanan dilakukan dengan tiga skala yaitu saat tuas keran diputar 30°, 60° dan 70°. Semakin besar sudut putar tuas keran, maka semakin besar pembebanan yang dihasilkan.

3.3 Perancangan *Neural Network*

Proses deteksi dan diagnosis dari kegagalan motor pada Tugas Akhir ini menggunakan metode *neural network*. Hal utama yang harus dilakukan ketika merancang *neural network* adalah menentukan struktur serta fungsi aktivasi yang digunakan. Jenis *network* yang digunakan yaitu *multilayer network* dengan algoritma *backpropagation*. Jumlah *neuron* pada *hidden layer* dapat diatur melalui program. Hal ini bertujuan untuk mengetahui perbedaan hasil keluran ketika menggunakan jumlah *neuron* tertentu. Jumlah *neuron* yang terdapat pada *input layer* adalah sebanyak 2 *neuron* yang merepresentasikan data arus serta kecepatan dari motor. *Output layer* terdiri dari 2 *neuron* dengan representasi seperti pada Tabel 3. 3. Fungsi aktivasi yang digunakan berupa fungsi *threshold* atau *hardlimit*. Hal ini disebabkan tujuan dari perancangan sistem adalah untuk mengklasifikasikan kondisi dari motor. struktur *neural network* dapat dilihat pada Gambar 3. 8.



Gambar 3. 8 Struktur *Neural network*

Tabel 3. 3 Representasi *Output layer*

No.	Yk1	Yk2	Keterangan
1	0	0	Kondisi motor normal

No.	Yk1	Yk2	Keterangan
2	0	1	Kondisi motor dengan beban
4	1	1	Kondisi kerusakan <i>bearing</i>

BAB IV

PENGUJIAN DAN ANALISA DATA

Pengujian dan analisis dilakukan untuk mengetahui keandalan dari sistem yang telah dirancang. Sebelum proses pengujian, dilakukan pengambilan data untuk mendapatkan parameter motor yang akan diolah. *Training neural network* secara *offline* dilakukan untuk mendapatkan nilai revisi bobot yang akan digunakan untuk proses deteksi kegagalan motor.

4.1 Pengambilan Data Kondisi Motor

Terdapat tiga jenis kondisi motor yang akan dianalisa, yaitu ketika kondisi motor tanpa beban, motor berbeban, dan motor dengan kerusakan pada bagian *bearing*. Data tersebut selanjutnya akan digunakan sebagai parameter sistem untuk mendeteksi kondisi dari motor. Jenis data yang diambil yaitu berupa arus masukan dan kecepatan dari motor.

4.1.1 Pengambilan Data Motor Tanpa Beban

Proses pengambilan data dilakukan dengan memberi masukan frekuensi yang berbeda, yaitu 10 Hz, 20 Hz, 25 Hz, 30 Hz, 35 Hz, 40 Hz, dan 50 Hz. Pembatasan frekuensi maksimal ditentukan berdasarkan spesifikasi dari motor. Pengambilan data bertujuan untuk melihat dampak dari perubahan frekuensi masukan terhadap arus dan kecepatan dari motor serta mendapatkan parameter saat motor bekerja dalam kondisi tanpa beban. Data arus diperoleh melalui pembacaan sensor ACS712. Keluaran dari sensor ini berupa tegangan DC. Oleh karena itu, proses linearisasi perlu dilakukan untuk mendapatkan nilai arus. Hasil pengambilan data dapat dilihat pada Tabel 4. 1.

Pengambilan data kembali dilakukan dengan frekuensi masukan 50 Hz sebanyak 30 set data. Hal ini bertujuan untuk mendapatkan nilai parameter untuk proses *training*. Waktu sampling yang digunakan selama pengambilan data adalah 5 ms selama 20 detik.

Tabel 4. 1 Data Motor Tanpa Beban

No	Frekuensi Masukan (Hz)	Kecepatan Motor (RPM)	Tegangan Tachogenerator (V)	Arus Motor (A)
1	0	0	0	0

No	Frekuensi Masukan (Hz)	Kecepatan Motor (RPM)	Tegangan Tachogenerator (V)	Arus Motor (A)
2	10	0	0	0.3902
3	20	0	0	0.5666
4	25	0	0	0.6536
5	30	1476.5	0.58284	0.6648
6	35	1857.6	0.88168	0.6604
7	40	2172.2	1.03058	0.686
8	50	2779.8	1.34912	0.739

Tabel 4. 2 Data Motor dengan Beban

	Skala Buka-an Keran Air	Arus Keluaran Inverter (A)	Tegangan RMS Keluaran Sensor ACS712 (V)	Tegangan RMS Tachogenerator (V)	Kecepatan Motor (RPM)
Pengambilan Data 1	1	0.695	2.563	3.0288	1921
	2	0.74	2.5697	2.7143	1847
	3	0.811	2.5177	2.5703	1743
Pengambilan Data 2	1	0.691	2.572	2.9967	1936
	2	0.755	2.5571	2.6829	1829
	3	0.709	2.4867	2.6495	1793
Pengambilan Data 3	1	0.685	2.5535	2.9878	1912
	2	0.728	2.4952	2.6944	1786
	3	0.774	2.5044	2.6449	1749
Pengambilan Data 4	1	0.703	2.543	2.9777	1924
	2	0.758	2.5267	2.7493	1803
	3	0.829	2.5573	2.6937	1686
Pengambilan Data 5	1	0.684	2.5583	2.9139	1922
	2	0.74	2.5421	2.7883	1787
	3	0.816	2.5728	2.6735	1672

Tabel 4. 3 Data Kerusakan *Bearing*

No	Frekuensi Masukan (Hz)	Kecepatan Motor (RPM)	Tegangan Tachogenerator (V)	Arus Motor (A)
1	0	0	0	0
2	10	0	0	0.4494
3	20	0	0	0.6606
4	25	0	0	0.7644
5	30	0	0	0.863

No	Frekuensi Masukan (Hz)	Kecepatan Motor (RPM)	Tegangan Tachogenerator (V)	Arus Motor (A)
6	35	0	0	0.9498
7	40	2211.8	2.33994	0.7838
8	50	2855.6	3.03932	0.8186

4.1.2 Pengambilan Data Motor dengan Beban

Pembebanan pada motor dilakukan menggunakan beban mekanis yang berupa aliran air. Hal ini dilakukan dengan pertimbangan motor yang digunakan berupa motor pompa air satu fasa. Terdapat tiga skala pembebanan motor, dimana dua diantaranya digunakan untuk merepresentasikan kondisi kerusakan dari motor. Pengambilan data dilakukan dengan memberikan frekuensi 50 Hz pada motor, kemudian mengubah skala keran air untuk pembebanan. Hasil pengambilan data dapat dilihat pada Tabel 4. 2. Pengambilan data kembali dilakukan sebanyak 30 set data untuk proses *training* pada *neural network*.

4.1.3 Pengambilan Data Motor dengan Kerusakan *Bearing*

Kondisi kerusakan *bearing* yang diberikan berupa cacat pada bagian pemisah antara bola *bearing* dan dengan mengurangi pelumas. Kerusakan pada *bearing* dapat mengakibatkan hambatan pada putaran motor sehingga dapat menimbulkan pembebanan. Proses pengambilan data dilakukan dengan memberi masukan frekuensi yang berbeda, yaitu 10 Hz, 20 Hz, 25 Hz, 30 Hz, 35 Hz, 40 Hz, dan 50 Hz, kemudian dilakukan pengukuran pada parameter. Hasil pengambilan data dapat dilihat pada

Tabel 4. 3. Pengambilan data kembali dilakukan sebanyak 30 set data untuk proses *training* pada *neural network*.

4.2 Training Data

Struktur *neural network* telah dirancang seperti pada Gambar 3. 8. Nilai parameter *input layer* diperoleh melalui proses pengambilan data sebelumnya. *Training* data dilakukan berdasarkan kondisi dari motor, yaitu motor tanpa beban, kondisi motor dengan beban, dan kondisi motor dengan kerusakan *bearing*. *Output layer* terdiri dari 2 *neuron* yang merepresentasikan kondisi motor.

Proses *training* dilakukan menggunakan tiga jenis kondisi motor yang diambil secara acak. Masing-masing kondisi terdiri dari 30 data arus dan 30 data tegangan keluaran dari tachogenerator. Proses *training* melewati lima tahapan, yang pertama yaitu inialisasi. Pada

proses ini ditentukan nilai bobot awal secara acak. Bobot ini yang akan digunakan pada proses *training*. Tahap inisialisasi juga dilakukan dengan menentukan jumlah *layer* dan jumlah *neuron* per *layer* yang akan digunakan. Pada Tugas Akhir ini struktur *neural network* yang dicoba adalah dengan jumlah *neuron* pada *input layer*, *hidden layer*, dan *output layer* berturut-turut 2-4-2; 2-10-2; 2-14-2; dan 2-20-2. Nilai *learning rate* juga diubah berturut-turut 0,1; 0,01; dan 0,001. Hal ini bertujuan untuk mengetahui dampak jumlah *hidden neuron layer* dan perbedaan *learning rate* terhadap nilai kesalahan pada proses *training*.

Tahapan selanjutnya yaitu presentasi. Tahap presentasi dilakukan dengan melakukan penghitungan dari *input layer*, yaitu mengalikan nilai masukan dengan nilai bobot acak yang telah ditentukan hingga diperoleh nilai *output*. Nilai keluaran dari tahap ini disebut dengan nilai aktual. Tahapan selanjutnya dilanjutkan dengan tahap penghitungan kesalahan. Pada struktur yang telah dijelaskan pada Bab 3, terdapat nilai target yang berfungsi untuk pembandingan terhadap nilai keluaran aktual. Selisih dari nilai target dikurangi nilai aktual disebut sebagai nilai kesalahan. Ketika terdapat nilai kesalahan, pada tahap selanjutnya dilakukan penghitungan balik pada *network*. Penghitungan balik berfungsi untuk merevisi nilai bobot acak awal hingga nilai kesalahan menjadi minimum. Nilai bobot yang direvisi berupa nilai bobot antara *input layer* dengan *hidden layer* dan antara *hidden layer* dengan *output layer*.

Tahap akhir merupakan tahap perulangan dari tahap presentasi hingga penghitungan balik yang bertujuan untuk mendapatkan nilai kesalahan yang minimal. Pada proses *training* data ini, iterasi yang digunakan adalah sebanyak 500 kali. Perubahan nilai kesalahan dan perubahan sampel bobot di plot dalam bentuk grafik untuk mengetahui perubahan nilai tersebut pada setiap *epoch*.

