

**PREDIKSI KUALITAS AIR DANAU TOBA
MENGUNAKAN *EXTREME LEARNING MACHINE***

SKRIPSI

ERIC SUWARNO

121402071



**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA**

2017

**PREDIKSI KUALITAS AIR DANAU TOBA
MENGUNAKAN *EXTREME LEARNING MACHINE***

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah
Sarjana Komputer

ERIC SUWARNO

121402071



**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA**

2017

PERSETUJUAN

Judul : PREDIKSI KUALITAS AIR DANAU TOBA
MENGUNAKAN EXTREME LEARNING
MACHINE

Kategori : SKRIPSI

Nama : ERIC SUWARNO

Nomor Induk Mahasiswa : 121402071

Program Studi : S1 TEKNOLOGI INFORMASI

Departemen : TEKNOLOGI INFORMASI

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA

Komisi Pembimbing :

Pembimbing 2, Pembimbing 1,

Dr. Maya Silvi Lydia, B.Sc., M.Sc.
NIP. 19740127 200212 2 001

Romi Fadillah Rahmat, B.Comp.Sc., M.Sc.
NIP. 19860303 201012 1 004

Diketahui/disetujui oleh
Program Studi S1 Teknologi Informasi
Ketua,

Romi Fadillah Rahmat, B.Comp.Sc., M.Sc.
NIP. 19860303 201012 1 004

PERNYATAAN**PREDIKSI KUALITAS AIR DANAU TOBA MENGGUNAKAN EXTREME
LEARNING MACHINE****SKRIPSI**

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sebelumnya.

Medan, Juli 2017

Eric Suwarno

121402071

UCAPAN TERIMA KASIH

Puji dan syukur penulis sampaikan kepada Tuhan Yesus Kristus, yang telah memberikan kesehatan kepada penulis serta menuntun penulis, sehingga penulis dapat menyelesaikan skripsi ini sebagai syarat untuk memperoleh gelar Sarjana Komputer.

Penulis juga ingin mengucapkan terima kasih kepada Bapak Romi Fadillah Rahmat, B.Comp.Sc., M.Sc. selaku pembimbing pertama dan Ibu Dr. Maya Silvi Lydia, B.Sc., M.Sc. selaku pembimbing kedua, yang telah meluangkan waktu dan tenaganya untuk membimbing penulis dalam penelitian serta penulisan skripsi ini. Tanpa inspirasi serta motivasi yang diberikan dari kedua pembimbing, tentunya penulis tidak akan dapat menyelesaikan skripsi ini. Penulis juga mengucapkan terima kasih kepada Ibu Sarah Purnamawati, ST, M.Sc. sebagai dosen pembimbing pertama dan Bapak Ainul Hizriadi, S.Kom., M.Sc. sebagai dosen pembimbing kedua, yang telah memberikan masukan serta kritik yang bermanfaat dalam penulisan skripsi ini. Ucapan terima kasih juga ditujukan kepada Ketua dan Sekretaris Program Studi Teknologi Informasi, Dekan dan Wakil Dekan Fakultas Ilmu Komputer dan Teknologi Informasi, dan semua dosen serta pegawai di lingkungan program studi Teknologi Informasi, yang telah membantu serta membimbing penulis selama proses perkuliahan.

Penulis tentunya tidak lupa berterima kasih kepada kedua orang tua penulis, yaitu Alm. Bapak Tjia Ju Hin dan Ibu Lie Ling Ling, yang telah membesarkan penulis dengan sabar dan penuh kasih sayang, serta doa dari mereka yang selalu menyertai penulis selama ini. Terima kasih juga tidak lupa penulis ucapkan kepada adik-adik penulis, Ruswan Efendi dan Sherly Natalia, yang selalu memberikan dukungan kepada penulis. Penulis juga berterima kasih kepada seluruh anggota keluarga penulis yang namanya tidak dapat disebutkan satu persatu.

Terima kasih juga penulis ucapkan kepada teman-teman yang telah memberikan dukungan, khususnya Athmanathan, Tika Hairani, Siti Fatimah, Muhammad Iqbal Rizki Siregar, Rona I Donna Vynaima Simbolon, Siti Hasanah Nainggolan, serta seluruh teman-

teman angkatan 2012 Program Studi Teknologi Informasi, yang telah menemani keseharian penulis selama perkuliahan hingga proses skripsi ini. Penulis juga mengucapkan terima kasih kepada abang kakak senior dan junior yang telah membantu penulis dalam penyusunan skripsi ini.

Akhir kata, penulis berharap agar skripsi ini berguna dan memberikan manfaat kepada seluruh pembaca.

Penulis,

Eric Suwarno

DAFTAR ISI

	Hal.
PERSETUJUAN	ii
PERNYATAAN.....	iii
UCAPAN TERIMA KASIH.....	iv
ABSTRAK	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xii
 BAB 1 PENDAHULUAN	
1.1. Latar Belakang	1
1.2. Rumusan Masalah.....	3
1.3. Batasan Masalah	3
1.4. Tujuan Penelitian	3
1.5. Manfaat Penelitian	4
1.6. Metodologi Penelitian.....	4
1.7. Sistematika Penulisan	5
 BAB 2 TINJAUAN PUSTAKA	
2.1. Pencemaran air	7
2.2. Indeks kualitas air (<i>Water Quality Index</i>).....	8
2.3. Artificial neural networks	10

	2.4. <i>Machine learning</i>	11
	2.5. <i>Extreme learning machine (ELM)</i>	13
	2.6. Penelitian Terdahulu	16
BAB 3	ANALISIS DAN PERANCANGAN	
	3.1. Data yang Digunakan.....	18
	3.2. Analisis Sistem.....	19
	3.2.1. <i>Preprocessing</i>	20
	3.2.2. Penentuan karakteristik <i>artificial neural network</i>	22
	3.2.2.1. Penentuan jumlah node pada <i>hidden layer</i>	22
	3.2.2.2. Penentuan fungsi aktivasi	23
	3.2.3. Proses <i>training</i>	24
	3.2.3.1. Pengacakan <i>input weight</i> dan <i>bias</i>	24
	3.2.3.2. Penghitungan <i>hidden layer output matrix</i>	25
	3.2.3.3. Penghitungan <i>output weight</i>	25
	3.2.4. Proses <i>testing</i>	25
	3.2.4.1. Pengolahan target keluaran.....	25
	3.2.4.2. Penghitungan keluaran	25
	3.3. Perancangan Antarmuka Sistem	26
	3.3.1. Perancangan menu sistem	26
	3.3.2. Rancangan tampilan halaman utama.....	26
	3.4. Hasil Akhir	28
BAB 4	IMPLEMENTASI DAN PENGUJIAN	
	4.1. Implementasi Sistem	29
	4.1.1. Spesifikasi perangkat keras dan perangkat lunak yang digunakan.....	29

4.1.2. Implementasi perancangan antarmuka	30
4.1.3. Implementasi data.....	32
4.2. Prosedur Operasional.....	32
4.3. Hasil Pengujian	35
4.3.1. Hasil pengujian terhadap <i>dataset</i> Ajibata	35
4.3.2. Hasil pengujian terhadap <i>dataset</i> Ambarita	40
4.3.3. Hasil pengujian terhadap <i>dataset</i> Haranggaol	44
4.3.4. Hasil pengujian terhadap <i>dataset</i> Parapat.....	49
4.3.5. Perbandingan dengan algoritma <i>backpropagation</i> (Ming, 2012).....	53
BAB 5 KESIMPULAN DAN SARAN	
5.1. Kesimpulan	55
5.2. Saran.....	56
DAFTAR PUSTAKA	57

DAFTAR TABEL

Tabel 2.1. Tabel penentuan indeks kualitas air (Sumber: Keputusan Menteri Lingkungan Hidup Nomor 115 Tahun 2003)	10
Tabel 2.2. Penelitian terdahulu	17
Tabel 3.1. Nilai normal, minimum, dan maksimum dari setiap parameter kualitas air (Lambrou et al., 2012; Peraturan Pemerintah Nomor 82 Tahun 2001)	21
Tabel 4.1. Perincian data yang digunakan dalam penelitian	32
Tabel 4.2. Parameter <i>artificial neural network</i> pada pengujian setiap <i>dataset</i>	35
Tabel 4.3. <i>Training error</i> terbaik dalam pengujian <i>dataset</i> Ajibata	36
Tabel 4.4. <i>Testing error</i> terbaik dalam pengujian <i>dataset</i> Ajibata	38
Tabel 4.5. <i>Training error</i> terbaik dalam pengujian <i>dataset</i> Ambarita	41
Tabel 4.6. <i>Testing error</i> terbaik dalam pengujian <i>dataset</i> Ambarita	42
Tabel 4.7. <i>Training error</i> terbaik dalam pengujian <i>dataset</i> Haranggaol	45
Tabel 4.8. <i>Testing error</i> terbaik pada pengujian <i>dataset</i> Haranggaol	46
Tabel 4.9. <i>Training error</i> terbaik dalam pengujian <i>dataset</i> Parapat	50
Tabel 4.10. <i>Testing error</i> terbaik dalam pengujian <i>dataset</i> Parapat	51

DAFTAR GAMBAR

Gambar 2.1. Struktur neuron pada otak makhluk hidup (Sumber: Heaton, 2008)	10
Gambar 2.2. Struktur sebuah neuron pada <i>artificial neural network</i> (Sumber: Jain <i>et al.</i> , 1996)	11
Gambar 2.3. Struktur umum ELM (Sumber: iopscience.iop.org)	14
Gambar 3.1. Bentuk <i>dataset</i> yang dihasilkan dari penelitian Rahmat <i>et al.</i> (2016)	18
Gambar 3.2. Arsitektur umum	19
Gambar 3.3. Contoh baris data pada <i>dataset</i> setelah proses normalisasi	21
Gambar 3.4. Ilustrasi <i>input weight</i> dan <i>bias</i>	24
Gambar 3.5. Struktur antarmuka sistem	26
Gambar 3.6. Rancangan halaman utama aplikasi	27
Gambar 3.7. Ilustrasi tampilan hasil prediksi	28
Gambar 4.1. Hasil rancangan antarmuka aplikasi	30
Gambar 4.2. Ilustrasi tampilan grafik hasil prediksi	31
Gambar 4.3. Ilustrasi grafik <i>testing error</i>	31
Gambar 4.4. Ilustrasi pemilihan <i>training dataset</i>	33
Gambar 4.5. Ilustrasi pemilihan <i>testing dataset</i>	33
Gambar 4.6. Ilustrasi pengaturan jumlah <i>hidden neuron</i> (a) dan fungsi aktivasi (b)	34
Gambar 4.7. Ilustrasi tombol “Start Prediction” pada aplikasi	34
Gambar 4.8. Grafik training error dalam setiap perulangan yang diperoleh pada pengujian kedelapan menggunakan dataset Ajibata	39

Gambar 4.9. Grafik testing error dalam setiap perulangan yang diperoleh pada pengujian kedelapan menggunakan dataset Ajibata	39
Gambar 4.10. Grafik hasil prediksi saat memperoleh testing error terbaik pada pengujian kelima menggunakan dataset Ajibata	40
Gambar 4.11. Grafik training error yang diperoleh pada setiap perulangan dalam pengujian menggunakan dataset Ambarita	43
Gambar 4.12. Grafik testing error yang diperoleh pada setiap perulangan dalam pengujian menggunakan dataset Ambarita	43
Gambar 4.13. Grafik hasil prediksi saat memperoleh testing error terbaik pada pengujian kelima menggunakan dataset Ambarita	44
Gambar 4.14. Grafik training error yang diperoleh pada setiap perulangan dalam pengujian kedelapan menggunakan dataset Haranggaol	47
Gambar 4.15. Grafik testing error yang diperoleh pada setiap perulangan dalam pengujian kedelapan menggunakan dataset Haranggaol	48
Gambar 4.16. Grafik hasil prediksi saat memperoleh testing error terbaik pada pengujian keenam menggunakan dataset Haranggaol	49
Gambar 4.17. Grafik training error yang diperoleh pada setiap perulangan dalam pengujian keenam menggunakan dataset Parapat	52
Gambar 4.18. Grafik testing error yang diperoleh pada setiap perulangan dalam pengujian keenam menggunakan dataset Parapat	52
Gambar 4.19. Grafik hasil prediksi saat memperoleh testing error terbaik pada pengujian keenam menggunakan dataset Parapat	53
Gambar 4.20. Grafik hasil pelatihan dari backpropagation neural network (Ming, 2012)	54

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Perairan Danau Toba merupakan perairan dengan kualitas air tercemar, terutama di sekitar wilayah Kecamatan Haranggaol Horison, Kabupaten Simalungun, Provinsi Sumatera Utara (Haro *et al.*, 2013). Tingkat pencemaran di perairan Danau Toba bervariasi dari ringan hingga sedang.

Pencemaran di perairan Danau Toba tidak hanya bersumber dari aktivitas rumah tangga, namun juga aktivitas lainnya, seperti tambak ikan, peternakan, dan industri. Selain itu, populasi eceng gondok dan limbah dari sungai yang bermuara ke Danau Toba juga menjadi sumber pencemaran pada perairan Danau Toba.

Pengukuran tingkat pencemaran pada perairan Danau Toba selama ini dilakukan dengan mengambil sampel air pada titik tertentu di perairan Danau Toba. Sampel ini kemudian akan dikirim ke laboratorium untuk diuji, dan didapat hasil pengujian sampel tersebut. Proses pengukuran ini memerlukan waktu yang relatif lama, terutama dalam proses pengambilan sampel dan pengujian pada laboratorium. Karena itu, dibutuhkan sebuah cara yang sesuai untuk mengatasi kendala tersebut melalui kecanggihan teknologi informasi dan komputer.

Perkembangan teknologi informasi dan komputer memungkinkan pengukuran kualitas air melalui peralatan seperti Arduino dan Raspberry Pi, dengan memanfaatkan sensor-sensor seperti sensor *dissolved oxygen* (DO), sensor keasaman, dan sensor suhu. Hal ini memungkinkan peneliti untuk merancang sebuah alat yang dapat digunakan untuk melakukan pengukuran secara berkelanjutan.

Selain perangkat keras tersebut, dibutuhkan juga sebuah metode yang dapat digunakan untuk melakukan prediksi data secara berkelanjutan, salah satunya dengan menggunakan jaringan saraf tiruan, atau *artificial neural networks* (Kasabov, 2007). *Artificial neural network* telah digunakan dalam memantau proses pemasangan baut secara otomatis (Lara *et al.*, 1999), pengaturan keseimbangan tegangan sistem listrik (Popovic *et al.*, 1998), dan sistem pembangkit listrik tenaga angin (Ata, 2015).

Permasalahan yang dihadapi dalam penggunaan *artificial neural network* adalah waktu yang digunakan dalam pemrosesan data, terutama dalam proses *learning*. Penelitian untuk meningkatkan kecepatan pemrosesan data dalam *artificial neural network* dimulai oleh Werbos (1974) dan Rumelhart *et al.* (1986), dengan menggunakan algoritma *back-propagation*, di mana nilai *error* akan dihitung berdasarkan data pada setiap node dan fungsi aktivasi.

Walaupun terjadi perbaikan pada waktu komputasi, algoritma ini memiliki kelemahan dalam memproses data dengan jumlah besar (Deng *et al.*, 2015). Chandra & Sharma mengembangkan *multilayer perceptron* dengan parameter yang diatur dengan fungsi trigonometri (2014) dan menerapkan parameter fungsi trigonometri pada *deep neural network* (2016). Hinton dan Teh (2006) memperkenalkan algoritma *learning* yang dapat digunakan untuk mempercepat waktu komputasi dalam *deep belief nets*.

Salah satu algoritma yang dapat digunakan untuk mempercepat waktu komputasi pada *artificial neural network* adalah *extreme learning machine* (ELM). *Extreme learning machine* dikemukakan oleh Huang *et al.* (2006) pada *single hidden layer feedforward neural network*. *Extreme learning machine* memiliki kemampuan meningkatkan kecepatan komputasi pada *artificial neural networks*.

Extreme learning machine telah digunakan dalam beberapa penelitian. Huang *et al.* (2013) menggunakan *extreme learning machine* untuk mendeteksi tumor liver. Fu *et al.* (2015) menggunakan *extreme learning machine* untuk melakukan prediksi pergerakan rotasi kapal. Pangaribuan & Suharjito (2014) menggunakan *extreme learning machine* untuk diagnosis diabetes mellitus. Zhai & Du (2008) menggunakan *extreme learning machine* untuk identifikasi spesies tanaman. Huang *et al.* (2016) menggabungkan *extreme learning machine* dan histogram gradien warna terorientasi (*histogram of oriented gradients/HOG*) dalam pengenalan rambu lalu lintas.

Berdasarkan latar belakang di atas, penulis mengajukan proposal penelitian dengan judul “**Prediksi Kualitas Air Danau Toba Menggunakan *Extreme Learning Machine***”. Hasil yang diharapkan dari penelitian ini adalah perkiraan kualitas air yang diukur di perairan Danau Toba.

1.2. Rumusan Masalah

Pemeriksaan mengenai kualitas air di Danau Toba selama ini hanya dilakukan melalui pemeriksaan sampel air pada laboratorium, sehingga dibutuhkan waktu yang relatif lebih lama untuk mengetahui kualitas air. Untuk mengantisipasi hal tersebut, diperlukan sebuah metode untuk melakukan prediksi kualitas air di perairan Danau Toba, agar dapat dimanfaatkan oleh pihak-pihak yang terkait untuk mengetahui tingkat pencemaran air di perairan Danau Toba.

1.3. Batasan Masalah

Untuk menghindari penyimpangan dan perluasan yang tidak diperlukan, penulis membuat batasan sebagai berikut:

1. Prediksi dilakukan berdasarkan data pengukuran kualitas air di perairan Danau Toba yang dilakukan oleh Rahmat *et al.* (2016), di mana pengukuran dilakukan di tiga titik, yaitu Parapat, Ajibata, dan Haranggaol;
2. Prediksi dilakukan berdasarkan hasil pengukuran kadar *dissolved oxygen*, temperatur air, suhu air, tingkat kelembaban, suhu udara, potensi reduksi oksidasi (*oxidation reduction potential/ORP*), dan tingkat keasaman (*pH*), sesuai dengan parameter pengukuran yang dilakukan oleh Rahmat *et al.* (2016); dan
3. *Extreme learning machine* diterapkan pada *single hidden layer feedforward neural networks*.

1.4. Tujuan Penelitian

Penelitian ini bertujuan untuk memperkirakan tingkat pencemaran yang terjadi di perairan Danau Toba melalui hasil pengukuran kadar kualitas air yang dilakukan oleh Rahmat *et al.* (2016), sehingga dapat digunakan sebagai alat bantu bagi instansi-instansi yang terlibat dalam pengelolaan Danau Toba untuk pengambilan kebijakan-kebijakan yang terkait dengan kualitas lingkungan Danau Toba.

1.5. Manfaat Penelitian

Manfaat yang diperoleh dari penelitian ini adalah sebagai berikut:

1. Melakukan proses perkiraan terhadap kualitas air di Danau Toba, dengan menggunakan *extreme learning machine* dan memaparkan hasilnya;
2. Membantu pihak-pihak terkait seperti pemerintah daerah setempat dalam menerapkan kebijakan terkait kondisi lingkungan di daerahnya, terutama di perairan Danau Toba; dan
3. Dapat digunakan sebagai referensi mengenai penerapan *artificial neural networks*, khususnya *extreme learning machines*, dalam proses prediksi kualitas air.

1.6. Metodologi Penelitian

Terdapat beberapa tahapan dalam penelitian ini untuk menghasilkan suatu sistem yang sesuai dengan yang diharapkan, yaitu:

1. Studi Literatur

Dalam kegiatan ini, peneliti mempelajari dokumen yang berkaitan dengan literatur dan teori sebagai referensi dalam penelitian, di mana literatur tersebut berhubungan dengan penelitian yang dilakukan. Literatur tersebut dapat berupa buku, jurnal, tesis, makalah, serta sumber lainnya yang diperoleh dari internet. Referensi yang dikumpulkan berkaitan dengan *artificial neural network*, *extreme learning machine*, data kualitas air di Danau Toba, serta indeks kualitas air.

