

## Penerapan Metode *Extreme Learning Machine* (ELM) Untuk Memprediksi Jumlah Produksi Pipa Yang Layak (Studi Kasus Pada PT. KHI Pipe Industries)

Nirzha Maulidya Ashar<sup>1</sup>, Imam Cholissodin<sup>2</sup>, Candra Dewi<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya  
Email: <sup>1</sup>modyashar@gmail.com, <sup>2</sup>imamcs@ub.ac.id, <sup>3</sup>dewi\_candra@ub.ac.id

### Abstrak

PT. KHI Pipe Industries merupakan perusahaan yang memproduksi pipa baja berkualitas tinggi. Dalam aktivitas produksinya, perusahaan ini melakukan proses produksi berdasarkan permintaan pelanggan dengan spesifikasi pengukuran diameter, ketebalan, dan panjang pipa. Dalam proses produksinya, hasil produksi pipa yang layak tidak mutlak sesuai dengan jumlah permintaan pelanggan, diperkirakan terdapat pipa yang rusak sehingga mengakibatkan pemenuhan permintaan pelanggan tidak dapat berjalan lancar. Perusahaan harus mengeluarkan biaya tambahan untuk bahan baku agar total produksi bertambah dan menutupi jumlah pipa yang rusak. Jumlah produksi pada setiap spesifikasi permintaan berbeda-beda sehingga menjadi kendala dalam mengambil kebijakan untuk memprediksi jumlah pipa yang layak. Dengan adanya prediksi jumlah pipa yang layak dapat membantu perusahaan dalam mengambil keputusan untuk menentukan target produksinya. Penelitian ini menerapkan metode dari Jaringan Saraf Tiruan (JST) yaitu *Extreme Learning Machine* (ELM) untuk memprediksi jumlah produksi pipa yang layak. Proses prediksi yaitu dengan normalisasi, *training*, *testing*, dan denormalisasi, serta melakukan perhitungan nilai *error* menggunakan *Mean Square Error* (MSE). Berdasarkan pengujian yang dilakukan, penggunaan 7 *hidden neuron*, 5 fitur, dan persentase perbandingan 80% data *training* 20% data *testing* menghasilkan nilai *error* terkecil dengan rata-rata 0,00372 dengan selisih  $\pm 1\%$  dari data aktual.

**Kata kunci:** produksi pipa, prediksi, ELM, JST, MSE

### Abstract

*KHI Pipe Industries is a company that specializes in producing high-quality steel pipes. This company produces its end product based on customer demand, with the measurement specifications which is diameter, thickness and the pipe length. In the production process, the amount of viable pipes do not always match with the number of customers demand since there were always a number of damaged pipes. Therefore, the company has always have to spend additional cost to cover the the damaged pipes. The number of production on each specifications varies so that it becomes a challenge for the company to predict the exact amount of pipes to produce. With the appropriate prediction of the number of pipes to produce can help the company to determine the production target. In this research applied method of Artificial Neural Network (ANN) that is Extreme Learning Machine (ELM) to predict the amount of approved pipe production. The prediction process is normalization, training, testing, and denormalization, and to calculate the error value using Mean Square Error (MSE). Based on evaluation performed, the use of 7 hidden neurons, 5 features, and percentage comparison 80% of training data 20% of testing data resulted in the smallest error average is 0,00372 with difference  $\pm 1\%$  to actual data.*

**Keywords:** pipe production, prediction, ELM, ANN, MSE

## 1. PENDAHULUAN

PT. KHI Pipe Industries adalah anggota dari Krakatau Steel Group yang juga dikenal sebagai perusahaan baja terbesar milik negara di Asia Tenggara. Dalam aktifitas produksinya PT. KHI

Pipe Industries memproduksi hanya berdasarkan permintaan pelanggan.

Produk pipa dalam proses produksi tersebut tidak mutlak lulus dalam pengujian spesifikasi. Diperkirakan tingkat kelulusan pipa menurun pada pengujian mengakibatkan berkurangnya

jumlah pipa yang berhasil produksi sesuai dengan target produksinya, maka diharapkan dapat melakukan proses produksi hingga finalisasi kelayakan dari pengujian yang sesuai dengan target produksi secara Rencana Kerja dan Anggaran Perusahaan (RKAP) pada PT. KHI Pipe Industries.

Konsumen melakukan permintaan produksi berdasarkan diameter, ketebalan, dan panjang pipa yang pada proses produksinya menghasilkan produk yang layak dengan jumlah berbeda-beda, dengan kisaran 40% hingga 96%. Agar jumlah produksi mencapai target, perusahaan seharusnya mengeluarkan biaya bahan baku tambahan dengan penambahan *Hot Rolled & Coiling* (HRC) agar total produksi bertambah dan meningkatkan jumlah produk yang layak pakai, untuk menutupi beberapa produk yang tidak lulus pengujian.