4.2.1 Training Data dengan Learning rate 0,1

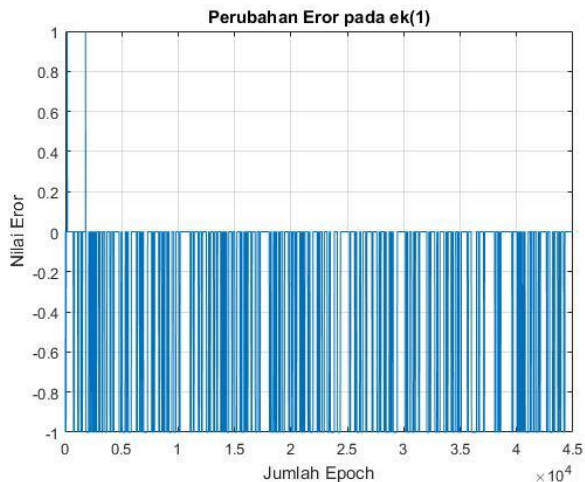
Proses *training* data ketika menggunakan *learning rate* sebesar 0,1 dilakukan dengan empat jumlah *hidden neuron* yang berbeda yaitu sebanyak 4, 10, 14, dan 20 *hidden neuron*. Hasil yang dicatat berupa waktu yang dibutuhkan untuk melakukan proses *training* terhadap data yang dapat dilihat pada Tabel 4. 4.

Berdasarkan Tabel 4. 4, jumlah *neuron* pada *hidden layer* berpengaruh terhadap waktu yang dibutuhkan untuk melakukan

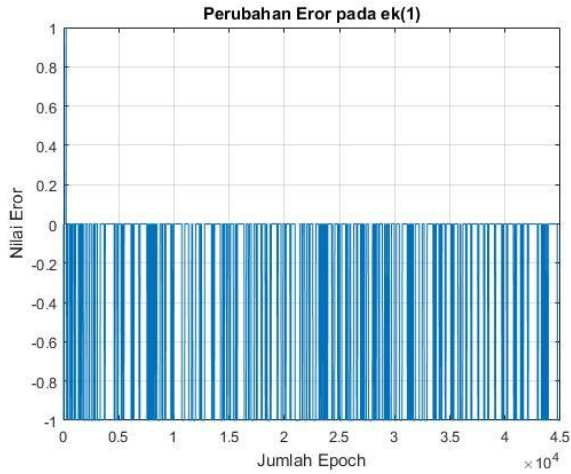
proses *training*. Semakin sedikit jumlah *neuron* pada *hidden layer*, maka waktu yang dibutuhkan untuk proses *training* akan semakin cepat. Sedangkan semakin banyak jumlah *neuron* pada *hidden layer*, maka waktu yang dibutuhkan untuk proses *training* akan semakin lama.

Tabel 4. 4 Catatan Waktu *Training Data* dengan *Learning rate* 0,1

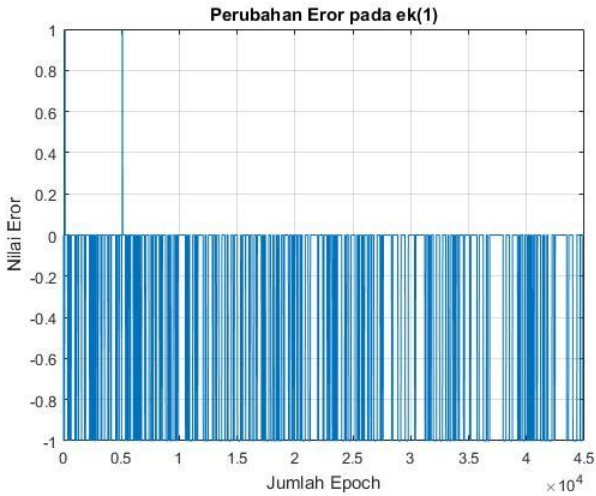
No.	Jumlah <i>Hidden neuron</i>	Percobaan ke-	Waktu yang Dibutuhkan
1	4	1	72.651 detik
		2	68.322 detik
		3	47.283 detik
2	10	1	90.600 detik
		2	89.505 detik
		3	89.505 detik
3	14	1	124.681 detik
		2	97.748 detik
		3	130.908 detik
4	20	1	155.919 detik
		2	154.643 detik
		3	145.927 detik



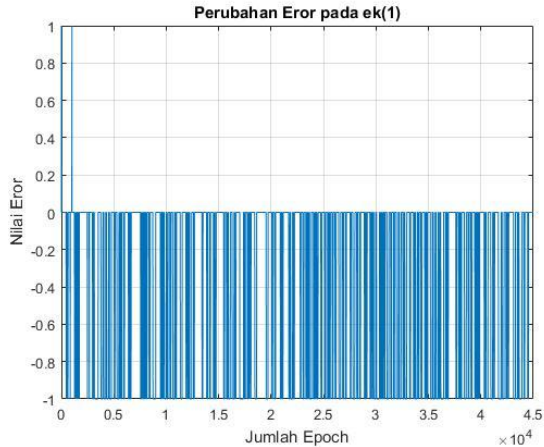
Gambar 4. 1 Perubahan Kesalahan pada $ek(1)$ dengan 4 *Hidden Neuron*



Gambar 4. 2 Perubahan Kesalahan pada ek(1) dengan 10 *Hidden Neuron*

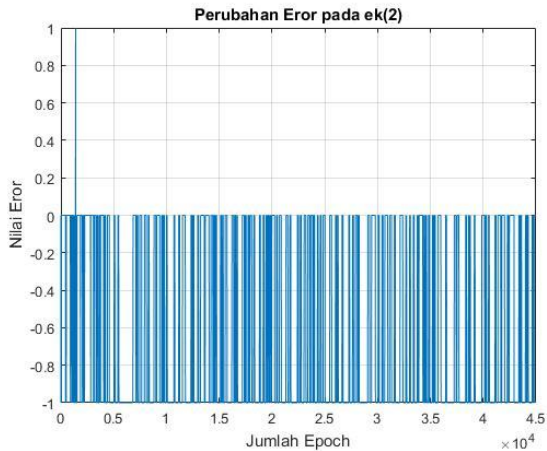


Gambar 4. 3 Perubahan Kesalahan pada ek(1) dengan 14 *Hidden Neuron*

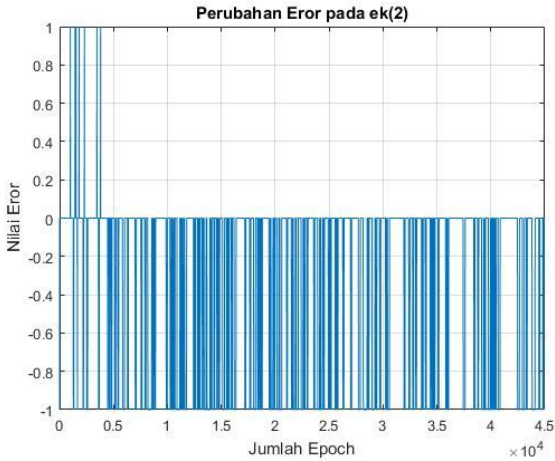


Gambar 4. 4 Perubahan Kesalahan pada ek(1) dengan 20 *Hidden Neuron*

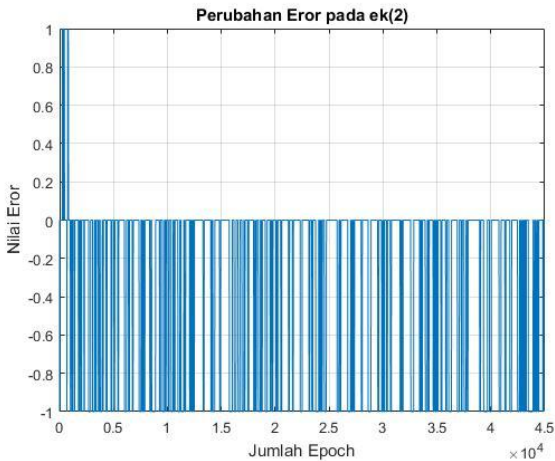
Gambar 4. 1, Gambar 4. 2, Gambar 4. 3, dan Gambar 4. 4 menunjukkan grafik perubahan kesalahan pada ek(1) ketika menggunakan *learning rate* sebesar 0,1. Berdasarkan keempat gambar tersebut, perubahan nilai kesalahan ketika proses *training* masih terjadi hingga iterasi terakhir. Hal ini berpengaruh pada proses revisi dari nilai bobot pada masing-masing *network*.



Gambar 4. 5 Perubahan Kesalahan pada ek(2) dengan 4 *Hidden Neuron*



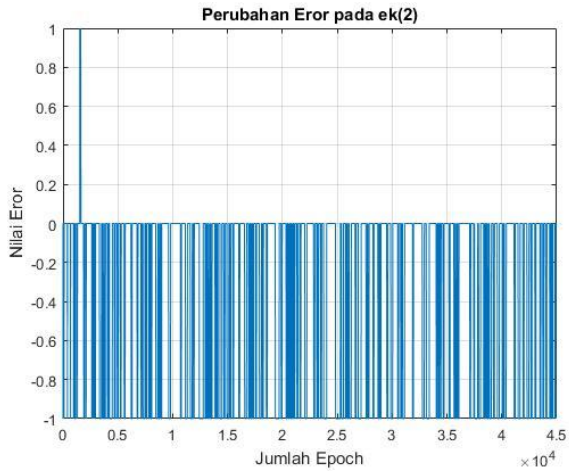
Gambar 4. 6 Perubahan Kesalahan pada ek(2) dengan 10 *Hidden Neuron*



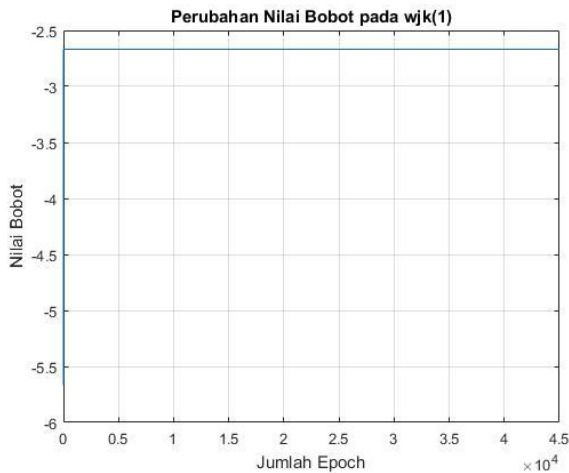
Gambar 4. 7 Perubahan Kesalahan pada ek(2) dengan 14 *Hidden Neuron*

Grafik perubahan kesalahan pada ek(2) ketika menggunakan *learning rate* sebesar 0,1 dapat dilihat pada Gambar 4. 5, Gambar 4. 6, Gambar 4. 7, dan Gambar 4. 8. Yang membedakan dari masing-masing gambar adalah jumlah *hidden neuron* yang digunakan. Berdasarkan keempat gambar tersebut, perubahan kesalahan masih

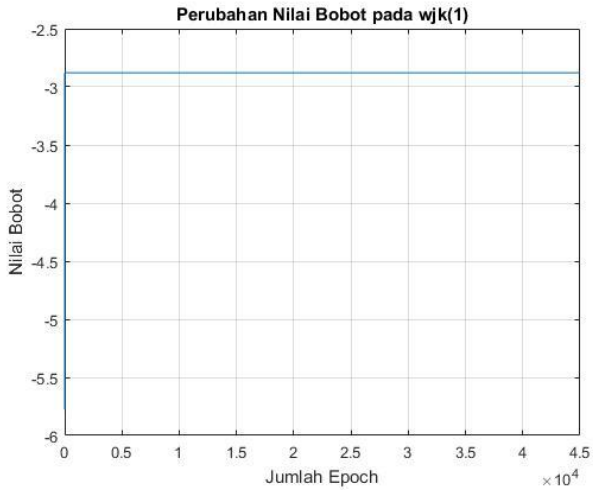
terjadi hingga iterasi berakhir. Sama seperti perubahan kesalahan pada ek(1), hal ini berpengaruh pula pada proses revisi bobot dari *network*. Sampel perubahan bobot dapat dilihat pada Gambar 4. 9, Gambar 4. 10, Gambar 4. 11, dan Gambar 4. 12.