2. Pengambilan Data

Setelah studi literatur dilakukan, penelitian akan dilanjutkan dengan pengambilan data. Data yang digunakan adalah data yang diperoleh dari penelitian yang dilakukan oleh Rahmat *et al.* (2016), yang dilakukan di perairan Danau Toba. Pengukuran dilakukan di tiga titik, yaitu di perairan Danau Toba sekitar Ajibata, Parapat, dan Haranggaol.

3. Analisis Permasalahan

Dalam tahap ini, dilakukan analisis terhadap data yang telah dikumpulkan. Hal ini bertujuan untuk mengetahui parameter-parameter yang dibutuhkan untuk kalkulasi kualitas air.

4. Pembangunan Program

Dalam tahap ini, dilakukan perancangan sistem yang dapat digunakan untuk melakukan pemantauan kualitas air di perairan Danau Toba.

5. Uji Coba Sistem

Dalam tahap ini, dilakukan uji coba terhadap sistem yang telah dibangun. Hal ini dilakukan untuk mengetahui apakah sistem sudah bekerja sesuai dengan fungsi yang diharapkan.

6. Dokumentasi dan Penyusunan Laporan

Dalam tahap ini, dilakukan dokumentasi dan menyusun laporan akhir dari penelitian yang telah dilakukan mengenai penerapan *extreme learning machine* dalam melakukan pemantauan kualitas air dan tingkat pencemaran air di perairan Danau Toba.

1.7. Sistematika Penulisan

Sistematika penulisan dari skripsi ini terdiri dari lima bagian utama sebagai berikut.

Bab 1 : Pendahuluan

Bab ini berisi latar belakang dari penelitian yang dilaksanakan, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, metodologi penelitian, serta sistematika penulisan.

Bab 2 : Tinjauan Pustaka

Bab ini berisi teori-teori yang diperlukan untuk memahami permasalahan yang dibahas pada penelitian ini. Teori-teori yang berhubungan dengan kualitas air, *artificial neural networks*, *machine learning*, dan *extreme learning machine*, akan dibahas pada bab ini.

Bab 3 : Analisis dan Perancangan

Bab ini menjabarkan arsitektur umum dari penelitian yang dilakukan. Setiap tahap yang dilakukan pada proses normalisasi, proses training, proses *testing*, dan perancangan aplikasi yang dibuat dalam penelitian ini, akan dijabarkan pada bab ini.

Bab 4 : Implementasi dan Pengujian

Bab ini berisi pembahasan tentang implementasi dari perancangan yang telah dijelaskan pada Bab 3. Selain itu, hasil yang didapatkan dari tahap pengujian terhadap implementasi yang dilakukan juga dijabarkan pada Bab ini.

Bab 5 : Kesimpulan dan Saran

Bab ini berisi ringkasan serta kesimpulan dari rancangan yang telah dibahas pada Bab 3, serta hasil penelitian yang dijabarkan pada Bab 4. Bagian akhir dari bab ini memuat saran-saran yang diajukan untuk pengembangan penelitian selanjutnya.

BAB 2

TINJAUAN PUSTAKA

2.1. Pencemaran air

Air merupakan salah satu komponen utama dalam lingkungan yang berperan penting dalam kehidupan (Warlina, 2004). Air, yang bersumber dari sungai, danau, gletser, air hujan, air tanah, dan sumber lainnya, memiliki peran penting, tidak hanya untuk dikonsumsi, tetapi juga digunakan dalam bidang pertanian, peternakan, perhutanan, kegiatan industri, perikanan, dan bidang-bidang lainnya (Effendi, 2015). Karena itu, air harus senantiasa terjaga kualitasnya, untuk mencegah terjadinya pencemaran air, yang dapat membahayakan bagi kehidupan.

Secara harafiah, pencemaran air dapat diartikan sebagai suatu kondisi di mana sebuah kawasan perairan berada dalam kondisi tercemar. Menurut Undang-Undang Nomor 32 Tahun 2009 tentang Perlindungan dan Pengelolaan Lingkungan Hidup, pencemaran lingkungan hidup adalah masuk atau dimasukkannya salah satu atau kombinasi antara makhluk hidup, zat, energi, dan komponen lain ke dalam lingkungan hidup oleh kegiatan manusia sehingga kualitasnya turun sehingga mencapai tingkat tertentu, yang menyebabkan lingkungan hidup tidak dapat berfungsi sesuai dengan peruntukannya. Jadi, dapat disimpulkan bahwa pencemaran air merujuk pada kondisi perairan yang mengandung komponen seperti makhluk hidup atau zat, yang pada kadar tertentu dapat mengganggu keseimbangan ekosistem yang berkembang di dalam kawasan perairan tersebut.

Menurut Warlina (2004), tingkat pencemaran air dapat diukur berdasarkan pengamatan secara fisis, kimiawi dan biologis. Pengamatan secara fisis dapat dilakukan dengan memperhatikan sifat-sifat fisik air, seperti tingkat kejernihan air, perubahan suhu dan warna; serta membandingkan perubahan warna, bau, dan rasa pada air. Peng-

amatan secara kimiawi dapat dilakukan dengan mengukur konsentrasi zat kimia yang terlarut dan tingkat keasaman. Pengamatan secara biologis dapat dilakukan dengan melakukan pengamatan atas mikroorganisme yang terkandung di dalam air, dengan memberikan perhatian khusus pada bakteri yang dapat memicu penyakit.

2.2. Indeks kualitas air (*Water Quality Index*)

Berbagai pendapat telah dikemukakan mengenai indeks kualitas air. Keputusan Menteri Negara Lingkungan Hidup Nomor 115 Tahun 2003 tentang Pedoman Penentuan Status Mutu Air mendefinisikan mutu air sebagai kondisi kualitas air yang diukur dan atau diuji berdasarkan parameter-parameter tertentu dan metode tertentu berdasarkan peraturan perundang-undangan yang berlaku. Efendi (2016) berpendapat bahwa indeks kualitas air adalah sebuah metode untuk mengukur mutu air, yang didapat dari hasil perhitungan parameter-parameter tertentu, di mana parameter tersebut dapat mencerminkan indikator kualitas air.

Penggunaan indeks kualitas air diperkenalkan oleh Horton (1965), di mana kalkulasi indeks kualitas air dilakukan melalui pengukuran parameter seperti *dissolved oxygen* (DO), tingkat keasaman, *coliform*, tingkat kemampuan konduksi, kadar klorin, dan parameter lainnya. Indeks kualitas air juga diperkenalkan oleh Brown *et al.* (1970), dengan menghitung *weight* dari setiap parameter yang diukur.

Menurut Abbasi & Abbasi (2012), indeks kualitas air, di mana setiap parameter dalam pengukurannya ditentukan oleh pihak yang berwenang dalam pengendalian kualitas sumber daya air, dapat memudahkan kegiatan-kegiatan yang membutuhkan pemantauan kualitas air, seperti penelitian sumber daya air, perencanaan lingkungan, pemurnian air, dan kegiatan lainnya. Manfaat dari penerapan indeks kualitas air dalam menentukan kualitas dari sumber daya air adalah sebagai berikut:

- a. Membantu pengambilan keputusan mengenai alokasi dana dan prioritas dalam pengelolaan sumber daya air;
- b. Berperan sebagai media pembanding kualitas sumber daya air antar wilayah geografis tertentu;
- c. Menentukan kesesuaian kualitas sumber daya air di suatu wilayah dengan parameter yang telah ditentukan oleh pihak berwenang;

- d. Memperkirakan perubahan kualitas sumber daya air di suatu wilayah sesuai pergerakan waktu;
- e. Berperan sebagai media informasi kepada publik mengenai tingkatan kualitas sumber daya air di suatu wilayah; dan
- f. Membantu kegiatan penelitian yang memerlukan pengukuran kualitas sumber daya air, seperti dampak aktivitas pembangunan terhadap kualitas sumber daya air, dan penelitian lainnya.

Ukuran kualitas air di Indonesia ditentukan oleh pemerintah melalui Keputusan Menteri Negara Lingkungan Hidup Nomor 115 Tahun 2003 tentang Pedoman Penentuan Status Mutu Air. Penentuan kualitas air dilakukan melalui penghitungan parameter-parameter dengan menggunakan metode Storet. Dalam metode Storet, data dari parameter kualitas air yang terukur akan dibandingkan dengan nilai baku yang disesuaikan dengan tujuan penggunaan air. Sistem nilai yang digunakan dalam menentukan kualitas air adalah sistem nilai dari US-EPA (*Environmental Protection Agency*), di mana kualitas air ditentukan dari empat kelas, yaitu:

- a. kelas A atau baik sekali, dengan skor 0, menunjukkan bahwa sumber daya air yang diukur telah memenuhi nilai baku mutu air;
- b. kelas B atau baik, dengan skor antara -1 hingga -10, menunjukkan bahwa sumber daya air yang diukur berada dalam kondisi tercemar dengan tingkat ringan;
- c. kelas C atau sedang, dengan skor antara -11 hingga -30, menunjukkan bahwa sumber daya air yang diukur berada dalam kondisi tercemar dengan tingkat sedang; dan
- d. kelas D atau buruk, dengan skor kurang dari -31, menunjukkan bahwa sumber daya air yang diukur berada dalam kondisi tercemar dengan tingkat berat.

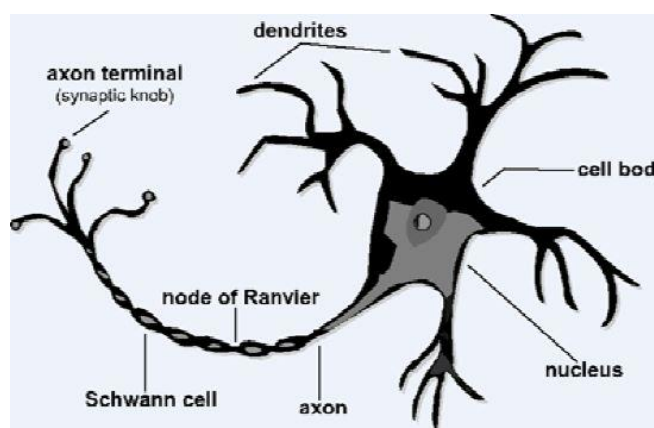
Pengukuran kualitas air menggunakan metode Storet dilakukan dengan mengumpulkan data kualitas air secara berkala. Hasil pengukuran yang didapat akan dibandingkan dengan baku mutu yang dibutuhkan sumber daya air untuk memenuhi tujuan penggunaan. Pengurangan indeks akan dilakukan apabila hasil pengukuran yang didapat tidak sesuai dengan standar baku mutu yang ditentukan. Besar dari pengurangan indeks dijelaskan oleh Tabel 2.1.

Tabel 2.1. Tabel penentuan indeks kualitas air (Sumber: Keputusan Menteri Lingkungan Hidup Nomor 115 Tahun 2003)

Jumlah parameter	Nilai	Parameter		
		Fisika	Kimia	Biologi
Di bawah 10	Maksimum	-1	-2	-3
	Minimum	-1	-2	-3
	Rata-rata	-3	-6	-9
10 ke atas	Maksimum	-2	-4	-6
	Minimum	-2	-4	-6
	Rata-rata	-6	-12	-18

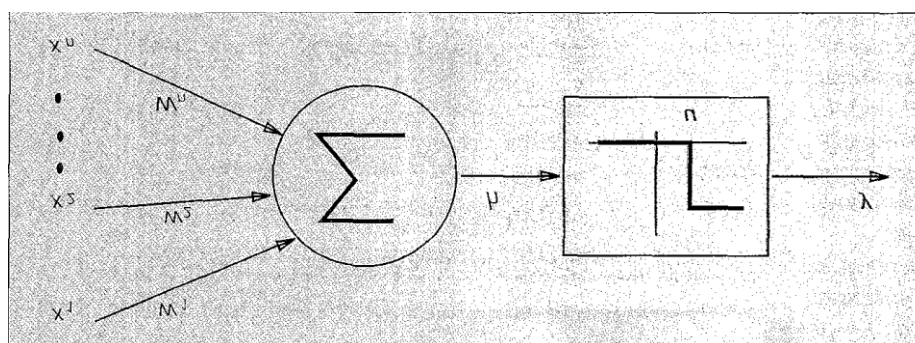
2.3. *Artificial neural networks*

Menurut Hammerstrom (1993), *artificial neural networks* merujuk pada teknologi komputasi yang dikembangkan berdasarkan cara kerja otak pada makhluk hidup. Seperti yang terlihat pada gambar 2.1, otak pada makhluk hidup terdiri atas kumpulan neuron yang saling terhubung, di mana setiap neuron memiliki soma, akson, sinapsis, dan dendrit. Melalui dendrit, sebuah neuron dapat terhubung ke ujung sinapsis dari neuron lainnya. Sinyal yang sampai ke neuron akan diproses di dalam soma sebagai pusat dari neuron, lalu diteruskan lagi melalui dendrit pada ujung lainnya. Dendrit juga dilengkapi oleh akson, yang berfungsi untuk transfer sinyal aktivasi ke neuron lainnya. Menurut Uhrig (1995), sebuah *artificial neural networks* terdiri dari kumpulan unit komputasi yang saling terhubung satu sama lain, melalui koneksi yang memiliki weight tertentu. Unit komputasi ini tersusun dalam rangkaian lapisan (*layer*) tertentu, yang pada umumnya terdiri dari tiga lapisan. Lapisan tersebut adalah *input layer*, *hidden layer*, dan *output layer*.



Gambar 2.1. Struktur neuron pada otak makhluk hidup (Sumber: Heaton, 2008)

Penelitian mengenai *artificial neural networks* dimulai ketika McCulloch & Pitts (1943) mengemukakan struktur neuron untuk melakukan kalkulasi logika. Seperti yang ditunjukkan oleh gambar 2.2, sebuah neuron dapat menerima n sinyal input. Sinyal input yang diterima oleh neuron akan dihitung sesuai fungsi yang berlaku pada neuron, yang kemudian dijumlahkan secara keseluruhan. Hasil penjumlahan yang didapat akan diuji berdasarkan fungsi batasan (*threshold function*). Sebuah neuron dinyatakan dalam kondisi aktif jika hasil penjumlahan sama atau melebihi fungsi batasan tersebut. Sebaliknya, jika hasil penjumlahan tidak melebihi fungsi batasan, maka neuron tersebut dinyatakan dalam kondisi tidak aktif.



Gambar 2.2. Struktur sebuah neuron pada *artificial neural network*

(Sumber: Jain *et al.*, 1996)

Penelitian mengenai peningkatan kinerja pada *artificial neural networks* dimulai ketika Rosenblatt (1958), yang kemudian dikembangkan oleh Rumelhart *et al.* (1986), memperkenalkan *backpropagation neural network*. Dalam *backpropagation neural network*, data yang diterima oleh sebuah node akan dikalkulasikan melalui fungsi aktivasi yang telah ditentukan. Selanjutnya, hasil kalkulasi, beserta nilai *error* yang didapat dari hasil kalkulasi tersebut, akan diteruskan ke neuron lainnya untuk diproses kembali.

2.4. Machine learning

Menurut Hertzmann & Fleet (2011), *machine learning* merujuk pada sebuah metode yang membuat komputer memiliki kemampuan dalam mempelajari dan melakukan sebuah pekerjaan secara otomatis. Proses *machine learning* akan dilakukan melalui

algoritma tertentu, sehingga pekerjaan yang diperintahkan kepada komputer dapat dilakukan secara otomatis.

Machine learning memiliki beberapa pengertian, yaitu sebagai berikut.

- a. *Machine learning* menurut sudut pandang kecerdasan buatan. Proses *learning* merupakan bagian yang penting dalam pembuatan sistem cerdas, namun sistem cerdas tidak dapat dibangun melalui aturan-aturan yang dirancang secara langsung, misalnya dengan menerapkan langkah-langkah prosedural, sehingga dibutuhkan proses yang memungkinkan sistem cerdas untuk mempelajari algoritma secara otomatis.
- b. *Machine learning* menurut sudut pandang teknis. *Machine learning* merupakan sebuah metode yang memungkinkan proses pemrograman komputer melalui sebuah sampel data tertentu.
- c. *Machine learning* menurut sudut pandang statistika. *Machine learning* merupakan suatu metode penyelesaian masalah yang bersifat statistik melalui teknik komputasi tertentu. *Machine learning* menyelesaikan masalah statistik melalui pendekatan yang sedikit berbeda, seperti lebih mengutamakan kecepatan pemrosesan data daripada akurasi hasil pemrosesan.

Machine learning dapat dilakukan melalui 2 fase, yaitu fase *training* dan fase *application*. Dalam fase *training*, model dari algoritma yang digunakan akan dipelajari oleh sistem melalui *training data*. Dalam fase *application*, model yang telah dipelajari sistem melalui fase *training* akan digunakan untuk menghasilkan sebuah keputusan tertentu, dengan menggunakan *testing data*.

Menurut van Heeswijk (2015), *machine learning* dapat dilakukan dengan dua cara, yaitu *supervised learning* dan *unsupervised learning*. Dalam *unsupervised learning*, pemrosesan *sample data* dilakukan tanpa mewajibkan hasil akhir memiliki bentuk yang sesuai dengan bentuk tertentu, dengan menggunakan beberapa *sample data* sekaligus. Penerapan *unsupervised learning* dapat ditemukan pada proses visualisasi, atau eksplorasi data. Sebaliknya, dalam *supervised learning*, *sample data x* akan diproses sedemikian rupa, sehingga menghasilkan bentuk keluaran yang sesuai dengan hasil akhir *y*. *Supervised learning* dapat diterapkan pada proses klasifikasi.

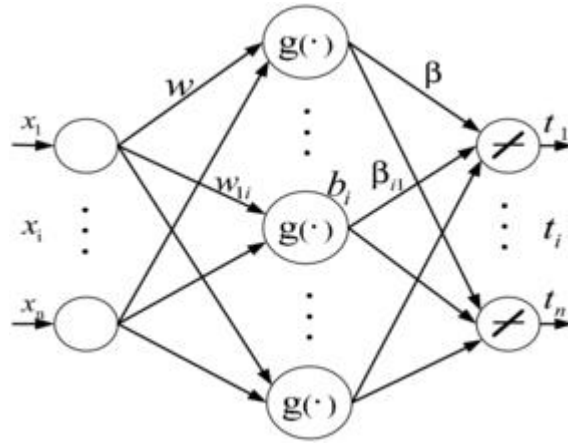
2.5. *Extreme learning machine (ELM)*

Extreme learning machine merupakan sebuah metode yang digunakan dalam jaringan saraf tiruan. Menurut Sun *et al.* (2008), *extreme learning machine* merupakan jaringan saraf tiruan yang bersifat *feedforward* dengan satu *hidden layer*. *Extreme learning machine* dikembangkan oleh Huang *et al.* (2006) untuk memperbaiki permasalahan *learning speed* pada metode-metode yang sebelumnya digunakan pada *feedforward neural networks*. *Feedforward neural networks* memiliki kelemahan dalam *learning speed* karena 2 hal berikut:

- a. Dalam proses *training*, *feedforward neural networks* menggunakan algoritma *slow gradient based learning*; dan
- b. *Input weight* dan *hidden bias* yang berhubungan antar *layer* pada jaringan saraf tiruan ditentukan secara berulang-ulang oleh algoritma *slow gradient based learning*.

Karena kedua hal tersebut, *feedforward neural networks* tidak hanya memiliki *learning speed* yang rendah, namun juga memiliki peluang yang tinggi untuk mencapai sebuah kondisi di mana jaringan saraf tiruan terjebak dalam *local minima* (Huang *et al.*, 2006). Hal ini diperbaiki pada ELM, di mana *input weight* dan *hidden bias* dipilih secara acak, sehingga dapat memperkecil *learning speed* dan bekerja dengan baik.

Struktur umum ELM dapat dilihat pada gambar 2.3. *Extreme learning machine* dijalankan pada *single hidden layer feedforward neural network*, yang digunakan oleh metode ELM. Dalam menjalankan proses *training*, ELM menggunakan teori invers matriks. Teori invers matriks yang digunakan ELM adalah teori *Moore-Penrose pseudoinverse*.



