

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)

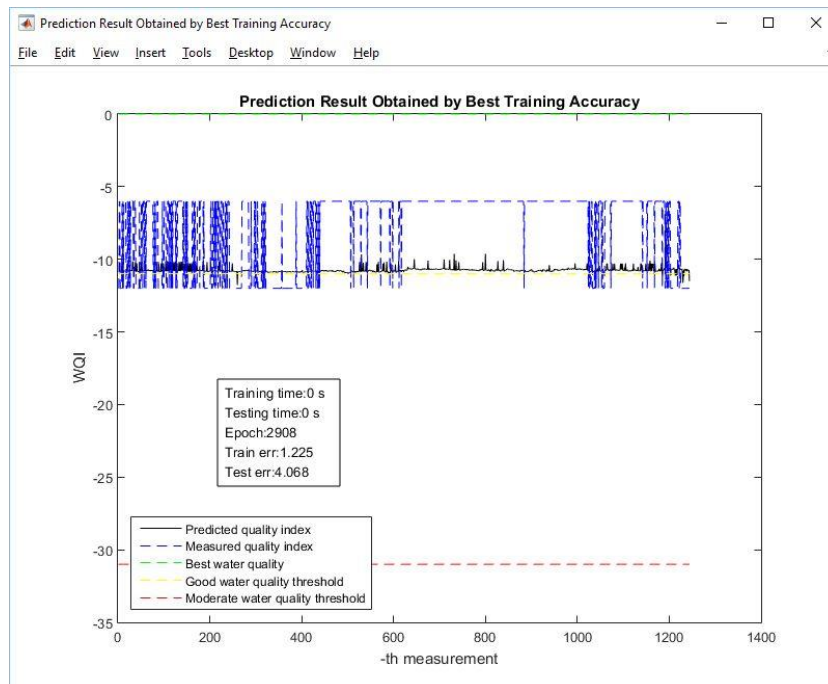
4.1.2. Implementasi perancangan antarmuka

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*.

The screenshot shows a web application window titled "PrediksiKualitasAirDanauToba". The main heading is "Lake Toba Water Quality Prediction using Extreme Learning Machine" with a copyright notice "© 2017 Eric Suwarno". The interface includes two sections for dataset selection: "Training dataset" and "Testing dataset", each with a text input field "Select the dataset file first." and a "Select File" button. Below these is the "Neural network characteristic" section, which includes a "Hidden Neuron" input set to "1" with "+" and "-" buttons, and an "Activation Function" dropdown menu currently showing "sigmoid". A large "Start Prediction" button is positioned below the neural network settings. At the bottom, a small text note states: "The training process will be iterated for 3,000 times."

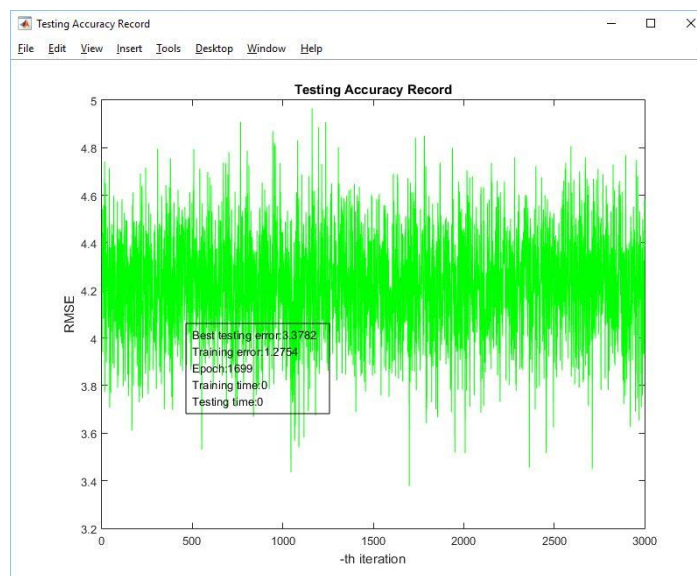
Gambar 4.1. Hasil rancangan antarmuka aplikasi