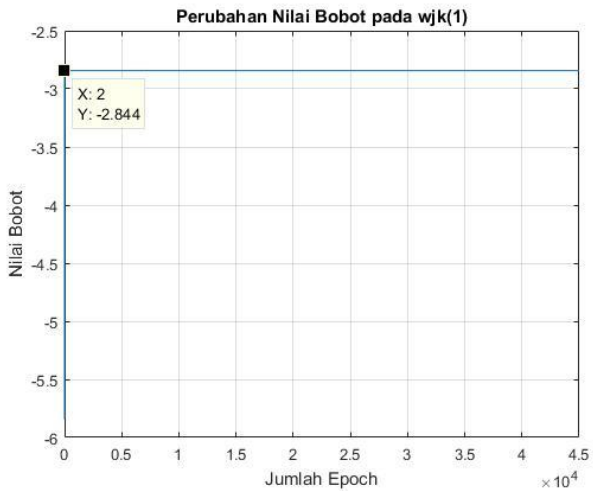
Gambar 4. 8 Grafik Perubahan Kesalahan pada ek(2) dengan 20 *Hidden Neuron*



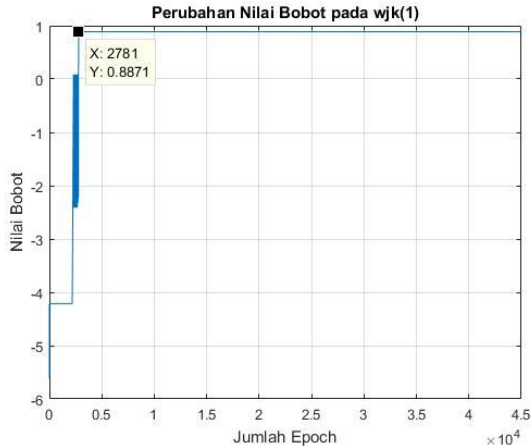
Gambar 4. 9 Perubahan Nilai Bobot pada wjk(1) dengan 4 *Hidden Neuron*



Gambar 4. 10 Perubahan Nilai Bobot pada wjk(1) dengan 10 *Hidden Neuron*



Gambar 4. 11 Perubahan Nilai Bobot pada wjk(1) dengan 14 *Hidden Neuron*



Gambar 4. 12 Perubahan Nilai Bobot pada wjk(1) dengan 20 *Hidden Neuron*

Berdasarkan keempat gambar grafik perubahan nilai bobot pada wjk(1), masing-masing memiliki nilai perubahan yang berbeda. Perubahan nilai bobot wjk(1) ketika menggunakan 4, 10, dan 14 *hidden neuron* mengalami perubahan sampai data mencapai *epoch* ke 2. Hal ini berbeda dengan perubahan nilai bobot wjk(1) ketika menggunakan 20 *hidden neuron* yang terus mengalami perubahan nilai hingga mencapai *epoch* ke 2781.

4.2.2 *Training Data dengan Learning rate 0,01*

Proses *training* data ketika menggunakan *learning rate* sebesar 0,1 dilakukan dengan empat jumlah *hidden neuron* yang berbeda yaitu sebanyak 4, 10, 14, dan 20 *hidden neuron*. Hasil yang dicatat berupa waktu yang dibutuhkan untuk melakukan proses *training* terhadap data yang dapat dilihat pada Tabel 4. 5.

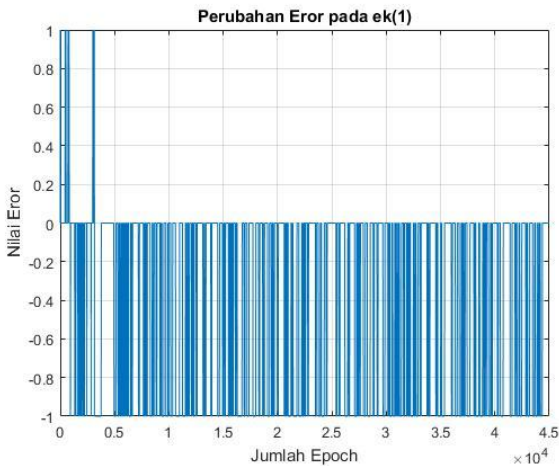
Tabel 4. 5 Catatan Waktu *Training* Data dengan *Learning rate* 0,01

No.	Jumlah <i>Hidden neuron</i>	Percobaan ke-	Waktu yang Dibutuhkan
1	4	1	27.868 detik
		2	26.98 detik
		3	38.059 detik
2	10	1	39.310 detik
		2	43.926 detik
		3	45.563 detik

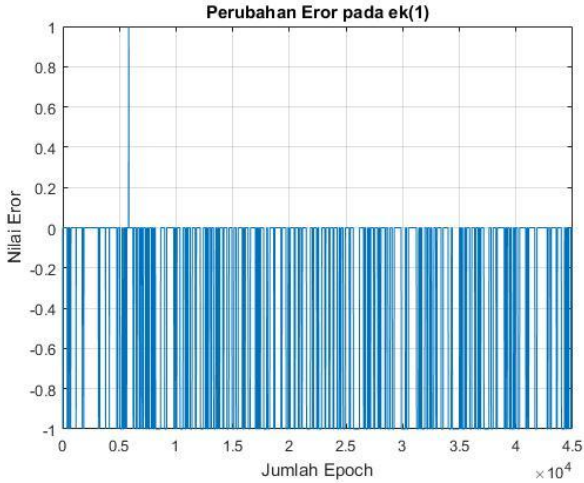
No.	Jumlah <i>Hidden neuron</i>	Percobaan ke-	Waktu yang Dibutuhkan
3	14	1	55.176 detik
		2	69.902 detik
		3	60.501 detik
4	20	1	65.272 detik
		2	86.391 detik
		3	59.178 detik

Berdasarkan Tabel 4. 5, jumlah *neuron* pada *hidden layer* berpengaruh terhadap waktu yang dibutuhkan untuk melakukan proses *training*. Semakin sedikit jumlah *neuron* pada *hidden layer*, maka waktu yang dibutuhkan untuk proses *training* akan semakin cepat. Sedangkan semakin banyak jumlah *neuron* pada *hidden layer*, maka waktu yang dibutuhkan untuk proses *training* akan semakin lama.

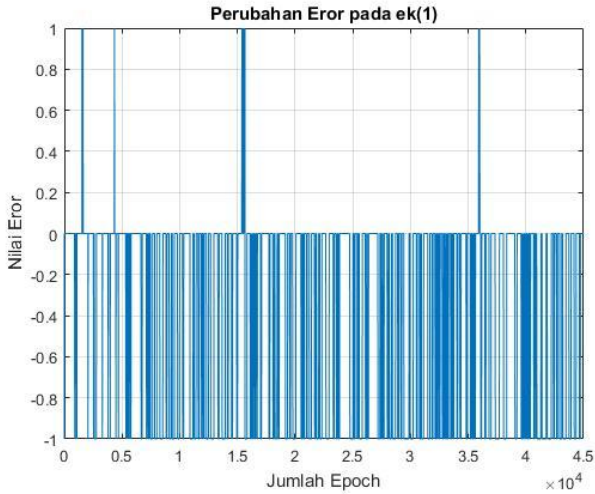
Gambar 4. 13, Gambar 4. 14, Gambar 4. 15, dan Gambar 4. 16 menunjukkan grafik perubahan kesalahan pada ek(1) ketika menggunakan *learning rate* sebesar 0,01. Berdasarkan keempat gambar tersebut, perubahan nilai kesalahan ketika proses *training* masih terjadi hingga iterasi terakhir. Hal ini berpengaruh pada proses revisi dari nilai bobot pada masing-masing *network*.



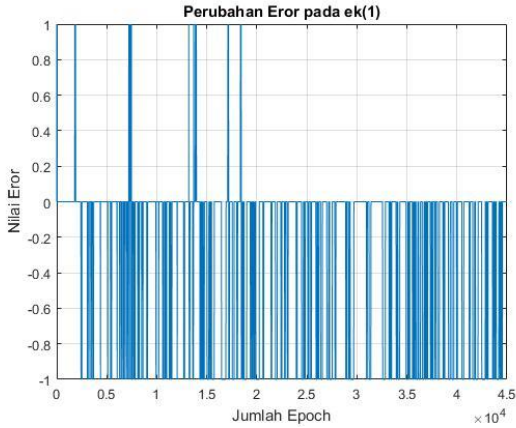
Gambar 4. 13 Perubahan Kesalahan pada ek(1) dengan 4 *Hidden Neuron*