Gambar 2.3. Struktur umum ELM (Sumber: iopscience.iop.org)

Jika diketahui sebuah sampel N berupa kumpulan data (x_i, t_i) , di mana $x_i = [x_{i1}, x_{i2}, \dots, x_{in}]^T \in R^n$, dan $x_i = [t_{i1}, t_{i2}, \dots, t_{im}]^T \in R^m$, maka sebuah model *single hidden layer feedforward neural network* yang memiliki fungsi aktivasi $g(x)$ dan N buah neuron pada *hidden layer*, dapat didefinisikan melalui persamaan 2.1:

$$\sum_{i=1}^N \beta_i g(w_i \cdot x_j + b_i) = o_j, j = 1, \dots, N \quad (2.1)$$

di mana $w_i = [w_{i1}, w_{i2}, \dots, w_{in}]^T$ merupakan input weight pada vektor yang terhubung antara *hidden neuron* ke- i dan neuron masukan, $\beta_i = [\beta_{i1}, \beta_{i2}, \dots, \beta_{im}]^T$ merupakan weight dari vektor yang menghubungkan *hidden neuron* ke- i dan neuron keluaran, dan b_i merupakan nilai *bias* dari *hidden neuron* ke- i . Jika $\sum_{j=1}^N \|o_j - t_j\| = 0$, maka persamaan 2.1 dapat juga didefinisikan melalui persamaan 2.2:

$$\sum_{i=1}^N \beta_i g(w_i \cdot x_j + b_i) = t_j, j = 1, \dots, N \quad (2.2)$$

Berdasarkan persamaan kedua, maka persamaan N dapat didefinisikan melalui persamaan 2.3:

$$H\beta = T \quad (2.3)$$

di mana tiap elemen H , β , dan T didefinisikan oleh persamaan 2.4, 2.5, dan 2.6:

$$H(w_1, \dots, w_{\bar{N}}, b_1, \dots, b_{\bar{N}}, x_1, \dots, x_N) = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \cdots & g(w_{\bar{N}} \cdot x_1 + b_{\bar{N}}) \\ \vdots & \ddots & \vdots \\ g(w_1 \cdot x_N + b_1) & \cdots & g(w_{\bar{N}} \cdot x_N + b_{\bar{N}}) \end{bmatrix} \quad (2.4)$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\bar{N}}^T \end{bmatrix} \quad (2.5)$$

$$T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix} \quad (2.6)$$

Proses pelatihan (*training*) menggunakan extreme learning machine dimulai dengan memberikan training set $\aleph = \{(x_i, t_i) | x_i \in R^n, t_i \in R^m, i = 1, \dots, N\}$, fungsi aktivasi $g(x)$, dan \bar{N} sebagai jumlah *hidden neuron*. Langkah-langkah yang dilakukan dalam proses training menggunakan extreme learning machine adalah sebagai berikut:

- a. Menentukan *input weight* w_i dan *bias* b_i secara acak;
- b. Menghitung nilai dari matriks *hidden layer output* H ; dan
- c. Menghitung nilai dari matriks keluaran β , yang ditentukan oleh persamaan 2.7:

$$\beta = H^\dagger T \quad (2.7)$$

di mana H^\dagger merupakan hasil *pseudoinvers* dari matriks H , dengan H^\dagger adalah matriks G yang memenuhi persamaan 2.8:

$$AGA = A, GAG = G, (AG)^T = AG, (GA)^T = GA \quad (2.8)$$

2.6. Penelitian Terdahulu

Penelitian mengenai analisis kualitas lingkungan, terutama kualitas udara dan kualitas air, telah dilakukan oleh beberapa orang. Ayyalasomayajula *et al.* (2016) melakukan simulasi kualitas udara menggunakan *big data* melalui Apache Hadoop. Hasil penelitian menunjukkan peningkatan kinerja sebesar 20 hingga 25 % dalam proses simulasi data. Khan & Chai (2016) melakukan prediksi tingkat pencemaran air menggunakan *artificial neural network*. *Artificial neural network* yang digunakan berupa *perceptron* dengan satu *hidden layer*. Qiao *et al.* (2008) melakukan evaluasi kualitas air di Sungai Changjiang menggunakan *fuzzy logic*. Ming (2012) melakukan asesmen kualitas air menggunakan *backpropagation neural network*. Pengujian dilakukan terhadap 10 sampel, di mana nilai *error* semakin mendekati 0 untuk setiap pengulangan.

Selain itu, penelitian mengenai penerapan *extreme learning machine* (ELM) sudah dilakukan dalam berbagai bidang. Huang *et al.* (2013) menggunakan *extreme learning machine* untuk mendeteksi tumor liver berdasarkan citra yang ditangkap dari proses *CT scan*. Penelitian ini memberikan hasil deteksi dengan volumetric overlap error sebesar 67,15 %. Pangaribuan & Suharjito (2014) menggunakan *extreme learning machine* untuk diagnosis diabetes mellitus. Dalam penelitian ini, proses *training* membutuhkan waktu rata-rata 0,1687 detik, dibandingkan dengan 0,9040 detik dengan menggunakan *backpropagation*. Selain itu, *error rate* yang dihasilkan mencapai 0,4036, dibandingkan 0,9425 dengan menggunakan *backpropagation*.

Pada penelitian ini, proses prediksi dilakukan menggunakan *extreme learning machine*. Selain itu, data yang digunakan tidak disimpan melalui kluster tertentu, melainkan menggunakan *file* data tersendiri. Data tersebut akan diproses menggunakan *extreme learning machine*, sehingga menghasilkan grafik yang menggambarkan perkiraan tingkat kualitas air. Rincian singkat dari penelitian yang pernah dilakukan dapat dilihat pada Tabel 2.2.

Tabel 2.2. Penelitian Terdahulu

No.	Peneliti	Tahun	Judul	Keterangan
1	Ayyalasomayajula <i>et al.</i>	2016	Air quality simulations using big data programming models	Peningkatan kinerja sebesar 20-25 %
2	Khan & Chai	2016	Predicting and analyzing water quality using machine learning: a comprehensive model	Training error: 0,0192 hingga 0,0444, testing error: 0,0181 hingga 0,0470
3	Qiao <i>et al.</i>	2008	The application of fuzzy comprehensive evaluation on the water quality of Changjiang river	-
4	Ming	2012	A New Water Quality Assessment Method Based on BP Neural Network	<i>Error rate</i> semakin mendekati 0 untuk setiap pengulangan
5	Huang <i>et al.</i>	2013	Liver tumor detection using kernel-based <i>extreme learning machine</i>	Volumetric overlap error rata-rata 67,15 %
6	Pangaribuan & Suharjito	2014	Diagnosis of diabetes mellitus using <i>extreme learning machine</i>	<i>Error rate</i> 0,4036 dan kecepatan training rata-rata 0,1687 detik

BAB 3

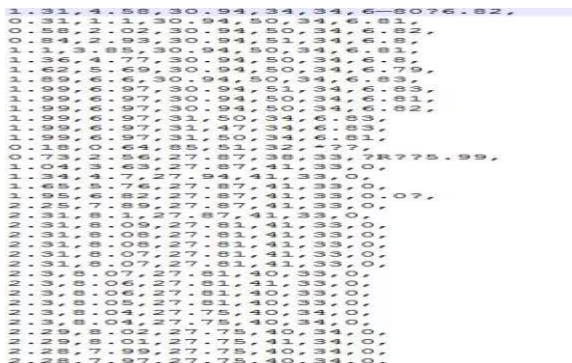
ANALISIS DAN PERANCANGAN

Bab ini membahas tentang implementasi *extreme learning machine* dalam proses prediksi kualitas air di Danau Toba. Bab ini juga membahas tentang data yang digunakan serta proses normalisasi data. Selain itu, bab ini juga membahas tentang proses *training* dan *testing* dari keluaran yang dihasilkan berdasarkan data yang dihasilkan dari proses normalisasi data.

3.1. Data yang Digunakan

Data yang digunakan dalam penelitian ini memiliki format CSV (*Comma Separated Value*), di mana nilai dari setiap parameter dipisahkan oleh sebuah karakter *semicolon* (titik koma), seperti yang ditunjukkan pada gambar 3.1. Data ini didapat dari penelitian yang dilakukan oleh Rahmat *et al.* (2016) mengenai pengukuran kualitas air di Danau Toba. Pengukuran dilakukan di empat lokasi, yaitu:

- Haranggaol Horison, Kabupaten Simalungun;
- Ambarita, Kabupaten Samosir;
- Ajibata, Kabupaten Toba Samosir; dan
- Parapat, Kabupaten Simalungun.

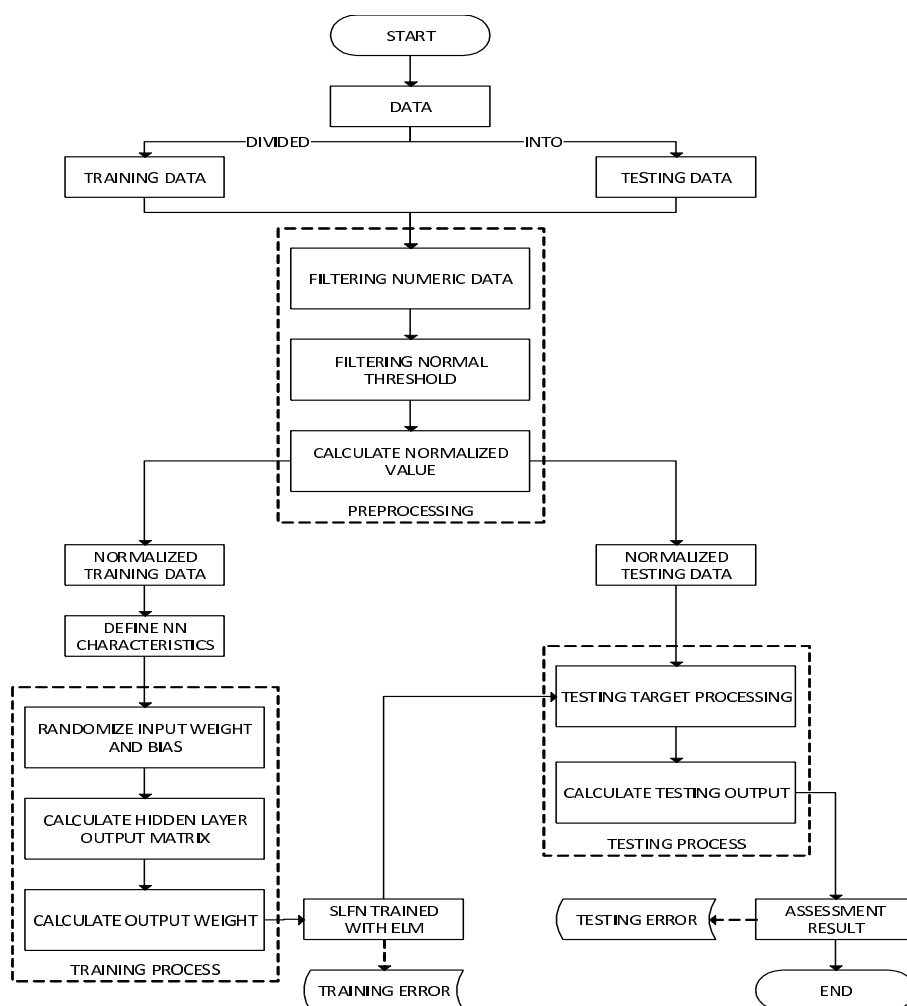


Gambar 3.1. Bentuk *dataset* yang dihasilkan dari penelitian Rahmat *et al.* (2016)

Data yang digunakan dalam penelitian ini merekam hasil pengukuran kualitas air melalui parameter fisika dan kimia pada air. Parameter tersebut antara lain suhu air, tingkat keasaman air, tingkat oksigen terlarut (*dissolved oxygen*), tingkat potensi reduksi oksidasi (*oxidation reduction potential*), suhu udara, dan kelembaban udara.

3.2. Analisis Sistem

Proses klasifikasi kualitas air Danau Toba berdasarkan data yang diperoleh dari Rahmat *et al.* (2016), dilakukan melalui beberapa langkah. Langkah-langkah tersebut ialah normalisasi data *training* dan data *testing*, penentuan jumlah neuron pada *hidden layer*, penentuan fungsi aktivasi, proses *training*, proses *testing*, dan proses validasi. Setiap langkah yang dilakukan akan dijelaskan dengan lebih terperinci pada bagian-bagian selanjutnya. Adapun arsitektur umum yang menggambarkan metode pada penelitian ini ditunjukkan pada Gambar 3.2.



Gambar 3.2. Arsitektur umum

3.2.1. Preprocessing

Tahap awal dari proses klasifikasi kualitas air Danau Toba pada penelitian ini dimulai dari proses preprocessing. Tahap preprocessing, yang juga dapat disebut sebagai tahap normalisasi, adalah sebuah proses di mana data disesuaikan sehingga memenuhi batasan nilai tertentu. (Shalabi *et al.*, 2006) Hasil dari proses normalisasi adalah kumpulan data yang dapat digunakan pada tahap selanjutnya. Pada penelitian ini, data yang diterima oleh sistem akan diolah terlebih dahulu, sehingga dihasilkan set data yang dapat digunakan oleh *extreme learning machine* dengan perbandingan 60:40. Ini berarti 60 % dari data yang diproses akan digunakan sebagai data *training*, sedangkan sisa dari data yang diproses akan digunakan sebagai data *testing*.

Pada penelitian ini, *preprocessing* dilakukan melalui tiga tahap. Tahap pertama dari langkah *preprocessing* pada penelitian ini adalah melakukan penyaringan terhadap tiap baris data, sehingga hanya data dengan nilai numerik yang akan digunakan untuk proses selanjutnya. Tahap ini dilanjutkan dengan menyaring data sesuai dengan batasan normal setiap parameter, sehingga dihasilkan data yang memiliki nilai yang sesuai dengan batasan setiap parameter. Tahap terakhir dari langkah *preprocessing* adalah menghitung nilai data dengan batasan $[C, D]$. Hasil dari proses preprocessing yang dilakukan adalah sebuah kumpulan data (dataset) yang sesuai dengan batasan yang ditentukan.

Metode normalisasi yang digunakan adalah *min-max normalization*, di mana normalisasi menghasilkan data yang memiliki batasan antara dua nilai tertentu (Patro & Sahu, 2015). Normalisasi akan dilakukan berdasarkan persamaan 3.1:

$$A' = \left(\frac{A - A_{min}}{A_{max} - A_{min}} \right) * (D - C) + C \quad (3.1)$$

di mana A' merupakan hasil normalisasi dari data bernilai A yang memiliki batasan antara $[C, D]$.

Hasil akhir dari langkah *preprocessing* pada penelitian ini adalah dataset latih dan dataset uji, yang telah disesuaikan pada rentang -1 hingga 1, sehingga dapat digunakan dalam proses *training*. Rentang nilai yang digunakan dalam proses kalkulasi dijelaskan pada Tabel 3.1. Penghitungan indeks kualitas air akan dilakukan berdasarkan kriteria yang dijelaskan pada Tabel 2.1.

Parameter *oxidation reduction potential* (ORP) yang digunakan dalam penelitian ini tidak diatur dalam Keputusan Menteri Negara Lingkungan Hidup Nomor 115 Tahun 2003 tentang Pedoman Penentuan Status Mutu Air. Sedangkan, menurut Lambrou *et al.* (2012), tingkat potensi oksidasi dan reduksi pada sumber daya air yang diperlukan untuk memenuhi syarat sebagai air minum adalah +650 hingga +800 mV. Karena itu, dalam proses *preprocessing*, hasil normalisasi untuk parameter ORP akan dihitung berdasarkan nilai normal tersebut.

Tabel 3.1. Nilai normal, minimum, dan maksimum dari setiap parameter kualitas air (Lambrou *et al.*, 2012; Peraturan Pemerintah Nomor 82 Tahun 2001)

Parameter	Satuan	Nilai normal	Nilai minimum	Nilai maksimum
<i>Dissolved Oxygen</i>	mg/L	$\geq 6,0$	0,0	18,0
Keasaman (pH)	-	6 – 9	0,1	14,0
ORP	mV	+650 - +800	-2.000,00	+2.000,00
Suhu air	$^{\circ}\text{C}$	deviasi 3	20,0	37,0
Suhu udara	$^{\circ}\text{C}$	-	20,0	37,0
Kelembaban udara	%	-	0,0	100,0

Tahap *preprocessing* akan menghasilkan dua buah kumpulan data (*dataset*), yaitu kumpulan data latih (*training*) dan data uji (*testing*). Setiap baris dari *dataset* yang dihasilkan dari tahap ini merepresentasikan target keluaran berupa hasil perhitungan indeks kualitas air menurut metode Storet, dan hasil pengukuran setiap sampel yang telah mengalami proses normalisasi sesuai rentang $[-1,1]$. Contoh dari beberapa baris data yang dihasilkan dari tahap preprocessing ditunjukkan oleh Gambar 3.3.

```

-21.000000 -1.000000 -1.000000 -0.682703 0.971775 0.058824 -0.120000
-15.000000 -1.000000 -1.000000 -0.678919 0.258814 0.058824 -0.140000
-15.000000 -1.000000 -1.000000 -0.678919 0.267604 -0.058824 -0.140000
-6.000000 -0.006667 -0.106000 -0.678919 0.274441 0.058824 -0.280000
-6.000000 0.003333 -0.097000 -0.675676 0.274441 0.058824 -0.300000
-6.000000 0.005556 -0.095000 -0.675676 0.275418 0.058824 -0.300000
-6.000000 0.010000 -0.091000 -0.678919 0.277371 0.058824 -0.280000
-6.000000 0.010000 -0.091000 -0.675676 0.277371 0.058824 -0.300000
-6.000000 0.011111 -0.090000 -0.678919 0.277371 0.058824 -0.300000
-6.000000 0.011111 -0.090000 -0.678919 0.276394 -0.176471 -0.140000
-6.000000 0.017778 -0.084000 -0.678919 0.276394 -0.058824 -0.140000
-6.000000 0.018889 -0.083000 -0.675676 0.277371 -0.058824 -0.140000
-6.000000 0.032222 -0.071000 -0.675676 0.277371 0.176471 -0.300000
-6.000000 0.025556 -0.077000 -0.678919 0.278347 -0.058824 -0.120000

```

Gambar 3.3. Contoh baris data pada *dataset* setelah proses normalisasi

3.2.2. Penentuan karakteristik artificial neural network

Pada tahap ini, ciri-ciri atau karakteristik dari *artificial neural network* yang akan digunakan pada *extreme learning machines*, akan ditentukan sebelum proses *training* dilakukan. Ciri-ciri atau karakteristik yang akan ditentukan pada penelitian ini adalah jumlah node pada *hidden layer* dan fungsi aktivasi yang digunakan pada setiap node.

3.2.2.1. Penentuan jumlah node pada hidden layer

Menurut Heaton (2008), penentuan jumlah node pada *artificial neural network*, khususnya pada *hidden layer*, merupakan hal yang penting sebelum menjalankan *training*, karena *hidden layer* berperan penting dalam penghitungan hasil akhir dari *artificial neural network*.

Jumlah node pada *hidden layer* yang tidak optimal dapat menyebabkan permasalahan tertentu pada proses *training*. *Hidden layer* yang memiliki jumlah node terlalu sedikit dapat menyebabkan kondisi *underfitting*, di mana node yang tersedia tidak dapat bekerja secara maksimal untuk mendeteksi sinyal yang diterima dari *input layer*. Sebaliknya, jumlah node yang terlalu banyak dapat berakibat pada bertambahnya waktu yang dibutuhkan *artificial neural network* untuk memproses data. Selain itu, jumlah node yang terlalu banyak juga dapat berakibat pada kondisi *overfitting*, di mana jumlah informasi yang diterima tidak cukup untuk diproses dalam *training* karena banyaknya kapasitas pemrosesan informasi yang dimiliki jaringan. Penentuan jumlah neuron pada *hidden layer* dapat memenuhi aturan-aturan berikut:

- a. Jumlah neuron pada *hidden layer* harus melebihi jumlah neuron di *input layer* dan tidak boleh melebihi jumlah neuron di *output layer*;
- b. Jumlah neuron pada *hidden layer* harus setara dengan $\frac{2}{3}$ (dua per tiga) dari jumlah neuron pada *input layer* dan *output layer*; dan
- c. Jumlah neuron pada *hidden layer* tidak boleh melebihi dua kali jumlah neuron pada *input layer*.