Berbedanya hasil produksi dengan setiap spesifikasi permintaan menjadi kendala dalam mengambil kebijakan untuk meramalkan jumlah pipa yang layak pakai karena perencanaan yang efektif dan efisien harus didukung oleh sistem prediksi yang akurat, maka dibutuhkan sebuah metode untuk mengatasi masalah tersebut.

Sebelumnya terdapat penelitian yang berkaitan dengan algoritme *Extreme Learning Machine* (ELM) yang membahas tentang prediksi *laser cutting* terkena zona panas oleh Aninic dkk. (2017). Dalam penelitiannya didapatkan bahwa algoritme *Extreme Learning Machine* mampu memberikan prediksi yang baik dan akurat dibandingkan dengan algoritme *Genetic Programming*. *Extreme Learning Machine* (ELM) juga diterapkan pada prediksi penjualan mie oleh Giusti dkk (2017) yang membahas tentang prediksi penjualan mie berdasarkan *time series* penjualan tiap harinya dan data sisa penjualan.

Penelitian terkait juga dilakukan oleh Mladenovic dkk. (2016) yang membahas tentang prediksi pertumbuhan ekonomi berdasarkan perdagangan, impor dan ekspor yang menghasilkan tingkat prediksi pada *Extreme Learning Machine* lebih tinggi dibandingkan dengan algoritme *Back Propagation*.

Penelitian lainnya masih dengan topik prediksi oleh Fachrony dkk (2018) yang membahas tentang prediksi beban pemanasan dan beban pendinginan berdasarkan *Relative Compactness*, *Surface Area*, dan *Wall Area* yang merupakan faktor desain bangunan. Penelitian ini menghasilkan prediksi *Heating Load* (HL)

dan *Cooling Load* (CL) dengan waktu proses 0,0176 detik.

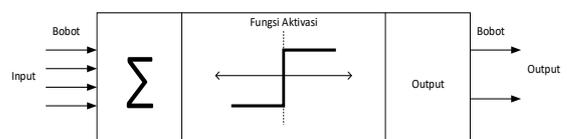
Berdasarkan permasalahan di atas, solusi yang memungkinkan adalah melakukan prediksi terhadap jumlah produk yang layak. Algoritme *Extreme Learning Machine* (ELM) adalah metode dari Jaringan Saraf Tiruan (JST), merupakan algoritme pembelajaran yang tepat untuk permasalahan ini, karena memiliki kelebihan pada *learning speed*, serta mempunyai tingkat kesalahan yang rendah dan diharapkan dapat memperoleh hasil prediksi yang lebih efektif.

## 2. DASAR TEORI

### 2.1. Jaringan Saraf Tiruan

Jaringan saraf tiruan adalah satu paradigma pemrosesan informasi yang diilhami dari cara kerja sistem ingatan makhluk hidup, seperti proses penyaluran informasi di dalam otak. Sistem ini terbuat dari beberapa elemen pemroses yang berhubungan satu sama lain yang bekerja untuk menyelesaikan suatu permasalahan. Sistem pembelajaran dalam kehidupan makhluk hidup melibatkan pembetulan (kesalahan) pada hubungan sinaptik yang terjadi di antara *neuron*.

Di dalam memproses informasi, JST melakukan pemrosesan informasi seperti yang dilakukan oleh otak manusia. JST dibuat dari sejumlah besar unsur pemroses yang saling berhubungan yang bekerja secara selaras yang berfungsi seperti neuron untuk menyelesaikan suatu masalah yang khusus. Gambar 1 merupakan struktur Jaringan Saraf Tiruan.



Gambar 1 Struktur Jaringan Saraf Tiruan  
Sumber: (Budiharto & Suhartono, 2014)

### 2.1. *Extreme Learning Machine* (ELM)

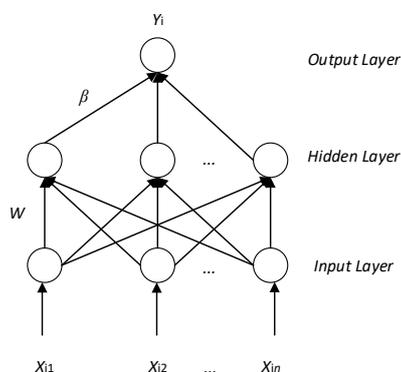
*Extreme Learning Machine* (ELM) adalah algoritme pelatihan dari jaringan saraf tiruan. ELM merupakan jaringan saraf tiruan (JST) *feedforward* dengan *single hidden layer* atau biasa disebut dengan *Single Hidden Layer Feedforward Neural Networks* (SLFNs) (Sun, et al., 2008).