Gambar 4. 14 Perubahan Kesalahan pada ek(1) dengan 10 *Hidden Neuron*

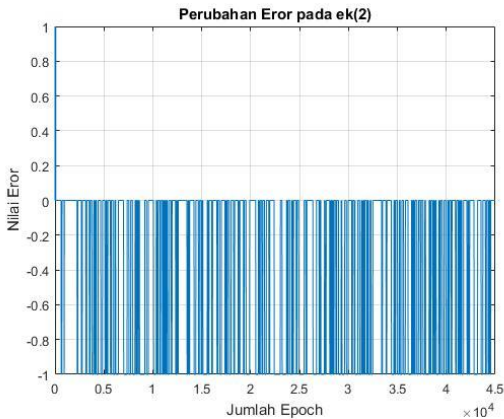


Gambar 4. 15 Perubahan Kesalahan pada ek(1) dengan 14 *Hidden Neuron*

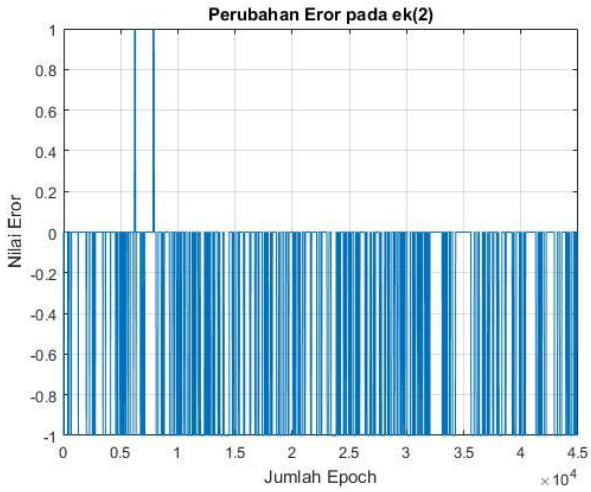


Gambar 4. 16 Perubahan Kesalahan pada ek(1) dengan 20 *Hidden Neuron*

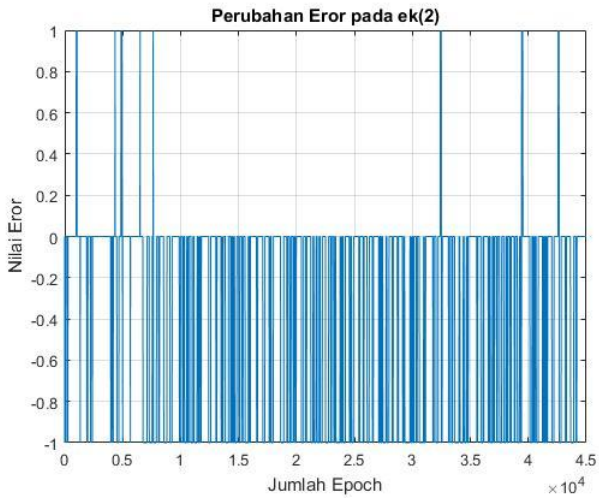
Grafik perubahan kesalahan pada ek(2) ketika menggunakan *learning rate* sebesar 0,1 dapat dilihat pada Gambar 4. 17, Gambar 4. 18, Gambar 4. 19, dan Gambar 4. 20. Hal yang membedakan dari masing-masing gambar adalah jumlah *hidden neuron* yang digunakan. Berdasarkan keempat gambar tersebut, perubahan kesalahan masih terjadi hingga iterasi berakhir. Sama seperti perubahan kesalahan pada ek(1), hal ini berpengaruh pula pada proses revisi bobot dari *network*.



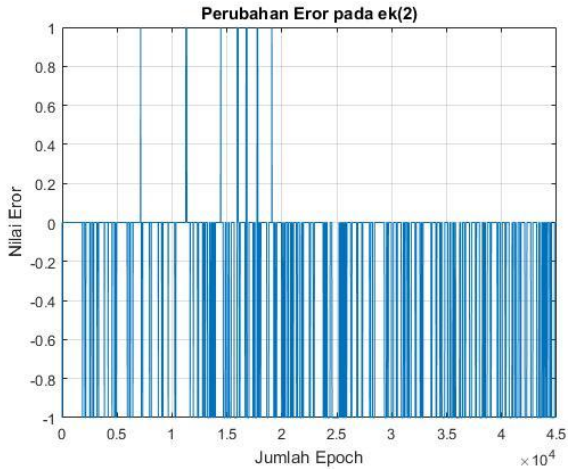
Gambar 4. 17 Perubahan Kesalahan pada ek(2) dengan 4 *Hidden Neuron*



Gambar 4. 18 Perubahan Kesalahan pada ek(2) dengan 10 *Hidden Neuron*

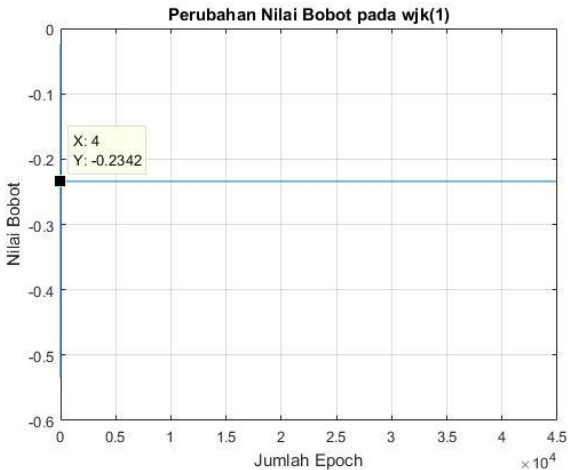


Gambar 4. 19 Perubahan Kesalahan pada ek(2) dengan 14 *Hidden Neuron*

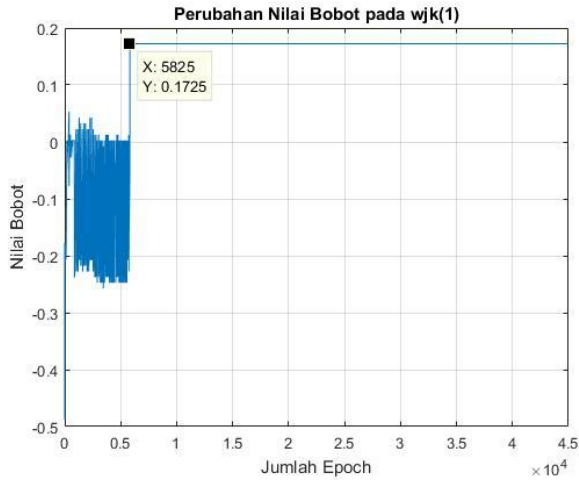


Gambar 4. 20 Perubahan Kesalahan pada ek(2) dengan 20 *Hidden Neuron*

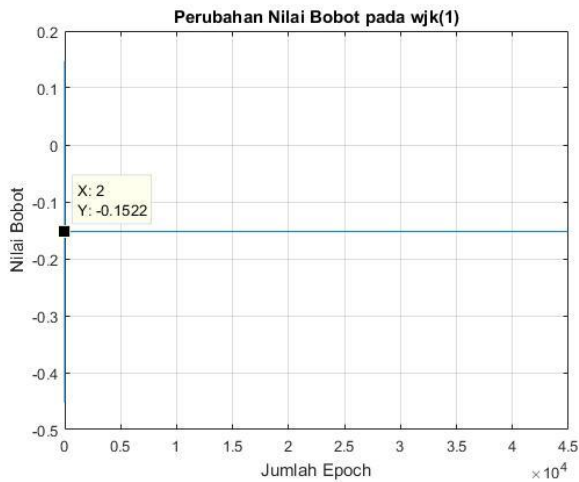
Sampel perubahan bobot dapat dilihat pada Gambar 4. 21, Gambar 4. 22, Gambar 4. 23, dan Gambar 4. 24. Masing-masing kondisi memiliki perubahan nilai bobot yang berbeda-beda. Ketika menggunakan 4 *hidden neuron*, nilai bobot berhenti mengalami perubahan pada *epoch* ke 4.



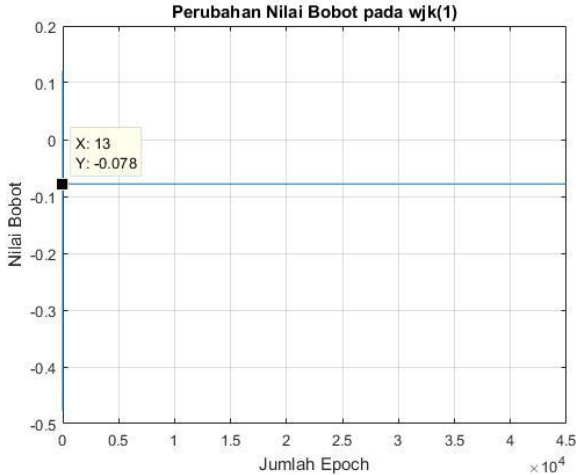
Gambar 4. 21 Perubahan Nilai Bobot pada wjk(1) dengan 4 *Hidden Neuron*



Gambar 4. 22 Perubahan Nilai Bobot pada wjk(1) dengan 10 *Hidden Neuron*



Gambar 4. 23 Perubahan Nilai Bobot pada wjk(1) dengan 14 *Hidden Neuron*



Gambar 4. 24 Perubahan Nilai Bobot pada wjk(1) dengan 20 *Hidden Neuron*

Hasil berbeda diperoleh ketika menggunakan 10 *hidden neuron*. Perubahan bobot masih terus terjadi hingga *epoch* ke 5825. Begitu juga ketika menggunakan 14 dan 20 *hidden neuron*. Perubahan bobot terjadi hingga *epoch* kedua pada 14 *hidden neuron* dan *epoch* ke 13 pada 20 *hidden neuron*. Perbedaan ini dapat dipengaruhi oleh nilai bobot acak awal yang berbeda pada tiap kondisi.

4.2.3 *Training Data dengan Learning rate 0,001*

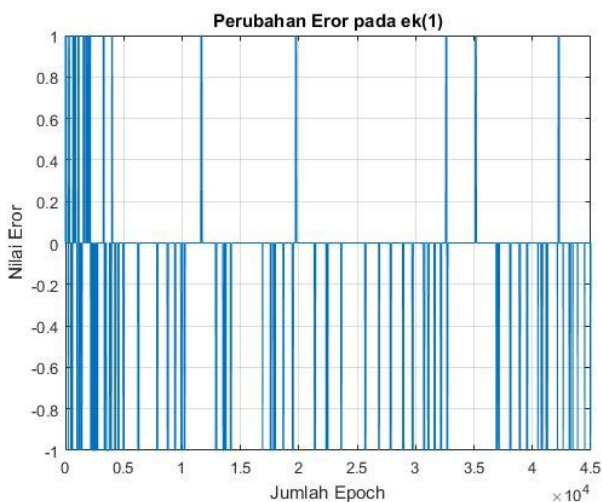
Proses *training* data ketika menggunakan *learning rate* sebesar 0,1 dilakukan dengan empat jumlah *hidden neuron* yang berbeda yaitu sebanyak 4, 10, 14, dan 20 *hidden neuron*. Hasil yang dicatat berupa waktu yang dibutuhkan untuk melakukan proses *training* terhadap data yang dapat dilihat pada Tabel 4. 6.