Aturan mengenai jumlah neuron pada *hidden layer* ini dapat digunakan sebagai pertimbangan. Walaupun begitu, proses menentukan jumlah neuron pada *hidden layer* merupakan proses *trial and error*. Hal ini dilakukan agar jaringan dapat menyesuaikan diri dengan masalah yang akan diselesaikan.

Pada penelitian ini, proses training akan dilakukan dengan jumlah neuron pada hidden layer bernilai n , di mana $n = 1, 2, 3, \dots, 98, 99, 100$. Hal ini dilakukan untuk mengetahui jumlah neuron pada *hidden layer* yang sesuai untuk proses prediksi.

3.2.2.2. Penentuan fungsi aktivasi

Tahap yang dilakukan setelah jumlah neuron pada *hidden layer* adalah menentukan fungsi aktivasi yang akan digunakan neuron dalam proses *training* dan proses *testing*. Fungsi aktivasi yang digunakan dalam penelitian ini adalah fungsi sigmoid, *sine*, *cosine*, dan *hardlim*.

Fungsi *sigmoid* merupakan fungsi aktivasi neuron yang umumnya digunakan pada algoritma *backpropagation*, yang bertujuan untuk mengurangi waktu komputasi (Karlik & Vehbi, 2011). Fungsi sigmoid pada *artificial neural network* didefinisikan oleh persamaan 3.2:

$$g(a, b, x) = \frac{1}{1 + \exp(-(a \cdot x + b))} \quad (3.2)$$

Fungsi *sine* sebagai fungsi aktivasi neuron pada *artificial neural network* didefinisikan oleh persamaan 3.3:

$$g(a, b, x) = \sin(a \cdot x + b) \quad (3.3)$$

Fungsi *cosine* sebagai fungsi aktivasi neuron pada *artificial neural network* didefinisikan oleh persamaan 3.4:

$$g(a, b, x) = \cos(a \cdot x + b) \quad (3.4)$$

Fungsi *hardlim* sebagai fungsi aktivasi neuron pada *artificial neural network* didefinisikan oleh persamaan 3.5:

$$g(a, b, x) = \begin{cases} 1, & \text{jika } a \cdot x - b \geq 0 \\ 0 & \end{cases} \quad (3.5)$$

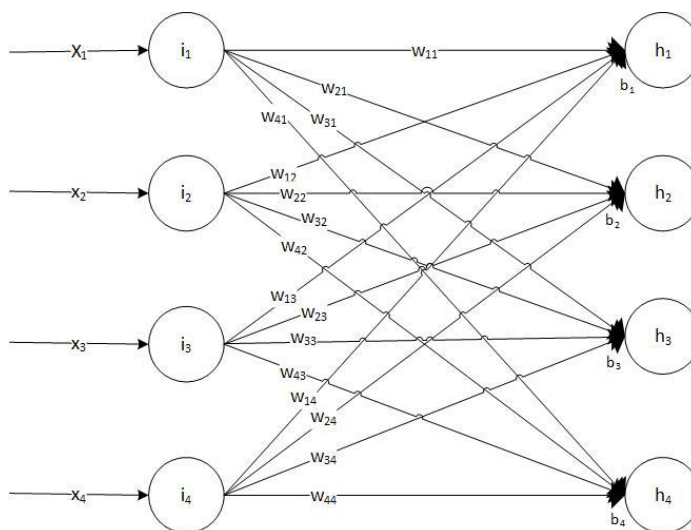
3.2.3. Proses training

Proses *training* merupakan tahap pertama yang dilakukan oleh *extreme learning machine* dalam menjalankan proses prediksi kualitas air. Proses *training* adalah sebuah proses di mana sebuah *artificial neural network* akan dilakukan konfigurasi untuk dapat menghasilkan keluaran yang dibutuhkan dengan memberikan sebuah kumpulan data tertentu. (Krose & van der Smagt, 1996) Hasil akhir dari proses ini adalah sebuah *artificial neural network* yang telah dilatih untuk memberikan hasil yang sesuai dengan data yang telah diberikan selama proses *training*.

Pada penelitian ini, sebuah *artificial neural network* dengan jumlah neuron pada hidden layer dan fungsi aktivasi neuron yang telah ditentukan pada langkah sebelumnya, akan dilatih menggunakan *extreme learning machine*. Proses *training* dilakukan melalui tiga tahap, yaitu pengacakan *input weight* dan *bias*, penghitungan *hidden layer output matrix*, dan penghitungan *output weight*. Proses *training* akan menghasilkan sebuah *artificial neural network* yang dapat digunakan untuk melakukan prediksi kualitas air.

3.2.3.1. Pengacakan input weight dan bias

Tahap pertama yang dilakukan dari proses *training* pada penelitian ini adalah pemberian nilai *input weight* dan *bias*. Jumlah neuron pada *input layer* yang digunakan pada penelitian ini disesuaikan dengan jumlah parameter yang diterima dari *dataset* yang digunakan. *Input weight* dan *bias* dari *artificial neural network* pada penelitian ini diberikan secara acak. Ilustrasi dari proses pengacakan *input weight* dan *bias* dapat dideskripsikan seperti pada Gambar 3.4.



Gambar 3.4. Ilustrasi *input weight* dan *bias*

3.2.3.2. Penghitungan *hidden layer output matrix*

Setelah tahap pengacakan *input weight* dan *bias* selesai dilakukan, tahap yang akan dilakukan adalah penghitungan *hidden layer output matrix*. *Hidden layer output matrix* merupakan hasil pengolahan dari masukan yang telah diterima oleh neuron pada *hidden layer* dari neuron pada *input layer*. Pengolahan dilakukan menggunakan fungsi aktivasi yang telah ditentukan pada tahap sebelumnya. Hasil dari proses ini adalah sebuah matriks H , yang sesuai dengan persamaan 2.4.

3.2.3.3. Penghitungan *output weight*

Penghitungan *output weight* dilakukan setelah proses penghitungan *hidden layer output matrix* telah selesai dilakukan. Pada tahap ini, *output weight* akan dihitung berdasarkan persamaan 2.7. Hasil dari proses ini adalah sebuah matriks yang merepresentasikan *weight* dari setiap neuron pada *output layer*.

3.2.4. Proses *testing*

Artificial neural network yang telah dihasilkan pada proses *training*, akan diuji pada proses *testing*. Proses *testing* dilakukan untuk mengetahui efektivitas dari metode *extreme learning machine* dalam melakukan proses prediksi kualitas air. Proses *testing* dilakukan melalui dua tahap, yaitu pengolahan target keluaran dan penghitungan keluaran.

3.2.4.1. Pengolahan target keluaran

Dalam proses ini, sistem akan mengambil nilai dari target keluaran yang terdapat pada data uji (*testing*). Target keluaran merupakan nilai pertama yang terdapat pada tiap baris dalam *dataset*.

3.2.4.2. Penghitungan keluaran

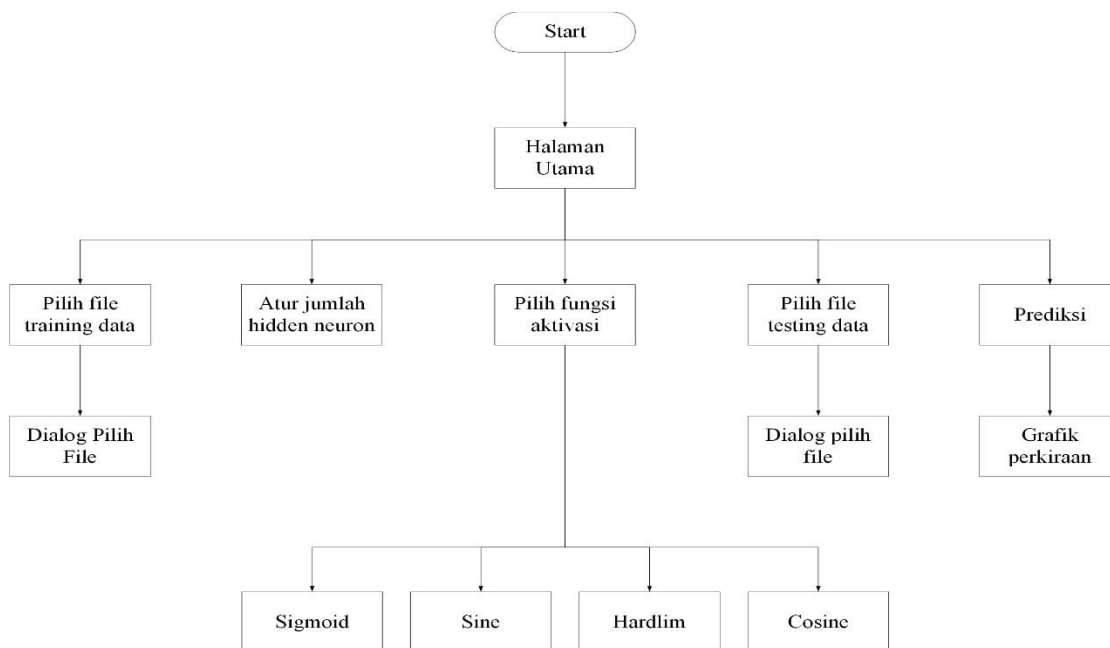
Setelah target keluaran berhasil didapatkan, tahap yang selanjutnya dilakukan adalah menghitung keluaran dari parameter yang diberikan. Proses penghitungan dilakukan menggunakan *artificial neural network* yang telah dilatih pada proses *training* sebelumnya. Hasil dari proses penghitungan akan diteruskan oleh sistem untuk ditampilkan dalam grafik yang menunjukkan pergerakan kualitas air dalam setiap kegiatan pengukuran.

3.3. Perancangan Antarmuka Sistem

Tampilan antarmuka sistem yang dibuat pada penelitian ini akan digunakan untuk proses pelatihan dan pengujian prediksi kualitas air.

3.3.1. Perancangan menu sistem

Struktur antarmuka yang akan dibuat pada sistem ditunjukkan oleh gambar 3.5.



Gambar 3.5. Struktur antarmuka sistem

3.3.2. Rancangan tampilan halaman utama

Halaman utama, yang merupakan tampilan awal dari aplikasi yang dirancang dalam penelitian ini, berisikan judul penelitian, nama peneliti, logo institusi, dan susunan menu. Menu yang ditampilkan dalam halaman utama aplikasi terdiri atas menu untuk memilih *file* data latih (*training*), mengatur jumlah *hidden neuron*, memilih fungsi aktivasi, memilih *file* data uji (*testing*), dan memulai proses prediksi, seperti yang ditunjukkan pada gambar 3.6. Rincian dari menu yang digunakan dalam aplikasi adalah sebagai berikut:

1. Menu pilih *file* data latih (*training*), adalah menu berupa kotak teks dan sebuah tombol “Pilih” untuk memilih *file* data latih, yang digunakan untuk memilih *file* data latih (*training*) yang akan digunakan dalam proses prediksi kualitas air. Tombol “Pilih” di dalam boks pilih *file* data latih akan membuka

sebuah dialog untuk memilih file. *File* yang dapat dipilih dalam dialog adalah *file* dengan ekstensi “.txt”.

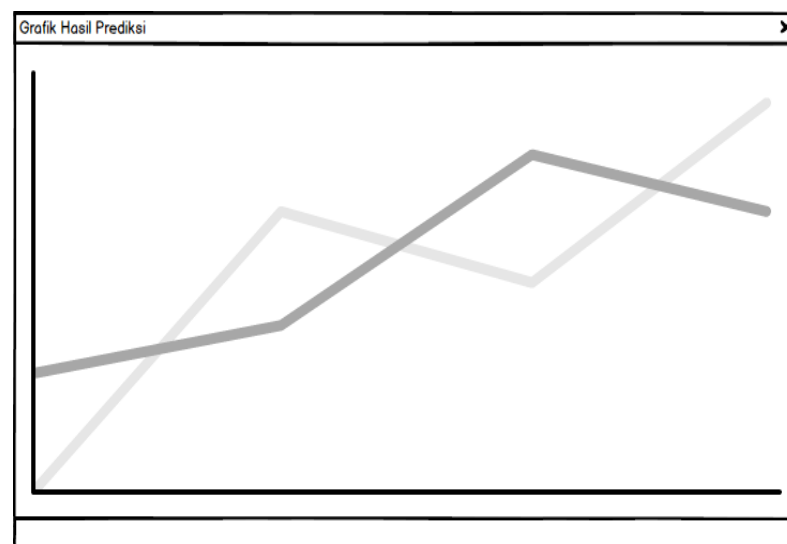
2. Menu atur jumlah hidden neuron, adalah menu berupa sebuah *text box* yang tidak dapat diubah nilainya dan dua buah tombol *plus* dan *minus*, yang berfungsi untuk mengatur jumlah neuron pada *hidden layer*. Nilai *default* dari jumlah neuron pada *hidden layer* dalam aplikasi ini adalah 1, dan dapat diubah dalam rentang 1 hingga 100.
3. Menu pilih fungsi aktivasi, adalah menu berupa sebuah *combo box*, yang digunakan untuk memilih fungsi aktivasi yang digunakan dalam proses prediksi kualitas air. Fungsi aktivasi yang dapat dipilih adalah fungsi *sigmoid*, fungsi *sine*, fungsi *hardlim*, dan fungsi *cosine*.
4. Menu pilih file data uji (testing), adalah menu berupa kotak teks dan tombol “Pilih”, yang akan digunakan untuk memilih *file* data uji (*testing*). Tombol “Pilih” pada menu ini akan membuka sebuah jendela dialog untuk memilih *file* data uji yang akan digunakan dalam proses prediksi. *File* data uji yang dapat digunakan dalam aplikasi adalah *file* dengan ekstensi “.txt”.
5. Menu mulai proses prediksi, adalah sebuah menu berupa tombol dengan label “Prediksi” pada aplikasi, yang akan memulai proses *training* dan proses *testing* menggunakan *extreme learning machine*.

The screenshot shows a Java Swing window titled "App". Inside, the title bar reads "Prediksi Kualitas Air di Danau Toba Menggunakan Extreme Learning Machine" and the author is "oleh: Eric Suwarno (121402071)". On the left is a circular "Logo institusi" placeholder. The main area contains two "File" selection sections: "File training" and "File testing", each with a text box labeled "Pilih file terlebih dahulu" and a "Pilih file" button. Below these is the "Karakteristik Neural Network" section, which includes a "Jumlah hidden neuron" field with a value of "1" and two "+" buttons, and a "Fungsi aktivasi" dropdown menu currently showing "sigmoid" with options for "sine", "hardlim", and "cosine". At the bottom is a large "Prediksi" button.

Gambar 3.6. Rancangan halaman utama aplikasi

3.4. Hasil Akhir

Hasil akhir yang didapatkan dari proses prediksi kualitas air pada penelitian ini adalah grafik yang menjelaskan perkiraan kualitas air untuk setiap baris data pada kumpulan data uji. Prediksi indeks kualitas air yang dihasilkan akan dibandingkan dengan indeks kualitas air yang didapat dari hasil pengukuran, untuk mengetahui tingkat akurasi dari hasil prediksi. Selain itu, juga akan ditampilkan grafik yang menunjukkan nilai error yang diperoleh pada saat proses training dan testing dalam setiap pengulangan. Ilustrasi dari grafik hasil perkiraan prediksi yang akan ditampilkan oleh aplikasi ditunjukkan oleh Gambar 3.7.



Gambar 3.7. Ilustrasi tampilan hasil prediksi

BAB 4

IMPLEMENTASI DAN PENGUJIAN

Bab ini akan membahas tentang hasil yang diperoleh dari implementasi metode yang diajukan, yaitu *extreme learning machine* (ELM), untuk melakukan proses prediksi kualitas air di Danau Toba. Bab ini akan menjabarkan hasil perancangan antarmuka yang digunakan dalam proses prediksi, prosedur operasional dari aplikasi yang dirancang, tahapan pengujian yang dilakukan, serta hasil yang didapat dari setiap pengujian.

4.1. Implementasi Sistem

Dalam penelitian ini, tahap *preprocessing* akan diimplementasikan ke dalam sistem menggunakan bahasa pemrograman Python, sedangkan tahap *training* dan *testing* serta visualisasi hasil pengujian akan diimplementasikan ke dalam sistem menggunakan bahasa pemrograman MATLAB.

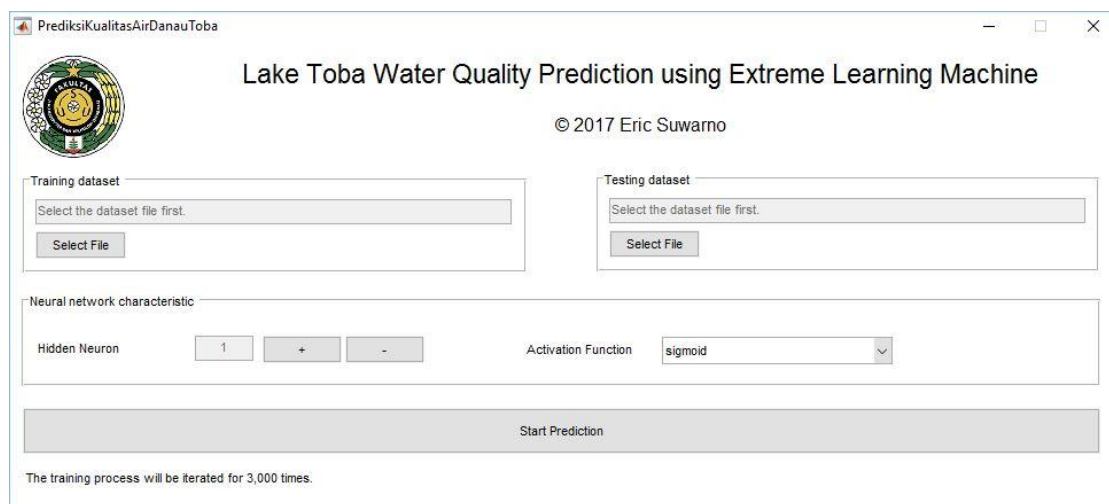
4.1.1. Spesifikasi perangkat keras dan perangkat lunak yang digunakan

Spesifikasi perangkat keras dan perangkat lunak yang digunakan untuk perancangan sistem pada penelitian ini adalah sebagai berikut:

1. Prosesor Intel(R) Core(TM) i5-3317U CPU @ 1.70 GHz
2. Kapasitas memori RAM sebesar 4 GB
3. Sistem operasi Windows 10 Pro 64-bit
4. *Hard drive* yang memiliki kapasitas sebesar 500 GB
5. *Software* yang digunakan adalah Python versi 2.7 dan MATLAB versi R2015a (8.5.0.197613)
6. *Library* yang digunakan adalah *library elm_train* dan *elm_predict* yang diterbitkan oleh Zhu *et al.* (2004)

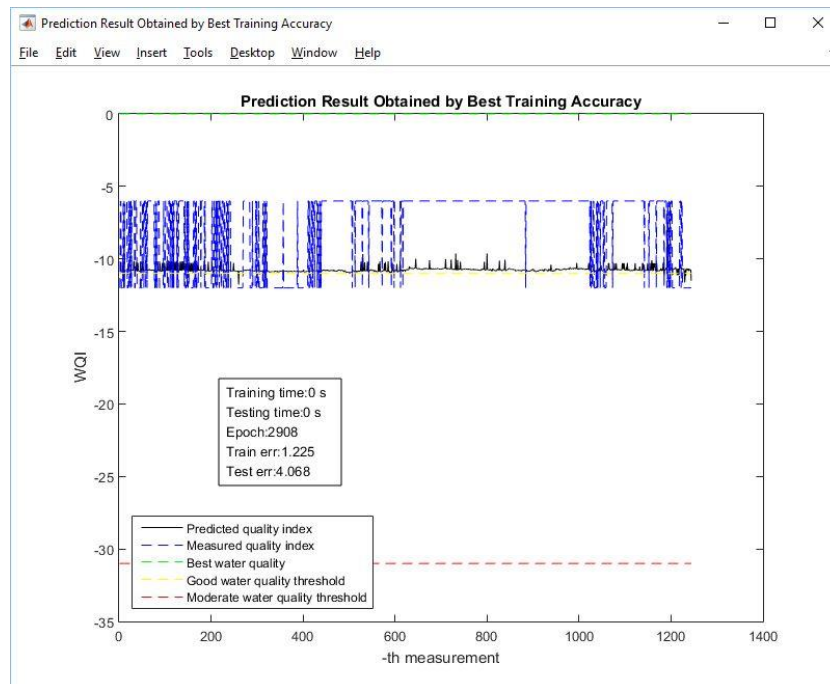
4.1.2. Implementasi perancangan antarmuka

Perancangan antarmuka sistem dibuat berdasarkan rancangan yang telah dipaparkan pada bab 3. Antarmuka aplikasi yang telah dirancang pada penelitian ini ditunjukkan oleh Gambar 4.1. Antarmuka yang dirancang terdiri dari halaman utama, di mana dalam halaman utama terdapat menu-menu yang digunakan untuk menjalankan proses prediksi kualitas air menggunakan *extreme learning machine*.



