

**PREDIKSI NILAI HARGA PATOKAN BATU BARA (HPB)  
UNTUK MEREK DAGANG GUNUNG BAYAN I  
DENGAN METODE *EXTREME LEARNING MACHINE* (ELM)**

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:  
Evilia Nur Harsanti  
NIM: 145150207111056



PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018

# PENGESAHAN

**PREDIKSI NILAI HARGA PATOKAN BATU BARA (HPB)  
UNTUK MEREK DAGANG GUNUNG BAYAN I  
DENGAN METODE *EXTREME LEARNING MACHINE* (ELM)**

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :  
**Evilia Nur Harsanti**  
NIM: 145150207111056

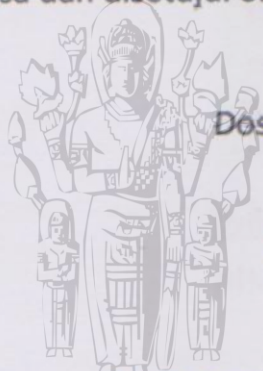
Skripsi ini telah diuji dan dinyatakan lulus pada  
23 Juli 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing 1

Dosen Pembimbing 2

Muhammad Tanzil Furgon S.Kom,  
M.CompSc  
NIP:19820930 200801 1 004



Putra Pandu Adikara S.Kom, M.Kom  
NIP: 19850725 200812 1 002

Mengetahui,

Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T, M.T, Ph.D  
NIP: 19710518 20012 1 001



## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

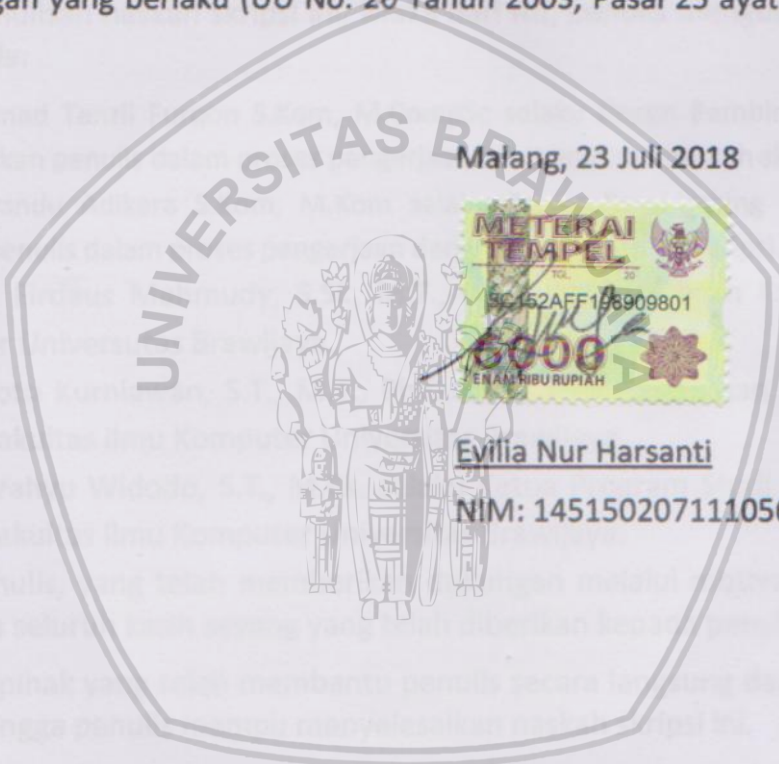
Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 23 Juli 2018



Evilia Nur Harsanti

NIM: 145150207111056



## KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT atas segala rahmat dan karunia-Nya, penulis dapat menyelesaikan skripsi dengan judul “Prediksi Nilai Harga Patokan Batu Bara (HPB) Untuk Merek Dagang Gunung Bayan I Dengan Metode *Extreme Learning Machine* (ELM)”

Tujuan dari penulisan naskah skripsi ini adalah sebagai syarat untuk memperoleh gelar Sarjana Komputer di Fakultas Ilmu Komputer Universitas Brawijaya.

Penulis menyadari keterbatasan pengetahuan yang penulis miliki, karena itu tanpa keterlibatan dan sumbangsih dari berbagai pihak, sulit bagi penulis untuk menyelesaikan penulisan naskah skripsi ini. Maka dari itu, penulis mengucapkan terima kasih kepada:

1. Bapak Muhammad Tanzil Furqon S.Kom, M.CompSc selaku Dosen Pembimbing I yang mengarahkan penulis dalam proses pengerjaan dan penulisan naskah skripsi.
2. Bapak Putra Pandu Adikara S.Kom, M.Kom selaku Dosen Pembimbing li yang mengarahkan penulis dalam proses pengerjaan dan penulisan naskah skripsi.
3. Bapak Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D. selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya.
4. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D. selaku Ketua Jurusan Teknik Informatika, Fakultas Ilmu Komputer Universitas Brawijaya.
5. Bapak Agus Wahyu Widodo, S.T., M.Cs. selaku Ketua Program Studi Teknik Informatika, Fakultas Ilmu Komputer Universitas Brawijaya.
6. Orang tua penulis, yang telah memberikan dukungan melalui motivasi dan material, serta seluruh kasih sayang yang telah diberikan kepada penulis.
7. Serta seluruh pihak yang telah membantu penulis secara langsung dan tidak langsung, sehingga penulis mampu menyelesaikan naskah skripsi ini.

Penulis menyadari bahwa selama penyusunan naskah skripsi, penulis masih memiliki banyak kekurangan. Untuk itu penulis mengharapkan kritik dan saran yang bersifat membangun dari pembaca demi penyempurnaan.

Akhir kata, penulis berharap semoga penulisan naskah skripsi ini dapat bermanfaat.

Malang, Juli 2018

Penulis

evilia.right@gmail.com

## ABSTRAK

**Evilia Nur Harsanti, Prediksi Nilai Harga Patokan Batu Bara (HPB) Untuk Merek Dagang Gunung Bayan I dengan Metode *Extreme Learning Machine* (ELM)**

**Dosen Pembimbing: Muhammad Tanzil Furqon dan Putra Pandu Adikara**

Batu bara merupakan bahan bakar fosil yang sering dimanfaatkan oleh perusahaan industri sebagai sumber energi pembangkit tenaga listrik maupun sebagai bahan baku pembuatan baja. Terdapat beberapa jenis batu bara yang dikelompokkan berdasarkan tingkat kematangan dan komposisi mineralnya. Komposisi mineral dalam jumlah tertentu akan sangat memengaruhi kualitas batu bara, khususnya untuk beberapa parameter seperti abu, sulfur, dan nilai panas batu bara.

Batu bara didapatkan oleh perusahaan industri melalui suatu transaksi jual beli yang dilakukan dengan perusahaan pertambangan batu bara. Harga merupakan faktor utama dalam proses transaksi tersebut, karena perusahaan industri perlu merancang anggaran pengeluaran setiap bulan sebelum melakukan transaksi. Oleh karena itu, prediksi harga batu bara akan sangat bermanfaat untuk perusahaan industri yang akan membeli produk batu bara untuk mengetahui perkiraan harga di masa yang akan datang. Metode yang digunakan untuk melakukan proses prediksi adalah metode *Extreme Learning Machine* (ELM). ELM memiliki keunggulan waktu komputasi yang cepat dan tingkat kesalahan yang kecil, sehingga ELM tidak memerlukan waktu yang lama dalam proses pembelajaran. Berdasarkan hasil penelitian, nilai *Means Absolute Percentage Error* (MAPE) terbaik adalah sebesar 3,926804% untuk proses *training* dan 7,360343% untuk proses *testing*. Hasil pengujian tersebut didapatkan dengan parameter jumlah fitur sebanyak 3, jumlah *neuron* sebanyak 7, dan jumlah fungsi aktivasi yang digunakan adalah fungsi *sigmoid biner*.

**Kata kunci:** Batu bara, Harga, Prediksi, *Extreme Learning Machine*, *Means Absolute Percentage Error*

## ABSTRACT

**Evilia Nur Harsanti, Prediksi Nilai Harga Patokan Batu Bara (HPB) Untuk Merek Dagang Gunung Bayan I dengan Metode *Extreme Learning Machine* (ELM)**

**Supervisors: Muhammad Tanzil Furqon dan Putra Pandu Adikara**

*Coal is a fossil fuel that is often used by industrial companies as a source of energy and power as a raw material for making steel. There are several types of coal that are grouped according to their maturity level and mineral composition. The composition of minerals in a certain amount will greatly affect the quality of coal, especially for some parameters such as ash, sulfur, and coal heat value.*

*Coal is obtained by industrial companies through a sale and purchase transactions conducted with coal mining companies. Price is a major factor in the transaction process, because industrial companies need to design an expenditure budget every month before making a transaction. Therefore, the prediction of coal price will be very beneficial for industrial companies that will buy coal products to find out the estimated price in the future. The method used to make the prediction process is the method of *Extreme Learning Machine* (ELM). ELM has the advantage of fast computing time and small error rate, so ELM does not require a long time in the learning process. Based on the result of research, the best *Means Absolute Percentage Error* (MAPE) score is 3,926804% for training process and 7,360343% for testing process. The test results were obtained with the feature number parameters of 3, the number of neurons as much as 7, and the number of activation functions used is a binary sigmoid function.*

**Keywords:** *Coal, Price, Prediction, Extreme Learning Machine, Means Absolute Percentage Error*

## DAFTAR ISI

PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
<i>ABSTRACT</i> .....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	xi
DAFTAR KODE PROGRAM .....	xiv
DAFTAR LAMPIRAN .....	xv
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah .....	2
1.3 Tujuan.....	3
1.4 Manfaat.....	3
1.5 Batasan masalah .....	3
1.6 Sistematika pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN .....	5
2.1 Kajian Pustaka .....	5
2.2 Batu Bara.....	8
2.2.1 Penetapan Harga Batu Bara .....	9
2.3 Kecerdasan Buatan .....	9
2.4 <i>Machine Learning</i> .....	10
2.5 Jaringan Saraf Tiruan.....	11
2.5.1 Definisi Jaringan Saraf Tiruan .....	11
2.5.2 Arsitektur Jaringan Saraf Tiruan .....	11
2.5.3 Faktor Bobot.....	12
2.5.4 Fungsi Aktivasi .....	13
2.6 <i>Time Series Forecasting</i> .....	13
2.7 <i>Extreme Learning Machine (ELM)</i> .....	14

2.7.1	Definisi <i>Extreme Learning Machine</i> (ELM)	14
2.7.2	Arsitektur <i>Extreme Learning Machine</i>	14
2.7.3	<i>Moore Penrose Pseudo Inverse</i>	15
2.7.4	Proses <i>Training</i>	15
2.7.5	Proses <i>Testing</i>	17
2.8	Normalisasi dan Denormalisasi Data	17
2.9	Nilai Evaluasi	18
2.10	<i>Cross Validation Time Series</i>	18
BAB 3 METODOLOGI PENELITIAN		20
3.1	Tipe Penelitian	20
3.2	Studi Kepustakaan	20
3.3	Metode Pengumpulan Data	20
3.4	Strategi dan Lokasi Penelitian	21
3.5	Peralatan Pendukung	21
3.6	Analisis dan Perancangan Sistem	22
3.6.1	Model Perancangan Sistem	22
3.6.2	Diagram Alir Sistem	23
3.7	Implementasi	24
3.8	Pengujian Sistem	24
3.9	Evaluasi dan Analisis Hasil	24
BAB 4 PERANCANGAN DAN IMPLEMENTASI		25
4.1	Formulasi Permasalahan	25
4.2	Arsitektur Perancangan Sistem	26
4.3	Alur Kerja Sistem	27
4.3.1	Normalisasi	27
4.3.2	Inisialisasi <i>Input Weight</i> dan Bias	29
4.3.3	Proses <i>Training</i>	31
4.3.3.1	Proses Menghitung Keluaran <i>Hidden Layer</i>	33
4.3.3.2	Proses Menghitung Aktivasi Keluaran <i>Hidden Layer</i>	36
4.3.3.3	Proses Menghitung <i>Moore-Penrose Pseudo Inverse</i>	37
4.3.3.4	Proses Perhitungan Matriks <i>Output Weight</i>	42
4.3.4	Proses <i>Testing</i>	44





4.3.4.1	Proses Perhitungan <i>Output</i> Peramalan .....	45
4.3.4.2	Proses Perhitungan Denormalisasi .....	46
4.3.4.3	Proses Perhitungan <i>Mean Absolute Percentage Error</i> .....	47
4.4	Perhitungan Manual .....	48
4.5	Perancangan <i>User Interface</i> .....	58
4.5.1	Perancangan Halaman Awal .....	58
4.5.2	Perancangan Tabel Data, Pembentukan Fitur, dan Normalisasi .....	59
4.5.3	Perancangan Halaman Masukan Pengguna .....	59
4.6	Perancangan Uji Coba dan Evaluasi .....	60
4.6.1	Pengujian Jumlah <i>Neuron</i> .....	60
4.6.2	Pengujian Jumlah Fitur Data .....	60
4.6.3	Pengujian Fungsi Aktivasi .....	61
4.7	Implementasi Sistem .....	62
4.7.1	Implementasi Normalisasi Data .....	62
4.7.2	Implementasi Inisialisasi <i>Input Weight</i> .....	63
4.7.3	Implementasi Inisialisasi Bias .....	64
4.7.4	Implementasi <i>Transpose Input Weight</i> .....	64
4.7.5	Implementasi Hitung Keluaran <i>Hidden Layer (Hinit)</i> .....	65
4.7.6	Implementasi Hitung Aktivasi Keluaran <i>Hidden Layer (H)</i> .....	66
4.7.7	Implementasi Hitung <i>Inverse</i> Matriks .....	67
4.7.8	Implementasi Hitung <i>Output Weight</i> .....	68
4.7.9	Implementasi Hitung <i>Output</i> Peramalan .....	69
4.7.10	Implementasi Denormalisasi Data .....	70
4.7.11	Implementasi Hitung <i>Mean Absolute Percentage Error</i> .....	71
4.8	Implementasi Antarmuka .....	71
BAB 5	Hasil dan Analisis .....	75
5.1	Pengujian Jumlah Fitur .....	75
5.2	Pengujian Jumlah <i>Neuron</i> .....	77
5.3	Pengujian Fungsi Aktivasi .....	79
5.4	Perbandingan Harga Aktual dan Hasil Prediksi .....	79
BAB 6	Penutup .....	81