Tabel 4. 6 Catatan Waktu *Training Data* dengan *Learning rate* 0,001

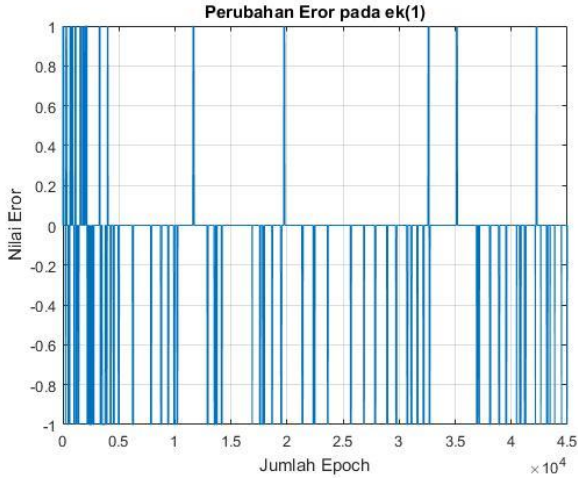
No.	Jumlah <i>Hidden neuron</i>	Percobaan ke-	Waktu yang Dibutuhkan
1	4	1	37.170 detik
		2	31.903 detik
		3	24.793 detik
2	10	1	35.919 detik
		2	31.578 detik
		3	30.559 detik

No.	Jumlah <i>Hidden neuron</i>	Percobaan ke-	Waktu yang Dibutuhkan
3	14	1	48.173 detik
		2	57.021 detik
		3	44.503 detik
4	20	1	119.757 detik
		2	72.355 detik
		3	73.051 detik

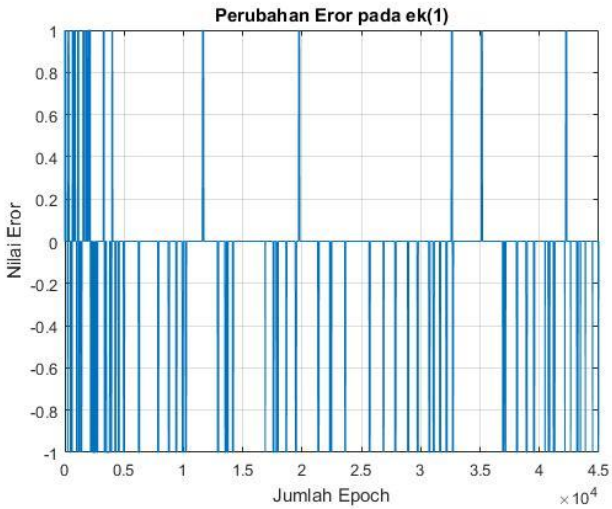
Berdasarkan Tabel 4. 6, jumlah *neuron* pada *hidden layer* berpengaruh terhadap waktu yang dibutuhkan untuk melakukan proses *training*. Semakin sedikit jumlah *neuron* pada *hidden layer*, maka waktu yang dibutuhkan untuk proses *training* akan semakin cepat. Sedangkan semakin banyak jumlah *neuron* pada *hidden layer*, maka waktu yang dibutuhkan untuk proses *training* akan semakin lama.



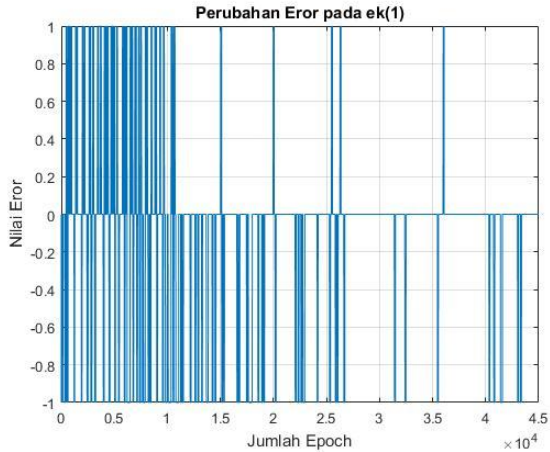
Gambar 4. 25 Perubahan Kesalahan pada ek(1) dengan 4 *Hidden Neuron*



Gambar 4. 26 Perubahan Kesalahan pada ek(1) dengan 10 *Hidden Neuron*

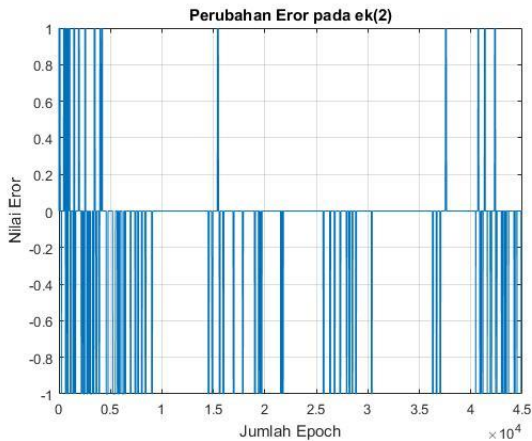


Gambar 4. 27 Perubahan Kesalahan pada ek(1) dengan 14 *Hidden Neuron*

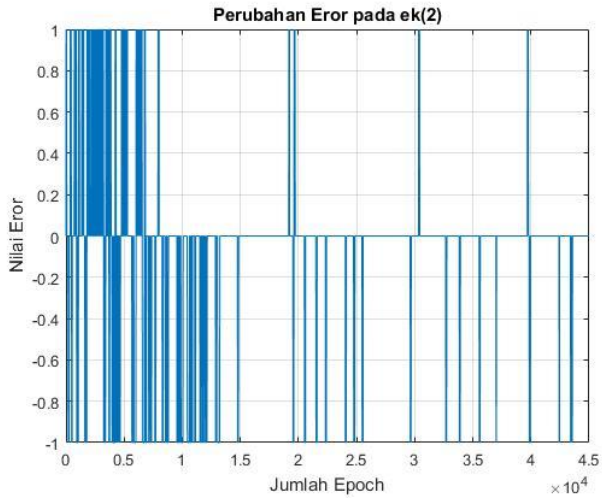


Gambar 4. 28 Perubahan Kesalahan pada ek(1) dengan 20 *Hidden Neuron*

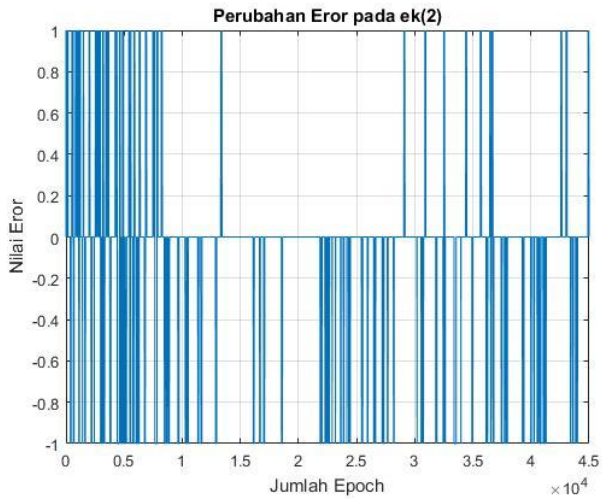
Gambar 4. 25, Gambar 4. 26, Gambar 4. 27, dan Gambar 4. 28 menunjukkan grafik perubahan kesalahan pada ek(1) ketika menggunakan *learning rate* sebesar 0,001. Perubahan kesalahan masih terjadi pada proses *training*, namun tidak sebanyak ketika menggunakan *learning rate* dengan nilai yang lebih besar dari 0,001. Hal ini dapat mempengaruhi nilai revisi dari bobot ketika proses *training* berlangsung.



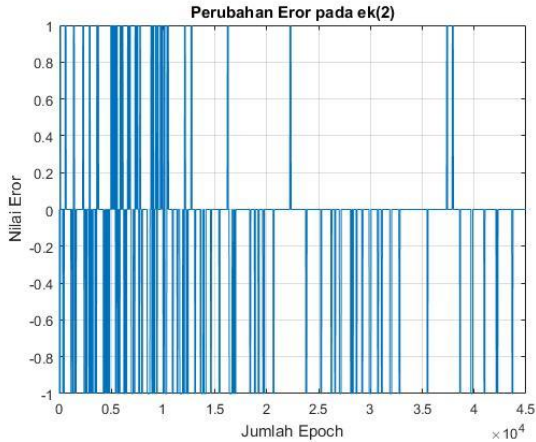
Gambar 4. 29 Perubahan Kesalahan pada ek(2) dengan 4 *Hidden Neuron*



Gambar 4. 30 Perubahan Kesalahan pada ek(2) dengan 10 *Hidden Neuron*

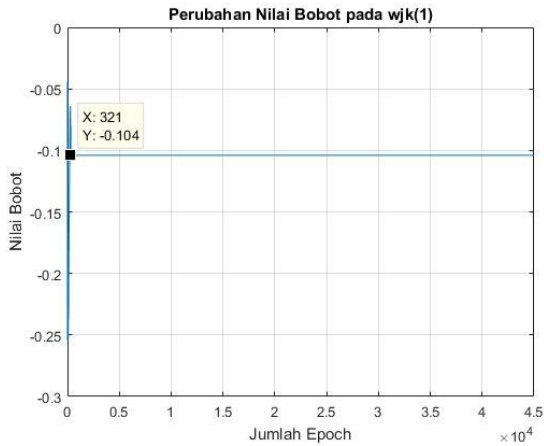


Gambar 4. 31 Perubahan Kesalahan pada ek(2) dengan 14 *Hidden Neuron*

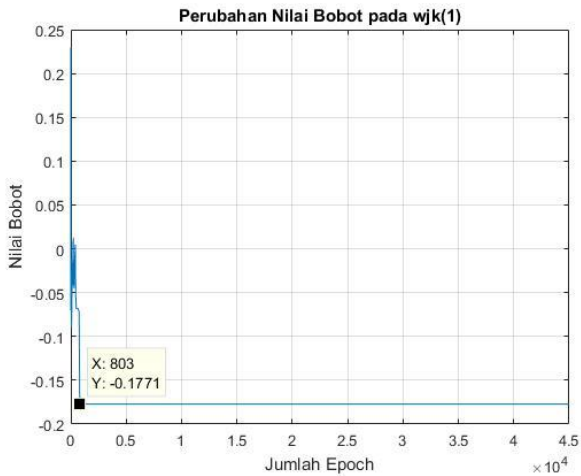


Gambar 4. 32 Perubahan Kesalahan pada ek(2) dengan 20 *Hidden Neuron*

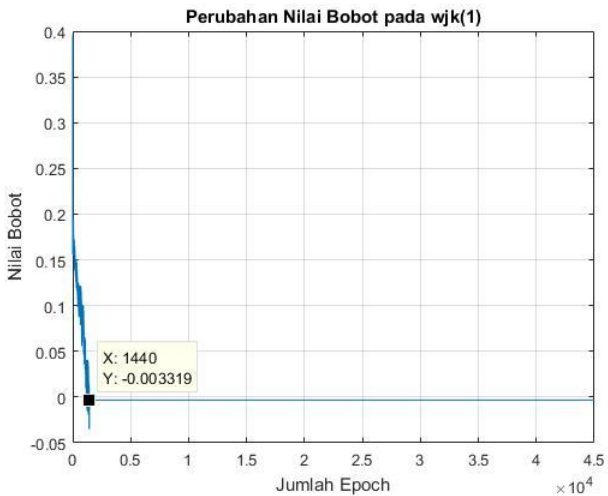
Grafik perubahan kesalahan pada ek(2) dapat dilihat pada Gambar 4. 29, Gambar 4. 30, Gambar 4. 31, dan Gambar 4. 32. Perubahan kesalahan masih terjadi pada tiap grafik, namun sama seperti pada ek(1), perubahan kesalahan yang terjadi tidak sebanyak ketika menggunakan *learning rate* dengan nilai yang lebih besar. Perubahan kesalahan berhubungan dengan revisi bobot yang terjadi ketika proses *training* berlangsung.