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

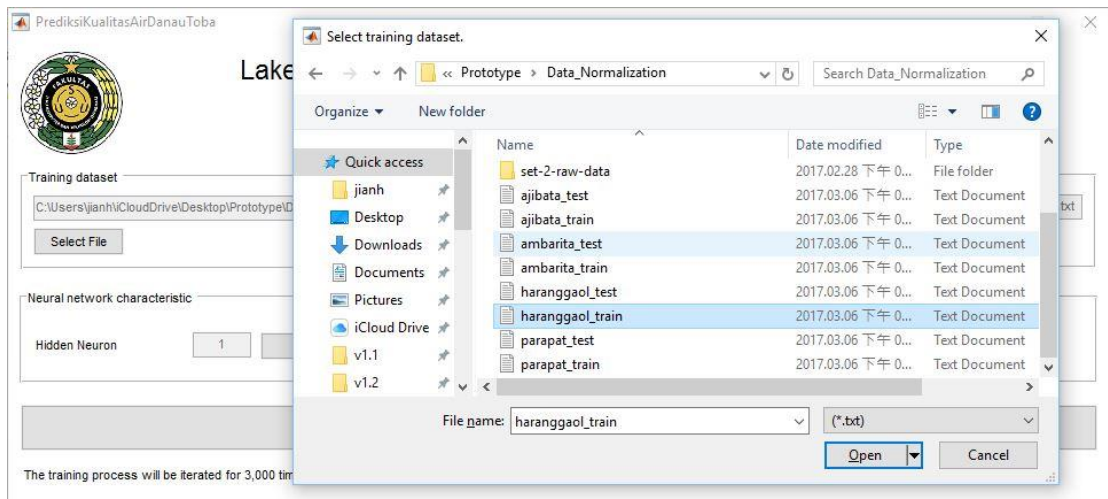
Data yang digunakan dalam penelitian ini bersumber dari penelitian yang dilakukan oleh Rahmat *et al.* (2016), di mana parameter-parameter kualitas air diukur dalam kurun waktu yang sedemikian rupa, dan disimpan dalam format dokumen teks, dengan hasil pengukuran setiap parameter dipisahkan oleh tanda 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 Haranggaol.txt	Haranggaol	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					

