ISSN 2088-4842 OPTIMASI SISTEM INDUSTRI

MEMINIMALISASI NILAI ERROR PERAMALANDENGAN ALGORITMA *EXTREME LEARNING MACHINE*

Rachmad Hidayat[1], Suprapto[2]
[1]Program Studi Teknik Industri, Fakultas Teknik, Universitas Trunojoyo Madura
Email: rachmad_trunojoyo@yahoo.co.id
[2]Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya Malang
Email: prapto@gmail.com

Dikirimkan 3 Pebruari 212 *Diterima 26 Pebruari 2012*

Abstract

This study uses a machine learning algorithm which is the extreme model one of the new learning method of neural networks. Determining the value of forecasting based on actual value. Output generated more quickly in this process, because the learning done in a fast speed and better accuracy rate than conventional forecasting methods. Epoch using the parameter changes and changes in the range of accuracy of test results demonstrate the value of error is quite good during the system testing.

Keywords: *neural networks, epoch, value of error.*

1. PENDAHULUAN

Peramalan adalah proses untuk memperkirakan beberapa kebutuhan dalam ukuran kuantitas,kualitas,waktu dan lokasi yang di butuhkan dalam rangka memenuhi permintaan barang ataupun jasa. Peramalan adalah kegiatan memperkirakan atau memprediksi apa yang terjadi pada waktu yang akan datang, sedangkan rencana merupakan penentuan apa yang akan dilakukan pada waktu yang akan datang. Peramalan merupakan proses untuk memperkirakan jumlah penjualan di masa datang yang meliputi kebutuhan pelayanan dan pemasaran yang tepat dalam mendapatkan jumlah pemakai yang berlimpah dan demi memenuhi keinginan pengguna. Untuk mengevaluasi harga parameter peramalan, digunakan ukuran kesalahan peramalan. Harga parameter peramalan yang terbaik adalah harga yang memberikan nilai kesalahan peramalan yang terkecil. Terdapat berbagai macam ukuran kesalahan yang dapat diklasifikasikan menjadi ukuran standar dalam statistik dan ukuran relatif. [1] .

Pada penelitiannya, Ai membandingkan beberapa metode untuk memperoleh nilai peramalan yang akurat. Pemilihan metode peramalan yang cocok digunakan untuk meramalkan volume penjualan adalah dengan melihat pola gerakan waktu dari volume penjualannya. Jika pola datanya cenderung naik/ cenderung turun, maka metode peramalan yang digunakan adalah metode dekomposisi. Jika pola datanya berpola fluktuatif, maka metode peramalan yang digunakan adalah metode *smoothing*.[1].

Peneliti lain menggunakan ELM untuk menentukan jumlah konsumen secara lebih akurat dan efektif dalam industri. Hasil uji coba dengan fungsi aktivasi dan jumlah *hidden neuron* yang berbeda, menghasilkan *output* optimal pada fungsi aktivasi *purelin* dengan jumlah *hidden neuron* lima untuk data kaos, dan jumlah *hidden neuron* tiga untuk data pin.[2].

Penelitian ini menggunakan algoritma *extreme learning machine* yang merupakan model salah satu metode pembelajaran baru dari jaringan syaraf tiruan.Metode *extreme learning machine* mempunyai kelebihan dalam *learning speed*, serta mempunyai tingkat akurasi yang lebih baik sehingga dengan menerapkan *extreme learning machine* pada *demand forecasting* diharapkan mampu menghasilkan ramalan yang lebih efektif [2].

2. TINJAUAN PUSTAKA

Teori ELM menunjukkan bahwa parameter simpul tersembunyi dapat benar-benar independen dari data pelatihan.(1) Dalam teori belajar konvensional dan implementasi, kita harus melihat data pelatihan sebelum menghasilkan parameter simpul tersembunyi. (2) Dalam teori pembelajaran dan implementasi ELM, seseorang dapat menghasilkan parameter simpul tersembunyi sebelum melihat data pelatihan.