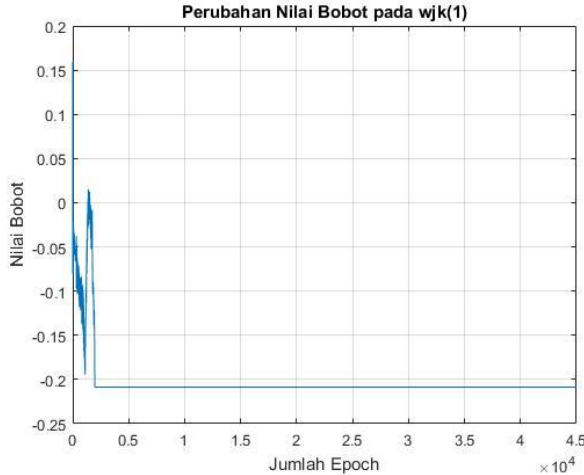
Gambar 4. 33 Perubahan Nilai Bobot pada wjk(1) dengan 4 *Hidden Neuron*



Gambar 4. 34 Perubahan Nilai Bobot pada wjk(1) dengan 10 *Hidden Neuron*



Gambar 4. 35 Perubahan Nilai Bobot pada wjk(1) dengan 14 *Hidden Neuron*



Gambar 4. 36 Perubahan Nilai Bobot pada wjk(1) dengan 20 *Hidden Neuron*

Contoh perubahan nilai bobot pada wjk(1) dapat dilihat pada Gambar 4. 33, Gambar 4. 34, Gambar 4. 35, dan Gambar 4. 36. Hal yang membedakan diantara keempat grafik adalah penggunaan jumlah *hidden neuron*-nya. Masing-masing kondisi memiliki perubahan nilai bobot yang berbeda. Nilai bobot wjk(1) dengan 4 *hidden neuron* terus mengalami perubahan hingga *epoch* yang ke 321. Berbeda dengan kondisi pertama, ketika menggunakan 10 *hidden neuron*, nilai bobot terus berubah hingga *epoch* yang ke 803. Sedangkan ketika menggunakan 14 *hidden neuron*, nilai bobot terus berubah hingga mencapai *epoch* yang ke 1440. Perbedaan ini dapat dipengaruhi oleh nilai bobot acak awal yang berbeda tiap kondisi.

4.3 Pengujian Klasifikasi

Pengujian dilakukan untuk mengetahui kemampuan sistem ketika melakukan klasifikasi terhadap kondisi motor. Proses ini dilakukan menggunakan struktur *forward* dari *neural network*. Parameter yang digunakan berupa arus dan kecepatan dari motor. bobot yang digunakan merupakan bobot terakhir (bobot terbaik) yang diperoleh melalui proses *training*. Pengujian dilakukan dengan data yang diurutkan tiap kondisi. Total data yang akan diklasifikasikan adalah sebanyak 90 data, dimana 30 data merupakan kondisi motor tanpa beban, 30 data untuk kondisi motor dengan beban, dan 30 data untuk

kondisi kerusakan bearing. Pengujian juga mengacu pada proses *training*, yaitu pengujian ketika menggunakan *learning rate* sebesar 0,1; 0,01; dan 0,001.

Tabel 4. 7 Nilai MSE dengan *Learning rate* 0,1

No.	Jumlah <i>Hidden neuron</i>	Pengujian ke-	Nilai MSE pada ek(1)	Nilai MSE pada ek(2)
1	4	1	0.3333333	0.3333333
		2	0.6333333	0.3333333
		3	0.3444444	0.3333333
2	10	1	0.3333333	0.3333333
		2	0.3333333	0.3333333
		3	0.3555556	0.3333333
3	14	1	0.3333333	0.3333333
		2	0.3333333	0.3333333
		3	0.3444444	0.3333333
4	20	1	0.3333333	0.3333333
		2	0.3333333	0.3333333
		3	0.3555556	0.3333333

Tabel 4. 7 menunjukkan nilai MSE (*Mean Square Error*) dengan *learning rate* sebesar 0,1. Kesalahan rata-rata yang diperoleh ketika pengujian terhadap *learning rate* 0,1 adalah sebesar 0,3638 untuk ek(1) dan 0,3333 untuk ek(2). Kesalahan yang terjadi masih cukup besar dan jumlah *neuron* pada *hidden layer* tidak berpengaruh pada nilai MSE.

Tabel 4. 8 Nilai MSE dengan *Learning rate* 0,01

No.	Jumlah <i>Hidden neuron</i>	Pengujian ke-	Nilai MSE pada ek(1)	Nilai MSE pada ek(2)
1	4	1	0.4222222	0.3333333
		2	0.3333333	0.3333333
		3	0.3777778	0.3333333
2	10	1	0.3666667	0.3333333
		2	0.3888889	0.3333333
		3	0.4	0.3222222
3	14	1	0.4111111	0.3222222
		2	0.4222222	0.3333333
		3	0.3777778	0.3333333

No.	Jumlah <i>Hidden neuron</i>	Pengujian ke-	Nilai MSE pada ek(1)	Nilai MSE pada ek(2)
4	20	1	0.3555556	0.3333333
		2	0.3444444	0.3333333
		3	0.3777778	0.3333333

Pengujian juga dilakukan pada hasil proses *training* dengan *learning rate* 0,001. Tabel 4. 8 menunjukkan nilai MSE (*Mean Square Error*) dengan *learning rate* sebesar 0,01. Kesalahan rata-rata yang diperoleh ketika pengujian terhadap *learning rate* 0,01 adalah sebesar 0,3814 untuk ek(1) dan 0,3314 untuk ek(2). Kesalahan yang terjadi masih cukup besar dan jumlah *neuron* pada *hidden layer* tidak berpengaruh pada nilai MSE.

Pengujian selanjutnya dilakukan dengan hasil *training* menggunakan *learning rate* sebesar 0,001. Tabel 4. 9 menunjukkan nilai MSE dengan *learning rate* sebesar 0,001. Kesalahan rata-rata dari ek(1) sebesar 0,1203, sedangkan pada ek(2) sebesar 0,0962. Dibandingkan dengan nilai *learning rate* yang lebih besar, kesalahan yang terjadi ketika menggunakan *learning rate* sebesar 0,001 bernilai paling kecil. Namun sama seperti sebelumnya, jumlah *neuron* pada *hidden layer* tidak berpengaruh pada selisih antara hasil.

Tabel 4. 9 Nilai MSE dengan *Learning rate* 0,001

No.	Jumlah <i>Hidden neuron</i>	Pengujian ke-	Nilai MSE pada ek(1)	Nilai MSE pada ek(2)
1	4	1	0.0444444	0.0444444
		2	0.4333333	0.3444444
		3	0.5222222	0.3222222
2	10	1	0.0444444	0.0444444
		2	0.0444444	0.0444444
		3	0.0444444	0.0444444
3	14	1	0.0555556	0.0555556
		2	0.0444444	0.0444444
		3	0.0444444	0.0444444
4	20	1	0.0555556	0.0555556
		2	0.0666667	0.0666667
		3	0.0444444	0.0444444

Halaman ini sengaja dikosongkan

BAB V

KESIMPULAN

Berdasarkan hasil pengujian, *Mean Square Error* (MSE) terkecil yaitu 0,044 diperoleh ketika dilakukan proses *training* pada data menggunakan *learning rate* sebesar 0,001. Nilai MSE untuk ek(1) dan ek(2) dengan *learning rate* 0,1; 0,01; dan 0,001 secara berturut-turut adalah 0,3638 dan 0,3333; 0,3814 dan 0,3314; dan 0,1203 dan 0,0962.

Perbedaan jumlah *hidden neuron* pada proses *training* dengan *learning rate* yang sama tidak berpengaruh terhadap nilai MSE, sedangkan perbedaan nilai *learning rate* berpengaruh terhadap nilai MSE. Hal ini dapat dilihat dari hasil *training* menggunakan *learning rate* 0,1 dengan jumlah *neuron* pada *hidden layer* yaitu 4, 10, 14 dan 20, nilai MSE diperoleh sama yaitu 0,3333.

Semakin besar jumlah *neuron* pada *hidden layer*, maka waktu yang dibutuhkan oleh sistem untuk melakukan proses *training* akan lebih lama. Ketika *neuron* pada *hidden layer* berjumlah 4, 10, 14, dan 20 saat dilakukan *training* dengan *learning rate* 0,01, maka waktu yang dibutuhkan masing-masing 27,868 detik; 39,310 detik; 55,176 detik; dan 65,272 detik.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] S. Kolla and L. Varatharasa, "Identifying Three-Phase Induction Motor Fault using Artificial Neural Networks," *ISA Transactions*, pp. 433-439, 2000.
- [2] V. N. Ghate and S. V. Dudul, "Cascade Neural-Network-Based Fault Classifier for Three-Phase Induction Motor," *IEEE Transaction on Industrial Electronics*, vol. 58, pp. 1555-1563, 2011.
- [3] K. Gupta and A. Kaur, "A Review on Fault Diagnosis of Induction Motor using Artificial Neural Networks," *International Journal of Science and Research (IJSR)*, vol. 3, no. 7, pp. 680-684, 2014.
- [4] P. S. Bhowmik, S. Pradhan and M. Prakash, "Fault Diagnostic and Monitoring Methods of Induction Motor: A Review," *International Journal of Applied Control, Electrical and Electronics Engineering (IJACEEE)*, vol. I, 2013.
- [5] J. S. Chapman, *Electric Machinery Fundamentals: Fourth Edition*, United States: The McGraw-Hill Companies, 2005.
- [6] S. Karmakar, "Induction Motor and Faults," in *Induction Motor Fault and Diagnosis*, Singapore, Springer Science, 2016.
- [7] S. Haykin, *Neural Networks and Learning Machines Third Edition*, New Jersey: Pearson Prentice Hall, 2009.
- [8] E. Tharani, V. Khartikeyan, S. Dharanith and S. Divya, "Alternator Performance Testing using PLC and SCADA," *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering (IJIREEICE)*, vol. IV, no. 5, pp. 344-347, 2016.

- [9] G. M. Ioannides, "Design and Implementation of PLC-Based Monitoring Control System for Induction Motor," *IEEE Transaction on Energy Conversion*, vol. IX, pp. 469-476, 2004.
- [10] P. S. Hamer, "Acceptance Testing of Electric Motors and Generators," *IEEE Transaction on Industry Application*, vol. 24, pp. 1138-1152, 1988.
- [11] L. Eren, A. Karahoca and M. J. Devaney, "Neural Network Based Motor Bearing Fault Detection," in *Instrumentation and Measurement Technology Conference*, Italy, 2004.