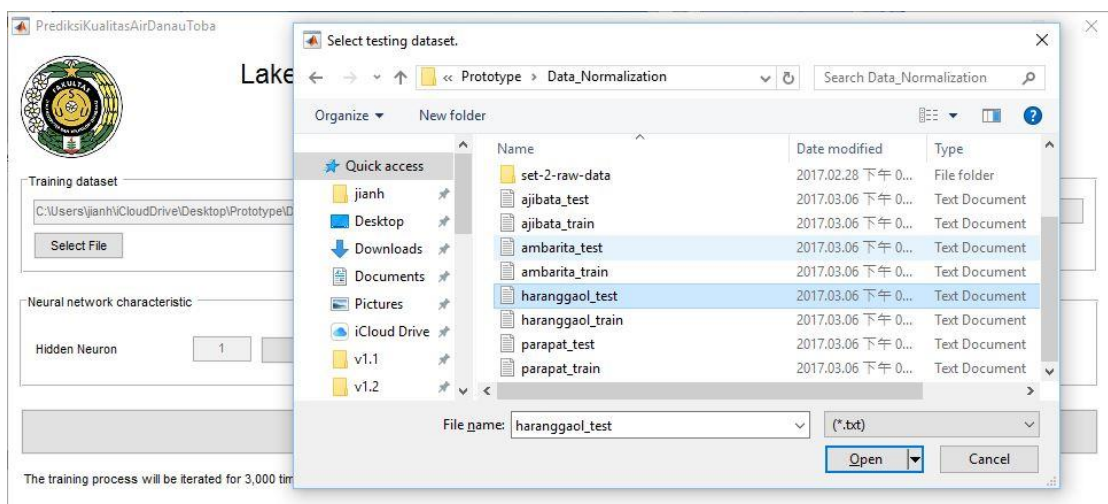
4.2. Prosedur Operasional

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*

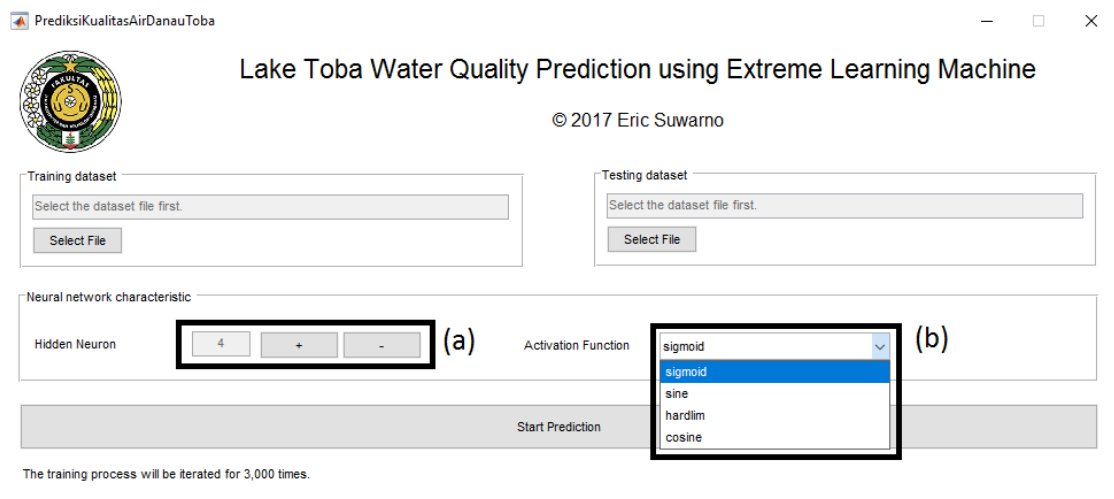
Setelah itu, pengguna juga dapat memilih *testing dataset* (himpunan data uji) dengan memilih tombol “*Select File*” pada bagian “*Testing dataset*”. Ketika dipilih, tombol ini akan memunculkan sebuah dialog pemilihan file, di mana pengguna dapat memilih *file* berupa dokumen teks dengan ekstensi “.txt”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



Gambar 4.5. Ilustrasi pemilihan *testing dataset*

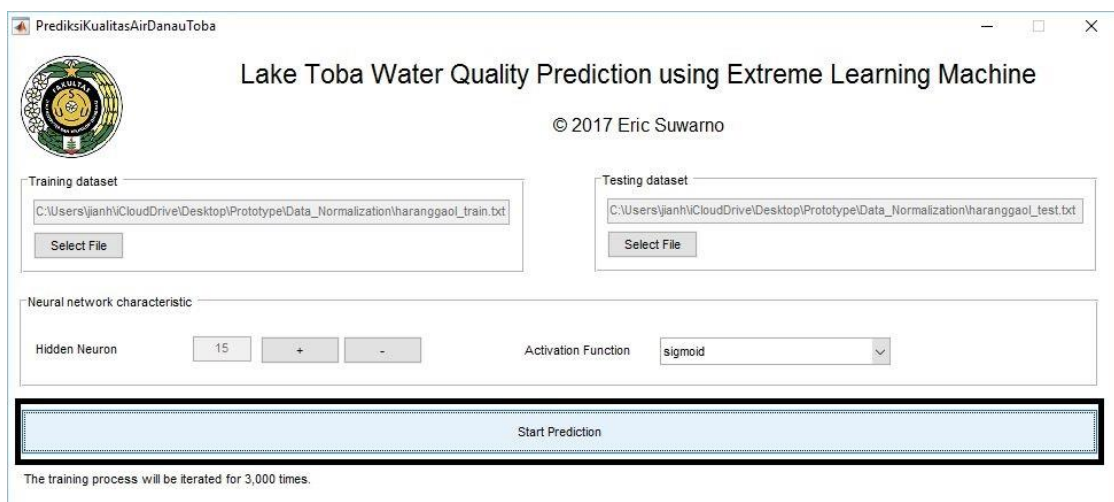
Setelah itu, pengguna dapat menentukan karakteristik dari *artificial neural network* yang akan digunakan dalam proses *training*, yaitu jumlah *hidden neuron* dan fungsi aktivasi neuron. Pengguna dapat mengubah jumlah *hidden neuron* melalui tombol *plus* dan *minus* yang terdapat pada bagian “*Hidden 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-	Training error terbaik				
	Training error (RMSE)	Testing error (RMSE)	Epoch ke-	Durasi training (s)	Durasi testing (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.