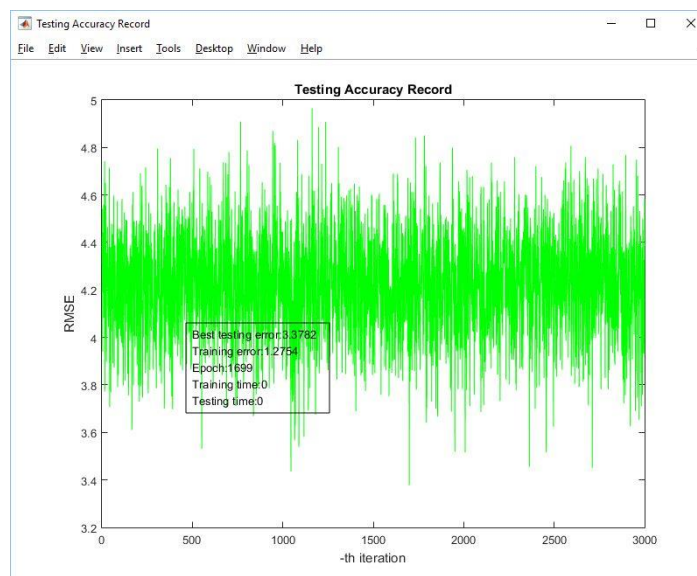
Gambar 4.1. Hasil rancangan antarmuka aplikasi

Setelah data latih, data uji, jumlah *hidden neuron*, dan fungsi aktivasi untuk proses prediksi telah diberikan kepada aplikasi, tahap *training* dan *testing* dalam proses prediksi akan dimulai, dan akan menghasilkan grafik hasil prediksi, yang menunjukkan pergerakan indeks kualitas air yang diperkirakan, dengan indeks kualitas air terukur sebagai perbandingan hasil prediksi. Ilustrasi dari grafik hasil prediksi yang akan ditampilkan setelah proses prediksi selesai ditunjukkan oleh Gambar 4.2. Grafik hasil prediksi akan ditampilkan pada dua jendela, berdasarkan *training error* dan *testing error* terendah. Grafik hasil prediksi juga menampilkan batas-batas tipe kualitas air, yaitu kualitas air tipe A (sangat baik), dengan nilai 0 skala Storet, kualitas air tipe B (baik), dengan batas bawah pada -11 skala Storet, dan kualitas air tipe C (sedang), dengan batas bawah pada -31 skala Storet.



Gambar 4.2. Ilustrasi tampilan grafik hasil prediksi

Selain itu, juga akan ditampilkan dua buah grafik, seperti yang ditunjukkan pada Gambar 4.3, yang menunjukkan nilai training error dan testing error yang diperoleh pada setiap perulangan. Grafik ini akan menunjukkan nilai *training error* dan *testing error* yang diperoleh dalam tiap perulangan dalam sebuah proses prediksi.



Gambar 4.3. Ilustrasi grafik *testing error*

4.1.3. Implementasi data

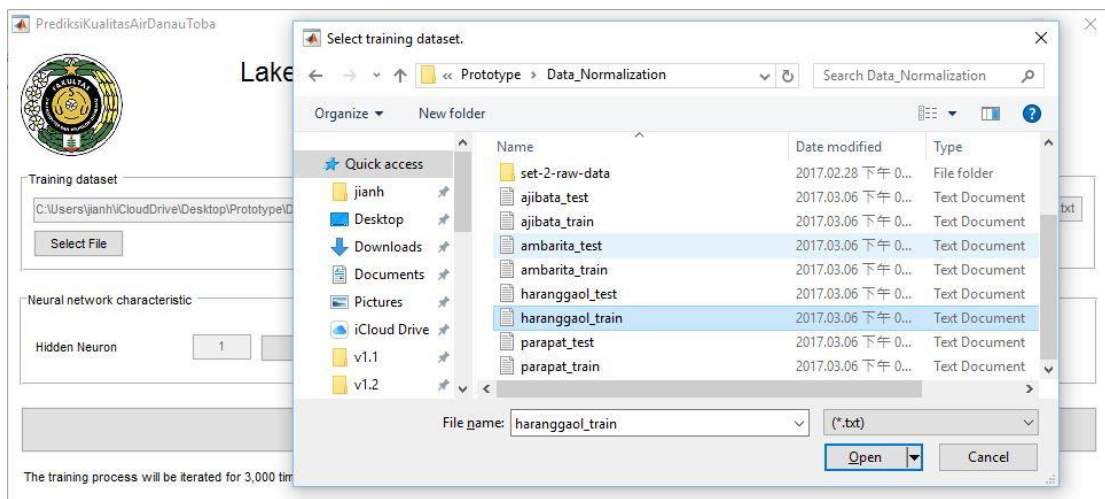
Data yang digunakan dalam penelitian ini bersumber dari penelitian yang dilakukan oleh Rahmat *et al.* (2016), di mana parameter-parameter kualitas air diukur dalam kurun waktu yang sedemikian rupa, dan disimpan dalam format dokumen teks, dengan hasil pengukuran setiap parameter dipisahkan oleh tanda titik koma. Pengukuran dilakukan pada beberapa lokasi, dan dikumpulkan dalam beberapa *file*. *File* tersebut akan diolah terlebih dahulu dalam tahap *preprocessing* sehingga dihasilkan *training dataset* dan *testing dataset* yang dapat digunakan oleh *extreme learning machine*. Perincian dari *file* yang digunakan dalam penelitian ini ditunjukkan oleh Tabel 4.1. Pemberian nama “*resume*” pada nama *file* awal menunjukkan bahwa proses pengukuran dilakukan pada waktu yang berbeda.

Tabel 4.1. Perincian data yang digunakan dalam penelitian

No.	Nama <i>file</i> awal	Lokasi	Jumlah baris data			
			Awal	Setelah penyaringan	Latih	Uji
1	DATA ajibata.txt	Ajibata	2203	2112	1268	844
2	DATA Haranggao1.txt	Haranggao1	6374	3532	2120	1412
3	DATA parapat.txt	Parapat	2446	1452	872	580
4	DATA parapat resume.txt					
5	DATA samosir.txt	Ambarita	6129	3113	1869	1244
6	DATA samosir resume.txt					

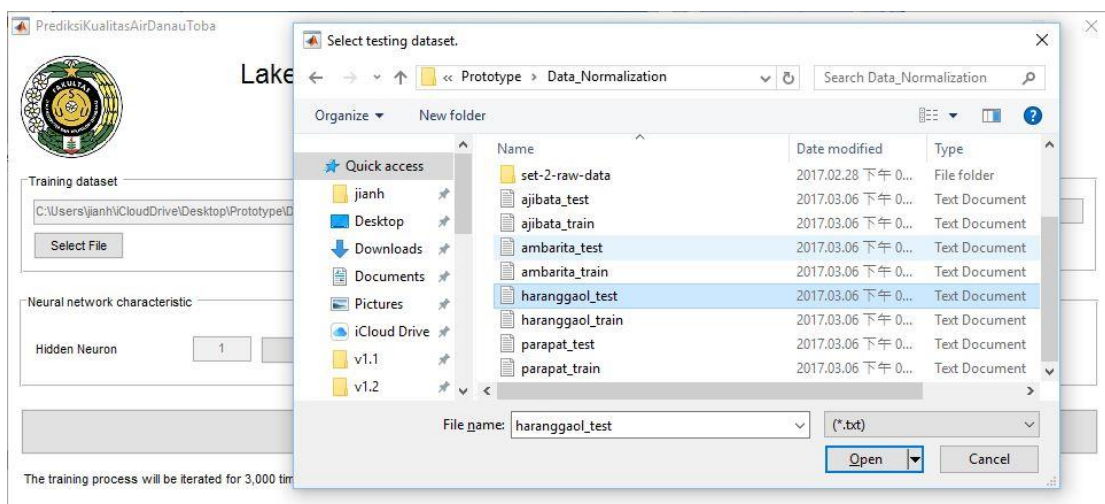
4.2. Prosedur Operasional

Bagian ini akan memaparkan prosedur yang harus dilakukan dalam menggunakan aplikasi yang dikembangkan dalam penelitian ini. Setelah antarmuka tampil di layar, pengguna dapat memilih *training dataset* (himpunan data latih) dengan memilih tombol “*Select File*” pada bagian “*Training dataset*”. Tombol “*Select File*” akan mengarahkan pengguna pada sebuah dialog pemilihan file, di mana file yang dapat dipilih di dalam dialog ini adalah dokumen teks dengan ekstensi “.txt”. Ilustrasi dari tahap pemilihan *training dataset* ditunjukkan oleh Gambar 4.4.



Gambar 4.4. Ilustrasi pemilihan *training dataset*

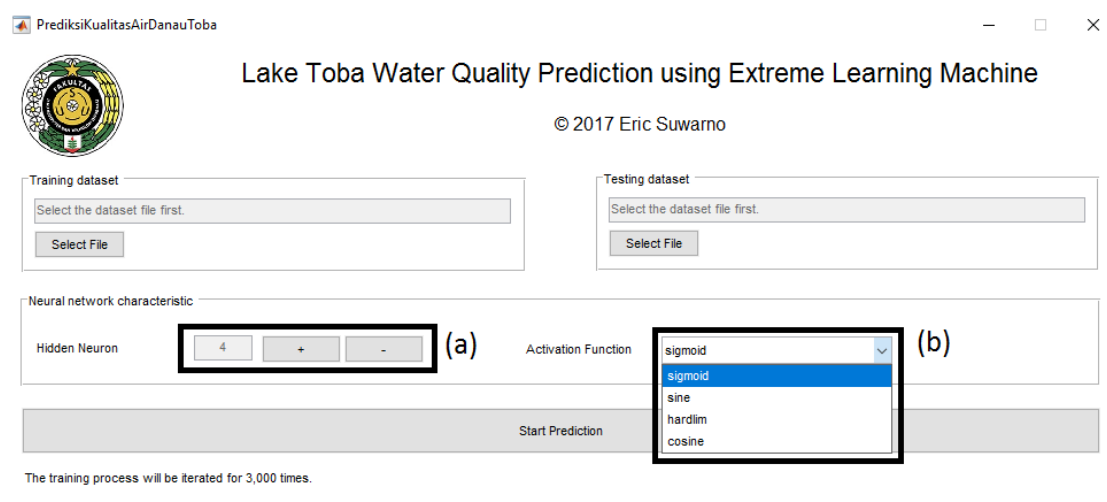
Setelah itu, pengguna juga dapat memilih *testing dataset* (himpunan data uji) dengan memilih tombol “Select File” pada bagian “Testing dataset”. Ketika dipilih, tombol ini akan memunculkan sebuah dialog pemilihan file, di mana pengguna dapat memilih *file* berupa dokumen teks dengan ekstensi “.txt” sebagai *testing dataset*. Ilustrasi dari tahap pemilihan testing dataset ditunjukkan oleh Gambar 4.5.



Gambar 4.5. Ilustrasi pemilihan *testing dataset*

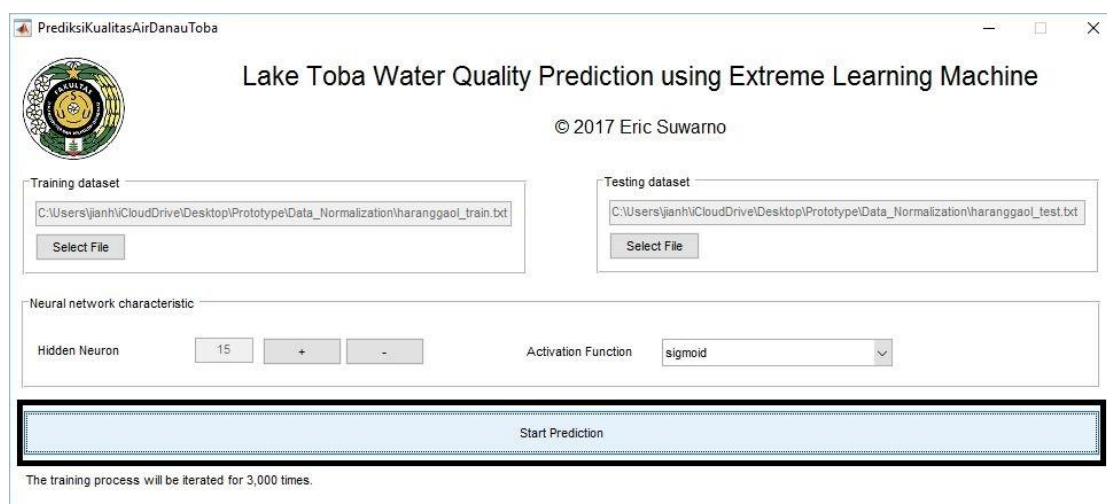
Setelah itu, pengguna dapat menentukan karakteristik dari *artificial neural network* yang akan digunakan dalam proses *training*, yaitu jumlah *hidden neuron* dan fungsi aktivasi neuron. Pengguna dapat mengubah jumlah *hidden neuron* melalui tombol *plus* dan *minus* yang terdapat pada bagian “Hidden Neuron”, sedangkan fungsi aktivasi yang digunakan oleh setiap neuron dapat dipilih dengan memilih salah satu

item yang terdapat dalam *combo box*. *Combo box* ini berada pada bagian “*Activation Function*”. Ilustrasi dari pemilihan hidden neuron dan fungsi aktivasi neuron ditunjukkan oleh Gambar 4.6.



Gambar 4.6. Ilustrasi pengaturan jumlah *hidden neuron* (a) dan fungsi aktivasi (b)

Setelah *training dataset*, *testing dataset*, dan karakteristik *artificial neural network* diberikan kepada sistem, pengguna dapat memulai proses prediksi dengan menekan tombol “*Start Prediction*”, seperti yang ditunjukkan pada Gambar 4.7. Sistem akan memulai proses prediksi, serta menampilkan grafik hasil prediksi berdasarkan *training error* dan *testing error* terendah.



Gambar 4.7. Ilustrasi tombol “*Start Prediction*” pada aplikasi

4.3. Hasil Pengujian

Bagian ini akan memaparkan hasil yang didapatkan dari implementasi *extreme learning machine* (ELM) dalam melakukan proses prediksi kualitas air di Danau Toba. Proses pengujian akan dilakukan pada setiap *dataset* dengan menggunakan parameter yang dijelaskan pada Tabel 4.2.

Tabel 4.2. Parameter *artificial neural network* pada pengujian setiap *dataset*

Pengujian ke-	Fungsi aktivasi	Jumlah neuron pada <i>hidden layer</i>
1	<i>Sigmoid</i>	15
2	<i>Sigmoid</i>	30
3	<i>Sine</i>	15
4	<i>Sine</i>	30
5	<i>Hardlim</i>	20
6	<i>Hardlim</i>	40
7	<i>Hardlim</i>	60
8	<i>Cosine</i>	15
9	<i>Cosine</i>	30

Pada setiap pengujian terhadap *dataset*, pengujian akan dilakukan sebanyak 3.000 kali perulangan. Tingkat akurasi yang diperoleh dari proses training dan testing dalam penelitian ini akan direpresentasikan melalui nilai *root mean square error* (RMSE). Nilai dari *root mean square error* (RMSE) dapat dihitung melalui persamaan 4.1:

$$RMSE = \sqrt{\frac{\sum (Y_{predicted}^i - Y_{expected}^i)^2}{N}} \quad (4.1)$$

di mana $Y_{predicted}^i$ adalah nilai hasil prediksi pada sampel ke- i , $Y_{expected}^i$ adalah nilai yang diharapkan pada sampel ke- i , $i = 1, 2, \dots, N$, dan N adalah jumlah sampel pengujian.

4.3.1. Hasil pengujian terhadap dataset Ajibata

Pengujian dilakukan terhadap *dataset* hasil pengukuran di Ajibata dengan menggunakan parameter *artificial neural network* yang ditentukan pada Tabel 4.2.

Training dataset hasil pengukuran di Ajibata memiliki 1268 baris data, sedangkan *testing dataset* hasil pengukuran di Ajibata memiliki 844 baris data. Hasil pengujian akan dipaparkan berdasarkan *training error* terbaik dan *testing error* terbaik yang diperoleh dalam setiap pengujian.

Training error terbaik yang diperoleh dalam setiap pengujian menggunakan *dataset* Ajibata dijelaskan pada Tabel 4.3. Menurut hasil pengujian pada tabel, diketahui bahwa penambahan jumlah neuron pada *hidden layer* mengakibatkan peningkatan terhadap tingkat akurasi terbaik yang dapat dihasilkan pada tahap *training*. Hal ini dapat dibuktikan dari perbandingan hasil pengujian yang dilakukan. Salah satu contoh dari hubungan antara jumlah neuron pada *hidden layer* dan *training error* terbaik dapat ditunjukkan oleh hasil pengujian pertama dan kedua pada Tabel 4.3. Parameter *artificial neural network* yang digunakan pada pengujian pertama adalah 15 neuron pada *hidden layer* dengan fungsi *sigmoid* sebagai fungsi aktivasi, sedangkan parameter *artificial neural network* yang digunakan pada pengujian kedua adalah 30 neuron pada *hidden layer* dengan fungsi *sigmoid* sebagai fungsi aktivasi. Pada pengujian pertama, *training error* terendah yang didapat adalah 0,12161 yang didapat pada perulangan ke-1353, dengan durasi *training* sebesar 0,046875 detik. Sedangkan pada pengujian kedua, *training error* terendah yang didapat adalah 0,053148 yang didapat pada perulangan ke-310, dengan durasi *training* sebesar 0,046875 detik.

Tabel 4.3. *Training error* terbaik dalam pengujian *dataset* Ajibata

Pengujian ke-	<i>Training error</i> terbaik				
	<i>Training error</i> (RMSE)	<i>Testing error</i> (RMSE)	<i>Epoch</i> ke-	Durasi <i>training</i> (s)	Durasi <i>testing</i> (s)
1	0,12161	1,0968	1353	0,046875	0,03125
2	0,053148	5,1126	310	0,046875	0,03125
3	0,11341	1,9972	123	0,0625	0,03125
4	0,05369	5,9416	1208	0,09375	0,046875
5	$2,2261 \times 10^{-14}$	$4,2301 \times 10^{-14}$	1582	0,046875	0,046875
6	$2,0764 \times 10^{-14}$	0,77258	2235	0,046875	0,046875
7	$1,9697 \times 10^{-14}$	1,3994	897	0,046875	0,03125
8	0,12742	0,99394	1394	0,078125	0,046875
9	0,055086	4,3224	125	0,0625	0,0625

Hasil pengujian yang dijelaskan pada Tabel 4.3 juga menunjukkan bahwa bertambahnya jumlah neuron pada *hidden layer* akan memperburuk kemampuan *artificial neural network* dalam melakukan prediksi. Hal ini dapat dibuktikan melalui bertambahnya *testing error* seiring dengan bertambahnya jumlah neuron pada *hidden layer*, yang menunjukkan penurunan tingkat akurasi pada proses prediksi.