6.1 Kesimpulan..... 81  
6.2 Saran ..... 81  
DAFTAR PUSTAKA..... 83



## DAFTAR TABEL

Tabel 2.1 Daftar Kajian Pustaka .....	6
Tabel 2.2 Kriteria Nilai MAPE .....	18
Tabel 3.1 Contoh Data Harga Patokan Batu Bara .....	21
Tabel 4.1 Data Sampel Historis Harga Patokan Batu Bara .....	26
Tabel 4.2 Representasi Data <i>Training</i> .....	48
Tabel 4.3 Representasi Data <i>Testing</i> .....	49
Tabel 4.4 Nilai <i>Maximum</i> dan <i>Minimum</i> .....	49
Tabel 4.5 Normalisasi Data <i>Training</i> .....	50
Tabel 4.6 Normalisasi Data <i>Testing</i> .....	50
Tabel 4.7 Matriks <i>Input Weight</i> .....	51
Tabel 4.8 Matriks Bias .....	51
Tabel 4.9 Matriks Keluaran <i>Hidden Layer (Hinit)</i> Proses <i>Training</i> .....	51
Tabel 4.10 Matriks Hasil Fungsi Aktivasi .....	52
Tabel 4.11 Transpose Matriks Keluaran <i>Hidden Layer</i> Proses <i>Training</i> .....	53
Tabel 4.12 Perkalian Matriks .....	53
Tabel 4.13 Matriks Invers .....	54
Tabel 4.14 Matriks <i>Moore-Penrose Pseudo Inverse</i> .....	55
Tabel 4.15 Matriks <i>Output Weight</i> .....	55
Tabel 4.16 Matriks Keluaran <i>Hidden Layer (Hinit)</i> Proses <i>Testing</i> .....	56
Tabel 4.17 Matriks Hasil Fungsi Aktivasi .....	56
Tabel 4.18 Matriks Nilai Hasil Peramalan .....	57
Tabel 4.19 Matriks Denormalisasi Data <i>Testing</i> .....	57
Tabel 4.20 Nilai MAPE Proses <i>Testing</i> .....	58
Tabel 4.21 Pengujian Jumlah <i>Neuron</i> .....	60
Tabel 4.22 Pengujian Jumlah Fitur Data .....	61
Tabel 4.23 Pengujian Fungsi Aktivasi .....	61
Tabel 5.1 Pengujian Jumlah Fitur Terhadap Nilai MAPE (%) .....	76
Tabel 5.2 Pengujian Jumlah <i>Neuron</i> Terhadap Nilai MAPE (%) .....	78
Tabel 5.3 Pengujian Fungsi Aktivasi Terhadap Nilai MAPE (%) .....	78
Tabel 5.4 Perbandingan Harga Aktual dan Hasil Prediksi .....	80

## DAFTAR GAMBAR

Gambar 2.1 Arsitektur Jaringan Saraf Tiruan.....	12
Gambar 2.2 Arsitektur <i>Extreme Learning Machine</i> .....	15
Gambar 2.3 <i>Validation Time Series</i> .....	19
Gambar 3.1 Model Perancangan Sistem Prediksi Harga Batu Bara.....	22
Gambar 3.2 Diagram Alir Sistem .....	23
Gambar 4.1 Arsitektur Perancangan Sistem.....	26
Gambar 4.2 Diagram Alir Proses <i>Getmax</i> .....	27
Gambar 4.3 Diagram Alir Proses <i>Getmin</i> .....	28
Gambar 4.4 Diagram Alir Normalisasi Data .....	29
Gambar 4.5 Diagram Alir Inisialaisasi <i>Input Weight</i> .....	30
Gambar 4.6 Diagram Alir Inisialisasi Bias .....	31
Gambar 4.7 Diagram Alir Proses <i>Training</i> .....	32
Gambar 4.8 Diagram Alir Perhitungan Keluaran <i>Hidden Layer</i> .....	33
Gambar 4.9 Diagram Alir Transpose Matriks <i>Input Weight</i> .....	34
Gambar 4.10 Diagram Alir Perhitungan Keluaran <i>Hidden Layer</i> .....	35
Gambar 4.11 Diagram Alir Aktivasi Keluaran <i>Hidden Layer</i> .....	36
Gambar 4.12 Diagram Alir Perhitungan <i>Moore-Penrose Pseudo Inverse</i> .....	37
Gambar 4.13 Diagram Alir <i>Inverse</i> Matriks.....	41
Gambar 4.14 Diagram Alir <i>Output Weight</i> .....	43
Gambar 4.15 Diagram Alir Proses <i>Testing</i> .....	44
Gambar 4.16 Diagram Alir <i>Output</i> Peramalan .....	45
Gambar 4.17 Diagram Alir Denormalisasi.....	46
Gambar 4.18 Diagram Alir <i>Mean Absolute Percentage Error</i> .....	47
Gambar 4.19 Perancangan Halaman Data.....	58
Gambar 4.20 Perancangan Tabel Data, Pembentukan Fitur, dan Normalisasi ....	59
Gambar 4.21 Perancangan Halaman Masukan Pengguna.....	59
Gambar 4.22 Implementasi Halaman Awal .....	72
Gambar 4.23 Implementasi Halaman Masukan Pengguna .....	72
Gambar 4.24 Implementasi Tabel Data .....	73
Gambar 4.25 Implementasi Pembentukan Fitur .....	73

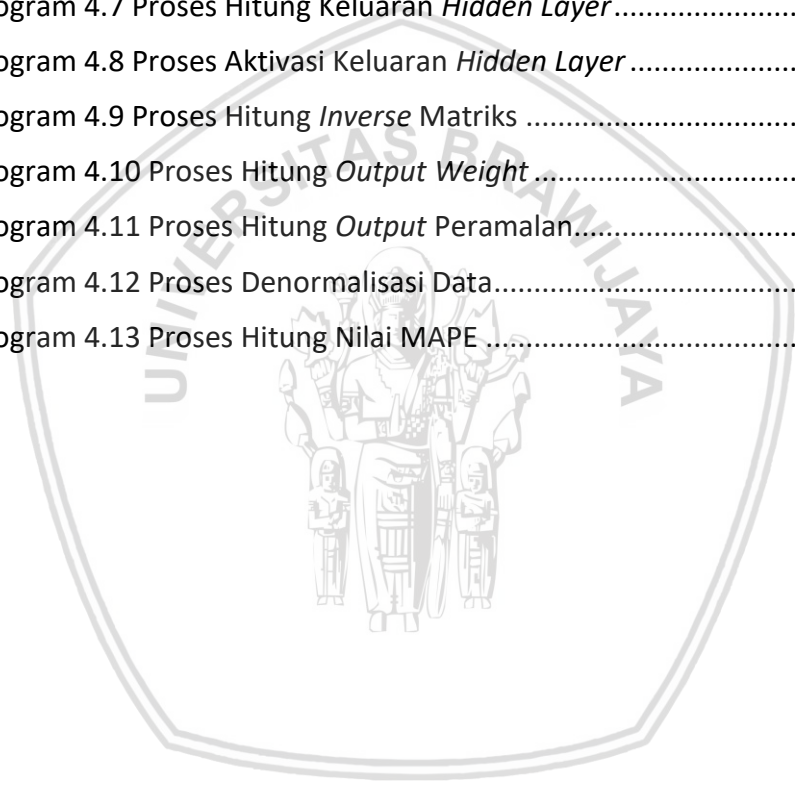


Gambar 4.26 Implementasi Normalisasi..... 74  
Gambar 5.1 Grafik Pengujian Jumlah Fitur Terhadap Nilai MAPE (%)..... 75  
Gambar 5.2 Grafik Pengujian Jumlah *Neuron* Terhadap Nilai MAPE (%) ..... 77  
Gambar 5.3 Grafik Pengujian Fungsi Aktivasi Terhadap Nilai MAPE (%)..... 79  
Gambar 5.4 Grafik Perbandingan Harga Aktual dan Hasil Prediksi ..... 80



## DAFTAR KODE PROGRAM

Kode Program 4.1 Proses <i>getmax()</i> .....	62
Kode Program 4.2 Proses <i>getmin()</i> .....	62
Kode Program 4.3 Proses Normalisasi Data.....	63
Kode Program 4.4 Proses Inisialisasi <i>Input Weight</i> .....	63
Kode Program 4.5 Proses Inisialisasi Bias .....	64
Kode Program 4.6 Proses <i>Transpose Input Weight</i> .....	65
Kode Program 4.7 Proses Hitung Keluaran <i>Hidden Layer</i> .....	66
Kode Program 4.8 Proses Aktivasi Keluaran <i>Hidden Layer</i> .....	66
Kode Program 4.9 Proses Hitung <i>Inverse</i> Matriks .....	68
Kode Program 4.10 Proses Hitung <i>Output Weight</i> .....	69
Kode Program 4.11 Proses Hitung <i>Output</i> Peramalan.....	70
Kode Program 4.12 Proses Denormalisasi Data.....	70
Kode Program 4.13 Proses Hitung Nilai MAPE .....	71



## DAFTAR LAMPIRAN

LAMPIRAN .....	85
A.1 Harga Patokan Batu Bara .....	85



## BAB 1 PENDAHULUAN

### 1.1 Latar belakang

Batu bara sebagai bahan bakar fosil yang memiliki berbagai macam kegunaan dalam dunia industri. Batuan ini diperkirakan telah terbentuk selama era *Carboniferous*, yaitu sekitar 298 – 358,9 juta tahun yang lalu dan terakumulasi di bawah badan air. Batuan ini berasal dari bahan organik dan tanaman mati.

Batu bara memiliki banyak kegunaan, salah satu kegunaannya yang sering dimanfaatkan oleh perusahaan industri adalah pemanfaatannya sebagai tenaga listrik. Dalam pemanfaatannya sebagai tenaga listrik, batu bara adalah sumber energi yang sangat terjangkau dan diandalkan untuk digunakan sebagai pembangkit tenaga listrik. Manfaat lain dari batu bara yang dapat digunakan dalam industri adalah pemanfaatan batu bara sebagai produk material logam, pupuk, bahan kimia, plastik, keramik, kertas, serta bahan baku dalam proses pembuatan baja.

Untuk mendapatkan batu bara, perusahaan industri harus memasok batu bara dengan cara membeli dari perusahaan pertambangan batu bara. Sebelum sebuah perusahaan industri membeli pasokan batu bara, maka perlu dilakukan perancangan terhadap anggaran pengeluaran perusahaan. Anggaran pengeluaran perusahaan biasanya disusun setiap bulan untuk merencanakan seluruh kegiatan perusahaan yang berlaku untuk jangka waktu tertentu di masa yang akan datang (Rahayuningsih, 2006). Untuk menyusun anggaran tersebut, maka dibutuhkan perkiraan harga batu bara untuk satu bulan berikutnya. Kebutuhan akan perkiraan harga batu bara dapat diatasi dengan melakukan proses prediksi yang dilakukan dengan proses pembelajaran data di masa lalu. Data yang digunakan untuk prediksi adalah data dalam bentuk *time series* (runtut waktu). Proses perhitungan harga patokan batu bara (HPB) saat ini mengacu pada harga batu bara acuan (HBA) dalam kesetaraan nilai kalor 6.322 kkal/kg GAR yang kemudian di standardisasi sesuai dengan kualitas dari masing-masing merek batu bara. HBA adalah rata-rata harga batu bara yang mengacu pada empat indeks harga batu bara, yaitu *new castle global coal index*, *new castle export index*, *platts index*, dan *Indonesia coal index*. Dengan menggunakan proses tersebut maka untuk dapat melakukan prediksi harga batu bara untuk merek dagang Gunung Bayan I harus menunggu nilai HBA terlebih dahulu untuk dapat di standardisasi sesuai dengan kualitas Gunung Bayan I. Nilai historis HPB yang sudah di standardisasi berdasarkan nilai HBA dapat digunakan sebagai acuan dalam proses prediksi harga untuk batu bara Gunung Bayan I. Metode pembelajaran seperti *machine learning* memiliki kemampuan yang baik untuk mengenali pola data masukan berupa data historis yang akan digunakan dalam proses prediksi.

Penelitian tentang prediksi harga batu bara telah dilakukan oleh Bo Zhang dan Junhai Ma (2011) dengan menggunakan teknik *partial least-squares regression*. Hasil yang didapatkan dengan menggunakan teknik tersebut adalah



hasil prediksi dengan nilai *error* antara 0,3% hingga 9,7%. Data yang digunakan adalah harga 20 hari sebelumnya, yang kemudian digunakan untuk memprediksi hari selanjutnya. Sebuah penelitian yang dilakukan oleh Saluza (2016), mengatakan bahwa data *time series* umumnya memiliki sifat yang tidak statis. Proses prediksi dengan sifat data yang tidak statis akan menghasilkan hasil akurasi yang baik apabila menggunakan sebuah *machine learning*.

*Machine learning* merupakan bagian dari kecerdasan buatan yang dapat digunakan untuk melakukan prediksi. *Machine learning* yang digunakan dalam penelitian ini yaitu metode *Extreme Learning Machine* (ELM) yang merupakan bagian dari jaringan saraf tiruan. Kinerja dari Jaringan saraf tiruan umumnya dapat bekerja dengan baik dalam menyelesaikan masalah prediksi, dikarenakan jaringan saraf tiruan dapat mencari pola dan hubungan antara data masukan dan data keluaran dengan jumlah variabel masukan yang banyak. Keunggulan yang dimiliki oleh metode ELM dibandingkan dengan metode *machine learning* yang lain adalah ELM menerapkan konsep *single-hidden layer feedforward network* (SLFN), proses pembelajaran *feedforward* yang cepat, serta memiliki kinerja generalisasi yang sangat baik.

Penelitian proses prediksi dengan menggunakan metode ELM telah dilakukan oleh Arifianti, dkk (2015) untuk peramalan indeks harga saham gabungan. Data yang digunakan berupa data *time series*. Hasil terbaik didapatkan dengan menggunakan fungsi aktivasi *sigmoid* serta jumlah *hidden neuron* sebanyak 7. Nilai *means square error* (MSE) yang didapatkan adalah sebesar 0,0082 untuk proses *training*, serta 0,0033 untuk proses *testing* dengan rata-rata waktu komputasi selama 0,0269 detik. Nilai MSE terbaik juga didapatkan pada penelitian yang dilakukan oleh Giusti, dkk (2017) untuk melakukan prediksi penjualan mie. Dengan menggunakan fitur data berupa sisa penjualan mie sebelumnya, didapatkan nilai MSE sebesar 0,0171.