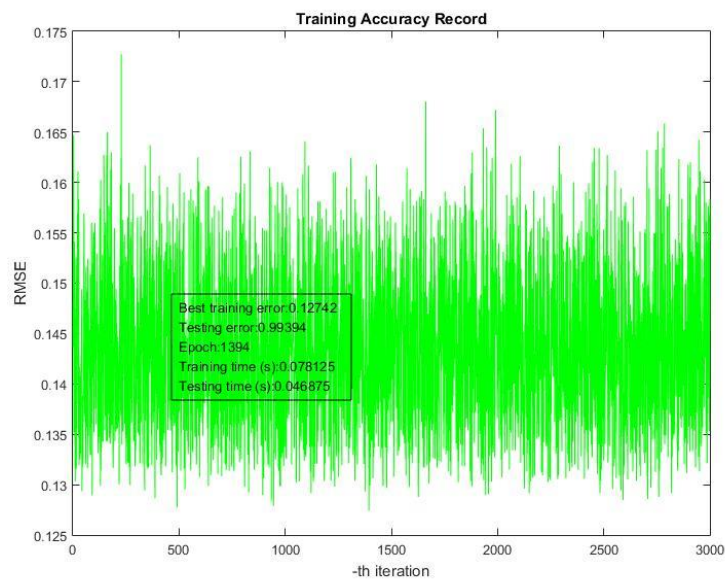
Tabel 4.4. *Testing error* terbaik dalam pengujian *dataset* Ajibata

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

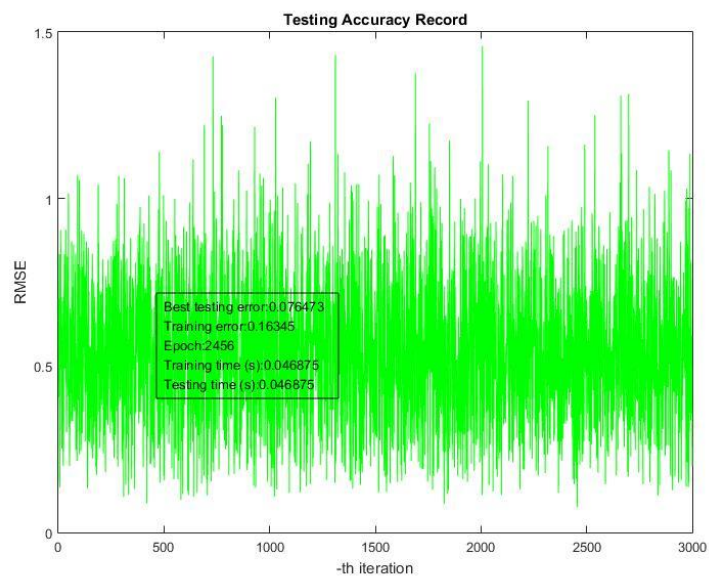
Hasil pengujian juga menunjukkan bahwa pada *extreme learning machine*, *training error* dan *testing error* yang diperoleh pada setiap pengulangan dipengaruhi oleh input weight dan bias yang dihasilkan dari proses training. Hal ini dikarenakan *input weight* dan *bias* pada *extreme learning machine*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

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

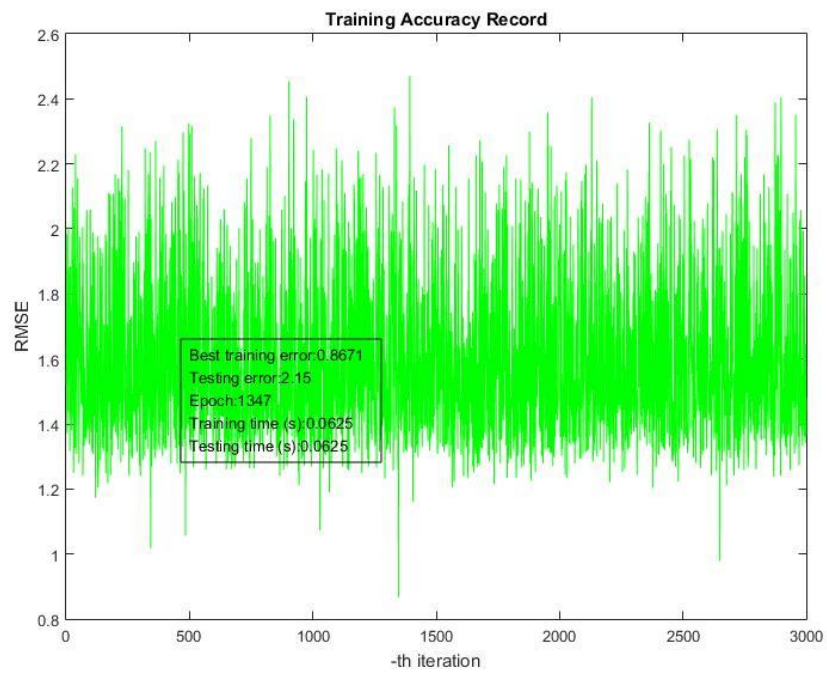
Dari hasil pengujian pada Tabel 4.5, diketahui juga bahwa dari empat fungsi aktivasi yang digunakan dalam pengujian, fungsi *hardlim* mampu memberikan hasil prediksi dengan tingkat *error* terendah. Hal ini dapat dibuktikan dari *training error* terendah yang diperoleh dari pengujian kelima, keenam, 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.