Metode pembelajaran ELM dibuat untuk mengatasi permasalahan yang disebabkan oleh

jaringan saraf tiruan *feedforward* terutama dalam hal *learning speed*. Huang dkk. pada tahun 2006 menjelaskan bahwa ada dua alasan mengapa JST *feedforward* memiliki *learning speed* rendah, yaitu:

1. Menggunakan *slow gradient based learning algorithm* dalam melakukan *training*.
2. Semua parameter pada jaringan ditentukan secara *iterative* dengan pembelajaran tersebut.

Pada metode pembelajaran *Conventional gradient based learning algorithm* seperti *Backpropagation* (BP) dan variannya *Lavenberg Marquadt* (LM) semua parameter pada JST *feedforward* harus ditentukan secara manual. Parameter yang dimaksud adalah *input weight* dan *hidden bias*. Sedangkan pada ELM parameter seperti *input weight* dan *hidden bias* dipilih secara acak, sehingga ELM mempunyai *learning speed* yang baik dan dapat memberikan *good generalization performance* (Huang, et al., 2006). Gambar 2 merupakan struktur dari ELM.



Gambar 2 Struktur *Extreme Learning Machine*  
 Sumber: (Giusti, et al., 2017)

Proses penyelesaian masalah prediksi dengan *Extreme Learning Machine* (ELM) diawali dengan normalisasi data, kemudian melakukan proses *training*, lalu proses *testing*, selanjutnya dilakukan denormalisasi data dan perhitungan nilai *error* dengan menggunakan *Mean Square Error* (MSE).

**a. Normalisasi Data**

Normalisasi data dilakukan karena rentang nilai *input* tidak sama. *Input* akan diproses ke nilai *output* yang kecil sehingga data yang digunakan harus disesuaikan agar dapat diproses untuk mendapatkan nilai normalisasi yang kecil. Dalam penelitian ini, data yang digunakan akan

disesuaikan dengan cara menormalisasi data. Persamaan 1 adalah proses normalisasi data menggunakan metode *Min-Max Normalization* (Jain & Bhandare, 2011):

$$d' = \frac{d - \min}{\max - \min} \tag{1}$$

Keterangan:

- $d'$  = nilai dari hasil normalisasi data
- $d$  = nilai asli data
- $\min$  = nilai terkecil pada data set fitur  $X$
- $\max$  = nilai terbesar pada data set fitur  $X$

**b. Proses Training**

Proses *training* harus terlebih dahulu dilakukan sebelum proses prediksi. Proses *training* bertujuan untuk mendapatkan nilai *output weight*. Langkah-langkah yang dilakukan dalam proses *training* adalah (Huang, et al., 2006):

1. Langkah pertama adalah menginisialisasi *input weight* dan bias. Nilai ini diinisialisasi secara acak dengan rentang nilai antara -1 hingga 1.
2. Langkah selanjutnya adalah menghitung keluaran *hidden layer* ( $H_{init}$ ) (Fachrony, et al., 2018). Persamaan 2 berikut untuk menghitung keluaran di *hidden layer*.

$$H_{init \ train} = X_{train} \cdot W^T + b \tag{2}$$

Keterangan:

- $H_{init \ train}$  = Matriks hasil keluaran *hidden layer* untuk proses *Training*
- $X_{train}$  = *Input data* menggunakan data *training*
- $W^T$  = *Transpose Input weight*
- $b$  = Nilai bias

Setelah nilai  $H_{init}$  didapatkan dan dihitung menggunakan fungsi aktivasi *sigmoid* biner, fungsi aktivasi ini sangat cocok untuk menyelesaikan permasalahan yang kompleks dan bersifat non-linier. Rumus fungsi aktivasi *sigmoid* biner diperlihatkan dalam Persamaan 3.

$$H = \frac{1}{1 + \exp(-H_{init \ train})} \tag{3}$$

Keterangan:

- $H$  = Fungsi aktivasi *sigmoid* biner
- $\exp$  = Eksponensial

$H_{init\ train}$  = Matriks keluaran *hidden layer* pada proses *training*

- Menghitung *output weight*. Agar mendapatkan hasil *output weight*, hal pertama yang harus dilakukan adalah menstranspose matriks hasil keluaran *hidden layer* dengan fungsi aktivasi. Setelah itu, matriks *transpose* tersebut dikalikan dengan matriks hasil keluaran *hidden layer* dengan fungsi aktivasi *sigmoid* biner biasa disebut matriks  $H$ . Setelah itu, langkah berikutnya adalah menghitung nilai *invers* dari matriks yang telah diperoleh sebelumnya. Setelah itu menghitung matriks *Moore-Penrose Generalized Invers* dari hasil keluaran *hidden layer* dengan fungsi aktivasi. Persamaan 4 berikut ini untuk mencari matriks *Moore-Penrose Generalized Inverse*.

$$H^+ = (H^T H)^{-1} H^T \tag{4}$$

Keterangan:

$H^+$  = *Moore-Penrose Generalized Inverse*

$H^T$  = Matriks  $H$  *transpose*

$H$  = Matriks  $H$  keluaran *hidden layer*

Persamaan 5 berikut ini untuk menghitung nilai *output weight*.

$$\hat{\beta} = H^+ \cdot Y \tag{5}$$

Keterangan:

$\hat{\beta}$  = Matriks *Output Weight*

$H^+$  = Matriks *Moore-Penrose Generalized Invers* dari matriks  $H$

$Y$  = Matriks *Target*

**c. Proses *Testing***

Proses ini bertujuan untuk mengevaluasi metode ELM dari hasil proses *training* sebelumnya. Proses *testing* menggunakan *input weight*, bias dan *output weight* yang didapatkan dari proses *training* dalam pelaksanaannya. Langkah-langkah yang dilakukan dalam proses *testing* adalah sebagai berikut:

- Langkah pertama adalah menginisialisasi *input weight* dan bias yang telah didapatkan dari proses *training*.
- Langkah selanjutnya adalah menghitung keluaran *hidden layer*. Proses perhitungan menggunakan Persamaan 6 dengan menggunakan data *testing*.

$$H_{init\ test} = X_{test} \cdot W^T + b \tag{6}$$

Keterangan:

$H_{init\ test}$  = Matriks hasil keluaran *hidden layer* untuk proses *Testing*

$X_{test}$  = *Input data* menggunakan data *testing*

$W^T$  = *Transpose Input weight*

$b$  = Nilai bias

- Keluaran di *hidden layer* dihitung menggunakan fungsi aktivasi dengan Persamaan 7.

$$H = \frac{1}{1 + \exp(-H_{init\ test})} \tag{7}$$

Keterangan:

$H$  = Fungsi aktivasi *sigmoid* biner

$exp$  = Eksponensial

$H_{init\ test}$  = Matriks keluaran *hidden layer* pada proses *testing*

- Nilai *output weight* yang telah didapatkan pada proses *training* digunakan untuk menghitung keluaran *output layer* yang merupakan hasil prediksi. Persamaan 8 berikut untuk menghitung nilai *output layer*.

$$\hat{Y} = H \cdot \hat{\beta} \tag{8}$$

Keterangan:

$\hat{Y}$  = *Output layer* yang merupakan hasil prediksi

$\hat{\beta}$  = Nilai *output weight* dari proses *training*

$H$  = Keluaran di *hidden layer* dihitung dengan fungsi aktivasi

**d. Denormalisasi**

Proses ini berfungsi untuk membangkitkan nilai yang telah dinormalisasi menjadi nilai asli. Persamaan 9 di bawah ini digunakan untuk melakukan proses denormalisasi data.

$$d = d'(\max - \min) + \min \tag{9}$$

Keterangan:

$d'$  = Nilai hasil prediksi sebelum didenormalisasi

$d$  = Nilai asli setelah di denormalisasi

$\min$  = Nilai terkecil pada data fitur  $X$

$\max$  = Nilai terbesar pada data fitur  $X$

**e. Mean Square Error (MSE)**

Hal terakhir yang harus dilakukan adalah menghitung nilai *error* semua *output layer* yang telah didenormalisasi. Nilai *error* ini

menunjukkan nilai kesalahan dari hasil prediksi yang didapatkan.