LAMPIRAN

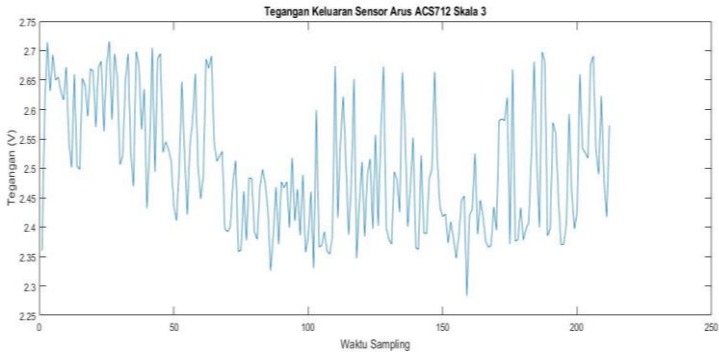
Lampiran 1. Data Kerusakan *Bearing*

No	Frekuensi Masukan (Hz)	Data 1			Data 2		
		Kecepatan Motor (RPM)	Tegangan Tachogenerator (V)	Arus Motor (A)	Kecepatan Motor (RPM)	Tegangan Tachogenerator (V)	Arus Motor (A)
1	0	0	0	0	0	0	0
2	10	0	0	0.499	0	0	0.448
3	20	0	0	0.735	0	0	0.655
4	25	0	0	0.842	0	0	0.764
5	30	0	0	0.94	0	0	0.858
6	35	0	0	1.025	0	0	0.971
7	40	2203	2.3335	0.805	2208	2.3204	0.798
8	50	2855	3.0275	0.826	2857	3.0378	0.816

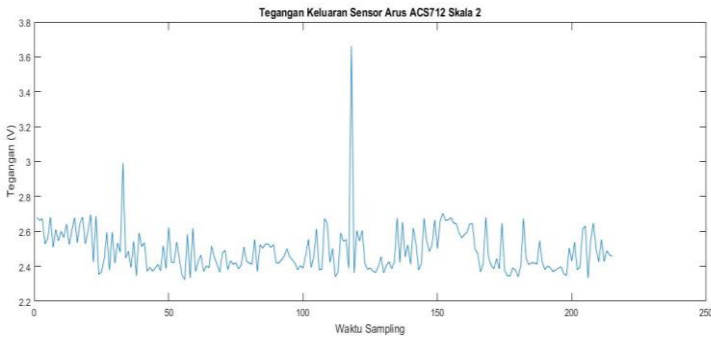
No	Frekuensi Masukan (Hz)	Data 3			Data 4		
		Kecepatan Motor (RPM)	Tegangan Tachogenerator (V)	Arus Motor (A)	Kecepatan Motor (RPM)	Tegangan Tachogenerator (V)	Arus Motor (A)
1	0	0	0	0	0	0	0
2	10	0	0	0.444	0	0	0.438
3	20	0	0	0.65	0	0	0.644
4	25	0	0	0.763	0	0	0.736
5	30	0	0	0.855	0	0	0.852
6	35	0	0	0.928	0	0	0.925
7	40	2222	2.3604	0.78	2202	2.3234	0.77
8	50	2854	3.0518	0.821	2852	3.0427	0.816

No	Frekuensi Masukan (Hz)	Data 5		
		Kecepatan Motor (RPM)	Tegangan Tachogenerator (V)	Arus Motor (A)
1	0	0	0	0
2	10	0	0	0.418
3	20	0	0	0.619
4	25	0	0	0.717
5	30	0	0	0.81
6	35	0	0	0.9
7	40	2224	2.362	0.766
8	50	2860	3.0368	0.814

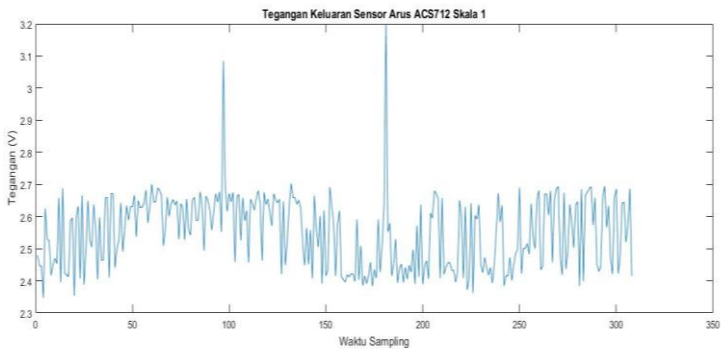
Lampiran 2. Pengambilan Data Arus Motor dengan Beban



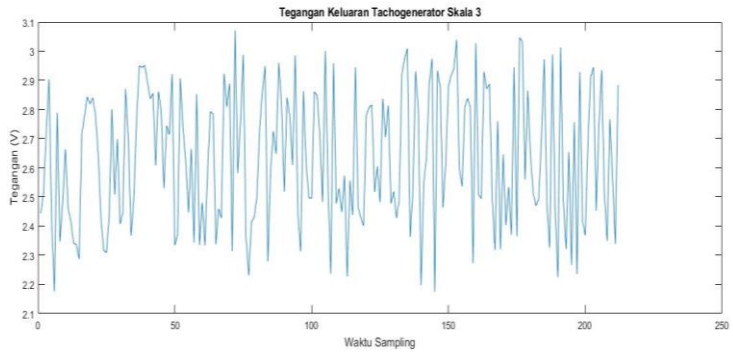
Gambar 1 Pembacaan Sensor Arus ACS712 saat Pembukaan Keran Skala 3



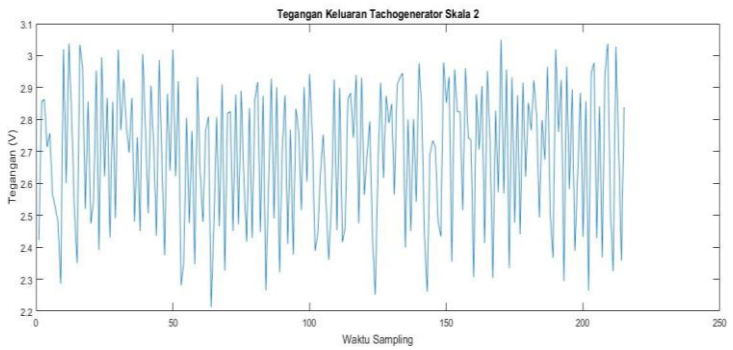
Gambar 2 Pembacaan Sensor Arus ACS712 saat Pembukaan Keran Skala 2



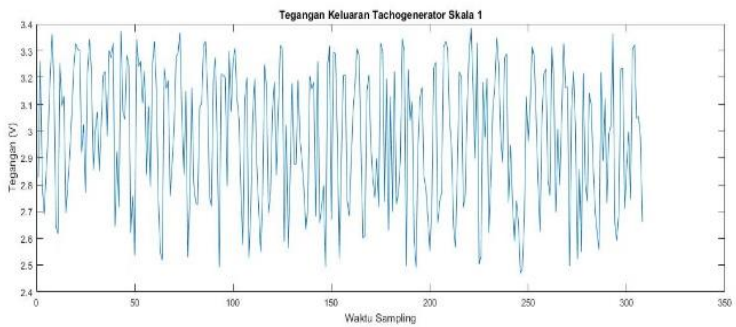
Gambar 3 Pembacaan Sensor Arus ACS712 saat Pembukaan Keran Skala 1



Gambar 4 Data Tegangan Tachogenerator saat Pembukaan Keran Skala 3



Gambar 5 Data Tegangan Tachogenerator saat Pembukaan Keran Skala 2



Gambar 6 Data Tegangan Tachogenerator saat Pembukaan Keran Skala 1

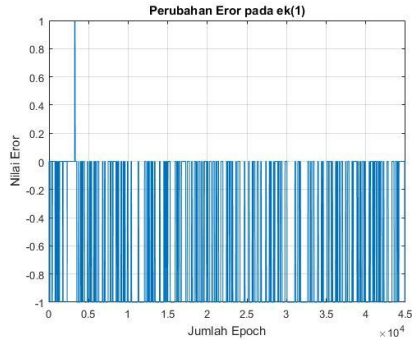
Lampiran 3. Data Kondisi Motor Normal

No	frekuensi masukan (Hz)	data 1			data 2		
		kecepatan motor (RPM)	tegangan tachogenerator (V)	arus motor (A)	kecepatan motor (RPM)	tegangan tachogenerator (V)	arus motor (A)
1	0	0	0	0	0	0	0
2	10	0	0	0.393	0	0	0.394
3	20	0	0	0.571	0	0	0.571
4	25	0	0	0.661	0	0	0.657
5	30	0	0	0.806	1369	0.73	0.64
6	35	1854	0.8521	0.674	1835	0.93	0.672
7	40	2160	1.0673	0.695	2031	1.0584	0.694
8	50	2748	1.3479	0.737	2652	1.3938	0.739

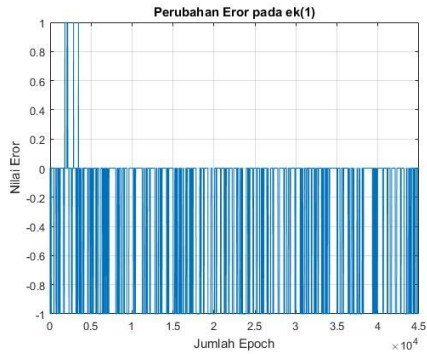
No	frekuensi masukan (Hz)	data 3			data 4		
		kecepatan motor (RPM)	tegangan tachogenerator (V)	arus motor (A)	kecepatan motor (RPM)	tegangan tachogenerator (V)	arus motor (A)
1	0	0	0	0	0	0	0
2	10	0	0	0.39	0	0	0.387
3	20	0	0	0.566	0	0	0.563
4	25	0	0	0.653	0	0	0.65
5	30	1517	0.7503	0.631	1517	0.7263	0.623
6	35	1843	0.8941	0.655	1858	0.8976	0.65
7	40	2186	1.0465	0.684	2225	0.9826	0.679
8	50	2798	1.3429	0.735	2867	1.3172	0.742

No	frekuensi masukan (Hz)	data 5		
		kecepatan motor (RPM)	tegangan tachogenerator (V)	arus motor (A)
1	0	0	0	0
2	10	0	0	0.387
3	20	0	0	0.562
4	25	0	0	0.647
5	30	1503	0.7076	0.624
6	35	1898	0.8346	0.651
7	40	2259	0.9981	0.678
8	50	2834	1.3438	0.742

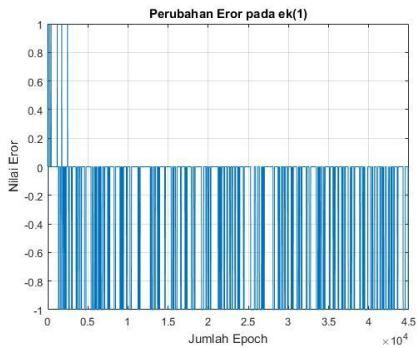
Lampiran 4. Perubahan Kesalahan pada ek(1)



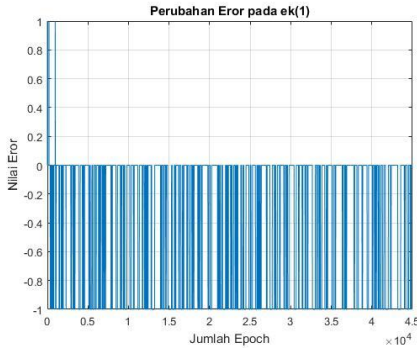
Gambar 7 Perubahan Kesalahan *Learning Rate* 0,1; Jumlah *Hidden Neuron*=4