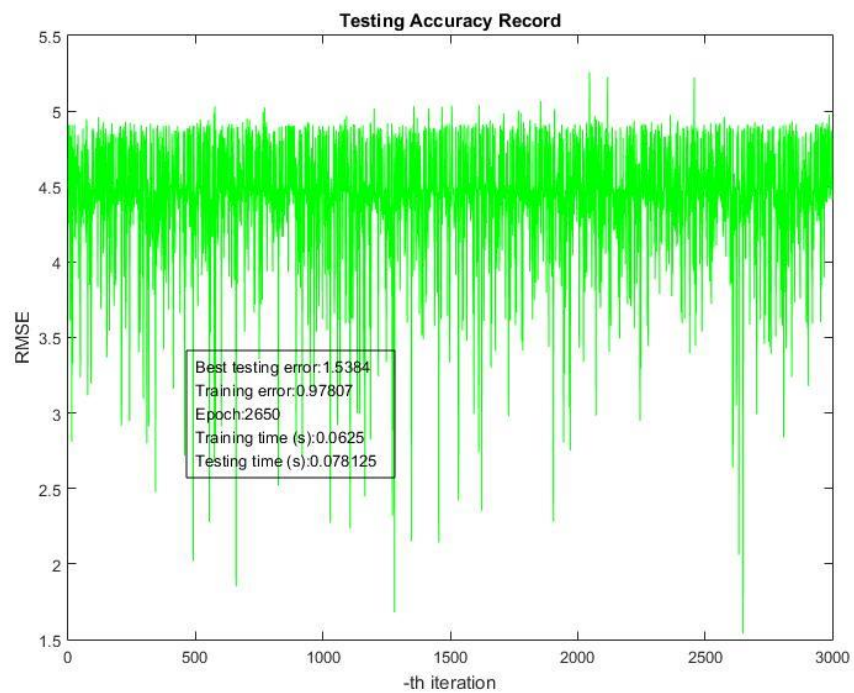
Tabel 4.6. *Testing error* terbaik dalam pengujian *dataset* Ambarita

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 4.10 dan 4.11, 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 4.10. Grafik *training error* yang diperoleh pada setiap perulangan dalam pengujian menggunakan *dataset* Ambarita



Gambar 4.11. Grafik *testing error* yang diperoleh pada setiap perulangan dalam pengujian 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.

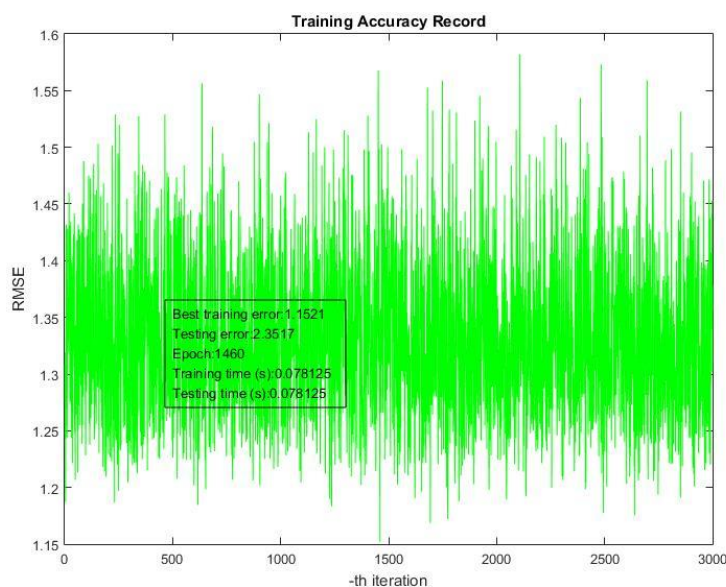
Tabel 4.8. *Testing error* terbaik pada pengujian *dataset* Haranggaol