Jika dibandingkan berdasarkan uraian di atas, maka dapat disimpulkan bahwa ELM memiliki kinerja yang lebih baik. ELM mampu menghasilkan nilai *error* yang kecil dengan waktu komputasi yang cepat. Oleh karena itu, berdasarkan permasalahan tersebut serta analisis terhadap kinerja algoritme ELM, maka penelitian ini mengangkat judul Prediksi Harga Patokan Batu Bara Untuk Merek Dagang Gunung Bayan I dengan Metode *Extreme Learning Machine* (ELM) dengan menggunakan parameter berupa harga batu bara sebelumnya.

## 1.2 Rumusan masalah

Berdasarkan uraian dari latar belakang, maka dapat dirumuskan permasalahan pada penelitian Prediksi Harga Patokan Batu Bara Untuk Merek Dagang Gunung Bayan I dengan Metode *Extreme Learning Machine* (ELM) sebagai berikut.

1. Bagaimana mengimplementasikan metode *Extreme Learning Machine* (ELM) sehingga dapat digunakan dalam memprediksi harga patokan batu bara di bulan berikutnya.

2. Bagaimana hasil evaluasi pengujian parameter dari metode *Extreme Learning Machine (ELM)* dalam memprediksi harga patokan batu bara di bulan berikutnya.

### 1.3 Tujuan

Tujuan yang diharapkan dari penelitian Prediksi Harga Patokan Batu Bara Untuk Merek Dagang Gunung Bayan I dengan Metode *Extreme Learning Machine (ELM)*, antara lain:

1. Merancang metode *Extreme Learning Machine (ELM)* agar bisa digunakan dalam prediksi harga patokan batu bara.
2. Menguji parameter dari metode *Extreme Learning Machine (ELM)* yang akan digunakan dalam prediksi harga patokan batu bara.

### 1.4 Manfaat

Manfaat yang diharapkan dari penelitian Prediksi Harga Patokan Batu Bara Untuk Merek Dagang Gunung Bayan I dengan Metode *Extreme Learning Machine (ELM)*, antara lain:

#### **Bagi perusahaan tambang batu bara:**

1. Membantu memberikan perkiraan harga batu bara untuk satu bulan ke depan sehingga para konsumen batu bara dapat menyusun anggaran pengeluaran perusahaan mereka.

#### **Bagi penulis:**

1. Sebagai bentuk mengaplikasikan ilmu selama kuliah dalam permasalahan di kehidupan masyarakat.
2. Sebagai bentuk untuk mengetahui bidang ilmu yang didalami dengan bidang ilmu yang lain.

#### **Bagi Universitas Brawijaya:**

1. Sebagai bentuk untuk mengetahui kemampuan mahasiswa dalam menerapkan ilmu yang telah didapatkan kedalam kehidupan bermasyarakat.
2. Sebagai bentuk untuk mengetahui kemampuan mahasiswa dalam penguasaan materi selama perkuliahan.

### 1.5 Batasan masalah

Penelitian ini disusun berdasarkan data-data yang diperoleh dalam ruang lingkup pertambangan batu bara. Ruang lingkup masalah akan dibatasi untuk mempersempit luasnya bidang yang dihadapi, yaitu:

1. Data yang digunakan dalam penelitian ini berupa daftar harga patokan batu bara untuk merek dagang Gunung Bayan I dari Januari 2009 hingga September 2017.

2. Metode yang digunakan hanya metode *Extreme Learning Machine* (ELM).
3. Pengujian dari penelitian prediksi harga patokan batu bara terdiri dari pengujian jumlah fitur, pengujian jumlah *neuron*, dan pengujian jenis fungsi aktivasi.

## 1.6 Sistematika pembahasan

Sistematika pembahasan ini bertujuan untuk memberikan suatu gambaran serta uraian dari suatu penelitian. Sistematika penulisan dari penelitian Prediksi Harga Patokan Batu Bara Untuk Merek Dagang Gunung Bayan I dengan Metode *Extreme Learning Machine* adalah sebagai berikut:

### BAB I Pendahuluan

Pendahuluan membahas mengenai latar belakang permasalahan, rumusan masalah, tujuan, batasan masalah, manfaat, serta sistematika isi pembahasan penelitian Prediksi Harga Patokan Batu Bara Untuk Merek Dagang Gunung Bayan I dengan Metode *Extreme Learning Machine*.

### BAB II Landasan Kepustakaan

Landasan Kepustakaan membahas mengenai teori dan referensi yang membahas mengenai Prediksi Harga Patokan Batu Bara Untuk Merek Dagang Gunung Bayan I dengan Metode *Extreme Learning Machine*.

### BAB III Metodologi Penelitian

Metodologi Penelitian membahas mengenai metode yang digunakan dalam penelitian. Hal tersebut terdiri dari tipe penelitian, studi kepustakaan, proses pengumpulan data, perancangan sistem perangkat lunak, pengujian, serta evaluasi dan analisis hasil penelitian Prediksi Harga Patokan Batu Bara Untuk Merek Dagang Gunung Bayan I dengan Metode *Extreme Learning Machine*.

### BAB IV Perancangan dan Implementasi

Perancangan dan implementasi membahas mengenai kebutuhan dan perancangan, serta bagaimana mengimplementasikan perangkat lunak sesuai dengan perancangan sistem yang telah dibuat dalam penelitian Prediksi Harga Patokan Batu Bara Untuk Merek Dagang Gunung Bayan I dengan Metode *Extreme Learning Machine*.

### BAB V Hasil dan Analisis

Hasil dan analisis membahas mengenai bagaimana evaluasi hasil dari proses pengujian. Pengujian yang dilakukan terdiri dari pengujian jumlah *neuron*, pengujian jumlah fitur data, serta pengujian fungsi aktivasi.

### Bab VI Penutup

Penutup membahas mengenai hasil dari suatu penelitian yang disampaikan secara singkat sebagai kesimpulan akhir dari suatu hasil penelitian, serta berisi saran untuk penelitian selanjutnya.

## BAB 2 LANDASAN KEPUSTAKAAN

### 2.1 Kajian Pustaka

Kajian pustaka penelitian ini membahas mengenai penelitian yang telah dilakukan sebelumnya. Isi yang terdapat dalam kajian pustaka meliputi referensi dari penelitian sebelumnya yang berkaitan dengan prediksi atau berkaitan dengan metode *Extreme Learning Machine* (ELM). Kajian yang dipilih berupa publikasi seperti *paper*, jurnal, tesis, skripsi, maupun hasil dari seminar.

Sejak pertama kali metode *Extreme Learning Machine* (ELM) diperkenalkan oleh Huang, dkk ada banyak penelitian yang menggunakan ELM sebagai metode untuk melakukan prediksi terhadap suatu permasalahan. Hasil prediksi yang baik dan kemampuan proses pembelajaran yang cepat, menjadikan ELM unggul dalam proses prediksi. Arifianty, dkk (2015) melakukan peramalan indeks harga saham gabungan dengan menggunakan data historis sebanyak 385 data sebagai data latih dan 96 data sebagai data uji. Dengan menggunakan aktivasi *sigmoid*, dilakukan pengujian terhadap banyaknya jumlah *hidden neuron* yang digunakan. Semakin banyak jumlah *hidden neuron*, maka nilai *means square error* (MSE) yang dihasilkan akan semakin kecil. Pada saat jumlah *hidden layer* yang digunakan adalah 3, hasil nilai MSE pada proses *training* adalah sebesar 0,0138. Sedangkan pada saat jumlah *hidden neuron* ditambah menjadi 7, nilai MSE yang dihasilkan menurun yaitu 0,0082. Hal tersebut juga berlaku pada hasil proses *testing*, nilai MSE dengan menggunakan *hidden layer* 7 memiliki nilai paling kecil bila dibandingkan *hidden layer* 1 hingga 6 (Arifianty, et al., 2015).

Untuk meningkatkan performa dari metode ELM, selain penentuan jumlah *hidden neuron*, perlu dilakukan juga penentuan perbandingan rasio jumlah data latih dan data uji. Seperti pada penelitian prediksi penjualan mie yang dilakukan oleh Giusti, dkk (2017), menyatakan bahwa jumlah perbandingan ratio sangat memengaruhi hasil dari prediksi. Perbandingan jumlah data latih dan data uji sebesar 80:20 memiliki rata-rata nilai *error* sebesar 0,0171. Nilai tersebut merupakan nilai terbaik yang dihasilkan apabila dibandingkan dengan menggunakan sedikit data latih dengan nilai *error* yang dapat mencapai 0,08. Nilai *error* yang tinggi tersebut, disebabkan karena adanya *underfitting*. *Underfitting* terjadi akibat ketidakmampuan dalam melakukan pengenalan pola pada saat proses *training* yang diakibatkan oleh jumlah data latih yang sedikit. Oleh karena itu, untuk mendapatkan hasil prediksi yang baik, jumlah perbandingan data latih harus lebih banyak dibandingkan dengan data uji (Giusti, et al., 2017).

Beberapa penelitian lain yang menjadi bahan referensi dalam penelitian ini, telah dirangkum dalam Tabel 2.1

Tabel 2.1 Daftar Kajian Pustaka

No	Penulis	Objek	Metode	Hasil
1.	(Arifianty, et al., 2015)	Penggunaan metode ELM untuk melakukan prediksi Indeks Harga Saham Gabungan (IHSG). Jumlah data yang digunakan adalah sebanyak 385 data latih yang dibuat menjadi 3 variabel dan sebanyak 96 data uji yang dibuat menjadi 1 variabel.	<p>Algoritme <i>Extreme Learning Machine</i> (ELM). Proses:</p> <ol style="list-style-type: none"> <li>Pembagian jumlah data latih dan data uji</li> <li>Normalisasi data</li> <li>Set fungsi aktivasi</li> <li>Hitung bobot <i>input</i>, bias, dan bobot <i>output</i></li> <li>Hitung nilai <i>error</i></li> <li>Denormalisasi hasil <i>training</i></li> <li>Melakukan proses <i>testing</i> dengan menggunakan bobot <i>input</i>, bias, dan bobot <i>output</i> dari proses <i>training</i></li> <li>Evaluasi nilai <i>error</i> dari hasil <i>testing</i></li> </ol>	Jenis fungsi aktivasi dan jumlah <i>hidden neuron</i> yang digunakan akan sangat memengaruhi hasil dari proses prediksi. Hasil terbaik dari penelitian tersebut didapatkan dengan menggunakan fungsi aktivasi berupa <i>sigmoid</i> dengan jumlah <i>hidden neuron</i> sebanyak 7. Nilai MSE yang didapatkan untuk masing-masing proses <i>training</i> dan <i>testing</i> adalah sebesar 0,0082 dan 0,0033 dengan waktu komputasi rata-rata sebesar 0,0269 detik.
2.	(Anggraini, 2017)	Penggunaan metode ELM untuk prediksi nilai tukar mata uang asing. Data yang digunakan merupakan data harian kurs jual dan kurs beli mulai 4 Januari 2016 hingga 31 Januari 2017.	<p>Algoritme <i>Extreme Learning Machine</i> (ELM). Proses:</p> <ol style="list-style-type: none"> <li>Normalisasi data</li> <li>Inisialisasi bobot dan bias, serta menentukan jumlah <i>hidden neuron</i></li> <li>Hitung <i>output</i> matriks <i>hidden layer</i></li> <li>Hitung bobot <i>output</i></li> <li>Denormalisasi</li> </ol>	Pengujian dilakukan dengan menggunakan 5 buah node <i>input</i> dan 25 <i>hidden layer</i> untuk masing-masing pembelajaran kurs jual dan kurs beli. Fungsi aktivasi yang digunakan adalah fungsi aktivasi <i>sigmoid</i> . Penelitian tersebut menghasilkan nilai <i>error</i> sebesar 0,001296 untuk kurs jual dan 0,001099 untuk kurs beli.
3.	(Syed & Anggarwal, 2016)	Penggunaan metode ELM untuk melakukan prediksi kekuatan angin. <i>Dataset</i> yang digunakan merupakan data untuk mengukur tenaga angin yang terdiri dari beberapa parameter, yaitu	<p>Algoritme <i>Extreme Learning Machine</i> (ELM). Proses:</p> <ol style="list-style-type: none"> <li>Inisialisasi nilai bobot dan bias</li> <li>Hitung <i>output</i> matriks <i>hidden layer</i> (H)</li> <li>Hitung bobot <i>hidden output</i></li> </ol>	Hasil pengujian terbaik didapatkan dengan membandingkan hasil dari <i>mean absolute error</i> (MAE) antara ELM model dengan <i>persistence model</i> . Dari pengujian terhadap tiga <i>dataset</i>



		kecepatan angin, arah angin, kekuatan angin. Terdapat 3 daerah yang akan digunakan sebagai <i>dataset</i> yaitu, Amarath, Alma, dan Kingsbridge.	layer (B)	didapatkan bahwa hasil prediksi dengan menggunakan ELM model selalu lebih baik dibandingkan <i>persistance model</i> . Berdasarkan hasil dari salah satu <i>dataset</i> didapatkan bahwa untuk <i>dataset</i> Amarath, nilai MAE pada 1 jam pertama untuk <i>persistance model</i> adalah sebesar 9,824, kemudian meningkat di 12 jam berikutnya menjadi 39,022. Sedangkan nilai MAE Amarath 1 jam pertama untuk ELM model adalah sebesar 11,0141, kemudian meningkat di 12 jam berikutnya menjadi 25,282.
4.	(Giusti, et al., 2017)	Penggunaan metode ELM untuk melakukan prediksi penjualan mie. Data yang digunakan merupakan data historis penjualan semua produk mie per hari, serta data sisa stok dari penjualan perhari selama 3 bulan yang berjumlah 89 data.	<p>Algoritme <i>Extreme Learning Machine</i> (ELM). Proses:</p> <ol style="list-style-type: none"> <li>Menentukan jumlah <i>hidden neuron</i></li> <li>Inialisasi <i>input weight</i> dan bias</li> <li>Normalisasi data</li> <li>Melakukan proses <i>training</i> dengan algoritme ELM</li> <li>Melakukan proses <i>testing</i> dengan algoritme ELM</li> <li>Denormalisasi hasil prediksi</li> <li>Hitung nilai <i>error</i> dengan MSE</li> </ol>	Penentuan jumlah <i>neuron</i> , jumlah data <i>training</i> , dan penentuan jumlah fitur, akan sangat memengaruhi hasil dari prediksi. Semakin banyak jumlah <i>neuron</i> , akan semakin memperkecil <i>error</i> , yaitu sebesar 0,00176. Penggunaan data <i>training</i> sebanyak 80% juga dapat memperkecil nilai <i>error</i> , yaitu sebesar 0,0171.
5.	(Umar, et al., 2014)	Penggunaan metode ELM untuk melakukan prediksi terhadap tingkat kebocoran pipa. Data yang digunakan berupa besarnya tekanan jaringan pipa	<p>Algoritme <i>Extreme Learning Machine</i> (ELM). Proses:</p> <ol style="list-style-type: none"> <li>Inialisasi jumlah <i>hidden neuron</i></li> <li>Inialisasi bobot</li> </ol>	Tingkat keakuratan prediksi pada penelitian tersebut dilakukan dengan membandingkan hasil prediksi dengan hasil aktual. Proses tersebut