Dibandingkan dengan Algoritma populer Back-Propagation (BP) dan Support Vector

Machine (SVM) / Least SVM Square (LS-SVM), ELM memiliki beberapa fitur penting yaitu (1) Kemudahan penggunaan. Tidak ada parameter yang perlu disetel secara manual, kecuali arsitektur jaringan standar. Pengguna tidak perlu menghabiskan beberapa jam atau tala hari dan mesin pelatihan pembelajaran. (2) Belajar cepat kecepatan. Kebanyakan pelatihan dapat diselesaikan dalam *milidetik, detik,* dan *menit* (untuk skala besar aplikasi yang kompleks). Ini bisa memperoleh kinerja yang lebih baik daripada BP generalisasi dalam banyak kasus, dan mencapai kinerja generalisasi yang mirip atau lebih baik dari SVM. Tampaknya ELM yang selalu memberikan lebih baik daripada kinerja generalisasi dari LS-SVM. (3) Cocok untuk hampir semua fungsi nonlinier. Hampir semua fungsi kontinyu bagian demi bagian (termasuk terputus, diferensial, diferensial non-fungsi) dapat digunakan sebagai fungsi aktivasi pada ELM. (4) Cocok untuk fungsi aktivasi yang kompleks. Sebaliknya fungsi kompleks juga dapat digunakan sebagai fungsi aktivasi pada ELM.

Extreme Learning Machine merupakan metode pembelajaran baru dari jaringan syaraf tiruan.*Extreme Learning Machine* merupakan jaringan syaraf tiruan feedforward dengan single hidden layer atau biasa disebut dengan *Single Hidden Layer Feedforward neural Networks* (SLFNs). Metode pembelajaran *Extreme Learning Machine* dibuat untuk mengatasi kelemahan-kelemahan dari jaringan syaraf tiruan feedforward terutama dalam hal *learning speed.* Huang *et al* mengemukaan dua alasan mengapa jaringan saraf tiruan feedforward lain mempunyai *leraning speed* rendah, yaitu : (1) Menggunakan *slow gradient based learning algorithm* untuk melakukan *training*. (2) Semua parameter pada jaringan ditentukan secara iterative denganmenggunakan metode pembelajaran tersebut. [2]

Pada *Extreme Learning Machine* parameter-parameter seperti *input weight* dan *hidden bias* dipilih secara random, sehingga *Extreme Learning Machine* memiliki learning speed yang cepat dan mampu menghasilkan *good generalization performance.* Metode *Extreme Learning Machine* mempunyai model matematis yang berbeda dari jaringan syaraf tiruan feedforward. Model matematis dari *Extreme Learning Machine* lebih sederhana dan efektif. Berikut model matrematis dari *Extreme Learning Machine* [2].

Untuk *N* jumlah sample yang berbeda (*Xi*,*ti*)

$Xi = [\ Xi_1, Xi_2 \ldots, Xi_n]^T \in R^n$ (1)

$Xt = [\ Xt_1, Xt_2 \ldots, Xt_n]^T \in R^n$ (2)

Standart SLFNs dengan jumlah hidden nodes sebanyak *N* dan *activation functiong*(*x*) dapat digambarkan secara matematis:

$$\sum_{i=1}^{\bar{N}} \beta igi(xj) = \sum_{i=1}^{N} \beta ig(W_i \bullet X_i + b_i) = 0_j \quad (3)$$

Dimana :

$j = 1,2,\ldots, N$

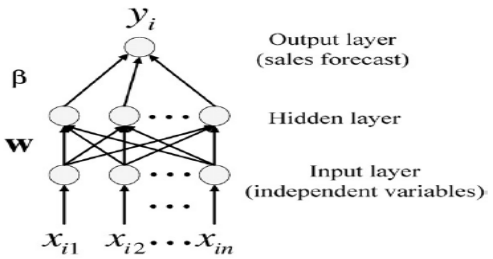
$w_i = (w_{i1}, w_{i2}, \ldots, w_{in})^T$ = merupakan vektor dari weight yang menghubungkan *i*th hidden nodes dan input nodes.

$\beta_i = (\beta_{i1}, \beta_{i2}, \ldots, \beta_{in})^T$ = merupakan *weight vector* yang menghubungkan*i* th *hidden* dan *output nodes.*

b_i = threshold dari *i* th *hidden nodes.*

$w_i x_j$=*inner* produk dari $w_i$dan x_j

Konfigurasi sederhana algoritma ELM dapat dijelaskan pada Gambar 1.