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

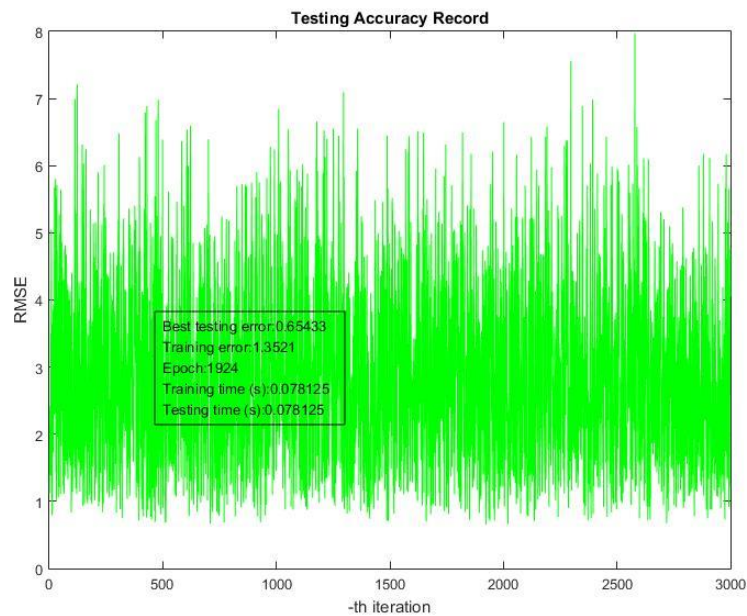
Hasil pengujian menggunakan *dataset* hasil pengukuran di Haranggaol menunjukkan bahwa *extreme learning machine* dapat memberikan hasil prediksi dengan waktu komputasi yang singkat. Berdasarkan Tabel 4.7, durasi *training* dan

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 4.12 dan 4.13 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 4.12. Grafik *training error* yang diperoleh pada setiap perulangan dalam pengujian kedelapan menggunakan *dataset* Haranggaol



Gambar 4.13. Grafik *testing error* yang diperoleh pada setiap perulangan dalam pengujian kedelapan 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.

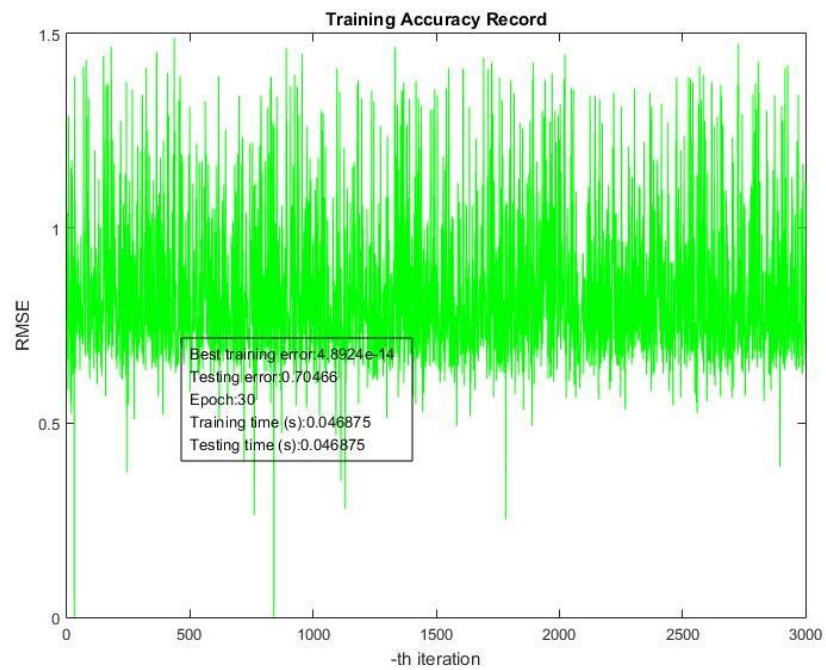
Testing error terbaik yang diperoleh dari pengujian menggunakan dataset hasil pengukuran di Parapat ditunjukkan pada Tabel 4.10. Hasil pengujian menunjukkan bahwa dengan menggunakan *extreme learning machine*, proses prediksi dapat

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.