Mean Square Error (MSE) digunakan untuk mengevaluasi hasil prediksi. Persamaan 10 berikut ini digunakan untuk menghitung nilai error pada hasil prediksi.

$$MSE = \frac{\sum_{i=1}^n e_i^2}{n} = \frac{\sum_{i=1}^n (y_i - t_i)^2}{n} \tag{10}$$

Keterangan:

$n$  = Jumlah data

$i$  = [1,2,...,  $n$ ],  $n$  adalah keseluruhan jumlah data

$e$  = Error

$y$  = Nilai output (prediksi)

$t$  = Nilai aktual

### 3. METODE

#### 3.1. Formulasi Data

Sampel data yang digunakan sebanyak 270 data permintaan dan modal. Prediksi jumlah produksi pipa yang layak menggunakan 5 fitur yaitu OD (kode untuk diameter pipa), THK (kode untuk ketebalan pipa), LENGTH (kode untuk panjang pipa), QR (kode untuk kuantitas permintaan pelanggan), dan COIL (kode untuk bahan baku pipa), serta menggunakan ACC (kode untuk jumlah pipa yang layak) sebagai target (*output layer*). Proses perhitungan pada ELM menggunakan parameter *hidden neuron*, *input weight*, *bias*, dan persentase perbandingan data *training* dan data *testing*. Tabel 1 merupakan sampel data dari penelitian ini.

Tabel 1 Sampel Data Permintaan dan Modal

OD (mm)	THK (mm)	LENGTH (m)	QR (kg)	COIL (kg)	ACC (kg)
216,3	7	5,35	115975,2	128340	122353,9
216,3	7	6,05	109291,3	104220	99892,2
216,3	4,8	6,45	40368,6	47040	44889,9
216,3	4,8	6,05	46498,4	70640	66945,6
216,3	6	6,05	282377,5	299520	285201,3

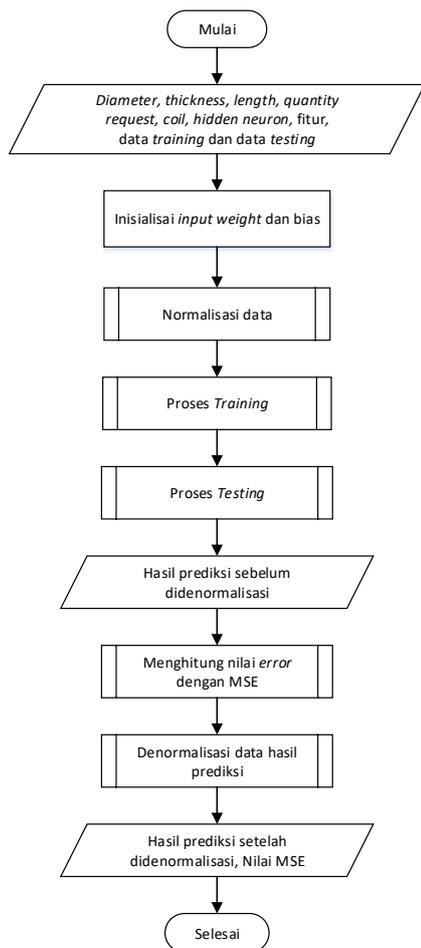
#### 3.1. Perancangan Sistem

Pada penelitian ini, proses penyelesaian masalah dengan menggunakan algoritme ELM dengan memasukkan data diameter, ketebalan, panjang, kuantitas, dan bahan baku, juga memasukkan jumlah fitur dan *hidden neuron*, serta persentase jumlah data *training* dan data *testing* sebagai input awal.

*Input weight* dan bias akan diproses secara acak dengan rentang nilai antara -1 sampai 1 berdasarkan jumlah *neuron* kemudian dilakukan *transpose* matriks. Selanjutnya melakukan normalisasi data menggunakan Persamaan 1. Untuk menghitung keluaran *hidden layer* digunakan Persamaan 2 dan menggunakan fungsi aktivasi pada Persamaan 3 untuk memetakan hasil keluaran *hidden layer* tersebut. Hasil matriks tersebut kemudian dikalikan dengan *transpose* matriksnya. Setelah itu menghitung matriks *Moore-Penrose Generalized Inverse* menggunakan Persamaan 4. Hasil matriks *Moore-Penrose Generalized Inverse* kemudian dikalikan dengan *transpose* matriks keluaran *hidden layer* yang telah diaktivasi. Selanjutnya menghitung hasil keluaran dari proses *training* menggunakan Persamaan 5, hasil keluaran berupa *output weight* digunakan pada proses *testing*.

Pada proses *testing*, *input weight* dan bias telah diperoleh dari proses *training*, keluaran *hidden layer* dihitung menggunakan Persamaan 6 serta menggunakan fungsi aktivasi pada Persamaan 7 menggunakan *input* data pada data *testing*. Nilai *output weight* yang telah didapatkan pada proses *training* digunakan pada proses *testing* untuk menghitung keluaran *output layer* yang merupakan hasil prediksi menggunakan Persamaan 8.