		yang kemudian diolah untuk mengetahui tingkat kebocoran pipa.	<p>dan bias</p> <p>c. Hitung bobot <i>output</i></p> <p>d. Hitung nilai <i>error</i></p>	dilakukan dengan menghitung nilai <i>root means square error</i> (RMSE). Hasil nilai RMSE untuk prediksi tingkat kebocoran adalah sebesar 0,15641.
6.	(Chandar, et al., 2016)	<p>Penggunaan metode ELM untuk melakukan prediksi terhadap harga emas. <i>Dataset</i> yang digunakan diambil mulai 1 Januari 2000 hingga 31 April 2014. Selain <i>dataset</i> harga emas, terdapat <i>dataset</i> lain yang digunakan dalam penelitian tersebut, yaitu:</p> <ol style="list-style-type: none"> <li>1. Harga minyak mentah</li> <li>2. Harga perak</li> <li>3. Harga dollar dalam satuan mata uang India</li> <li>4. <i>Standard &amp; Poor's 500 stock index</i></li> </ol>	<p>Algoritme <i>Extreme Learning Machine</i> (ELM). Proses:</p> <ol style="list-style-type: none"> <li>a. Inisialisasi nilai bobot dan bias</li> <li>b. Hitung <i>output</i> matriks <i>hidden layer</i> (H)</li> <li>c. Hitung bobot <i>hidden output layer</i> (B)</li> </ol>	<p>Hasil pengujian terbaik setelah dilakukan perbandingan antara metode ELM dengan metode <i>Feed Forward Network</i>, <i>Feed Forward Back Propagation Network</i>, <i>Radial Basis Function</i>, dan metode <i>Elman Network</i> adalah metode ELM dapat menghasilkan nilai akurasi yang paling besar. Nilai akurasi dari metode ELM untuk proses <i>training</i> dan <i>testing</i> adalah sebesar 97,65% dan 93,82%</p>
7.	(Zhang & Ma, 2011)	<p>Penggunaan metode <i>partial least-square regression</i> untuk prediksi indeks harga batu bara. Penelitian tersebut menggunakan <i>dataset</i> yang diambil dari <i>Qinhuangdao Coal Information Network</i> mulai 2 September 2009 hingga 2 Maret 2011.</p>	<p>Algoritme <i>Partial Least-Square Regression</i>. Proses:</p> <ol style="list-style-type: none"> <li>a. Melakukan proses ekstraksi pada komponen utama algoritme</li> <li>b. Melakukan proses regresi <i>nonlinear</i> pada beberapa variabel tidak bebas dan variabel utama</li> <li>c. Melakukan iterasi terhadap regresi <i>nonlinear</i></li> </ol>	<p>Hasil pengujian dari penelitian indeks harga batu bara menghasilkan nilai <i>error</i> minimum sebesar 0,3% dan nilai <i>error</i> maksimum sebesar 9,7%.</p>

## 2.2 Batu Bara

Batu bara merupakan sebuah batuan sedimen yang tersusun dari campuran bahan organik dan bahan anorganik. Kandungan bahan organik yang ada didalamnya dapat mencapai 50% hingga 75%. Marsenal adalah sebutan untuk



bahan organik yang berasal dari sisa tumbuhan yang telah mengalami berbagai tingkat dekomposisi serta mengalami perubahan sifat fisik dan kimia. Sedangkan untuk bahan anorganik, biasa disebut dengan mineral atau mineral *matter*. Komposisi mineral dalam jumlah tertentu dapat memengaruhi kualitas batu bara, terutama untuk beberapa parameter seperti abu, sulfur, dan nilai panas batu bara (Finkelman, 1993).

Terdapat beberapa jenis batu bara yang dikelompokkan berdasarkan tingkat kematangan dari komposisi mineralnya. Jenis batu bara tersebut terbagi menjadi subbituminus, semi antrasit, antrasit, dan meta antrasit (Stach, 1982).

### 2.2.1 Penetapan Harga Batu Bara

Berdasarkan peraturan yang ditetapkan oleh Direktur Jendral Mineral dan Batu Bara No 480 tahun 2014 terdapat dua jenis harga batu bara, yaitu Harga Batu Bara Acuan (HBA) dan Harga Patokan Batu Bara (HPB). HBA adalah nilai rata-rata dari empat indeks yang terdiri dari, *new castle global coal index*, *new castle export index*, *platts index*, dan *Indonesia coal index*. Dua indeks pertama merupakan indeks harga batu bara internasional, sedangkan dua indeks berikutnya merupakan indeks harga batu bara domestik.

HPB adalah harga patokan batu bara untuk masing-masing merek dagang yang didapatkan dengan melakukan standardisasi harga berdasarkan nilai HBA yang disesuaikan dengan kualitas dari masing-masing merek batu bara. Ada banyak merek dagang batu bara seperti Gunung Bayan I, Prima Coal, Melawan Coal, Envirocoal, Pinang 6150, dan lainnya. HBA memiliki kesetaraan nilai kalor berupa 6.322 kkal/kg GAR dengan kandungan abu 15%, kandungan air 8%, dan kandungan belerang 0,8%. Sedangkan untuk Gunung Bayan I memiliki kesetaraan nilai kalor berupa 7.000 kkal/kg GAR dengan kandungan abu 15%, kandungan air 10%, dan kandungan belerang 1%. Formula yang digunakan dalam menghitung HPB untuk merek dagang Gunung Bayan I ditunjukkan pada Persamaan 2.1.

$$HPB \text{ Maker} = (HBA * K * A) - (B + U) \quad (2.1)$$

Keterangan:

$HPB$  = Nilai harga patokan batu bara

$K$  = Nilai kalor batu bara / 6322

$A$  =  $(100 - \text{kandungan air batu bara}) / (100 - 8)$

$B$  =  $(\text{Kandungan belerang batu bara} - 0.8) * 4$

$U$  =  $(\text{Kandungan abu batu bara} - 15) * 0.4$

### 2.3 Kecerdasan Buatan

Kecerdasan buatan (*artificial intelligence*) merupakan sebuah tingkah laku mesin yang mampu berfikir, mampu mengambil keputusan, serta mampu menentukan tindakan yang akan dilakukan seperti tingkah laku manusia. Beberapa komponen kecerdasan buatan terdiri dari: (Sutojo, et al., 2011)

- *Knowledge representation* (representasi pengetahuan)



- *Theorem proving* (membuktikan teorema)
- *Natural language processing* (pemrosesan bahasa alami)
- *Computer vision*
- *Problem solving* (penyelesaian masalah)
- *Handling uncertain situations* (mengatasi situasi yang tidak tentu)
- *Game playing* (permainan game)
- *Expert system* (sistem pakar)
- *Dealing philosophical issue* (menangani persoalan filosofis)

Kelebihan dari kecerdasan buatan, yaitu: (Dahria, 2008)

- Tidak dapat berubah selama tidak ada perubahan pada program dan sistem komputernya
- Memudahkan dalam penyebaran informasi pengetahuan antara satu komputer dengan komputer lainnya
- Aktivitas yang dilakukan oleh sebuah sistem cerdas dapat dengan mudah di dokumentasikan
- Waktu proses sistem cerdas tergolong cepat
- Dapat menghasilkan solusi terbaik

## 2.4 Machine Learning

*Machine learning* merupakan representasi buatan pada proses pembelajaran program komputer yang mampu menyelesaikan sejumlah proses perhitungan dengan mengadopsi proses pembelajaran pada otak manusia. Proses pembelajaran diawali dengan memasukkan sejumlah informasi yang akan dilewatkan dari suatu *neuron* ke *neuron* yang lainnya dalam bentuk rangsangan listrik melalui dendrit. Apabila rangsangan tersebut diterima oleh suatu *neuron*, maka proses dilanjutkan dengan *neuron* membangkitkan *output* ke semua *neuron* yang berhubungan dengannya. Terdapat dua proses pembelajaran dalam jaringan saraf tiruan, yaitu:

### 1. Pembelajaran terawasi (*Supervised learning*)

Metode pembelajaran terawasi dilakukan apabila nilai *output* yang diharapkan sudah diketahui sebelumnya. Pada proses pembelajaran, satu pola *input* akan diberikan ke satu *neuron* pada lapisan *input*. Pola ini akan dirambatkan di sepanjang jaringan saraf hingga ke *neuron* lapisan *output*. Lapisan *output* akan membangkitkan pola *output* yang nantinya akan dicocokkan dengan pola *output* target. Apabila terjadi perbedaan, maka itu berarti terjadi *error*. Jika nilai *error* yang dihasilkan cukup besar, maka masih perlu dilakukan proses pembelajaran lebih banyak lagi.

### 2. Pembelajaran tak terawasi (*Unsupervised learning*)

Pembelajaran tak terawasi tidak memerlukan target *output* dalam proses pembelajaran. Nilai bobot akan disusun dalam suatu rentang tertentu sesuai dengan nilai *input* yang diberikan pada saat proses pembelajaran. Tujuan

dari proses pembelajaran ini adalah untuk mengelompokkan unit-unit yang serupa dalam suatu area tertentu.

## 2.5 Jaringan Saraf Tiruan

Pada sub bab ini akan dijelaskan definisi dari jaringan saraf tiruan, lapisan penyusun jaringan saraf tiruan, serta arsitektur jaringan saraf tiruan.

### 2.5.1 Definisi Jaringan Saraf Tiruan

Jaringan saraf tiruan (JST) merupakan sebuah pemodelan dalam pemrosesan suatu informasi yang diadaptasi dari sistem sel saraf pada biologi. Penerapan JST dalam pemodelan matematika dibentuk dengan asumsi bahwa: (Agustin & Prahasto, 2012)

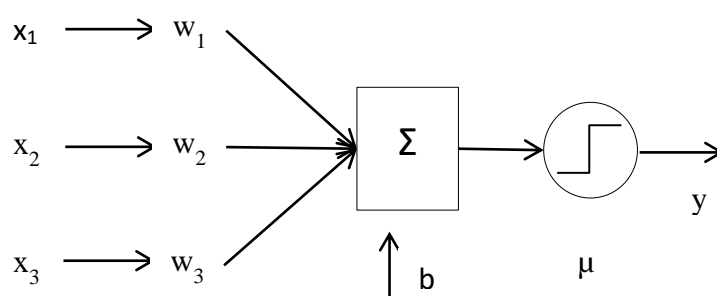
1. Pemrosesan suatu informasi terbentuk pada suatu elemen sederhana (*neuron*).
2. Melalui suatu penghubung, sinyal akan dikirimkan di antara *neuron-neuron*.
3. Bobot yang dimiliki oleh penghubung antar *neuron* dapat memperlemah atau memperkuat sinyal.
4. Nilai *output* ditentukan dengan menggunakan suatu fungsi aktivasi. *Output* yang dihasilkan selanjutnya akan dibandingkan dengan batas ambang.

Istilah "*fault tolerant*" sering dikaitkan dengan istilah *neuron* biologi. Hal ini disebabkan karena *neuron* biologi pada manusia dapat mengenali masukan yang berbeda dari proses pembelajarannya. Selain itu, suatu *neuron* biologi tetap dapat bekerja dengan baik apabila terdapat kerusakan pada sebuah *neuron*, karena sebuah *neuron* lain dapat ditugaskan untuk menggantikan fungsi *neuron* yang rusak (Siang, 2004).

Tujuan utama dari sebuah pemrosesan informasi dengan menggunakan konsep JST adalah untuk meningkatkan keseimbangan antara kemampuan generalisasi dan kemampuan memorisasi. Kemampuan memorisasi merupakan kemampuan dalam menerapkan konsep JST untuk mengambil informasi terhadap pola yang telah dipelajari. Sedangkan kemampuan generalisasi merupakan kemampuan dalam menerapkan konsep JST untuk menghasilkan respon terhadap hasil memorisasi. Dengan menerapkan konsep tersebut dalam proses JST, maka apabila diberi suatu informasi baru, JST akan tetap dapat menghasilkan *output* yang baik (Puspitaningrum, 2006).

### 2.5.2 Arsitektur Jaringan Saraf Tiruan

Lapisan penyusun yang terdapat pada jaringan saraf tiruan umumnya terbagi menjadi dua jenis, yaitu jaringan dengan lapisan tunggal (*single layer network*) dan jaringan dengan banyak layer (*multi layer network*). Bentuk arsitektur dari jaringan saraf tiruan dapat dilihat pada Gambar 2.1. terdapat tiga komponen utama dalam arsitektur tersebut, yaitu *input, neuron* (sebagai fungsi penjumlahan dan fungsi aktivasi), serta *output* yang saling terhubung.



**Gambar 2.1** Arsitektur Jaringan Saraf Tiruan

Sumber: (Giusti, et al., 2017)

Cara kerja dari jaringan saraf tiruan, umumnya mengacu pada cara kerja jaringan saraf makhluk hidup yang mengumpulkan informasi melalui lapisan *input layer*. Informasi tersebut akan dikirimkan ke *neuron* dengan bobot *input* tertentu. *Input* tersebut kemudian diproses dengan suatu Persamaan. Persamaan tersebut digunakan untuk menjumlahkan semua nilai bobot yang datang. Hasil penjumlahan tersebut kemudian akan dibandingkan dengan dengan sebuah fungsi aktivasi dengan suatu nilai ambang tertentu. Apabila *input* tersebut sudah melewati suatu nilai ambang tersebut, maka *neuron* akan dalam aktif. Jika tidak, maka *neuron* tersebut tidak akan diaktifkan. *Neuron* yang aktif, kemudian akan mengirimkan *output* melalui bobot *output* ke semua *neuron*. Hasil tersebut dapat ditemui pada lapisan *output layer*.

Untuk dapat digunakan dalam menyelesaikan masalah, JST juga memerlukan pelatihan seperti cara kerja jaringan saraf makhluk hidup. JST melakukan proses pembelajaran berdasarkan contoh-contoh pelatihan yang diberikan. Proses pelatihan dilakukan untuk mendapatkan model yang optimal, sehingga *output* yang dihasilkan juga dapat mendekati optimal. Lamanya waktu pembelajaran yang dibutuhkan untuk setiap proses pada JST dipengaruhi oleh bobot antar layer, banyaknya jumlah *neuron*, metode pembelajaran yang digunakan, serta fungsi aktivasi (Atmojo, et al., 2013).