Gambar 1.*Struktur Extreme Learning Machine*

Neural network (NN) adalah suatu metode analisis data yang menggunakan prinsip dasar jaringan syaraf di dalam otak manusia.Model syaraf merupakan pembentuk utama NN yang di susun berdasarkan pilihan arsitektur dan konsep learning.Kemampuan dasar NN adalah mampu mempelajari contoh input dan output yang di berikan sehingga dapat memecahkan masalah-masalah yang tidak dipecahkan dengan metode konvensional.

NN terdiri dari unit utama dan neuron-neuron. Beberapa neuron terkombinasi membentuk jaringan. Jaringan tersebut terdiri dari sebuah lapisan input (input layer),sebuah output (output layer) dan kemungkina satu atau lebih lapisan atau disebut juga lapisan tersembunyi(hidden layer).

Langkah-langkah *Extreme Learning Machine*adalah: (1) Pembagian data training dan testing. (2) Training *Extreme Learning Machine.* (3) Testing *Extreme Learning Machine.* (4) Analisa hasil peramalan. Pembagian data menjadi data training dan testing. Proses training dan testing mutlak diperlukan pada proses peramalan dengan *Extreme Learning Machine*. Adapun proses pertama yang harus adalah proses pelatihan

tujuan dari proses ini adalah untuk mendapatkan input weight, bias dan output weight dengan tingkat kesalahan yang rendah. Selanjutnya proses training untuk mengembangkan model dari *Extreme Learning Machine*, sehingga data dibagi menjadi dua yaitu *data training* dan *data testing*.Beberapa peneliti membagi data training dan testing dengan komposisi:(1) Data training sebanyak 80% dari total data. (2) Data testing 20% dari total data. [3].

Pada proses *training*jumlah *hidden neuron*dan fungsi aktivasi dari *Extreme Learning Machine* harus ditentukan terlebih dahulu. Pada proses ini, dilakukan uji coba dengan menggunakan fungsi aktivasi *log sigmoid* atau *tan sigmoid.*Kedua fungsi tersebut yang paling sering digunakan pada permasalahan *forecasting*, disamping fungsi transfer *purelin* karena data yang diramalkan bersifat stasioner. Fungsi transfer linier memiliki kelemahan pada pola data yang memiliki trend. [3]. Untuk jumlah hidden neuron, *Extreme Learning Machine* menghasilkan output peramalan yang stabil dengan jumlah hidden neuron 0-30. Namun jika output yang didapatkan dari *Extreme Learning Machine*kurang optimal, maka akan digunakan alternative fungsi transfer yang lain atau merubah jumlah hidden neuron. [4].

Menghitung *input weight*, *bias of hidden neuron* dan *output weight* yaitu *output* dari proses pelatihan *Extreme Learning Machine* adalah *input* dan *output weight* serta *bias* dari *hidden neuron* dengan tingkat kesalahan rendah yang diukur dengan MSE dan MAPE. *Input weight* ditentukan secara random, sedangkan *output weight* merupakan *invers* dari matrix *hidden layer* dan *output*. Secara matematis adalah:

$$H = (w_i,....,w_N,b_i,....,b_N,x_i,....,x_N)$$

$$= \begin{matrix} g(w_1x_1+b_1)\cdots g(w_Nx_1+b_N) \\ \vdots \quad \vdots \quad \vdots \\ g(w_1x_N+b_1)\cdots g(w_Nx_N+b_N) \end{matrix} \quad (4)$$