Sebelum proses denormalisasi, terlebih dahulu menghitung nilai *error*, perhitungan nilai *error* pada semua *output layer* yang belum dinormalisasi dengan data aktual. Nilai *error* ini menunjukkan nilai kesalahan dari hasil prediksi yang didapatkan. Pengujian terhadap nilai *error* menggunakan *Mean Square Error* (MSE) dengan Persamaan 10. Hal terakhir yang harus dilakukan adalah melakukan denormalisasi menggunakan Persamaan 9 untuk membangkitkan nilai yang sebelumnya telah dinormalisasi dapat kembali menjadi nilai asli. Gambar 3 merupakan diagram alir proses penyelesaian permasalahan dengan ELM.



Gambar 3 Diagram Alir Proses ELM

#### 4. PENGUJIAN DAN ANALISIS

Pada dasarnya, pengujian dilakukan untuk mendapatkan pola pengenalan yang lebih baik dengan menggunakan parameter terbaik. Dalam penelitian ini, proses pengujian dibagi menjadi 3 bagian pengujian, yaitu pengujian jumlah *neuron* pada *hidden layer*, pengujian jumlah fitur data, dan pengujian jumlah data *training* dan data *testing*.

Pada pengujian ini, proses evaluasi yang dilakukan didasari oleh nilai MSE yang didapatkan setiap menjalankan program. Nilai *error* yang didapatkan selalu berbeda, hal ini disebabkan karena nilai dari *input weight* dan bias diinisialisasi secara acak. Maka dari itu, pada pengujian ini dilakukan 10 kali percobaan pengujian dan dihitung rata-rata dari semua percobaan pengujian.

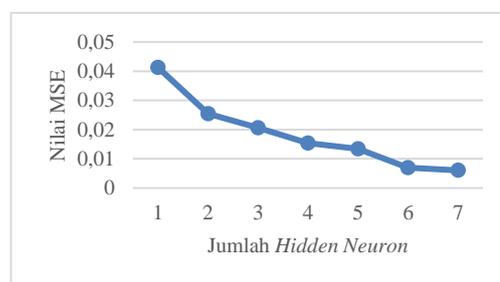
##### 4.1. Pengujian Jumlah *Neuron* pada *Hidden Layer*

Hasil dari pengujian jumlah *neuron* pada

*hidden layer* memperlihatkan bahwa semakin banyak jumlah *neuron*, maka nilai *error* yang dihasilkan akan semakin kecil. Hal itu dapat dibuktikan dengan hasil yang didapatkan ketika pengujian menggunakan jumlah *neuron* 7, nilai rata-rata MSE yang dihasilkan adalah sebesar 0,0060. Padahal, ketika jumlah *neuron* yang digunakan pada *hidden layer* 1 didapatkan rata-rata nilai MSE sebesar 0,0413 yang berarti nilai tersebut tinggi.

Hal ini dapat terjadi karena adanya *underfitting*. *Underfitting* terjadi ketika proses *training* tidak dapat menangkap pola pembelajaran dengan baik karena pemroses *input* atau *hidden neuron* berjumlah sedikit, sehingga ketika proses *training* dilakukan pola yang ditangkap menjadi terbatas dan menyebabkan *error* yang tinggi ketika proses *testing* dijalankan (Sheela & Deepa, 2013).

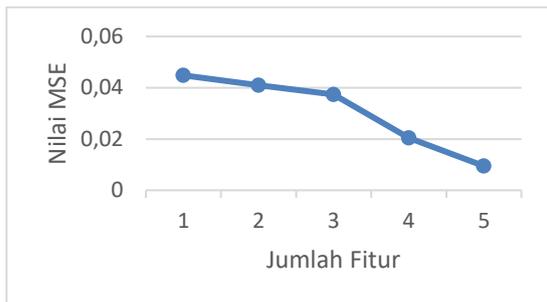
Kesimpulannya, semakin banyak jumlah *neuron*, semakin banyak pula penghubung antara *input layer* ke *output layer* sehingga pengenalan pola yang dilakukan oleh sistem akan semakin baik. Gambar 4 menunjukkan grafik hasil pengujian jumlah *neuron* pada *hidden layer*.



Gambar 4 Hasil Pengujian Jumlah *Neuron* pada *Hidden Layer*

##### 4.2. Pengujian Jumlah Fitur

Pada pengujian jumlah fitur, nilai MSE terendah didapatkan pada jumlah fitur sebanyak 5 sebesar 0,00946. Nilai MSE tinggi pada saat pengujian dengan jumlah fitur 1 sebesar 0,04483 dikarenakan pengenalan pola yang diperoleh terlalu sedikit sehingga sistem sulit melakukan pembelajaran dari pola tersebut dan mendapatkan hasil prediksi yang kurang optimal dengan tingkat kesalahan yang tinggi. Kondisi yang terjadi ini disebut *underfitting*. *Underfitting* mengakibatkan sistem tidak dapat melakukan pengenalan dengan baik. Gambar 5 merupakan grafik hasil pengujian jumlah fitur.