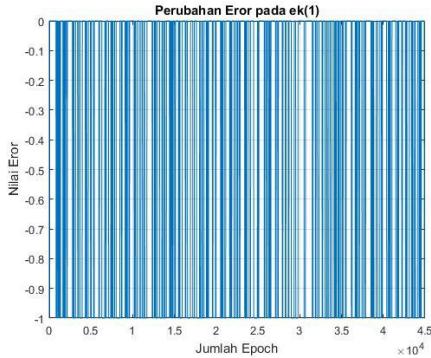
Gambar 8 Perubahan Kesalahan *Learning Rate* 0,1; *Hidden Neuron*=10



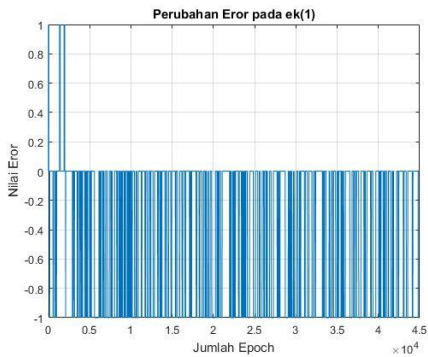
Gambar 9 Perubahan Kesalahan *Learning Rate* 0,1; *Hidden Neuron*=14



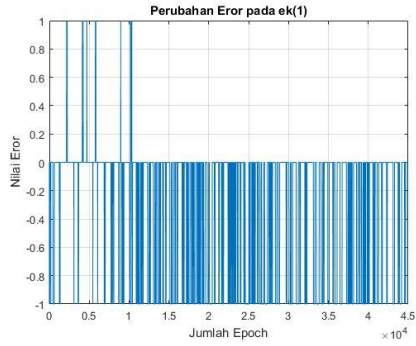
Gambar 10 Perubahan Kesalahan *Learning Rate* 0,1; *Hidden Neuron*=20



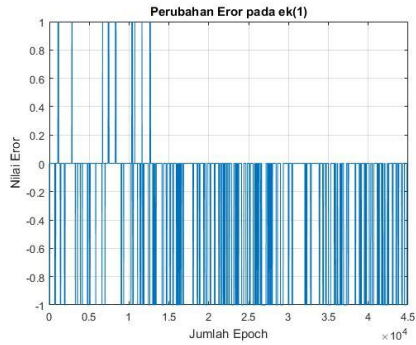
Gambar 11 Perubahan Kesalahan *Learning Rate* 0,01; *Hidden Neuron*=4



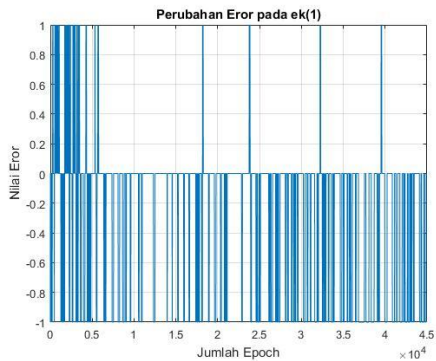
Gambar 12 Perubahan Kesalahan *Learning Rate* 0,01; *Hidden Neuron*=10



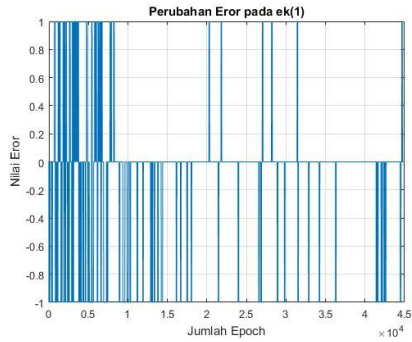
Gambar 13 Perubahan Kesalahan *Learning Rate* 0,01; *Hidden Neuron*=14



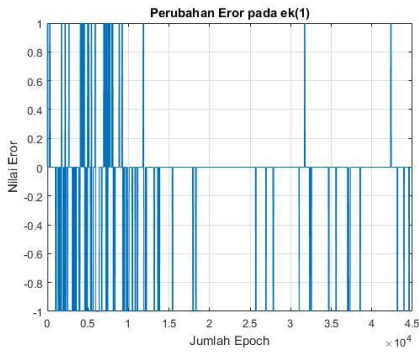
Gambar 14 Perubahan Kesalahan *Learning Rate* 0,01; *Hidden Neuron*=20



Gambar 15 Perubahan Kesalahan *Learning Rate* 0,001; *Hidden Neuron*=4

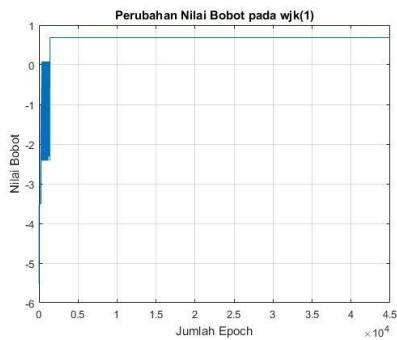


Gambar 16 Perubahan Kesalahan *Learning Rate* 0,001; *Hidden Neuron*=10

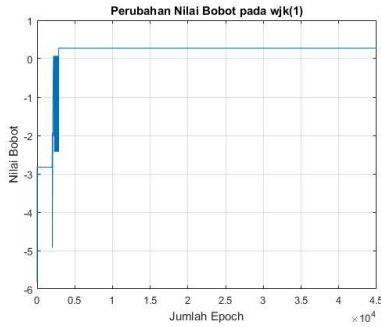


Gambar 17 Perubahan Kesalahan *Learning Rate* 0,001; *Hidden Neuron*=14

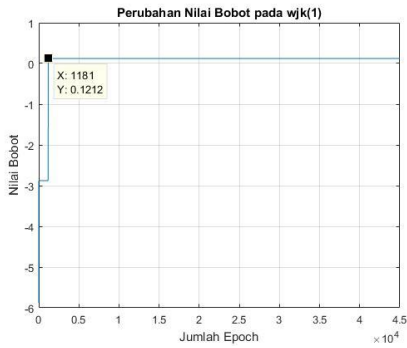
Lampiran 5. Perubahan Nilai Bobot wjk(1)



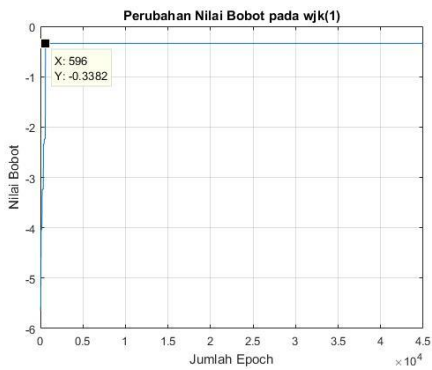
Gambar 18 Perubahan Bobot, *Learning Rate* 0,1; Jumlah *Hidden Neuron*=4



Gambar 19 Perubahan Bobot *Learning Rate* 0,1; Jumlah *Hidden Neuron*=10



Gambar 20 Perubahan Bobot *Learning Rate* 0,1; Jumlah *Hidden Neuron*=14



Gambar 21 Perubahan Bobot *Learning Rate* 0,1; Jumlah *Hidden Neuron*=20

Lampiran 6. Program *Neural network*

```
clear rmss1 rmss2 ero1 ero2 erm1 erm2 hh ek yii err
rmz erz1 erz2
clc
jd=90;
in= 2;
hn= 10;
on= 2;
n=0.005
wij=(rand(in,hn))/2;
wjk=(rand(hn,on))/2;
bot1=wij;
bot2=wjk;
%tk=[0 0;0 1;1 1];
%erm1=1
%erm2=1
itt=0
%%-----update weight-----
-----%%

%normalisasi
inor1=(datanya(1:90,1)-
repmat(min(min(datanya(1:90,1))),size(datanya(1:90,
1))))/(max(max(datanya(1:90,1)))-
min(min(datanya(1:90,1))));
inor2=(datanya(1:90,2)-
repmat(min(min(datanya(1:90,2))),size(datanya(1:90,
2))))/(max(max(datanya(1:90,2)))-
min(min(datanya(1:90,2))));

for k=1:500 %iterasi
    dat=randperm(jd);
    for nf=1:jd %jumlah data
        %dat=randperm(jd); %pengambilan data acak
        yi=[inor1(dat) inor2(dat)];
        vj=yi*wij;
        yj=hardlim(vj);
        vk=yj*wjk;
        yk=hardlim(vk);
        ek=[datanya(dat,3) datanya(dat,4)]-yk;
        wij=wij+(n*yi'*ek*wjk');
        wjk=wjk+(n*yj'*ek); %gausah pake if geng,
data dijadikan satu antara input dan output
```

```

        itt=itt+1;
        %erm1(itt+1)=erm1(itt)+ek(1)^2;
        %erm2(itt+1)=erm2(itt)+ek(2)^2;
        %rmss1(itt)=(erm1(itt)/itt)^0.5;
        %rmss2(itt)=(erm2(itt)/itt)^0.5;
        hh(itt)=wjk(1);
        erol(itt)=ek(1);
        ero2(itt)=ek(2);
    end

end
itt

%pengujian shiaaap
%dn=0;
%erz1=1;
%erz2=1;
for du=1:jd
    datt=1:jd;
    in=[inor1(datt) inor2(datt)];
    vjj=in*wij;
    oh=hardlim(vjj);
    vkk=oh*wjk;
    ok=hardlim(vkk);
    err=[datanya(datt,3) datanya(datt,4)]-ok;

end
erz1=((sum((err(1:90,1).^2)))/jd);
erz2=((sum((err(1:90,2).^2)))/jd);
rmz=[erz1 erz2]

figure(3);
plot(hh)
grid on
title('Perubahan Nilai Bobot pada wjk(1)')
xlabel('Jumlah Epoch')
ylabel('Nilai Bobot')
figure(1);
plot(erol)
grid on
title('Perubahan Kesalahan pada ek(1)')
xlabel('Jumlah Epoch')
ylabel('Nilai Kesalahan')
figure(2);

```

```
plot(ero2)
grid on
title('Perubahan Kesalahan pada ek(2)')
xlabel('Jumlah Epoch')
ylabel('Nilai Kesalahan')
```

RIWAYAT PENULIS



Diandra Tamara Ismira Rendra lahir di kota Denpasar pada tanggal 20 Desember 1995 merupakan anak pertama dari dua bersaudara. Penulis menempuh Pendidikan formal di SDK Anugrah Denpasar (2001-2007), SMP Negeri 1 Denpasar (2007-2010) dan SMA Negeri 1 Denpasar (2010-2013). Penulis melanjutkan pendidikannya di jurusan D3 Teknik Elektro Institut Teknologi Sepuluh Nopember Surabaya, program studi *Computer Control* (2013-2016). Penulis kembali melanjutkan studinya melalui program lintas jalur di jurusan Teknik Elektro Institut Teknologi Sepuluh Nopember Surabaya bidang studi Teknik Sistem Pengaturan. Motto hidup penulis yaitu “bersukacitalah dalam pengharapan, bersabarlah dalam kesesakkan, dan bertekunlah dalam doa”.

E-mail: tamaradiandra@gmail.com