Salah satu sampel dari hasil pengujian yang membuktikan pernyataan tersebut adalah pengujian kelima, keenam, dan ketujuh. Pengujian kelima, keenam, dan ketujuh menggunakan fungsi *hardlim* sebagai fungsi aktivasi dengan jumlah *hidden neuron* yang berbeda-beda. Jumlah *hidden neuron* yang digunakan pada pengujian kelima, keenam, dan ketujuh, masing-masing adalah 20, 40, dan 60 neuron. *Testing error* yang didapat pada saat mencapai *training error* terbaik dalam pengujian kelima adalah $4,2301 \times 10^{-14}$ dengan durasi *testing* sebesar 0,046875 detik. *Testing error* yang didapat pada saat mencapai *training error* terbaik dalam pengujian keenam adalah 0,77258 dengan durasi *testing* sebesar 0,046875 detik. Sedangkan pada pengujian ketujuh, *testing error* yang didapat saat mencapai *training error* terbaik adalah 1,3994 dengan durasi *testing* sebesar 0,03125 detik.

Hasil *testing error* terbaik yang didapat dari setiap pengujian menggunakan *dataset* Ajibata dijelaskan pada Tabel 4.4. Seperti yang ditunjukkan pada Tabel 4.3, hasil pengujian pada tabel menunjukkan bahwa jumlah *hidden neuron* yang digunakan pada *artificial neural network* mempengaruhi tingkat akurasi terbaik yang mampu diperoleh dari proses prediksi pada setiap pengujian. Hasil pengujian pada tabel juga menunjukkan bahwa *testing error* yang didapat pada setiap perulangan tidak selalu berkaitan dengan *training error* yang didapat. Sebagai perbandingan, pada pengujian ketiga, *training error* terbaik yang didapat adalah 0,11341 dengan *testing error* sebesar 1,9972 yang didapat pada perulangan ke-123. Sedangkan, *testing error* terbaik yang didapat pada pengujian ketiga adalah 0,076934 dengan *training error* sebesar 0,15285 yang didapat pada pengulangan ke-975.

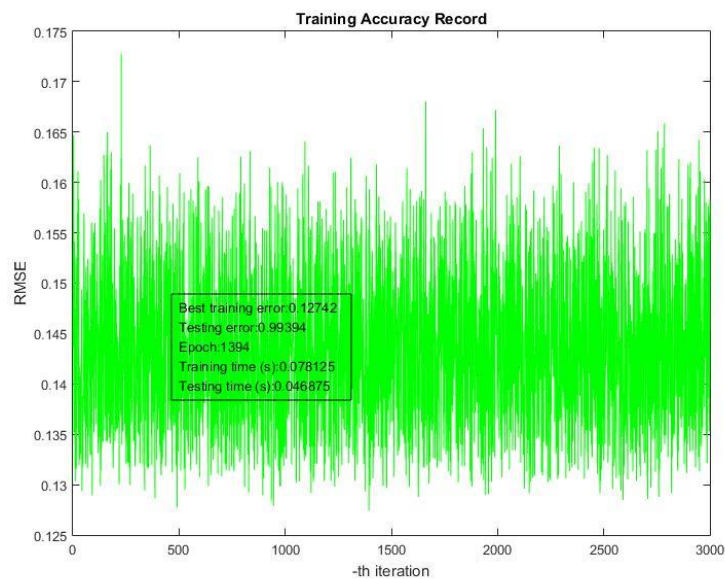
Tabel 4.4. *Testing error* terbaik dalam pengujian *dataset* Ajibata

Pengujian ke-	<i>Testing error</i> terbaik				
	<i>Training error</i> (RMSE)	<i>Testing error</i> (RMSE)	<i>Epoch</i> ke-	Durasi <i>training</i> (s)	Durasi <i>testing</i> (s)
1	0,16083	0,10341	382	0,046875	0,046875
2	0,12143	0,40694	1503	0,046875	0,046875
3	0,15285	0,076934	975	0,0625	0,03125
4	0,11242	0,36725	2822	0,046875	0,03125
5	$2,2261 \times 10^{-14}$	$4,2301 \times 10^{-14}$	1582	0,046875	0,046875
6	$2,039 \times 10^{-13}$	$1,9932 \times 10^{-13}$	1318	0,0625	0,03125
7	$1,0297 \times 10^{-13}$	$1,1631 \times 10^{-13}$	2366	0,0625	0,0625
8	0,16345	0,076473	2456	0,046875	0,046875
9	0,11972	0,32694	2899	0,0625	0,03125

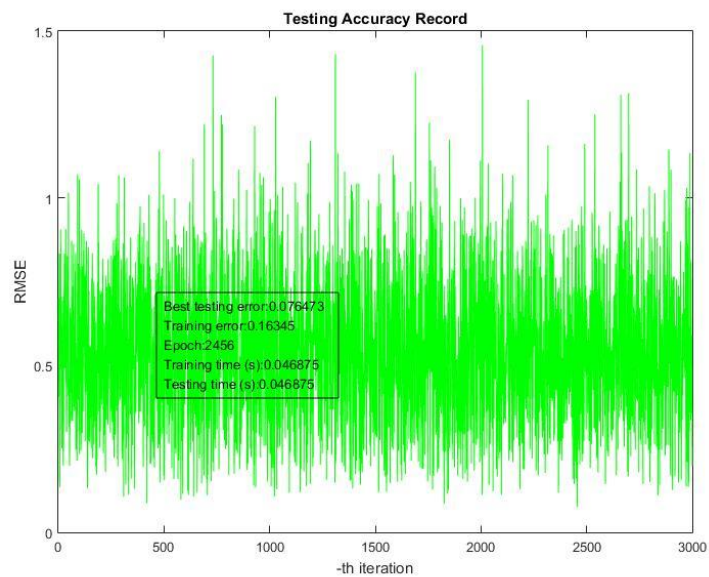
Hasil pengujian juga menunjukkan bahwa pada *extreme learning machine*, *training error* dan *testing error* yang diperoleh pada setiap pengulangan dipengaruhi oleh input weight dan bias yang dihasilkan dari proses training. Hal ini dikarenakan *input weight* dan *bias* pada *extreme learning machine* diperoleh melalui proses pengacakan.

Pengujian menggunakan *dataset* Ajibata menunjukkan bahwa *extreme learning machine* dapat melakukan pelatihan dan memberikan hasil prediksi dengan waktu komputasi yang singkat. Berdasarkan hasil pengujian pada Tabel 4.3, durasi *training* yang diperoleh ketika mencapai *training error* terbaik memiliki rentang antara 0,046875 hingga 0,09375 detik, dan durasi *testing* memiliki rentang antara 0,03125 detik hingga 0,0625 detik. Berdasarkan hasil pengujian pada Tabel 4.4, durasi *training* yang diperoleh ketika mencapai *testing error* terbaik memiliki rentang antara 0,046875 hingga 0,0625 detik, dan durasi *testing* memiliki rentang antara 0,03125 hingga 0,0625 detik.

Seperti yang ditunjukkan pada Gambar 4.8 dan Gambar 4.9 pada pengujian kedelapan, *training error* dan *testing error* cenderung berubah-ubah untuk setiap perulangan. Hal ini menyebabkan hasil pengujian terbaik dapat diperoleh dengan waktu yang lebih singkat.



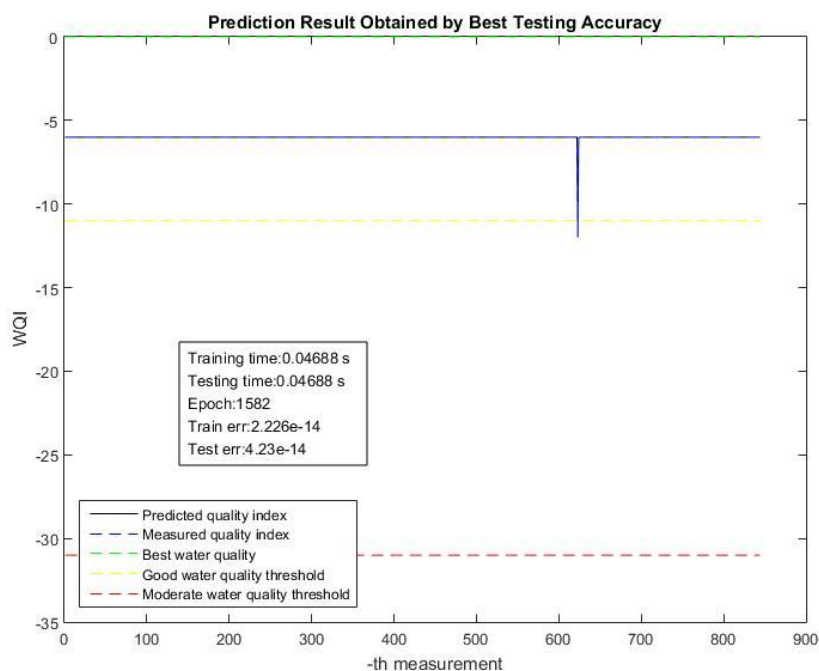
Gambar 4.8. Grafik *training error* dalam setiap perulangan yang diperoleh pada pengujian kedelapan menggunakan *dataset* Ajibata



Gambar 4.9. Grafik *testing error* dalam setiap perulangan yang diperoleh pada pengujian kedelapan menggunakan *dataset* Ajibata

Grafik hasil prediksi yang diperoleh ketika memperoleh *testing error* terbaik pada pengujian kelima terhadap *dataset* Ajibata ditunjukkan pada Gambar 4.10. Pada gambar ini, garis biru pada grafik menunjukkan indeks kualitas air yang didapat pada saat *preprocessing*, dan garis hitam pada grafik menunjukkan indeks kualitas air yang

didapat menggunakan *extreme learning machine*. Berdasarkan grafik pada Gambar 4.10, dapat diketahui bahwa indeks kualitas air yang terukur pada permukaan danau Toba di Ajibata memiliki rentang antara -6 hingga -11. Dengan hasil tersebut, dapat disimpulkan bahwa dengan mengacu pada Keputusan Menteri Negara Lingkungan Hidup Nomor 115 Tahun 2003 Tentang Pedoman Penentuan Status Mutu Air, kualitas air danau Toba yang terukur di wilayah Ajibata berada pada skala B (baik) hingga C (cukup).



Gambar 4.10. Grafik hasil prediksi saat memperoleh *testing error* terbaik pada pengujian kelima menggunakan *dataset* Ajibata

4.3.2. Hasil pengujian terhadap *dataset* Ambarita

Pengujian dilakukan terhadap *dataset* hasil pengukuran di Ambarita dengan menggunakan parameter *artificial neural network* yang ditentukan pada Tabel 4.2. *Training dataset* yang diperoleh dari hasil pengukuran di Ambarita memiliki 1869 baris data, sedangkan *testing dataset* hasil pengukuran di Ambarita memiliki 1244 baris data. Hasil pengujian akan dipaparkan berdasarkan *training error* terbaik dan *testing error* terbaik yang diperoleh dalam setiap pengujian.

Tabel 4.5 menunjukkan *training error* terbaik yang berhasil diperoleh dalam setiap pengujian menggunakan *dataset* Ambarita. Seperti hasil yang diperoleh pada pengujian menggunakan *dataset* Ajibata, hasil pengujian yang ditunjukkan pada tabel

menunjukkan bahwa penambahan jumlah *hidden neuron* mempengaruhi tingkat akurasi terbaik yang diperoleh pada saat *training*.

Tabel 4.5. *Training error* terbaik dalam pengujian *dataset* Ambarita

Pengujian ke-	<i>Training error</i> terbaik				
	<i>Training error</i> (RMSE)	<i>Testing error</i> (RMSE)	<i>Epoch</i> ke-	Durasi <i>training</i> (s)	Durasi <i>testing</i> (s)
1	1,20750	3,6577	1802	0,078125	0,046875
2	1,14940	3,5106	2507	0,062500	0,046875
3	1,20940	3,7979	918	0,062500	0,062500
4	1,15340	3,5509	2348	0,078125	0,062500
5	0,86710	2,1500	1347	0,062500	0,062500
6	0,85058	2,9877	1552	0,093750	0,093750
7	0,79288	4,7899	1836	0,093750	0,031250
8	1,21270	3,6513	514	0,078125	0,109380
9	1,15280	3,6166	1384	0,078125	0,062500

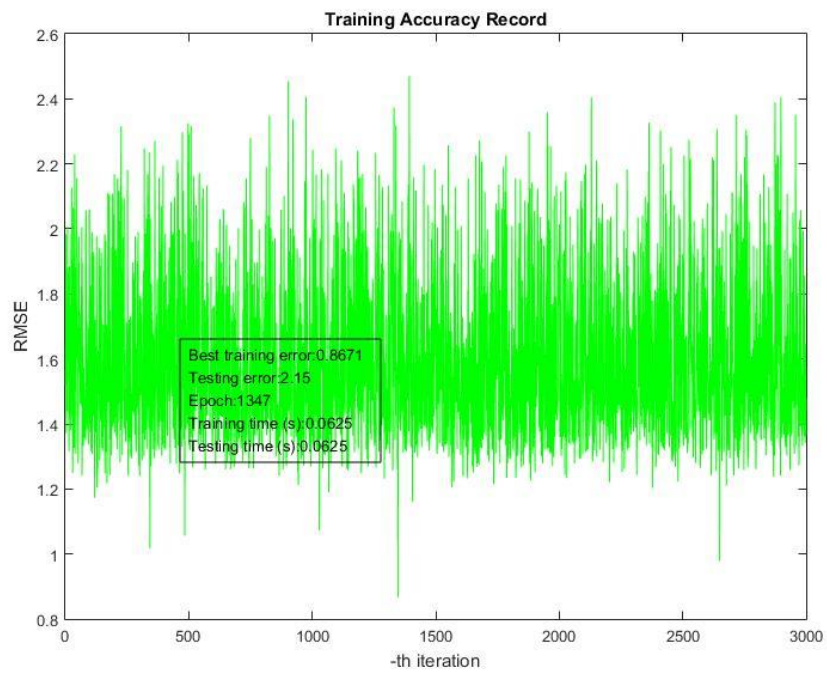
Dari hasil pengujian pada Tabel 4.5, diketahui juga bahwa dari empat fungsi aktivasi yang digunakan dalam pengujian, fungsi *hardlim* mampu memberikan hasil prediksi dengan tingkat *error* terendah. Hal ini dapat dibuktikan dari *training error* terendah yang diperoleh dari pengujian kelima, keenam, dan ketujuh menggunakan *dataset* Ambarita. *Training error* terbaik yang dihasilkan dari pengujian kelima, keenam, dan ketujuh lebih rendah dibandingkan dengan *training error* terbaik yang diperoleh pada pengujian lainnya.

Testing error terendah yang diperoleh dari pengujian menggunakan *dataset* hasil pengukuran di Ambarita ditunjukkan pada Tabel 4.6. Seperti pada *testing error* terendah yang diperoleh menggunakan *dataset* Ajibata, diketahui juga bahwa dari hasil pengujian *dataset*, *testing error* yang diperoleh tidak berhubungan erat dengan *training error* yang dihasilkan untuk setiap perulangan. Namun, *training error* yang lebih rendah akan meningkatkan kemungkinan untuk mendapatkan hasil prediksi dengan *testing error* yang lebih rendah, yang akan meningkatkan tingkat akurasi dari proses prediksi.

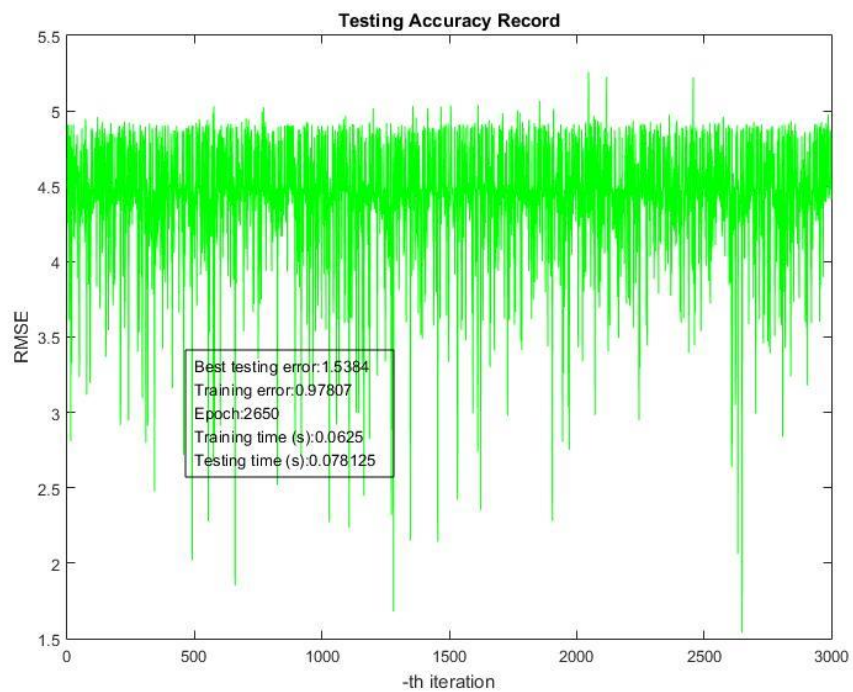
Tabel 4.6. *Testing error* terbaik dalam pengujian *dataset* Ambarita

Pengujian ke-	<i>Testing error</i> terbaik				
	<i>Training error</i> (RMSE)	<i>Testing error</i> (RMSE)	<i>Epoch</i> ke-	Durasi <i>training</i> (s)	Durasi <i>testing</i> (s)
1	1,27030	3,4397	1554	0,046875	0,031250
2	1,20190	3,3278	2654	0,062500	0,046875
3	1,23950	3,4364	2827	0,046875	0,046875
4	1,18190	3,3549	2056	0,062500	0,093750
5	0,97807	1,5384	2650	0,062500	0,078125
6	1,16880	1,3560	1315	0,093750	0,078125
7	1,12140	1,4468	274	0,125000	0,046875
8	1,24690	3,4402	2419	0,046875	0,031250
9	1,18810	3,3000	452	0,078125	0,031250

Seperti hasil pengujian terhadap *dataset* hasil pengukuran di Ajibata, nilai *training error* dan *testing error* yang diperoleh pada setiap perulangan dalam pengujian sangat dipengaruhi oleh nilai *input weight* dan *bias* yang diperoleh dari proses pengacakan. Sehingga, hasil dengan *testing error* terendah dapat diperoleh tanpa memerlukan waktu komputasi yang lebih lama, seperti yang ditunjukkan pada Gambar 4.10 dan 4.11, yang menunjukkan *training error* dan *testing error* yang diperoleh pada pengujian kelima. Dari Gambar 4.11 dan 4.12, dapat dibuktikan bahwa dalam pengujian kelima, hasil dengan *training error* terendah dapat diperoleh pada perulangan ke-1.347 dalam 3.000 perulangan. Begitu juga dengan hasil dengan *testing error* terendah, yang dapat diperoleh pada perulangan ke-2.650 dalam 3.000 perulangan.