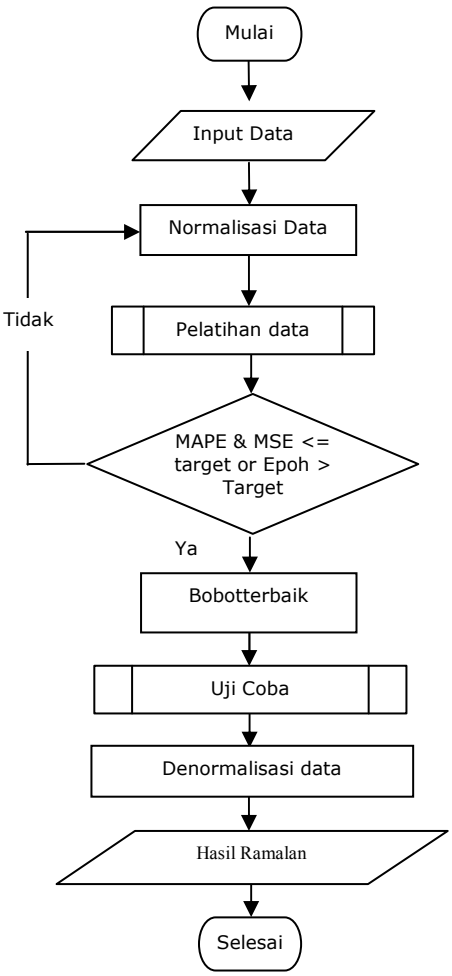
$$\beta = \begin{matrix} \beta_1^T \\ \vdots \\ \beta_N^T \end{matrix} \quad (5)$$

$$T = \begin{matrix} t_1^T \\ \vdots \\ t_N^T \end{matrix} \quad (6)$$

$$\beta = H^{\tau}T \quad (7)$$

H adalah *hidden layeroutput* matrix $g(w_ix_i+b_1)$ menunjukkan *output*dari *hidden neuron*yang berhubungan dengan *input*. β merupakan matrix dari *output weight* dan T matrix dari target atau output. Pada *Extreme Learning Machine, input weight* dan *hidden bias* ditentukan secara acak, maka *output weight* yang berhubungan dengan *hidden layer*.

Proses testing *Extreme Learning Machine*berdasarkan *inputweight* dan *output weight* yangdidapatkan dari proses *training*. Tahap selanjutnyaadalah melakukan peramalan dengan *Extreme Learning Machine*. Data yangdigunakan adalah data *testing* sebanyak 20% dari data. Pada tahap ini data *input* dinormalisasi terlebih dahuludengan *range* dan rumus normalisasi yang sama dengandata *training*. Secara otomatis *output* dari proses ini jugaharus melalui proses dinormalisasi.[3].



Gambar 2. Flowchart umum ELM

3. METODOLOGI PENELITIAN

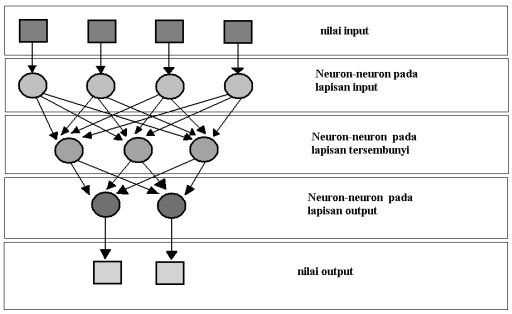
3.1. Flowchart Umum Extreme Learning Machine (ELM)

Flowchart adalah gambaran dalam bentuk diagram alir dari algoritma-algoritma

dalam suatu program, yang menyatakan arah alur program tersebut.

3.2. Jaringan Saraf Tiruan (JST)

JST adalahpemodelan datastatistik non-linier. JST dapat digunakan untuk memodelkan hubungan yang kompleks antara input dan output untuk menemukan pola-pola pada data. Jaringan saraf tiruan pada umumnya digunakan untuk tugas atau pekerjaan yang kurang praktis jika dikerjakan secara manual[5].



Gambar 3. Jaringan syaraf dengan 3 lapisan

3.3. Transformasi Data

Sebab-sebab utama data ditransformasi adalah agar kestabilan taburan data dicapai. Selain itu berguna untuk menyesuaikan nilai data dengan *range* fungsi aktivasi yang digunakan dalam jaringan. Ada beberapa transformasi yaitu transformasi polinomial, transformasi normal, transformasi linear.

3.4. Transformasi Normal (Normalisasi)

Suatu teknik untuk mengorganisasikan data ke dalam table-tabel untuk memenuhi kebutuhan pemakai di dalam suatu ogranisasi. Data-data yang ada dilakukan normalisasi dengan membagi nilai data tersebut dengan nilai range data (nilai data maksimum- nilai data minimum).Tujuan dari Normalisasi yaitu:

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

$$X_n = X_n = \frac{X_o - X_{\min}}{X_{\max} - X_{\min}} \qquad (5)$$

dengan,

x_n = nilai data normal.
x_0 =nilai data aktual.
Xmin= nilai minimum data aktual keseluruhan.
Xmax= nilai maksimum data aktual keseluruhan[2].