### 2.5.3 Faktor Bobot

Bobot merupakan suatu nilai yang mendefinisikan tingkat hubungan antara suatu *node* dengan *node* lainnya. Semakin besar nilai bobot, maka hubungan antar *node* yang terjadi akan semakin kuat. Bobot dapat berisi bilangan *real* atau *integer*, sesuai dengan permasalahan dan model yang digunakan. Nilai bobot dapat ditentukan untuk berada dalam suatu rentang tertentu agar selama proses pelatihan bobot dapat menyesuaikan dengan pola-pola *input*.

Pada saat proses pembelajaran, jaringan akan terus memperbaiki diri. Setiap ada masalah baru, maka jaringan tersebut akan melakukan proses pembelajaran dengan mengatur kembali nilai bobot untuk menyesuaikan karakter nilai (Puspitaningrum, 2006).

## 2.5.4 Fungsi Aktivasi

Setiap *neuron* memiliki sebuah keadaan internal. Keadaan tersebut disebut level aktivasi atau level aktivitas dari suatu fungsi *input* yang diterima. Suatu *neuron* dapat mengirimkan aktivitasnya ke beberapa *neuron* lain sebagai sinyal. Sinyal yang dikirimkan dari *neuron* satu ke *neuron* lainnya hanya berupa satu sinyal sesaat.

Ada beberapa jenis fungsi aktivasi dalam JST. Dalam proses pembelajaran, karakteristik yang harus dimiliki fungsi aktivasi adalah bersifat kontinu, *diferensiabel*, dan tidak menurun secara monoton. Fungsi aktivasi diharapkan mampu mendekati nilai-nilai maksimum dan minimum secara baik. Berikut merupakan beberapa jenis fungsi aktivasi dalam JST.

### 1. Fungsi *Sigmoid Biner*

Fungsi *sigmoid biner* merupakan sebuah fungsi yang menghasilkan *output* dalam rentang 0 sampai 1. Bentuk persamaan fungsi *sigmoid biner* ditunjukkan pada Persamaan 2.2.

$$H = \frac{1}{(1 + \exp^{-H_{init}})} \quad (2.2)$$

### 2. Fungsi *Sigmoid Bipolar*

Fungsi *sigmoid bipolar* merupakan sebuah fungsi yang menghasilkan *output* dalam rentang -1 sampai 1. Bentuk persamaan fungsi *sigmoid bipolar* ditunjukkan pada Persamaan 2.3.

$$H = \frac{(1 - \exp^{-H_{init}})}{(1 + \exp^{-H_{init}})} \quad (2.3)$$

### 3. Fungsi *Sin*

Bentuk persamaan fungsi *sin* ditunjukkan pada Persamaan 2.4.

$$H = \sin(H_{init}) \quad (2.4)$$

## 2.6 Time Series Forecasting

Peramalan merupakan suatu kegiatan untuk memperkirakan apa yang terjadi di masa yang akan datang. Perkiraan dilakukan berdasarkan data yang relevan pada masa lalu untuk ditempatkan ke masa yang akan datang dengan suatu model matematis. Data tersebut biasa disebut dengan data *time series*.

Data *time series* merupakan suatu representasi dari suatu variabel acak yang biasanya mempunyai interval waktu yang sama dan diamati pada suatu periode tertentu. Keunggulan dari data *time series* adalah kemampuannya dalam mengestimasi, memperkirakan, dan meramal nilai data pada waktu berikutnya berdasarkan data sebelumnya (Bahtiar, 2016).

## 2.7 Extreme Learning Machine (ELM)

Pada sub bab ini akan dijelaskan definisi dari metode ELM dan proses dari metode ELM.

### 2.7.1 Definisi Extreme Learning Machine (ELM)

Metode *Extreme Learning Machine* (ELM) merupakan salah satu konsep dari metode jaringan saraf tiruan dengan konsep pembelajaran *feed-forward single hidden layer* (SLFN). Metode pembelajaran dengan ELM dibuat untuk mengatasi kelemahan dari pembelajaran berbasis *feedforward* terutama dalam hal *learning speed*. Terdapat dua alasan yang dikemukakan oleh Huang, dkk (2006) mengapa JST *feedforward* mempunyai *learning speed* yang rendah, yaitu:

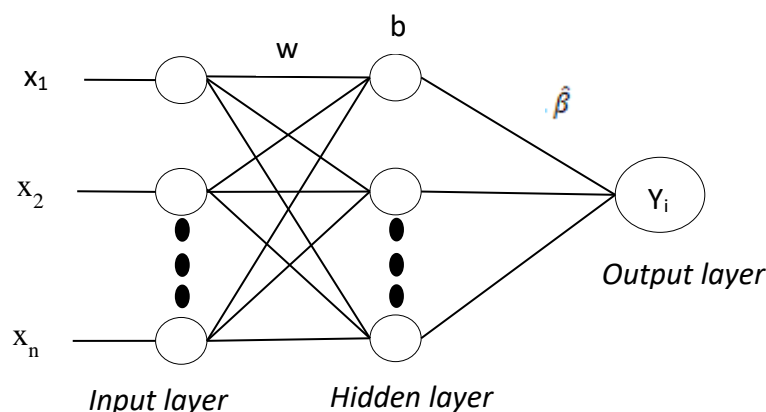
1. Menerapkan *slow gradient based learning algorithm* dalam melakukan proses *training*
2. Semua parameter ditentukan secara *iterative* berdasarkan metode pembelajaran tersebut

Proses pembelajaran dengan menggunakan *conventional gradient based learning algorithm*, umumnya menentukan semua parameternya secara manual. Parameter tersebut adalah *input weight* dan *hidden bias*. Parameter tersebut akan saling berhubungan pada saat proses pembelajaran, sehingga membutuhkan waktu pembelajaran yang lama dan sering terjebak pada *local minima*. Pada pembelajaran dengan ELM, proses penentuan parameter seperti *input weight* dan *hidden bias* dilakukan secara acak. Hal ini bertujuan untuk mempercepat dalam proses pembelajaran, serta menghasilkan performa generalisasi yang baik.

### 2.7.2 Arsitektur Extreme Learning Machine

*Extreme Learning Machine* (ELM) adalah salah satu metode jaringan saraf tiruan yang memanfaatkan teori *inverse* matriks untuk proses pembelajarannya, yaitu teori *Moore Penrose pseudoinverse*. Bentuk algoritme pembelajaran ELM sering disebut dengan model pembelajaran *single hidden layer feedforward network* (SLFN) (Atmojo, et al., 2013). Bentuk arsitektur dari ELM dapat dilihat pada Gambar 2.2.

Sebuah model jaringan saraf tiruan SLFN terdiri dari *input*  $n$ , *neuron hidden layer*  $m$ , serta fungsi aktivasi  $g(x)$ . Jika terdapat  $X = [x_1, x_2, \dots, x_n]$  dengan  $x_1$  adalah nilai *input* pada jaringan tersebut dan nilai  $\alpha$  adalah matriks bobot antara *input layer* dan *hidden layer*, maka sebuah matriks  $\alpha$  akan mempunyai ukuran berupa  $n \times m$ . Penentuan nilai elemen matriks akan dilakukan secara acak. Kemudian, setiap nilai *input* akan diolah pada *hidden layer* menggunakan fungsi aktivasi. Nilai tersebut kemudian akan dikumpulkan dalam sebuah matriks  $H$  dengan *ordo*  $1 \times m$  ( $H=[h_1, h_2, \dots, h_m]$ ). Teori *Moore Penrose pseudo inverse* akan digunakan untuk menentukan hasil nilai bobot antara *hidden layer* dengan *output layer* (Atmojo, et al., 2013).



**Gambar 2.2** Arsitektur *Extreme Learning Machine*

Sumber: (Giusti, et al., 2017)

### 2.7.3 Moore Penrose Pseudo Inverse

*Moore Penrose Pseudo Inverse* merupakan suatu konsep *inverse* matriks yang umumnya terbatas hanya untuk matriks persegi dengan ordo  $n \times n$  dan *non singular*. Untuk matriks *singular* dengan ordo  $m \times n$  umumnya tidak memiliki nilai *inverse*. Akan tetapi, ada sebuah matriks yang seolah-olah mampu menjadi *inverse* untuk matriks dengan ordo  $m \times n$ . Matriks tersebut adalah matriks *Moore Penrose pseudo inverse*.

Definisi dari *Moore Penrose pseudo inverse* adalah jika  $A \in M_{m,n}$  maka terdapat  $A^+ \in M_{m,n}$  yang unik serta memenuhi empat kondisi *penrose* yaitu (Fikriya, et al., 2017):

1.  $AA^+A = A$
2.  $A^+AA^+ = A^+$
3.  $A^+A = (A^+A)^*$
4.  $AA^+ = (AA^+)^*$

$A^*$  menyatakan konjungsi *transpose* dari matriks  $A$ .

### 2.7.4 Proses Training

Proses *training* dilakukan dengan menyiapkan data *input* ( $D$ ) sebagai *training set* dan *target output* ( $T$ ) sebagai hasil prediksi. Tujuan dari proses *training* adalah untuk mendapatkan nilai *output weight* yang baik. Berikut merupakan langkah-langkah *training* pada ELM (Huang, et al., 2006).

- a. Melakukan inialisasi terhadap bobot ( $W_{JK}$ ), bias ( $b$ ), serta jumlah *neuron hidden layer* yang akan digunakan dalam proses prediksi. Nilai acak untuk matriks bobot dibuat dengan rentang  $[-1,1]$  dalam bentuk *array* ukuran  $m$  (jumlah *hidden neuron*)  $\times$   $n$  (jumlah *input neuron*). Sedangkan nilai acak untuk bias dibuat dengan rentang  $[0,1]$  dalam ukuran  $1 \times$  [jumlah *hidden neuron*].

- b. Menghitung matriks keluaran pada *hidden layer*. Dalam proses ini, data sejumlah  $N$  yang berisi hasil normalisasi dalam bentuk matriks yang akan dikalikan dengan *input weight* dan ditambahkan dengan bias. Proses perhitungan keluaran *hidden layer* adalah menggunakan persamaan 2.5.

$$H_{init} = X.W^T + b \quad (2.5)$$

Keterangan:

$H_{init}$  = matriks keluaran *hidden neuron*  
 $X$  = matriks normalisasi data tanpa target  
 $W^T$  = matriks hasil *transpose* bobot *input*  
 $B$  = matriks bias

3. Menghitung matriks *hidden neuron* menggunakan fungsi aktivasi. Fungsi aktivasi yang digunakan adalah fungsi aktivasi *sigmoid biner* ( $H$ ) dengan tujuan untuk memetakan nilai matriks keluaran *hidden layer*. Hasil yang didapatkan dengan menggunakan fungsi aktivasi adalah berupa nilai *output* yang terletak pada interval 0 sampai 1. Bentuk persamaan untuk menghitung matriks *hidden neuron* dapat dilihat pada Persamaan 2.6.

$$H = \frac{1}{(1+exp^{-X.W^T+b})} \quad (2.6)$$

Keterangan:

$H$  = matriks hasil fungsi aktivasi *sigmoid biner*  
 $exp$  = eksponensial data

4. Mengitung matriks *Moore-Penrose Pseudo Inverse*. Proses perhitungan matriks *Moore-Penrose Pseudo Inverse* dilakukan dengan melakukan perkalian antara matriks *inverse* dan matriks *transpose* dari hasil keluaran *hidden layer* dengan fungsi aktivasi. Bentuk persamaan untuk menghitung matriks *Moore-Penrose pseudo inverse* dapat dilihat pada Persamaan 2.7.

$$H^+ = (H^T.H)^{-1}.H^T \quad (2.7)$$

Keterangan:

$H^+$  = matriks hasil perhitungan *Moore-Penrose Pseudo Inverse*  
 $H$  = matriks hasil perhitungan fungsi aktivasi

5. Menghitung nilai *output weight*. Bentuk persamaan untuk menghitung nilai *output weight* dapat dilihat pada Persamaan 2.8.

$$\hat{\beta} = H^+.T \quad (2.8)$$

Keterangan:

$\beta$  = matriks *output weight*  
 $H^+$  = matriks hasil perhitungan *Moore-Penrose Pseudo Inverse*  
 $T$  = matriks target

6. Menghitung nilai  $Y$  prediksi sebagai hasil dari proses *training* dengan Persamaan 2.9.



$$\hat{Y} = H \cdot \hat{\beta} \quad (2.9)$$

Keterangan:

$Y$  = hasil prediksi

### 2.7.5 Proses *Testing*

Proses *testing* adalah proses untuk mengevaluasi metode ELM berdasarkan hasil dari proses *training*. Proses ini menggunakan *input weight*, *output weight* dan bias yang didapatkan pada proses *training*. Berikut merupakan langkah-langkah dari proses *testing* (Huang, et al., 2006).

- Inisialisasi bias dan *input weight* yang didapatkan dari proses *training*
- Hitung matriks *output hidden neuron* ( $H_{init}$ ) dengan menggunakan persamaan seperti pada proses *training*
- Hitung nilai  $Y$  prediksi dengan menggunakan persamaan seperti pada proses *training*
- Evaluasi hasil prediksi dengan menghitung nilai *error* menggunakan *Mean Absolute Percentage Error* (MAPE).

### 2.8 Normalisasi dan Denormalisasi Data

Normalisasi data merupakan suatu proses *preprocessing* data. Proses ini perlu dilakukan sebelum melakukan proses prediksi harga batu bara. Normalisasi dilakukan untuk melakukan standardisasi data yang akan digunakan dalam proses perhitungan dengan mengubah data untuk berada dalam jarak tertentu. Proses normalisasi data yang digunakan dalam penelitian ini menggunakan *Min Max Normalization* yang ditunjukkan dengan Persamaan 2.10.

$$d' = \frac{d - \min}{\max - \min} \quad (2.10)$$

Keterangan:

$d'$  = nilai dari hasil normalisasi

$d$  = nilai asli data

$\min$  = nilai minimum pada data set fitur  $X$

$\max$  = nilai maximum pada data set fitur  $X$

Proses normalisasi dalam penelitian ini bertujuan untuk menyamakan nilai rentang data dengan fungsi aktivasi dalam ELM. Untuk mendapatkan nilai *output* yang berada pada rentang 0 sampai 1, maka perlu dilakukan normalisasi data yang untuk mengubah nilai rentang data menjadi 0 sampai 1 (Siwi, 2016).