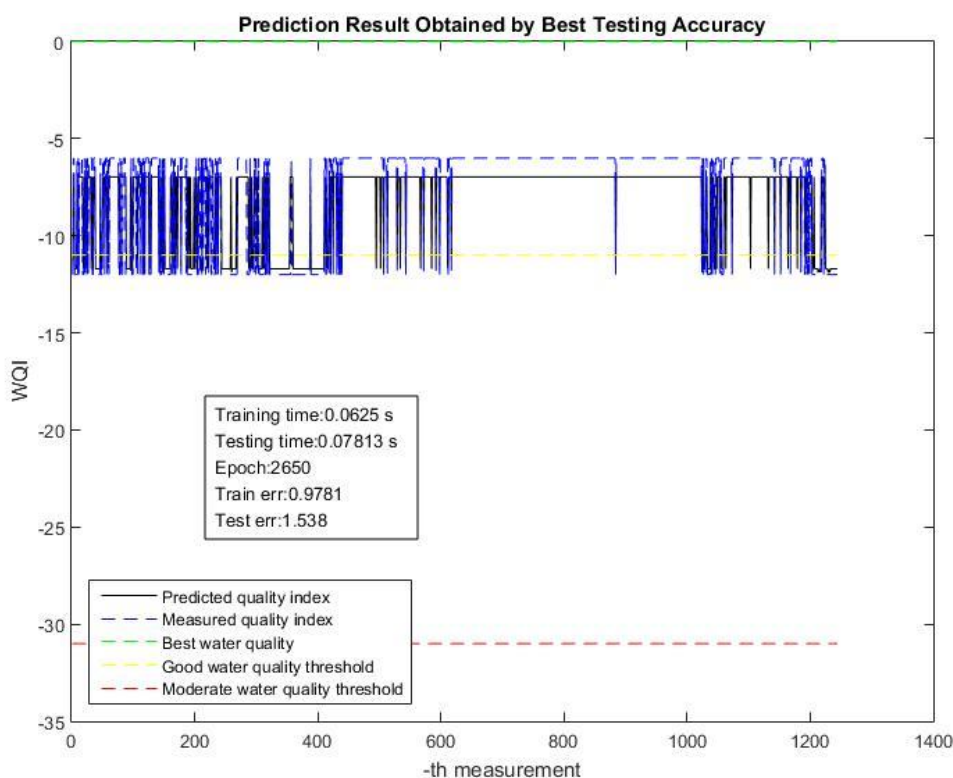


Gambar 4.11. Grafik *training error* yang diperoleh pada setiap perulangan dalam pengujian menggunakan *dataset* Ambarita



Gambar 4.12. Grafik *testing error* yang diperoleh pada setiap perulangan dalam pengujian menggunakan *dataset* Ambarita

Grafik yang terdapat pada Gambar 4.13 menunjukkan hasil prediksi yang diperoleh saat pengujian kelima menggunakan dataset Ambarita. Testing error terbaik yang diperoleh pada pengujian kelima menggunakan dataset ini adalah 1,538, yang didapat pada perulangan ke-2.650. Berdasarkan hasil prediksi yang diperoleh, diketahui bahwa indeks kualitas air yang diperoleh menggunakan *extreme learning machine* memiliki rentang antara -7 hingga -11 menurut metode perhitungan Storet. Karena itu, dapat disimpulkan bahwa dengan mengacu pada Keputusan Menteri Negara Lingkungan Hidup Nomor 115 Tahun 2003 Tentang Pedoman Penentuan Status Mutu Air, kualitas air danau Toba yang diukur di wilayah Ambarita bervariasi antara kelas B (baik) dan C (cukup).



Gambar 4.13. Grafik hasil prediksi saat memperoleh *testing error* terbaik pada pengujian kelima menggunakan *dataset* Ambarita

4.3.3. Hasil pengujian terhadap dataset Haranggaol

Pengujian dilakukan terhadap *dataset* hasil pengukuran di Ambarita dengan menggunakan parameter *artificial neural network* yang ditentukan pada Tabel 4.2. *Training dataset* Haranggaol pada penelitian ini memiliki 2120 baris data, sedangkan

testing dataset Haranggaol memiliki 1412 baris data. Hasil pengujian akan dipaparkan berdasarkan *training error* terbaik dan *testing error* terbaik yang diperoleh dalam setiap pengujian.

Pada Tabel 4.7 dijelaskan *training error* terbaik yang diperoleh dalam pengujian menggunakan *dataset* hasil pengukuran di Haranggaol. Training error terendah yang ditunjukkan pada tabel menunjukkan bahwa bertambahnya jumlah *hidden neuron* berkaitan dengan meningkatnya tingkat akurasi terbaik yang dihasilkan pada training, di mana dalam *training error* terbaik yang diperoleh menunjukkan penurunan. Hasil ini sesuai dengan hasil pengujian terhadap dataset Ajibata dan Ambarita, yang menyatakan bahwa peningkatan jumlah *hidden neuron* dapat meningkatkan tingkat akurasi pada tahap *training*.

Tabel 4.7. *Training error* terbaik dalam pengujian *dataset* Haranggaol

Pengujian ke-	<i>Training error</i> terbaik				
	<i>Training error</i> (RMSE)	<i>Testing error</i> (RMSE)	<i>Epoch</i> ke-	Durasi <i>training</i> (s)	Durasi <i>testing</i> (s)
1	1,1534	1,4118	2243	0,0625	0,09375
2	1,0785	5,6017	2905	0,078125	0,0625
3	1,1614	1,1454	1461	0,09375	0,078125
4	1,0812	3,5407	1885	0,078125	0,03125
5	1,3355	2,8366	822	0,09375	0,078125
6	1,1583	2,6632	339	0,09375	0,0625
7	1,1214	1,1182	1752	0,10938	0,09375
8	1,1521	2,3517	1460	0,078125	0,078125
9	1,0741	5,1926	2124	0,078125	0,09375

Hasil pengujian pada Tabel 4.7 juga menunjukkan bahwa *training error* yang rendah tidak memiliki hubungan secara langsung dengan rendahnya *testing error* yang dihasilkan. Hal ini dapat dilihat pada hasil pengujian ketiga dan keempat, yang menggunakan fungsi *sine* sebagai fungsi aktivasi pada neuron. Pengujian ketiga menggunakan 15 *hidden neuron*, sedangkan pengujian keempat menggunakan 30

hidden neuron. *Testing error* yang diperoleh ketika mencapai *training error* terendah pada pengujian keempat lebih tinggi daripada *testing error* yang diperoleh ketika mencapai *training error* terendah pada pengujian ketiga. Hal ini menunjukkan bahwa tingkat akurasi prediksi pada pengujian keempat lebih rendah dibandingkan dengan tingkat akurasi prediksi pada pengujian ketiga.

Testing error terbaik yang diperoleh dalam setiap pengujian menggunakan *dataset* hasil pengukuran di Haranggaol ditunjukkan pada Tabel 4.8. Berdasarkan hasil pengujian pada tabel, *training error* yang rendah tidak memastikan secara absolut bahwa *testing error* yang dihasilkan juga rendah. Namun, *training error* rendah yang diperoleh pada saat *training* akan meningkatkan peluang untuk menghasilkan sebuah prediksi dengan *testing error* yang rendah.

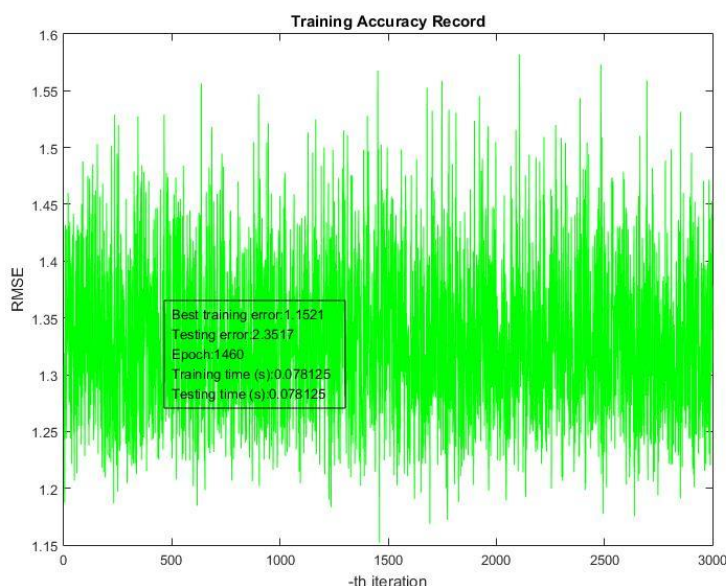
Tabel 4.8. *Testing error* terbaik pada pengujian *dataset* Haranggaol

Pengujian ke-	<i>Testing error</i> terbaik				
	<i>Training error</i> (RMSE)	<i>Testing error</i> (RMSE)	<i>Epoch</i> ke-	Durasi <i>training</i> (s)	Durasi <i>testing</i> (s)
1	1,5075	0,66059	743	0,09375	0,0625
2	1,1246	0,64494	1472	0,078125	0,078125
3	1,4945	0,63876	2269	0,0625	0,09375
4	1,1082	0,66183	2603	0,078125	0,078125
5	1,6723	0,57691	1238	0,10938	0,046875
6	1,5634	0,52108	500	0,0625	0,078125
7	1,6651	0,60052	2664	0,078125	0,09375
8	1,3521	0,65433	1924	0,078125	0,078125
9	1,142	0,70699	2900	0,078125	0,078125

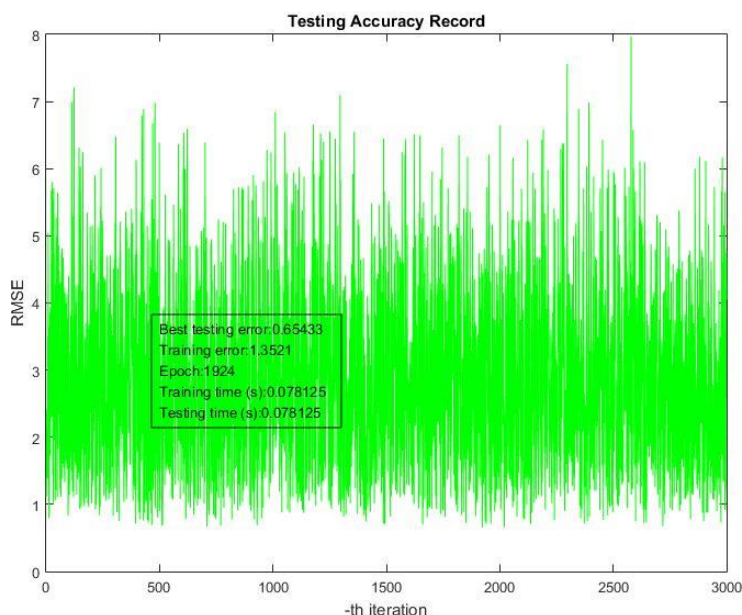
Hasil pengujian menggunakan *dataset* hasil pengukuran di Haranggaol menunjukkan bahwa *extreme learning machine* dapat memberikan hasil prediksi dengan waktu komputasi yang singkat. Berdasarkan Tabel 4.7, durasi *training* dan *testing* yang diperoleh memiliki rentang antara 0,03125 hingga 0,10938 detik.

Sedangkan menurut hasil pengujian pada Tabel 4.8, durasi *training* dan *testing* yang diperoleh memiliki rentang antara 0,046875 hingga 0,10938 detik.

Gambar 4.14 dan 4.15 menunjukkan *training error* dan *testing error* yang diperoleh pada tiap pengulangan dalam pengujian kedelapan dalam proses prediksi menggunakan *dataset* hasil pengukuran di Haranggaol. Sesuai dengan hasil pengujian yang diperoleh menggunakan *dataset* hasil pengukuran di Ajibata dan Ambarita, hasil pengujian menggunakan *dataset* hasil pengukuran di Haranggaol menunjukkan bahwa penerapan *extreme learning machine* pada proses pelatihan menyebabkan *training error* yang dihasilkan dalam proses *training* dipengaruhi oleh *input weight* dan *bias* yang diperoleh, di mana *input weight* dan *bias* dihasilkan secara acak. Hal ini juga menyebabkan perubahan *testing error* dalam setiap perulangan.

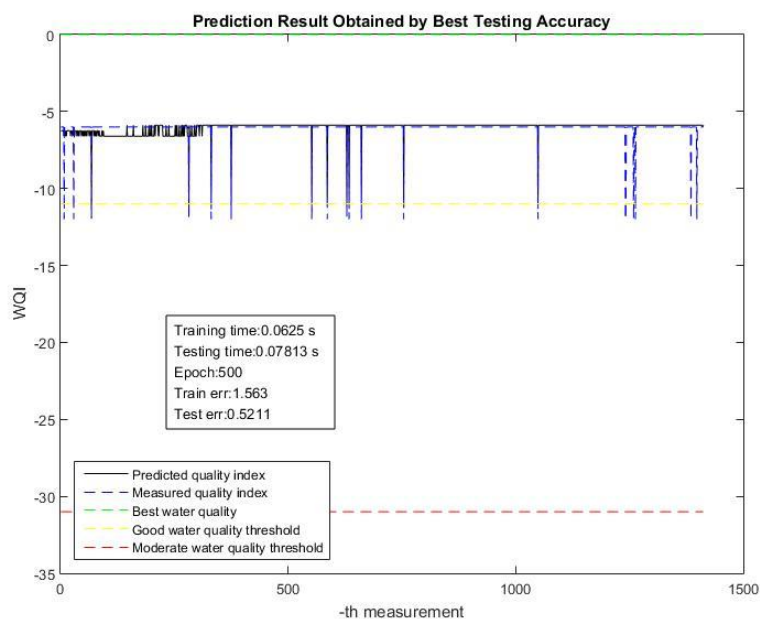


Gambar 4.14. Grafik *training error* yang diperoleh pada setiap perulangan dalam pengujian kedelapan menggunakan *dataset* Haranggaol



Gambar 4.15. Grafik *testing error* yang diperoleh pada setiap perulangan dalam pengujian kedelapan menggunakan *dataset* Haranggaol

Grafik yang ditunjukkan pada Gambar 4.16 menampilkan hasil prediksi yang diperoleh pada pengujian keenam menggunakan *dataset* Haranggaol. Garis biru pada grafik menunjukkan indeks kualitas air yang diperoleh pada tahap *preprocessing*, sedangkan garis hitam pada grafik menunjukkan hasil prediksi yang diperoleh menggunakan *extreme learning machine*. Berdasarkan hasil yang ditunjukkan pada grafik, diketahui bahwa indeks kualitas air yang diperoleh menggunakan *extreme learning machine* memiliki rentang antara -6 hingga -11 menurut metode pengukuran Storet. Dengan itu, dapat disimpulkan bahwa dengan mengacu pada Keputusan Menteri Negara Lingkungan Hidup Nomor 115 Tahun 2003 Tentang Pedoman Penentuan Status Mutu Air, kualitas air danau Toba yang terukur di wilayah Haranggaol bervariasi antara kelas B (baik) dan kelas C (cukup).



Gambar 4.16. Grafik hasil prediksi saat memperoleh *testing error* terbaik pada pengujian keenam menggunakan *dataset* Haranggaol

4.3.4. Hasil pengujian terhadap *dataset* Parapat

Pengujian dilakukan terhadap *dataset* hasil pengukuran di Parapat dengan menggunakan parameter *artificial neural network* yang ditentukan pada Tabel 4.2. *Training dataset* Parapat pada penelitian ini memiliki 872 baris data, sedangkan *testing dataset* Parapat memiliki 580 baris data. Hasil pengujian akan dipaparkan berdasarkan *training error* terbaik dan *testing error* terbaik yang diperoleh dalam setiap pengujian.

Pada Tabel 4.9 dijelaskan hasil *training error* terbaik yang diperoleh pada setiap pengujian menggunakan *dataset* hasil pengukuran di Parapat. Sesuai dengan hasil pengujian terhadap *dataset* hasil pengukuran di Ajibata, Ambarita, dan Haranggaol, hasil pengujian terhadap *dataset* hasil pengukuran di Parapat menunjukkan bahwa penambahan jumlah *hidden neuron* berdampak pada penurunan *training error*, yang menunjukkan bahwa akurasi yang diperoleh pada proses *training* lebih tinggi. Namun, penurunan *training error* yang dihasilkan dari penambahan jumlah *hidden neuron* tidak berakibat pada penurunan *testing error*, yang menunjukkan bahwa peningkatan akurasi *training* tidak secara absolut menyatakan bahwa akurasi hasil prediksi, yang diperoleh pada saat *testing*, juga mengalami peningkatan.

Tabel 4.9. *Training error* terbaik dalam pengujian *dataset* Parapat

Pengujian ke-	<i>Training error</i> terbaik				
	<i>Training error</i> (RMSE)	<i>Testing error</i> (RMSE)	<i>Epoch</i> ke-	Durasi <i>training</i> (s)	Durasi <i>testing</i> (s)
1	0,3769	3,0576	848	0,0625	0,046875
2	0,17352	3,4329	735	0,0625	0,0625
3	0,31253	2,6038	2142	0,09375	0,078125
4	0,17505	3,548	855	0,046875	0,03125
5	$8,6368 \times 10^{-14}$	0,69024	1196	0,046875	0,03125
6	$4,8924 \times 10^{-14}$	0,70466	30	0,046875	0,046875
7	$3,6388 \times 10^{-14}$	0,51141	612	0,046875	0,015625
8	0,36496	2,4245	1980	0,046875	0,0625
9	0,17532	6,0092	2142	0,0625	0,03125

Training error terbaik yang diperoleh pada pengujian menggunakan *dataset* Parapat sesuai Tabel 4.9 juga menunjukkan bahwa dari sembilan pengujian yang dilakukan, pengujian kelima, keenam, dan ketujuh mampu memberikan *training error* terbaik, di mana pengujian kelima, keenam, dan ketujuh memberikan *training error* terbaik dengan nilai masing-masing $8,6368 \times 10^{-14}$, $4,8924 \times 10^{-14}$, dan $3,6388 \times 10^{-14}$. Ketiga pengujian tersebut dilakukan dengan menggunakan fungsi *hardlim* sebagai fungsi aktivasi pada setiap neuron. Jumlah *hidden neuron* yang digunakan pada pengujian kelima, keenam, dan ketujuh, masing-masing adalah 20, 40, dan 60 neuron.

Dari hasil pengujian yang ditunjukkan pada Tabel 4.9, juga diketahui bahwa proses prediksi menggunakan *extreme learning machine* memiliki durasi training dan testing yang singkat. Durasi training pada saat mencapai *training error* terbaik memiliki rentang antara 0,046875 hingga 0,09375 detik. Sedangkan, durasi *testing* pada saat mencapai *training error* terbaik memiliki rentang antara 0,015625 hingga 0,078125 detik.

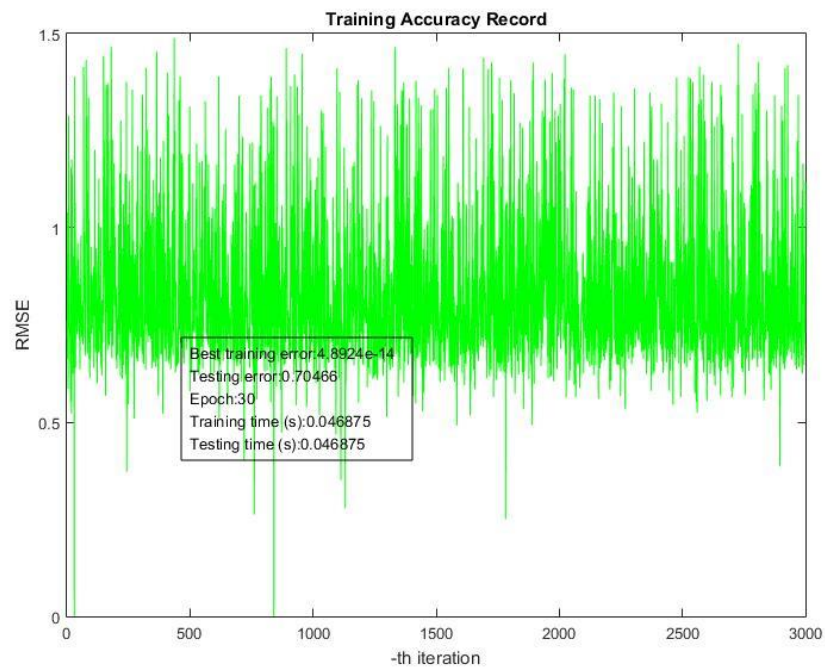
Testing error terbaik yang diperoleh dari pengujian menggunakan dataset hasil pengukuran di Parapat ditunjukkan pada Tabel 4.10. Hasil pengujian menunjukkan bahwa dengan menggunakan *extreme learning machine*, proses prediksi dapat

dilakukan dengan *testing error* terbaik yang diperoleh pada pengujian keenam dan ketujuh. Pengujian keenam dan ketujuh menggunakan fungsi *hardlim* sebagai fungsi aktivasi pada neuron, dengan jumlah *hidden neuron* masing-masing sebesar 40 dan 60 neuron. *Testing error* terbaik yang dihasilkan adalah 0,24914, di mana nilai *testing error* tersebut diperoleh pada perulangan ke-841 pada pengujian keenam, dan perulangan ke-1.510 pada pengujian ketujuh.