Normalisasi data input bertujuan untuk menyesuaikan nilai range data dengan fungsi aktivasi dalam sistem ELM. Ini berarti nilai kudrat input harus berada pada range 0 sampai 1. Sehingga range input yang memenuhi syarat adalah nilai data input dari 0 sampai 1. Oleh karena itu output yang dihasilkan pun akan berada pada range 0 sampai 1. kemudian untuk mendapatkan nilai sebenarnya dari output perlu dilakukan proses denormalisasi.

3.5. Denormalisasi

Denormalisasi memberikan atau mengembalikan data, sehingga didapatkan *predicted sales* dari data *training*. Output yang dihasilkan oleh jaringan berkisar antara 0 sampai dengan 1 sehingga perlu dilakukan denormalisasi yang berguna untuk mengkonversikan kembali hasil output jaringan menjadi harga material normal. setelah itu akan dilakukan perbandingan antara data sebenarnya dengan data hasil prediksi, sehingga dapat dihitung error atau prosentase errornya [4]. Berikut rumus denormaliasi yang digunakan:

$$\boldsymbol{X = 0.5(X_p + 1)(max\{X_p\} - min\{X_p\}) + min\{X_p\}} \quad (6)$$

Dimana :

X = nilai data setelah denormalisasi.
X_p= data *output* sebelum denormalisasi.
$\min(X_p)$= data minimum pada data set sebelum normalisasi.
$\max(X_p)$= data maksimun pada data set sebelum normalisasi.

3.6. Analisis dan Perancangan Sistem

Perancangan sistem ini berupa *use case diagram, activity diagram* dan *sequence diagram*.

4. Hasil dan Pembahasan

4.1. Skenario Uji Coba

Pelaksanaan ujicoba perangkat lunak ini didefinisikan parameter yang digunakan, seperti: perubahan *epoch*, dan perubahan *range.* Proses uji coba dilakukan untuk menentukan keakuratan sistem dalam melakukan proses prediksi jumlah penjualan berdasar *MSE* dan *MAPE*.Pembagian data untuk proses training dan testing sebagai berikut: Skenario 1 = data uji 10, pelatihan 38, Skenario 2 = data uji 15, pelatihan 33, dan Skenario 3 = data uji 18, pelatihan 30.

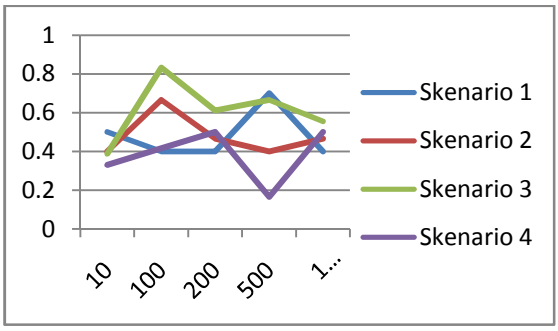
Dengan ketetapan yang di uji cobakan dan dengan set parameter *e-poch* dan *range*.Dengan ketetapan yang di uji cobakan dan dengan set parameter *perubahan e-poch* dan perubahan *range*.

Pelaksanaan Ujicoba Skenario 1 – 4.
Batasan yang digunakan :

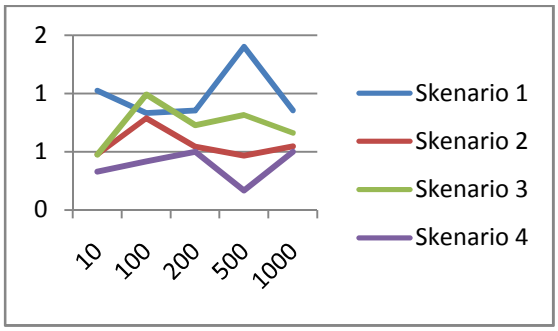
- Input Layer = 2

- Hidden Layer = 2
- Output Layer = 1
- MSE Target = 0,1
- MAPE = 0,1
- Max epoch atau iterasi = 1000
- Range antara -0,04 sampai 0,04
- Epoch dari 10 sampai 1000