Setelah proses normalisasi, maka perlu dilakukan proses denormalisasi untuk mengembalikan nilai data menjadi nilai sebenarnya berdasarkan hasil peramalan. Bentuk persamaan yang digunakan dalam proses denormalisasi ditunjukkan pada Persamaan 2.11 .



$$d = d'(max - min) + min \tag{2.11}$$

Keterangan:

- $d'$  = nilai prediksi sebelum denormalisasi
- $d$  = nilai hasil denormalisasi
- $min$  = nilai minimum pada data set fitur X
- $max$  = nilai maksimum pada data set fitur X

## 2.9 Nilai Evaluasi

Hasil prediksi dievaluasi menggunakan *Mean Absolute Percentage Error* (MAPE). Tujuan yang didapatkan dengan melakukan evaluasi nilai *error* adalah untuk membandingkan selisih antara nilai ramalan dengan nilai aktual. MAPE merupakan skala *independen* yang mudah diinterpretasikan, sehingga MAPE banyak direkomendasikan di beberapa *textbooks*. Proses perhitungan dilakukan menggunakan Persamaan 2.12 dengan kriteria nilai MAPE yang ditunjukkan pada Tabel 2.2 (Chang, et al., 2007).

$$MAPE = \frac{1}{N} \sum_{i=1}^n \left| \frac{y_i - t_i}{y_i} \right| \times 100 \tag{2.12}$$

Keterangan:

- $n$  = Jumlah data
- $y_i$  = Nilai prediksi
- $t_i$  = Nilai aktual

**Tabel 2.2 Kriteria Nilai MAPE**

Nilai MAPE	Kriteria
< 10%	Sangat Baik
10% - 20%	Baik
20% - 50%	Cukup
>50%	Buruk

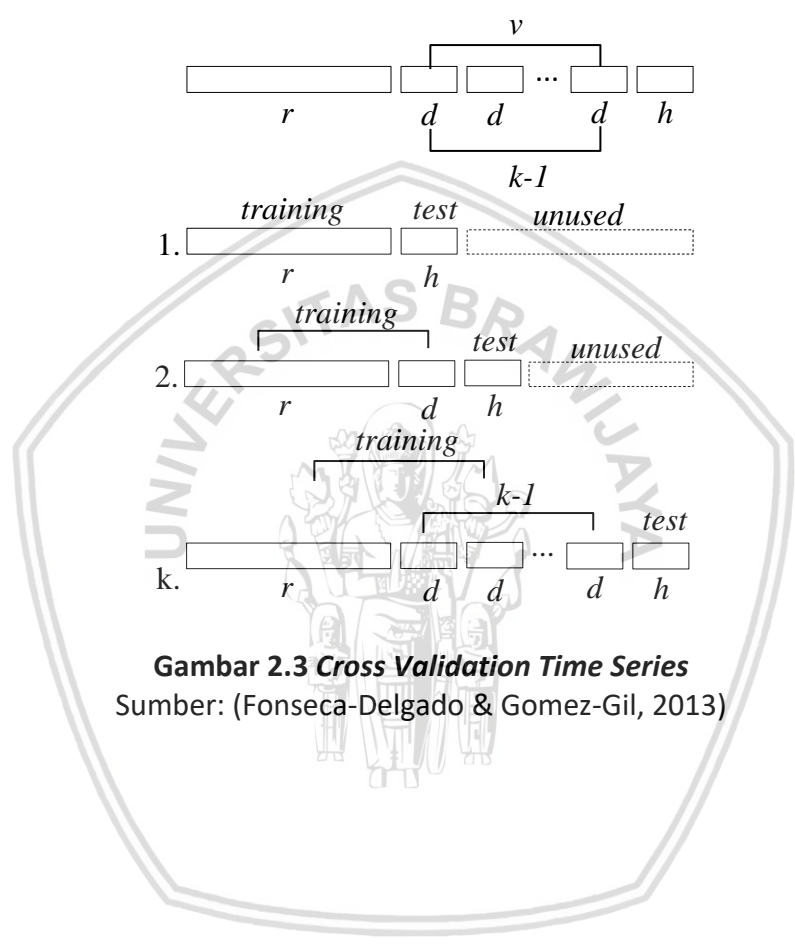
## 2.10 Cross Validation Time Series

*Cross validation* merupakan sebuah metode statistik yang dapat digunakan untuk mengevaluasi kinerja suatu model atau algoritme yang memisahkan data menjadi dua *subset*, yaitu data *training* dan data *testing*. Model atau algoritme akan dilatih oleh *subset training* dan divalidasi oleh *subset testing*.

Gambar 2.3 menunjukkan serangkaian proses validasi menggunakan *cross validation* untuk data *time series*. *Dataset* akan dibagi menjadi dua *subset* yaitu *subset r* dan *subset v*. *Subset r* menunjukkan data yang akan digunakan sebagai data *training*. Kemudian *subset v* akan dibagi menjadi *k-fold* yang akan divalidasi (Fonseca-Delgado & Gomez-Gil, 2013).



Proses validasi pertama akan menggunakan subset  $r$  sebagai data training dan satu data setelah data  $training$  ( $h$ ) akan digunakan sebagai data  $testing$ . Sedangkan data sisanya tidak akan digunakan. Untuk validasi kedua, data yang digunakan sebagai data  $testing$  pada proses validasi pertama, akan dimasukkan ke data  $training$ . Untuk data  $testing$  berisi data setelah data  $training$  tersebut. Begitu pula untuk proses validasi berikutnya seperti pada Gambar 2.3. Tingkat akurasi dari proses peramalan tersebut akan didapatkan dari nilai rata-rata proses  $testing$  untuk setiap  $k$ -fold.



**Gambar 2.3 Cross Validation Time Series**  
 Sumber: (Fonseca-Delgado & Gomez-Gil, 2013)



## BAB 3 METODOLOGI PENELITIAN

Bab ini berisi tentang langkah-langkah yang akan diambil dalam penelitian yang dilakukan. Terdapat beberapa tahapan penelitian yang akan dilakukan. Tahapan tersebut terdiri dari analisis tipe penelitian, studi kepustakaan, pengumpulan data, strategi dan lokasi penelitian, peralatan pendukung, analisis dan perancangan sistem, implementasi sistem, pengujian sistem, serta evaluasi dan analisis hasil.

### 3.1 Tipe Penelitian

Tipe penelitian yang diangkat dalam penelitian ini adalah tipe penelitian nonimplementatif yang bersifat analitik. Hal ini disebabkan karena penelitian menitikberatkan pada investigasi terhadap suatu fenomena yaitu peramalan harga patokan batu bara. Proses investigasi permasalahan dilakukan dengan melakukan wawancara terhadap salah satu karyawan perusahaan tambang batu bara. Hasil yang diharapkan dalam penelitian ini adalah berupa analisis hasil prediksi harga patokan batu bara yang nantinya dapat menjadi salah satu solusi dalam menentukan keputusan penetapan harga patokan batu bara.

### 3.2 Studi Kepustakaan

Tahapan studi kepustakaan dilakukan sebagai pembelajaran untuk memahami, mengenal, dan mengetahui dasar-dasar teori dari bidang ilmu yang memiliki hubungan dengan penelitian Prediksi Harga Patokan Batu Bara Untuk Merek Dagang Gunung Bayan I dengan Metode *Extreme Learning Machine* (ELM). Berikut merupakan studi kepustakaan yang digunakan.

1. Jaringan saraf tiruan
2. Teori Metode *Extreme Learning Machine* (ELM)
3. Identifikasi terhadap pergerakan harga batu bara

Literatur yang diperoleh tentunya perlu memiliki dasar yang kuat, sehingga dapat dipertanggungjawabkan nilai kebenarannya. Sumber pembelajaran diperoleh dari sumber seperti buku, jurnal, *paper*, *e-book*, serta penelitian yang telah dilakukan sebelumnya.

### 3.3 Metode Pengumpulan Data

Pengumpulan data yang digunakan dalam penelitian ini berupa data sekunder. Data didapatkan dari laman resmi Ditjen Minerba, Kementerian ESDM. Data yang digunakan dalam penelitian merupakan data *time series* harga patokan batu bara dengan merek dagang Gunung Bayan I. Data berbentuk runtut waktu per bulan dari Januari 2009 hingga September 2017. Daftar harga tersebut, nantinya akan digunakan sebagai fitur dalam proses prediksi harga batu bara. Contoh data harga patokan batu bara untuk merek dagang Gunung Bayan I dapat ditunjukkan pada Tabel 3.1.

Tabel 3.1 Contoh Data Harga Patokan Batu Bara

Tanggal	HPB Gunung Bayan I (US\$/ton)
Januari 2009	84,65
Februari 2009	87,51
Maret 2009	80,75
April 2009	67,72
Mei 2009	67,45
Juni 2009	68,58
Juli 2009	76,62
Agustus 2009	76,82
September 2009	75,70
Oktober 2009	71,66
November 2009	74,13
Desember 2009	80,11
Januari 2010	83,22
Februari 2010	94,61
Maret 2010	93,25

### 3.4 Strategi dan Lokasi Penelitian

Strategi penelitian dilakukan dengan melakukan eksperimen terhadap parameter data serta parameter dari algoritme ELM. Proses eksperimen dilakukan dengan melakukan percobaan sebanyak 10 kali. Hal ini disebabkan karena hasil perhitungan ELM akan menghasilkan nilai yang berbeda pada setiap eksekusi yang diakibatkan oleh *random* bobot dan bias. Untuk mendapatkan kualitas yang baik dari hasil pengujian, maka pengujian akan dilakukan sebanyak 10 kali.

Eksperimen parameter data dilakukan dengan mengubah banyaknya jumlah fitur data yang digunakan dalam proses prediksi. Fitur data berisi data historis harga patokan batu bara yang disusun sebagai data masukan dalam proses prediksi. Eksperimen parameter algoritme ELM dilakukan dengan mengubah banyaknya jumlah *neuron* dan jenis fungsi aktivasi yang digunakan. Hasil eksperimen yang mampu menghasilkan nilai *error* terkecil, merupakan hasil terbaik dari proses prediksi menggunakan algoritme ELM.

Lokasi untuk melakukan penelitian dilakukan di Laboratorium Riset Fakultas Ilmu Komputer Universitas Brawijaya.

### 3.5 Peralatan Pendukung

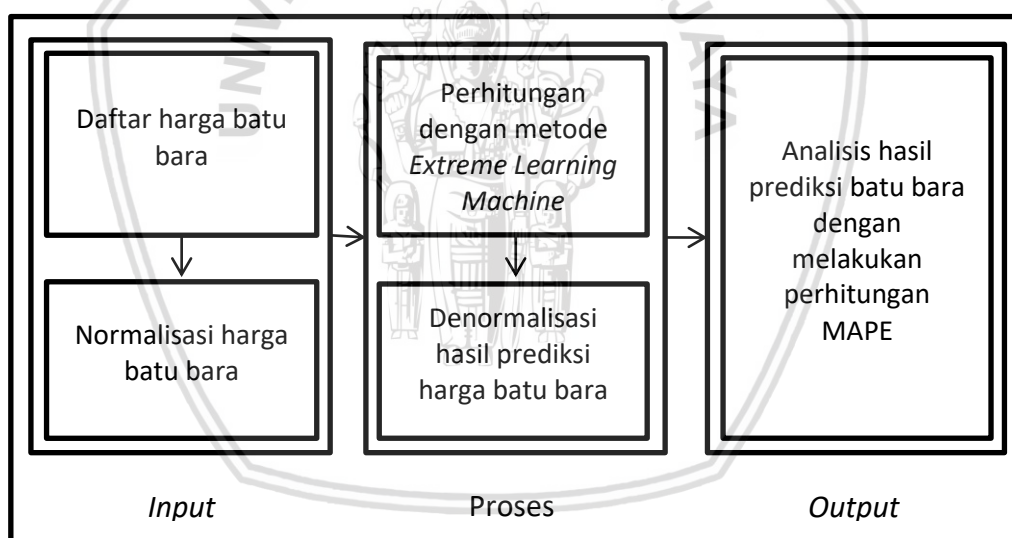
Tahapan analisis kebutuhan dilakukan untuk merumuskan kebutuhan-kebutuhan pembangunan suatu sistem prediksi. Berikut merupakan kebutuhan dalam pemodelan sistem prediksi harga patokan batu bara untuk merek dagang Gunung Bayan I.

1. Kebutuhan *Hardware*: Laptop dengan spesifikasi prosesor *Intel Core™ i5-5200U@2.7GHz*, memori 4 GB, kapasitas *HDD* 500 GB, kartu grafik *NVIDIA GeForce 930M*.
2. Kebutuhan *Software*:
  - Sistem Operasi Windows 8.1 64 bit
  - Netbeans
  - JDK Java
3. Data yang dibutuhkan berupa daftar historis harga patokan batu bara.

### 3.6 Analisis dan Perancangan Sistem

Tahapan ini dilakukan untuk menganalisis dan merancang sistem dari segi model maupun arsitekturnya, terkait hal apa saja yang dibutuhkan sistem dalam proses pembuatannya. Hal tersebut terdiri dari deskripsi sistem, perancangan program aplikasi, perhitungan manual, desain *interface* atau antarmuka, serta perancangan pengujian. Metode pengerjaan akan disesuaikan dengan arsitektur yang telah dirancang.

#### 3.6.1 Model Perancangan Sistem



**Gambar 3.1 Model Perancangan Sistem Prediksi Harga Batu Bara**

Model perancangan sistem merepresentasikan struktur kinerja sistem, mulai dari proses *input* hingga proses menghasilkan *output*. Proses yang berlangsung dalam Gambar 3.1 diatas terdiri dari:

1. Masukan (*Input*)

Masukan dari sistem berupa banyaknya jumlah fitur, jumlah *neuron*, serta jenis fungsi aktivasi yang digunakan. Banyaknya jumlah fitur akan memengaruhi dalam proses penyusunan data historis harga batu bara sebagai masukan sistem dalam proses prediksi. Sebagai contoh, apabila di masukkan jumlah fitur sebanyak 3, maka akan digunakan daftar harga batu

bara satu bulan terakhir, dua bulan terakhir, dan tiga bulan terakhir, untuk menentukan prediksi harga batu bara bulan depan.

2. Proses

Proses perhitungan dalam penelitian ini menggunakan metode *Extreme Learning Machine* (ELM), dengan langkah-langkah sebagai berikut.