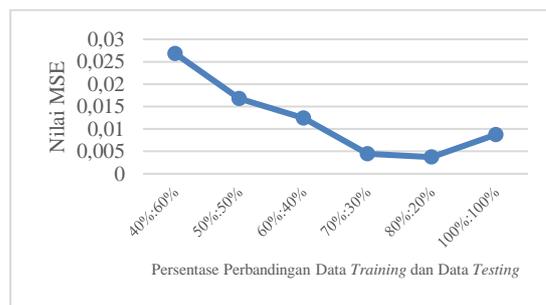
Gambar 5 Hasil Pengujian Jumlah Fitur

### 4.3. Pengujian Jumlah Data Training dan Data Testing

Pada pengujian jumlah data *training* dan data *testing*, nilai MSE terkecil didapatkan ketika jumlah data *training* sebesar 80% terhadap data *testing* sebesar 20% dari *dataset* dengan rata-rata nilai *error* sebesar 0,00372. Dari hasil pengujian ini dapat disimpulkan bahwa jumlah data *training* berpengaruh terhadap nilai *error* yang dihasilkan untuk jumlah data *testing* sebesar 20%. Selain itu, jika jumlah data *training* lebih kecil daripada data

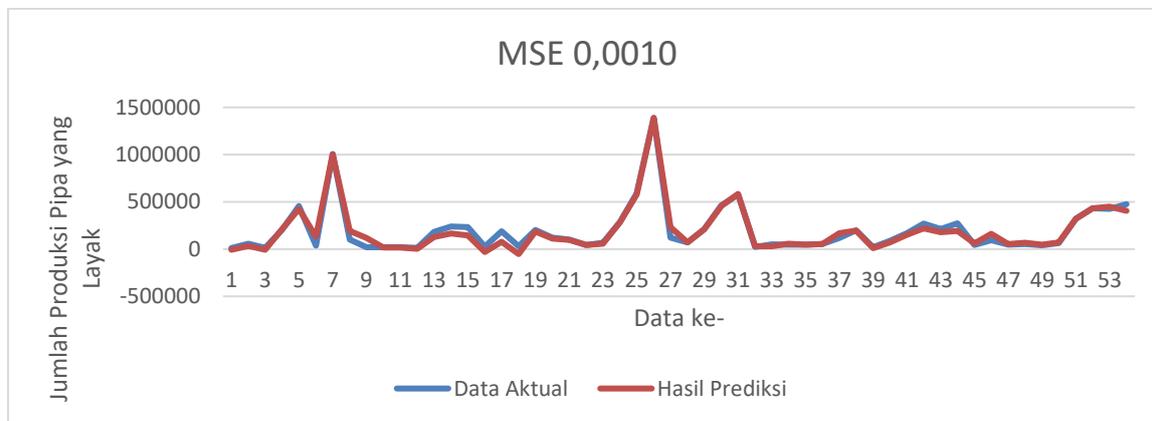
*testing*, maka nilai *error* akan semakin meningkat karena terjadinya *underfitting*.

*Underfitting* terjadi karena jumlah data *training* yang lebih sedikit dibandingkan jumlah data *testing* menyebabkan proses *training* tidak mampu melakukan pengenalan pola dengan baik. Metode ELM merupakan metode pembelajaran, banyaknya jumlah data *training* akan memengaruhi seberapa baiknya prediksi yang dihasilkan. Gambar 7 merupakan hasil pengujian jumlah data *training* dan data *testing*.



Gambar 6 Hasil Pengujian Jumlah Data Training dan Data Testing

### 4.4 Perbandingan Data Aktual dan Hasil Prediksi



Gambar 7 Grafik Perbandingan Data Aktual dan Hasil Prediksi

Gambar 7 merupakan perbandingan data aktual dan hasil prediksi dengan nilai MSE 0.0010 dan jumlah data *testing* sebanyak 20% dari total keseluruhan data dengan jumlah *neuron* pada *hidden layer* sejumlah 7, 5 fitur, dan data *training* yang optimal yaitu 80%. Grafik tersebut menunjukkan pola hasil prediksi cenderung mengikuti pola dari data aktual. Hasil prediksi jumlah produksi pipa yang layak pada data *testing* ke-52 sebanyak 432.055,5 Kg memiliki selisih terendah yaitu 170,46 Kg dengan data aktual sebanyak 432.225,96 Kg.