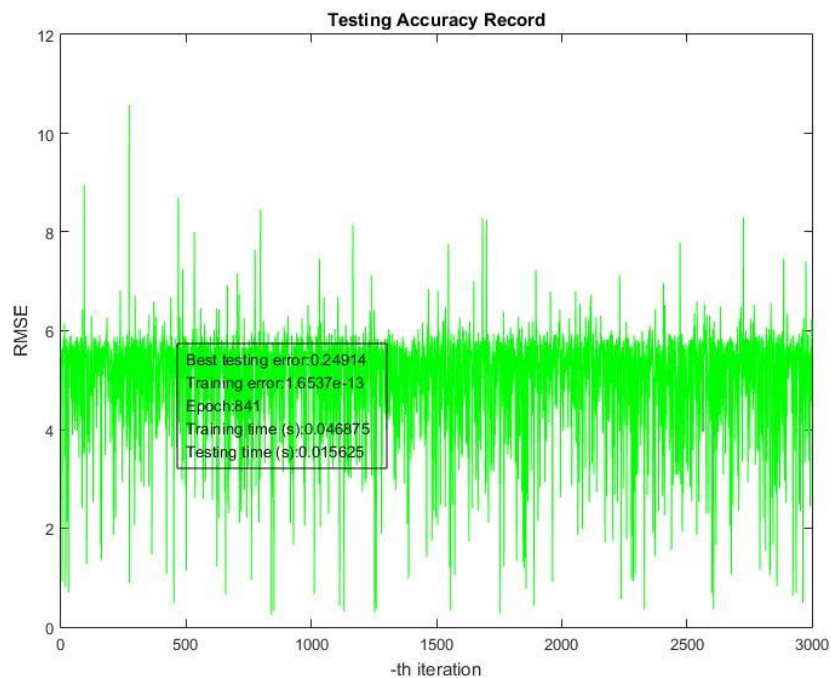
Tabel 4.10. *Testing error* terbaik dalam pengujian *dataset* Parapat

Pengujian ke-	<i>Testing error</i> terbaik				
	<i>Training error</i> (RMSE)	<i>Testing error</i> (RMSE)	<i>Epoch</i> ke-	Durasi <i>training</i> (s)	Durasi <i>testing</i> (s)
1	0,4866	0,62242	2922	0,078125	0,0625
2	0,2593	0,88564	2901	0,046875	0,0625
3	0,46413	0,55857	1849	0,078125	0,046875
4	0,32993	0,97932	756	0,03125	0,03125
5	0,687	0,25066	1369	0,046875	0,03125
6	$1,6537 \times 10^{-13}$	0,24914	841	0,046875	0,015625
7	$2,6007 \times 10^{-13}$	0,24914	1510	0,03125	0,046875
8	0,51283	0,57402	1641	0,0625	0,03125
9	0,23654	0,92673	2317	0,0625	0,015625

Hasil pengujian menggunakan *dataset* hasil pengukuran di Parapat menunjukkan bahwa penerapan *extreme learning machine* dalam proses prediksi kualitas air menyebabkan diperolehnya *training error* dan *testing error* yang berbeda-beda untuk setiap perulangan dalam sebuah pengujian. Hal ini dikarenakan *input weight* dan *bias*, yang dihasilkan secara acak, mempengaruhi proses *training* dan *testing* dalam setiap perulangan. Kondisi ini diperlihatkan pada Gambar 4.17 dan Gambar 4.18, yang diperoleh pada pengujian keenam menggunakan *dataset* Parapat.

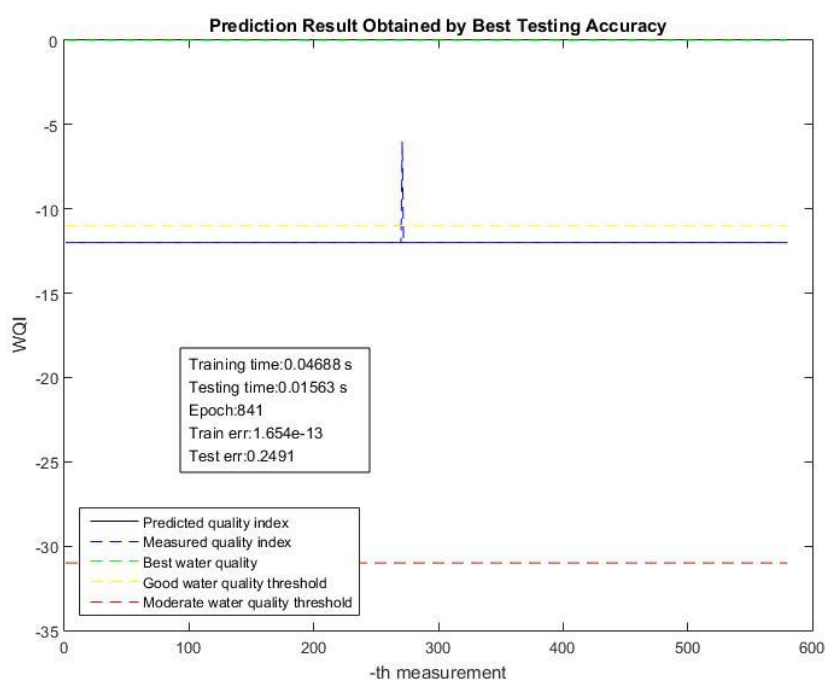


Gambar 4.17. Grafik *training error* yang diperoleh pada setiap perulangan dalam pengujian keenam menggunakan *dataset* Parapat



Gambar 4.18. Grafik *testing error* yang diperoleh pada setiap perulangan dalam pengujian keenam menggunakan *dataset* Parapat

Grafik yang ditunjukkan pada Gambar 4.19 menunjukkan hasil prediksi yang diperoleh saat memperoleh testing error terbaik pada pengujian keenam menggunakan dataset Parapat. Pada grafik, garis biru menunjukkan indeks kualitas air yang dihitung dari setiap hasil pengukuran. Sedangkan, garis hitam pada grafik menunjukkan indeks kualitas air yang diperkirakan oleh extreme learning machine. Hasil prediksi yang didapatkan dari pengujian keenam menggunakan dataset Parapat pada Gambar 4.19 menunjukkan bahwa indeks kualitas yang diperoleh pada pengukuran menggunakan dataset Parapat memiliki rentang antara -6 hingga -11. Karena itu, dapat disimpulkan bahwa dengan mengacu pada Keputusan Menteri Negara Lingkungan Hidup Nomor 115 Tahun 2003 Tentang Pedoman Penentuan Status Mutu Air, kualitas air danau Toba yang terukur pada wilayah Parapat bervariasi antara tingkat kualitas air tingkat baik (kelas B) hingga tingkat sedang (kelas C).



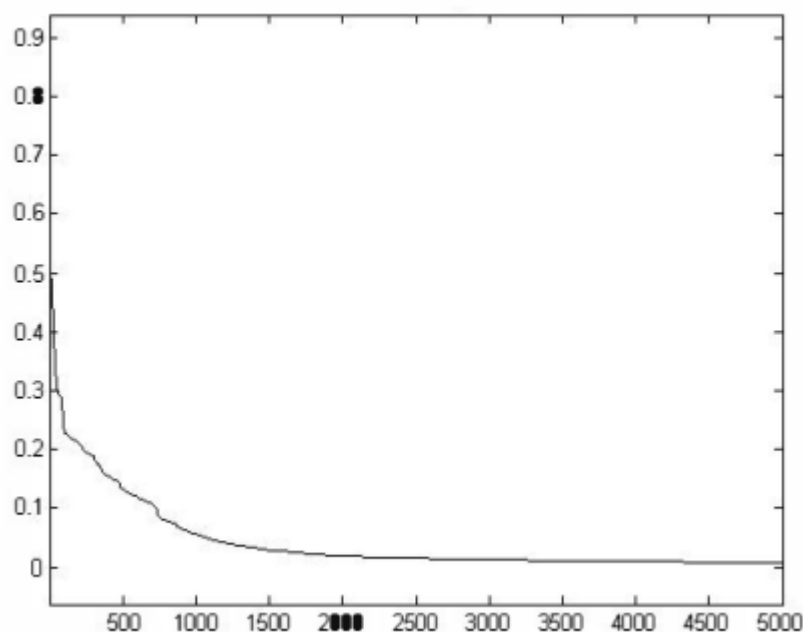
Gambar 4.19. Grafik hasil prediksi saat memperoleh *testing error* terbaik pada pengujian keenam menggunakan *dataset* Parapat

4.3.5. Perbandingan dengan algoritma *backpropagation* (Ming, 2012)

Secara umum, hasil pengujian dalam penelitian ini menunjukkan bahwa proses prediksi kualitas air menggunakan extreme learning mampu memberikan hasil dengan training error dan testing error yang rendah, dan waktu komputasi yang singkat. Indeks kualitas

air yang diperoleh, sesuai dengan metode perhitungan Storet yang digunakan dalam Keputusan Menteri Negara Lingkungan Hidup Nomor 115 Tahun 2003 tentang Pedoman Penentuan Status Mutu Air menunjukkan bahwa kualitas air di Danau Toba memiliki kategori antara kualitas air kelas B (baik) hingga kelas C (sedang).

Sedangkan, Ming (2012) melakukan penelitian mengenai prediksi kualitas air menggunakan algoritma *backpropagation*. Sampel data yang digunakan menyimpan informasi mengenai hasil pengukuran kadar *dissolved oxygen*, *biological oxygen demand*, indeks permanganat, kadar zat phosphor, amonia, dan nitrat. Hasil penelitian dari Ming menunjukkan bahwa proses prediksi kualitas air menggunakan algoritma *backpropagation* dapat memberikan hasil terbaik dengan learning rate sebesar 0,7 dan impuls sebesar 0,5. Grafik *training error* yang diperoleh dalam percobaan yang dilakukan oleh Ming ditunjukkan pada Gambar 4.20.



Gambar 4.20. Grafik hasil pelatihan dari *backpropagation neural network* (Ming, 2012)

BAB 5

KESIMPULAN DAN SARAN

Bab ini membahas tentang kesimpulan dari metode yang diajukan untuk melakukan prediksi kualitas air Danau Toba serta saran-saran yang dapat dipertimbangkan untuk pengembangan dalam penelitian berikutnya.

5.1. Kesimpulan

Berdasarkan hasil pengujian dari metode yang diajukan, yakni *extreme learning machine*, dalam proses prediksi kualitas air di Danau Toba, didapat beberapa kesimpulan, yaitu:

1. Hasil akhir yang didapat pada pengujian menunjukkan bahwa kualitas air di Danau Toba berada pada tingkat tercemar ringan hingga tercemar sedang, dengan kualitas air berkisar di kategori kelas B (baik) hingga kelas C (sedang);
2. Pada *extreme learning machine*, nilai *error* yang dihasilkan dipengaruhi oleh nilai *input weight* dan *bias* yang dihasilkan secara acak;
3. Fungsi aktivasi yang digunakan pada setiap neuron dalam *artificial neural network* mempengaruhi nilai *error* yang dihasilkan;
4. Dalam proses prediksi kualitas air Danau Toba, *extreme learning machine* mampu melakukan *training* dan *testing* dalam durasi yang singkat;
5. *Training error* yang rendah pada proses *training* menggunakan *extreme learning machine* tidak berbanding lurus dengan *testing error* yang diperoleh pada proses *testing*; dan
6. Penambahan jumlah neuron pada *hidden layer* dapat memberikan hasil *training* yang lebih baik, namun tidak selalu berbanding lurus dengan hasil *testing* yang didapatkan.

5.2. Saran

Saran yang dapat diberikan penulis untuk pengembangan selanjutnya adalah sebagai berikut:

1. Penambahan parameter-parameter kualitas air, seperti kadar zat padat terlarut (*total dissolved solids/TDS*), daya konduktivitas, dan parameter lainnya, dapat dilakukan agar dapat memberikan hasil yang lebih akurat;
2. Penerapan berbagai arsitektur *artificial neural network* menggunakan *extreme learning machine* dalam proses prediksi kualitas air danau Toba; dan
3. Berbagai metode penghitungan dapat diterapkan dalam proses perhitungan indeks kualitas air, seperti *Oregon Water Quality Index* (Dinius, 1987).

DAFTAR PUSTAKA

- Abbasi, T. & Abbasi, S.A. 2012. Why Water Quality Indices. In *Water Quality Indices*. pp. 3 - 7. Elsevier: Amsterdam.
- Aptula, O.A., Jeliaskova, N.G., Schultz, T.W. & Cronin, M.T.D. 2005. The Better Predictive Model: High q^2 for the Training Set or Low Root Mean Square Error of Prediction for the Test Set? *QSAR Combinational Science* 24: 385-396.
- Ata, R. 2015. Artificial neural networks applications in wind energy systems: a review. *Renewable and Sustainable Energy Reviews* 49: 534-562.
- Ayyalasomayajula, H., Gabriel, E., Lindner, P. & Price, D. 2016. Air quality simulations using big data programming models. *2016 IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService)*, pp. 182-184.
- Becker, D., King, T. D. & McMullen, B. 2015. Big data, big data quality problem. *Proceedings of 2015 IEEE International Conference on Big Data*: 2644-2653.
- Brown, R.M., McClelland, N.I., Deininger, R.A. and Tozer, R.G. 1970. Water quality index-do we dare? *Water Sewage Works* 117 (10): 339-343.
- Chai, T. & Draxler, R. R. 2014. Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature. *Geoscientific Model Development* 7 (3): 1247-1250.
- Chandra, B. & Sharma, R. K. 2014. Fast learning for big data applications using parameterized multilayer perceptron. *Proceedings of 2014 IEEE International Conferences on Big Data*, pp. 17-22.

- Chandra, B. & Sharma, R. K. 2016. Fast learning in deep neural networks. *Neurocomputing* 171: 1205-1215.
- Deng, C.W., Huang, G.B., Xu, J. & Tang, J.X. 2015. Extreme learning machines: new trends and applications. *Science China Information Sciences* 58: 020301:1-020301:16.
- Dinius, S.H. 1987. Design of an index of water quality. *Water Resources Bulletin* 23 (5): 833–843.
- Effendi, H. 2016. River water quality preliminary rapid assessment using pollution index. *Environmental Sciences* 33: 562-567.
- Fu, H., Wang, Y. & Zhang, H. 2015. Ship rolling motion prediction based on extreme learning machine. *Proceedings of the 34th Chinese Control Conference*, pp. 3468-3472.
- Hammerstrom, D. 1993. Neural networks at work. *IEEE Spectrum* 30 (6), June: 26-32.
- Haro, D. D., Yunasfi, & Harahap, Z.A. 2013. Kondisi Kualitas Air Danau Toba di Kecamatan Haranggaol Horison Kabupaten Simalungun Sumatera Utara (Toba Lake Water Quality Conditions in Sub-District Haranggaol Horison Simalungun Regency of North Sumatra). *Aqua Coast Marine* 1 (1).
- Heaton, J. 2008. Introduction to Neural Networks for Java, 2nd Edition. Heaton Research: St. Louis.
- Hertzmann, A & Fleet, D. 2011. Machine learning and data mining lecture notes. (Online) <https://www.dgp.toronto.edu/~hertzman/411notes.pdf> (5 Agustus 2016)
- Hinton, G. E. & Teh, Y.W. 2006. A fast learning algorithm for deep belief nets. *Neural Computation* 18: 1527-1554.
- Horton, R. K. 1965. An index number system for rating water quality. *Journal of Water Pollution Control Federation* 37 (3): 300-306.

- Huang, G.-B. 2014. An insight into extreme learning machines: random neurons, random features and kernels. *Cognitive Computation* 6: 376-390.
- Huang, G.B., Zhu, Q.Y. & Siew, C.K. 2006. Extreme learning machine: theory and applications. *Neurocomputing* 70 (1-3): 489-501.
- Huang, W., Li, N., Lin, Z., Huang, G.-B., Zong, W., Zhou, J. & Duan, Y. 2013. Liver tumor detection and segmentation using kernel-based extreme learning machine. *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 3662-3665.
- Huang, Z.Y., Yu, Y.L., Gu, J. & Liu, H.P. 2016. An efficient method for traffic sign recognition based on extreme learning machine. *IEEE Transactions on Cybernetics* PP(99): 1-14.
- Kasabov, N. 2007. *Evolving Connectionist Systems*. 2nd Edition. Springer: London.
- Khan, Y & Chai, S S. 2016. Predicting and analyzing water quality using machine learning: a comprehensive model. *2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, pp. 1-6.
- Krose, B. & van der Smagt, P. 1996. *An Introduction to Neural Networks*. 8th Edition. The University of Amsterdam: Amsterdam.
- Lambrou, T P, Panayiotou, C G & Anastasiou, C C. 2012. A Low-Cost System for Real Time Monitoring and Assessment of Potable Water Quality at Consumer Sites. *2012 IEEE SENSORS Proceedings*, pp. 1-4.
- Lara, B., Althoefer, K. & Seneviratne, L. D. 1999. Use of artificial neural networks for the monitoring of screw insertions. *Proceedings of the 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 579-584.
- McCulloch, W S & Pitts, W H. 1943. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* 5: 115-133.
- Ming, X. 2012. A new water quality assessment method based on BP neural network. *World Automation Congress (WAC) 2012*, pp. 1-4.

- Patro, S. G. K. & Sahu, K. K. 2015. Normalization: A Preprocessing Stage. (*Online*) <https://arxiv.org/abs/1503.06462> (23 Januari 2017)
- Popovic, D., Kukolj, D. & Kulic, F. 1998. Monitoring and assessment of voltage stability margins using artificial neural networks with a reduced input set. *IEEE Proceedings - Generation, Transmission and Distribution* 145(1), pp. 355-362.
- Pangaribuan, J. J. & Suharjito. 2014. Diagnosis of diabetes mellitus using extreme learning machine. *Proceedings of International Conference on Information Technology and Innovation (ICITSI) 2014*, pp. 33-38.
- Qiao, J.-J., Zhen, X.-W. & Zhang, Y.-R. 2008. The application of fuzzy comprehensive evaluation on the water quality of Changjiang river. *Proceedings of the Seventh International Conference on Machine Learning and Cybernetics*, Kunming, 2008, pp. 1467-1473.
- Rahmat, R F, Athmanathan, Syahputra, M F, Lydia, M S. 2016. *Real Time Monitoring System for Water Pollution in Lake Toba. International Conference on Informatics and Computing 2016*, pp. 383-388.
- Republik Indonesia. 2001. Peraturan Pemerintah Nomor 82 Tahun 2001 Tentang Pengelolaan Kualitas Air dan Pengendalian Pencemaran Air. Lembaran Negara Republik Indonesia Tahun 2001, Nomor 153. Sekretariat Negara. Jakarta.
- Republik Indonesia. 2003. Keputusan Menteri Negara Lingkungan Hidup Nomor 115 Tahun 2003 Tentang Pedoman Penentuan Status Mutu Air. Sekretariat Negara. Jakarta.
- Republik Indonesia. 2009. Undang-Undang Nomor 32 Tahun 2009 Tentang Perlindungan dan Pengelolaan Lingkungan Hidup. Lembaran Negara Republik Indonesia Tahun 2009, Nomor 140. Sekretariat Negara. Jakarta.
- Rumelhart, D. E., Hinton, G. E. & Williams, R. J. 1986. Learning representations by back-propagating errors. *Nature* 323: 533-536.
- Shalabi, L. A., Shaaban, Z. & Kasasbeh, B. 2006. Data mining: a preprocessing engine. *Journal of Computer Science* 2 (9): 735-739.

- Sun, Z.L., Choi, T.M., Au, K.F. & Yu, Y. 2008. Sales forecasting using extreme learning machine with applications in fashion retailing. *Decision Support Systems* 46 (1): 411-419.
- Uhrig, R. E. 1995. Introduction to artificial neural networks. *Proceedings of the 1995 IEEE IECON 21st International Conference on Industrial Electronics, Control, and Instrumentation*, pp. 33-37.
- van Heeswijk, M. 2015. Advances in extreme learning machines. Disertasi D.Sc. Aalto University.
- Warlina, L. 2004. Pencemaran air: sumber, dampak dan penanggulangannya. (Online) http://www.rudycr.com/PPS702-ipb/08234/lina_warlina.pdf (28 Juli 2016)
- Werbos, P. 1974. Beyond regression: new tools for prediction and analysis in the behavioral sciences. Disertasi Ph.D. Harvard University.
- Zhai, C.M. & Du, J.X. 2008. Applying extreme learning machine to plant species identification. *Proceedings of the 2008 IEEE International Conference on Information and Automation*, pp. 879-884.
- Zhu, Q.-Y. & Huang, G.-B. 2004. MATLAB Codes of ELM Algorithm. (Online) http://www.ntu.edu.sg/home/egbhuang/elm_random_hidden_nodes.html (27 Februari 2017)