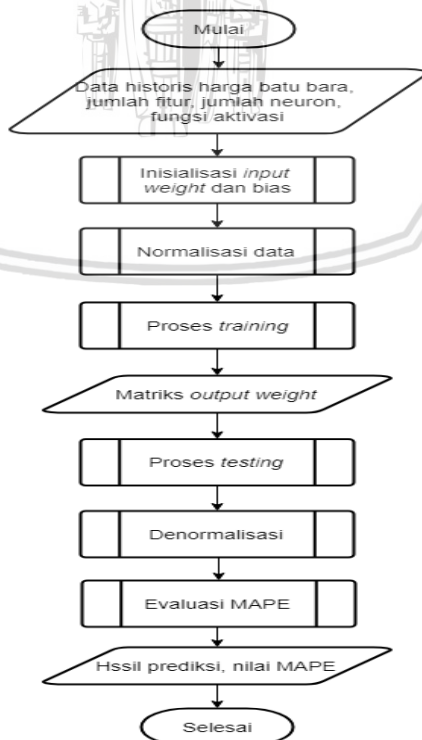
- Pengambilan nilai harga batu bara satu bulan terakhir, dua bulan terakhir, dan tiga bulan terakhir yang dimasukkan oleh pengguna
- Normalisasi nilai harga batu bara tersebut
- Inisialisasi bobot ( $w$ ) dan bias ( $b$ )
- Perhitungan matriks *output hidden neuron* ( $H_{init}$ )
- Perhitungan matriks *hidden neuron* menggunakan fungsi aktivasi ( $H$ )
- Perhitungan *pseudo-inverse* dengan *Moore-Penrose* ( $H^+$ )
- Perhitungan nilai bobot *output* ( $\hat{\beta}$ )
- Perhitungan nilai  $Y$  prediksi ( $\hat{Y}$ )
- Denormalisasi nilai  $Y$  prediksi ( $d$ )

3. Keluaran (*Output*)

Keluaran yang dihasilkan oleh sistem berupa prediksi harga batu bara dengan merek Gunung Bayan I untuk satu bulan ke depan.

**3.6.2 Diagram Alir Sistem**

Diagram alir pada Gambar 3.2 berisi gambaran dari alur proses kerja sistem. Diagram alir sistem terdiri dari proses memasukkan data, normalisasi data, proses *training*, proses *testing*, denormalisasi, serta proses evaluasi.



**Gambar 3.2 Diagram Alir Sistem**



### 3.7 Implementasi

Tahapan implementasi merupakan tahapan pembentuk sistem yang didasari oleh hal-hal yang telah dipelajari sebelumnya untuk membuat dan membangun sistem prediksi harga batu bara. Pembuatan sistem, didasari oleh tahapan pada perancangan sistem. Pengembangan dilakukan dengan menggunakan bahasa pemrograman Java. Berikut merupakan tahapan dalam implementasi sistem.

1. Implementasi GUI dengan menggunakan Java pada Netbeans
2. Implementasi algoritme menggunakan metode *Extreme Learning Machine* (ELM) yang diimplementasikan dengan bahasa pemrograman Java
3. *Output* sistem berupa informasi hasil prediksi harga batu bara untuk merek dagang Gunung Bayan I. Harga tersebut merupakan harga prediksi untuk satu bulan ke depan.

### 3.8 Pengujian Sistem

Pengujian sistem ini dilakukan untuk memastikan bahwa suatu aplikasi dalam penelitian ini dapat bekerja sesuai dengan tujuan yang diharapkan. Pengujian dilakukan dengan beberapa tahap, yaitu:

1. Pengujian jumlah *neuron*  
Pengujian ini bertujuan untuk mencari jumlah *neuron* terbaik. Hasil yang diharapkan dari pengujian ini adalah untuk mendapatkan jumlah *neuron* terbaik yang dapat menghasilkan nilai *error* yang kecil.
2. Pengujian fitur data  
Pengujian ini bertujuan untuk mencari kombinasi fitur terbaik yang dapat menghasilkan prediksi terbaik dengan nilai *error* yang kecil.
3. Pengujian fungsi aktivasi  
Pengujian ini bertujuan untuk mencari fungsi aktivasi terbaik yang sesuai dengan karakteristik data penelitian. Fungsi aktivasi yang akan diuji terdiri dari fungsi aktivasi *sigmoid biner*, *sigmoid bipolar*, dan fungsi *sin*.

### 3.9 Evaluasi dan Analisis Hasil

Tahapan ini dilakukan setelah seluruh proses pengujian sistem dilakukan. Tahapan ini bertujuan untuk membandingkan dan mempertimbangkan tingkat kesuksesan aplikasi dari implementasi metode dalam menyelesaikan permasalahan dalam melakukan prediksi harga batu bara untuk merek dagang Gunung Bayan I dengan metode *Extreme Learning Machine* (ELM). Tahapan akhir dari seluruh rangkaian proses ini adalah pengambilan kesimpulan dan saran terhadap hasil yang diperoleh, sehingga hal ini nantinya dapat dijadikan acuan dan bahan pembelajaran untuk pembaca serta pengembang aplikasi lainnya.

## BAB 4 PERANCANGAN DAN IMPLEMENTASI

Bab ini berisi formulasi permasalahan, arsitektur perancangan sistem, diagram alir sistem, siklus penyelesaian prediksi harga patokan batu bara untuk merek dagang Gunung Bayan I menggunakan *Extreme Learning Machine* (ELM), perhitungan manual, perancangan *user interface*, perancangan skenario pengujian, serta bentuk implementasi sistem.

### 4.1 Formulasi Permasalahan

Permasalahan yang akan diselesaikan dalam penelitian ini adalah permasalahan dalam melakukan prediksi harga patokan batu bara untuk merek dagang Gunung Bayan I. Metode peramalan yang digunakan adalah *Extreme Learning Machine* (ELM). Metode tersebut diketahui mampu memberikan hasil peramalan dengan tingkat akurasi yang baik apabila dibandingkan dengan metode lain, serta memiliki *learning speed* yang tinggi. Hasil prediksi dapat dilihat berdasarkan besarnya tingkat kesalahan. Semakin kecil nilai kesalahan yang dihasilkan maka hasil prediksi semakin baik.

Sistem peramalan ini memiliki data masukan berupa daftar harga batu bara serta beberapa parameter perhitungan yang terdiri dari jumlah *neuron* pada *hidden layer*, banyaknya data *training* dan data *testing*, *input weight* ( $w$ ), serta bias ( $b$ ). Banyaknya *input weight* ditentukan dari banyaknya *input node* dan *hidden neuron*. Penelitian ini akan menggunakan tiga buah masukan yang digunakan untuk memprediksi harga patokan batu bara (HPB) bulan berikutnya. Masukan tersebut terdiri dari, HPB satu bulan terakhir, HPB dua bulan terakhir, dan HPB tiga bulan terakhir. Data masukan tersebut akan diproses dengan menggunakan 3 *hidden neuron*, sehingga diperoleh matriks dengan ordo  $3 \times 3 = 9$  bobot yang terletak antara *input layer* dan *hidden layer*. Dalam pengerjaannya, terdapat dua proses utama yang akan dikerjakan, yaitu proses *training* dan proses *testing*. Sebelum melakukan proses prediksi maka perlu dilakukan proses *preprocessing* terhadap data untuk mengubah data kedalam rentang  $[0,1]$ . Hal tersebut dilakukan untuk menyamakan *output* yang akan dihasilkan oleh proses aktivasi. Proses *training* diawali dengan melakukan proses perhitungan keluaran *hidden neuron* menggunakan fungsi aktivasi yang kemudian didapatkan hasilnya dalam bentuk matriks. Proses perhitungan dilanjutkan hingga mendapatkan nilai *output weight*. Nilai *output weight* yang didapatkan dari proses *training* akan digunakan kembali pada proses *testing* untuk mendapatkan *output* berupa hasil prediksi. Setelah didapatkan *output* dari sistem, maka akan dihitung tingkat kesalahannya dengan menggunakan perhitungan *mean absolute percentage error* (MAPE).

Adapun sampel data historis harga patokan batu bara untuk merek dagang gunung bayan I yang digunakan dalam penelitian ini ditunjukkan pada Tabel 4.1.

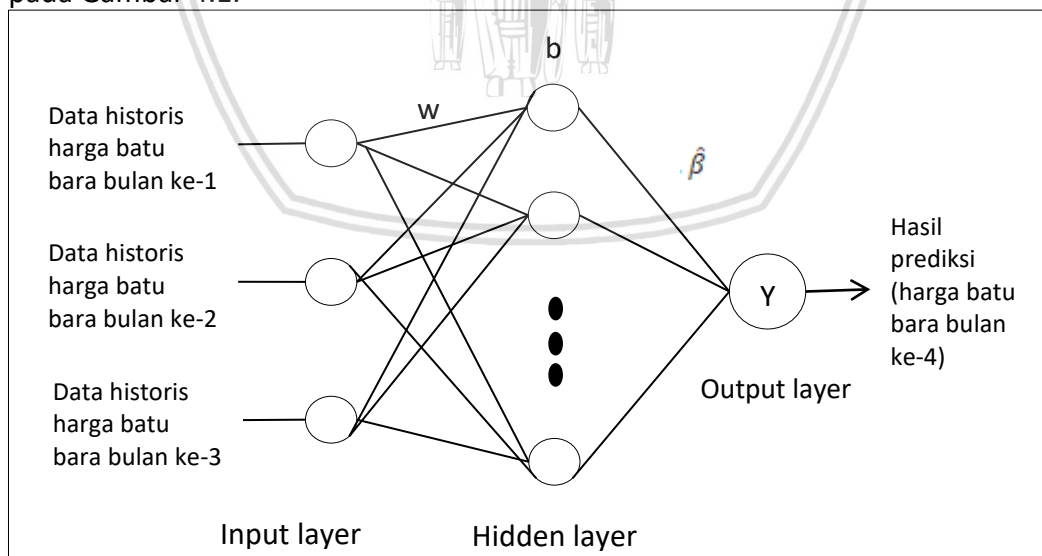


Tabel 4.1 Data Sampel Historis Harga Patokan Batu Bara

Tanggal	HPB Gunung Bayan I (US\$/ton)
Desember 2016	109,35
Januari 2017	92,60
Febuari 2017	89,45
Maret 2017	87,91
April 2017	88,57
Mei 2017	89,98
Juni 2017	80,94
Juli 2017	84,72
Agustus 2017	90,15
September 2017	98,88

#### 4.2 Arsitektur Perancangan Sistem

Arsitektur perancangan sistem berisi gambaran sistem secara umum dalam proses prediksi harga patokan batu bara dengan menggunakan metode *Extreme Learning Machine* (ELM). Data masukan yang digunakan untuk proses prediksi terdiri dari harga tiga bulan terakhir. Berikut merupakan bentuk arsitektur dari jaringan ELM untuk proses prediksi harga patokan batu bara yang ditunjukkan pada Gambar 4.1.



Gambar 4.1 Arsitektur Perancangan Sistem

Berdasarkan arsitektur perancangan, setiap *neuron* pada *hidden layer* akan terhubung pada *hidden neuron* pada *hidden layer*. *Neuron* tersebut dihubungkan melalui *input weight* dengan nilai yang berbeda. Setiap *neuron* pada *hidden layer*



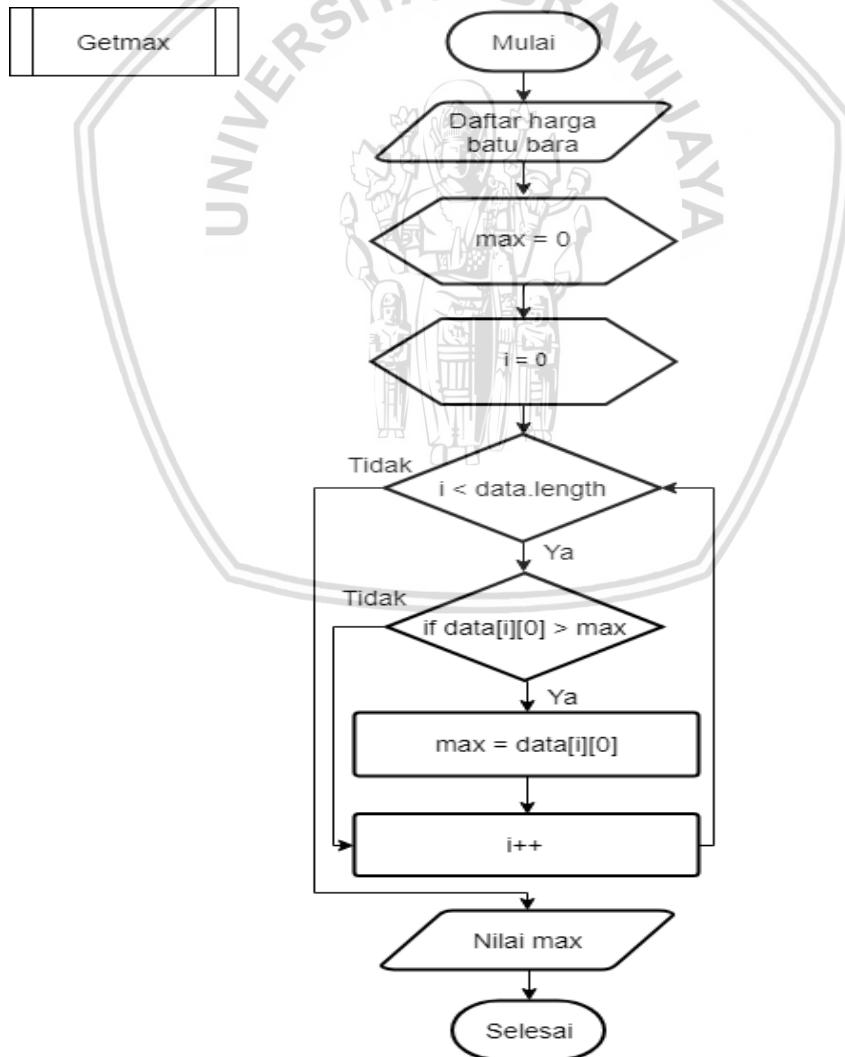
dihubungkan dengan *output layer* menggunakan *output weight*. Hasil prediksi sistem berupa satu nilai *output* yang dihubungkan dari semua *hidden layer*.

### 4.3 Alur Kerja Sistem

Alur kerja sistem berisi gambaran dari alur proses kerja sistem sesuai dengan diagram alir sistem pada Gambar 3.2. Alur kerja sistem terdiri dari proses memasukkan data, normalisasi data, proses *training*, proses *testing*, denormalisasi, serta proses evaluasi.

#### 4.3.1 Normalisasi

Proses normalisasi yang ditunjukkan pada Gambar 4.4 dilakukan untuk melakukan standardisasi terhadap semua data yang digunakan. Hal ini bertujuan untuk mengubah data agar berada pada rentang tertentu. Metode normalisasi yang digunakan dalam penelitian ini adalah *min max normalization*. Untuk melakukan proses normalisasi, dibutuhkan sebuah proses untuk mendapatkan nilai *maximum* dan *minimum* dari data seperti pada Gambar 4.2 dan Gambar 4.3.