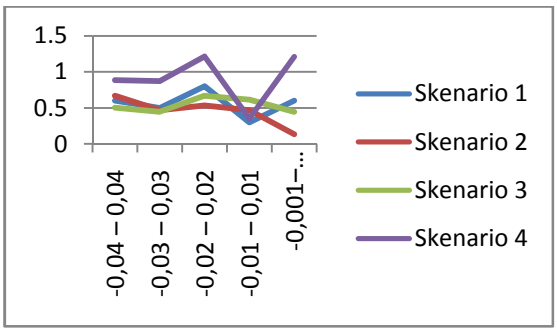
Uji Coba Data *ELM* dengan perubahan Epoch untuk parameter MSE dan MAPE seperti Gambar 1 dan 2. Sedangkan uji coba *ELM*dengan perubahan parameter *range*untuk data MSE dan MAPE seperti Gambar 3 dan 4.



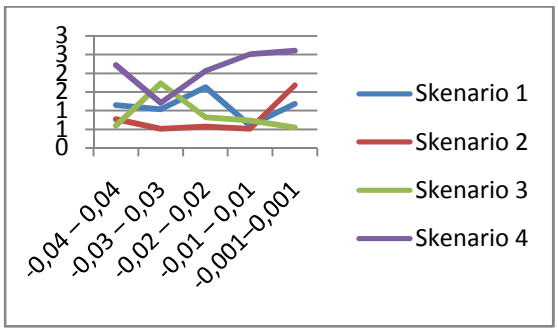
Gambar 1. Hasil uji Coba Data *ELM* dengan perubahan Epoch untuk parameter MSE.



Gambar 2. Hasil Uji Coba Data *ELM* dengan perubahan Epoch untuk parameter MAPE.



Gambar 3. Hasil uji Coba Data *ELM* dengan perubahan Range untuk parameter MSE.



Gambar 1. Hasil uji Coba Data *ELM* dengan perubahan Range untuk parameter MAPE.

Hasil uji coba 1, *ELM* untuk set perubahan *epoch*menunjukkan hasil terbaik adalah MSE mencapai 0,4 dan *MAPE* mencapai 0,831%pada Epoch ke -100. Hasil uji coba *ELM* untuk set perubahan parameter *range*menunjukkan hasil terbaik mtode *ELM* adalah MSE mencapai 0,3 dan *MAPE* mencapai 0,592 % set *range* [-0,01 – 0,01].

Hasil uji coba 2, *ELM* untuk set perubahan *epoch*menunjukkan bahwa MSE mencapai 0,4 dan *MAPE* mencapai 0,466% pada Epoch ke -500. Sedangkan untuk set perubahan *range,* MSE mencapai 0,133 dan *MAPE* mencapai 1,673 % set *range* ke [-0,001 – 0,001].

Hasil uji coba 3, *ELM* untuk set perubahan *menunjukkan* bahwa hasil terbaik MSE mencapai 0,388 dan *MAPE* mencapai 0,475 % pada Epoch ke -10. Sedangkan untuk set perubahan *range,* MSE mencapai 0,444 dan *MAPE* mencapai 0,541 % set *range* ke [-0,001 – 0,001].

Hasil uji coba 4, *ELM* untuk set perubahan *epoch* adalah MSE mencapai 0,166 dan *MAPE* mencapai 0,352% pada Epoch ke -500. Sedangkan untuk set perubahan parameter *range,* hasil terbaik adalah MSE mencapai 0,3 dan *MAPE* mencapai 0,592 % set *range* ke [-0,01 – 0,01].