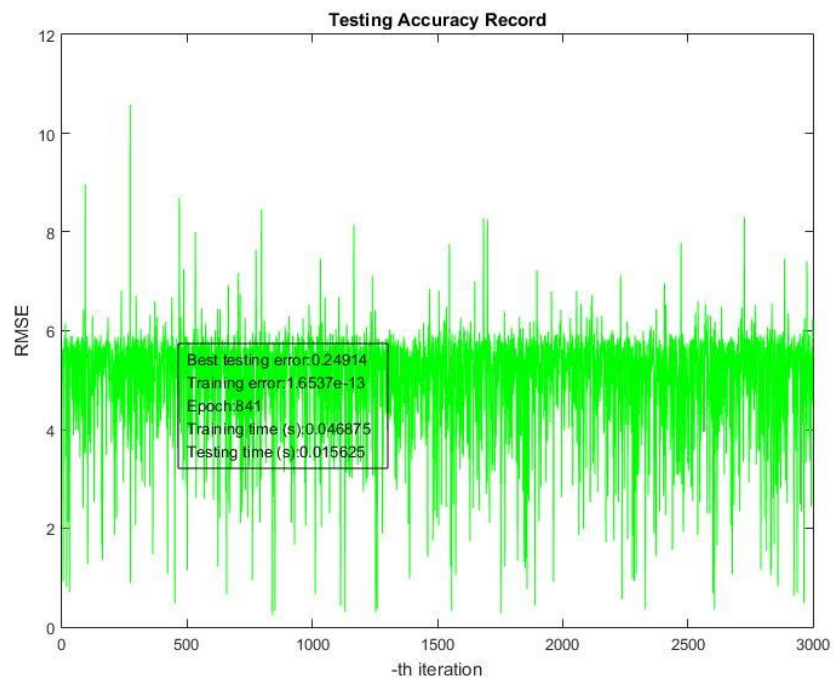
Tabel 4.10. *Testing error* terbaik dalam pengujian *dataset* Parapat

Pengujian ke-	<i>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	0,4866	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 4.14 dan Gambar 4.15, yang diperoleh pada pengujian keenam menggunakan *dataset* Parapat.



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

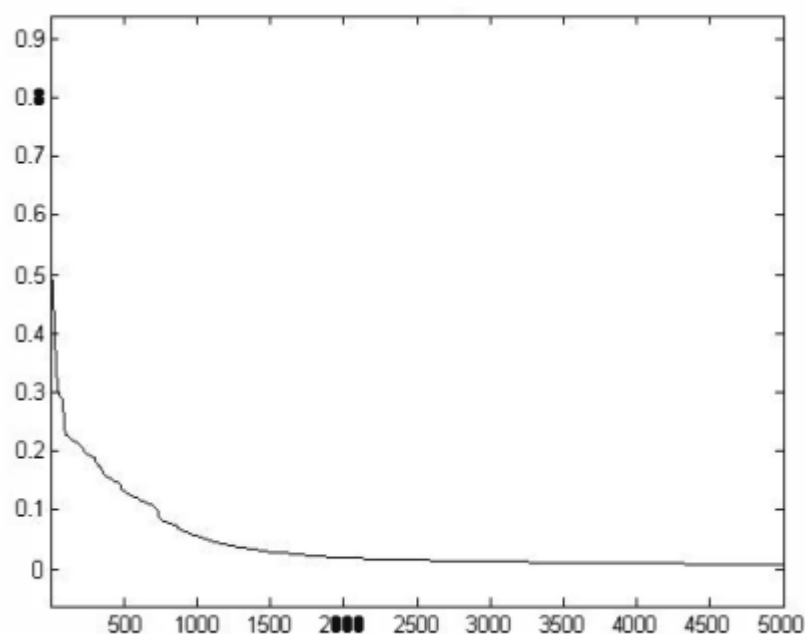


Gambar 4.15. Grafik *testing error* yang diperoleh pada setiap perulangan dalam pengujian keenam menggunakan *dataset* Parapat

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 4.16.



Gambar 4.16. Grafik hasil pelatihan dari *backpropagation neural network* (Ming, 2012)