Sedangkan data *testing* ke-17 dengan hasil prediksi 76.591,04 Kg dan data aktual 189.932,82 Kg memiliki selisih tertinggi yaitu 103.256Kg. Serta dengan selisih rata-rata ± 1% atau sejumlah 29.008,33 Kg

### KESIMPULAN

Berdasarkan hasil pengujian dan pembahasan dari prediksi jumlah produksi pipa yang layak menggunakan metode *Extreme Learning Machine* (ELM) maka diperoleh beberapa kesimpulan, yaitu:

1. Algoritme *Extreme Learning Machine* (ELM) mampu menyelesaikan permasalahan prediksi jumlah pipa yang layak dengan nilai *error* yang rendah. Proses implementasi algoritme *Extreme Learning Machine* (ELM) bermula dari inialisasi *input weight* dan bias secara acak, lalu melakukan normalisasi untuk menyetarakan nilai pada data, selanjutnya melakukan proses *training*, dilanjutkan dengan proses *testing*, juga menghitung nilai *error* dengan menggunakan *Mean Square Error* (MSE), dan terakhir melakukan denormalisasi untuk mengembalikan nilai aktual setelah proses perhitungan.
2. Tingkat kesalahan berdasarkan hasil pengujian yang diukur dengan *Mean Square Error* (MSE) pada hasil prediksi jumlah produksi pipa yang layak adalah:
  - c. Jumlah *neuron* mempengaruhi hasil perhitungan. Hal ini dibuktikan dengan jumlah *neuron* sebesar 7 menghasilkan nilai *error* yang rendah, yaitu sebesar 0,00606.
  - d. Banyaknya jumlah variasi fitur data akan menghasilkan prediksi yang baik, karena proses prediksi menggunakan ELM mengandalkan objek dan fitur yang digunakan. Nilai *error* tertinggi diperoleh dengan jumlah fitur 1 sejumlah 0,04483 dan turun pada jumlah fitur 5 sejumlah 0,00946.
  - e. Jumlah data *training* berpengaruh terhadap hasil keluaran prediksi untuk data *testing* sebesar 20%. Berdasarkan pengujian dan pembahasan yang telah dilakukan, jumlah data *training* 80% menghasilkan rata-rata kesalahan terkecil sejumlah 0,00372.
3. Perbandingan data aktual dan hasil prediksi 20% dari total 270 keseluruhan data memiliki selisih rata-rata  $\pm 1\%$  atau sejumlah 29.008,33 Kg dengan nilai MSE 0,0010.
4. Pada penelitian mendatang diharapkan adanya penggabungan metode dengan *Extreme Learning Machine* (ELM) seperti metode optimasi atau metode kecerdasan buatan lainnya untuk memperoleh hasil prediksi yang lebih optimal dengan selisih yang minimal.

## DAFTAR PUSTAKA

- Anicic, O., Jović, S., Skrijelj, H. & Nedić, B., 2017. Prediction of laser cutting heat affected zone by extreme learning machine. *Optics and Lasers in Engineering*, pp. 1-4.
- Budiharto, W. & Suhartono, D., 2014. *Artificial Intelligence Konsep dan Penerapannya*. Yogyakarta: Andi.
- Fachrony, A., Cholissodin, I. & Santoso, E., 2018. Implementasi Algoritme *Extreme Learning Machine* (ELM) untuk Prediksi Beban Pemanasan dan Pendinginan Bangunan. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(9), pp. 3045-3046.
- Giusti, A., Widodo, A. W. & Adinugroho, S., 2017. Prediksi Penjualan Mi Menggunakan Metode *Extreme Learning Machine* (ELM) di Kober Mie Setan Cabang Soekarno Hatta. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Agustus, Volume 2, pp. 2972-2978.
- Huang, G. B., Zhu, Q. Y. & Siew, C. K., 2006. *Extreme learning machine: Theory and applications*. *Neurocomputing*, Volume 70, pp. 489-501.
- Jain, Y. K. & Bhandare, S. K., 2011. Min Max Normalization Based Data Perturbation Method for Privacy Protection. *International Journal of Computer & Communication Technology*, Volume 2, pp. 45-50.
- Mladenovic, S. S., Milovancevic, M., Mladenovic, I. & Alizamir, M., 2016. Economic growth forecasting by artificial neural network with extreme learning machine based on trade, import, and export parameters. *Computers in Human Behavior*, pp. 43-45.
- Sheela, K. G. & Deepa, S. N., 2013. Review on methods to fix number of hidden neurons in neural networks. *Mathematical Problems in Engineering*.
- 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*, Agustus, pp. 411-419.