4.2. Uji Coba Data *ELM* dengan perubahan Hidden

Pada pelaksanaan ujicoba sistem, sebelum melakukan perhitungan user harus menentukan beberapa batasan, yaitu:

- Input Layer = 2
- Output Layer = 1
- MSE Target = 0,1
- MAPE = 0,1
- Max epoch atau iterasi = 1000

Tabel 1. Hidden yang diinginkan*ELM*

Range	**Hidden**	***MSE***	***MAPE***
-0,01 – 0,01	3	0,333	0,807
-0,01 – 0,01	4	0,5	1,1
-0,01 – 0,01	5	0,5	1,015
-0,01 – 0,01	6	0,583	1,357
-0,01 – 0,01	7	0,25	0,582
-0,01 – 0,01	8	0,416	0,964

-0,01 – 0,01	9	0,666	1,516
-0,01 – 0,01	10	0,416	0,938

Pada tabel 1 ditunjukkan hasil uji coba *ELM* untuk set perubahan *hidden*. Pada tabel diatas ditunjukkan bahwa dari 5 *run*, hasil terbaik metode *ELM* adalah MSE mencapai 0,25 dan *MAPE* mencapai 0,582%pada hidden ke -7.

4.2. Analisis Performansi

Tabel 2. Analisis skenario berdasarkan *MSE*dan*MAPE* untuk *ELM*

Skenario	MSE			MAPE		
	1	2	3	1	2	3
Epoch	0,4	0,133	0,277	0,831	0,275	0,68
Range	0,3	0,4	0,388	0,592	0,999	0,872

Tabel 2 memperlihatkan perbedaan pada setiap skenario untuk nilai MSE dan nilai MAPE. Rekap hasil uji coba skenario1,2, dan 3 tentang hasil akurasi perubahan *Epoch* dan *range*. Nilai MSE terkecil pada perubahan Epoch terdapat pada skenario 2 yaitu dengan data uji 15 dan data pelatihan 33 dengan nilai MSE 0,133, sedangkan nilai MAPE terkecil terdapat pada skenario 2 yaitu dengan data uji 15 dan data pelatihan 33 dengan nilai MAPE 0,275%.

Perubahan range nilai MSE terkecil terdapat pada skenario 1 yaitu dengan data uji 10 dan data pelatihan 38, sehingga di dapatkan nilai MSE 0,3.Untuk nilai MAPE terkecil terdapat pada skenario 1 yaitu dengan data uji 10 dan data pelatihan 38 dengan nilai MAPE 0,592%.

5. Kesimpulan

*Extreme Learning Machine (ELM)*hanya menggunakan*Hidden* nodes sebanyakNdan H (*hidden layeroutput* matriks), bobot input &*hidden* bias ditentukan secara acak untuk mengontrol hasil peramalannya. Yang di insialisasikan pada awal perhitungan sangat mempengaruhi pada nilai keluaran. Menentukan nilai peramalan berdasarkan nilai aktual. Dan *output* yang dihasilkan lebih cepat dalam proses ini, karena *learning speed* di lakukan secara cepat dan tingkat akurasinya lebih baik daripada metode konvensional dan JST lainnya.

Dengan menggunakan parameter perubahan epoch dan perubahan range maka hasil pengujian menunjukkan nilai akurasi error yang cukup baik saat sistem melakukan pengujian. Untuk peneliti berikutnya, penelitian ini dapat dikembangkan dengan menggunakan metode peramalan lain dengan penambahan parameter yang lain.

DAFTAR PUSTAKA

[1] J. Ai, "Optimasi Peramalan Pemulusan Exponensial satu Parameter Dengan Menggunakan Algoritma Non-Linear Programing", *Jurnal Teknologi Industri*, Bandung. 1999.

[2] G.B. Huang, , Q.Y. Zhu, and C.K. Siew, "Extreme Learning Machine: Theory and applications', *Neurocomputing*, vol. 70, pp. 489-501, 2006.

[3] G. Zhang, , B. E. Pattuwo, , and M.Y. Hu, "Forecasting with Artificial Neural Networks: The State of the Art", *International Journal of Forecasting*, vol.14, pp. 35-62, 1998.

[4] Z.L. Sun, T.M. Choi, , K.F. Au, and Y. Yu, "Sales Forecasting using Extreme Learning Machine with Application in Fashion Retailing", *Decision Support Systems*, vol. 46, pp. 411-419, 2008.

[5] N-Y. Liang. *Classification of mental tasks from eeg signals Using extreme learning machine.* School of electrical and electronic engineering, Nanyang technological University, Nanyang avenue, Singapore. 2006.