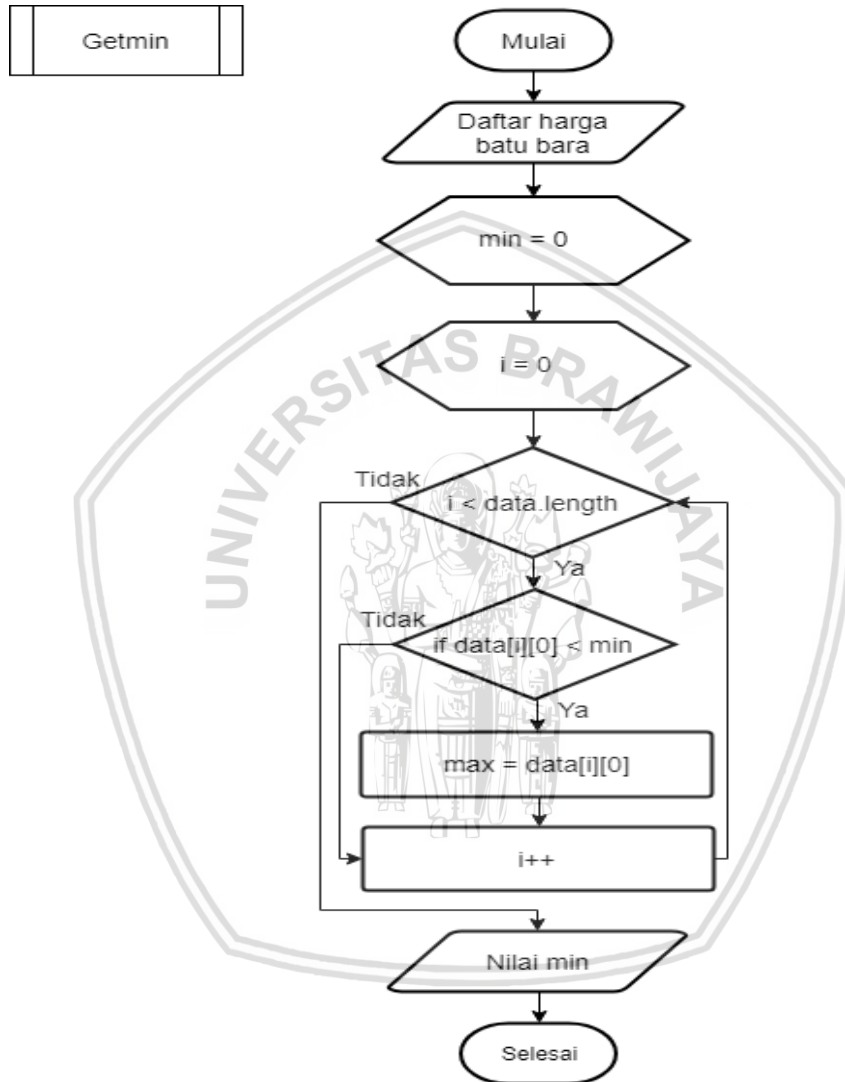


Gambar 4.2 Diagram Alir Proses *Getmax*



Berikut merupakan langkah-langkah proses pencarian nilai *maximum* dari data berdasarkan Gambar 4.2.

1. Masukan berupa data harga patokan batu bara
2. Melakukan pencarian nilai *maximum* dengan membandingkan nilai data satu dengan data lainnya
3. Menampilkan nilai *maximum* dari data.

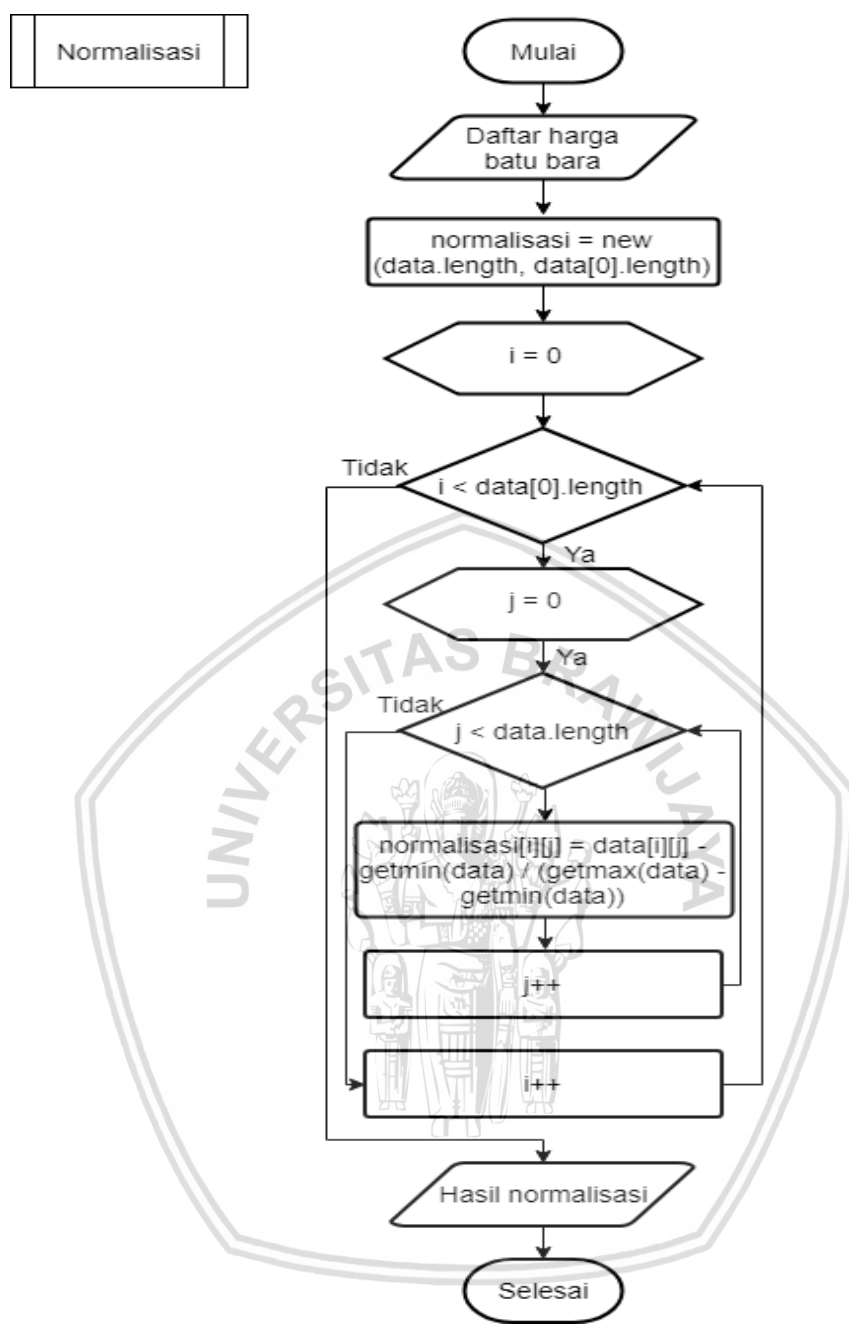


**Gambar 4.3 Diagram Alir Proses Getmin**

Berikut merupakan langkah-langkah proses pencarian nilai *minimum* dari data berdasarkan Gambar 4.3.

1. Masukan berupa data harga patokan batu bara
2. Melakukan pencarian nilai *minimum* dengan membandingkan nilai data satu dengan data lainnya
3. Menampilkan nilai *minimum* dari data.





Gambar 4.4 Diagram Alir Normalisasi Data

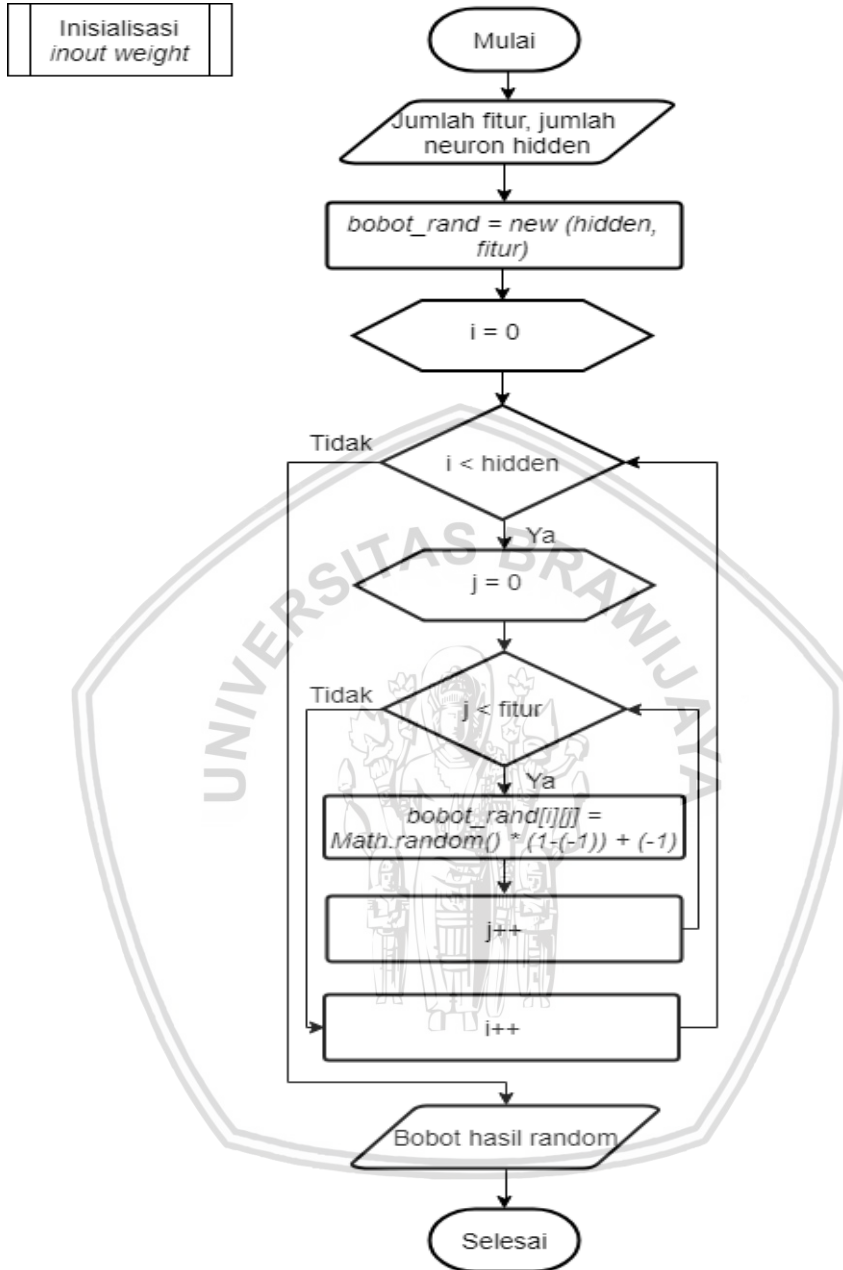
Berikut merupakan langkah-langkah proses normalisasi data menggunakan *Min-Max Normalization* berdasarkan Gambar 4.4.

1. Masukan berupa data harga patokan batu bara
2. Melakukan proses perhitungan normalisasi dengan menggunakan Persamaan 2.10
3. Menampilkan hasil proses normalisasi.

**4.3.2 Inisialisasi *Input Weight* dan Bias**

Inisialisasi *input weight* (w) dan bias (b) dilakukan secara acak dengan rentang [-1,1] untuk inisialisasi *input weight* dan [0,1] untuk inisialisasi bias.

Banyaknya jumlah inialisasi untuk *input weight* dan bias ditentukan berdasarkan jumlah *neuron* pada *hidden layer*.

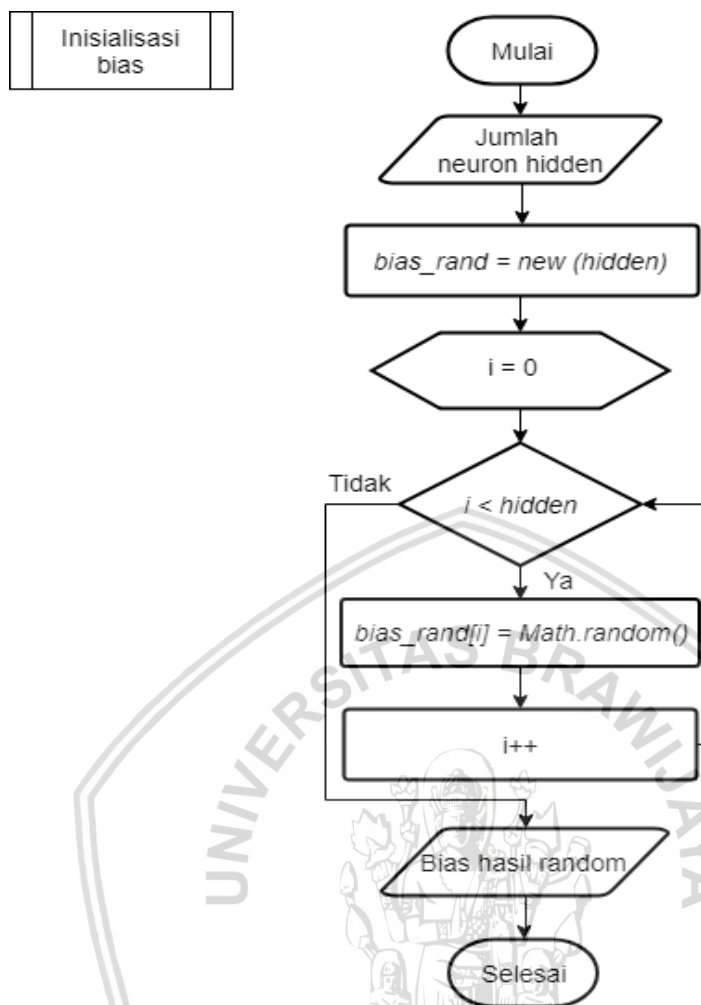


**Gambar 4.5 Diagram Alir Inisialaisasi *Input Weight***

Berikut merupakan langkah-langkah proses inialisasi *input weight* ( $w$ ) berdasarkan Gambar 4.5.

1. Masukan berupa jumlah fitur dan jumlah *neuron hidden*
2. Melakukan proses inialisasi *input weight* ( $w$ ) dengan rentang  $[-1,1]$
3. Menampilkan hasil inialisasi *input weight* ( $w$ ).





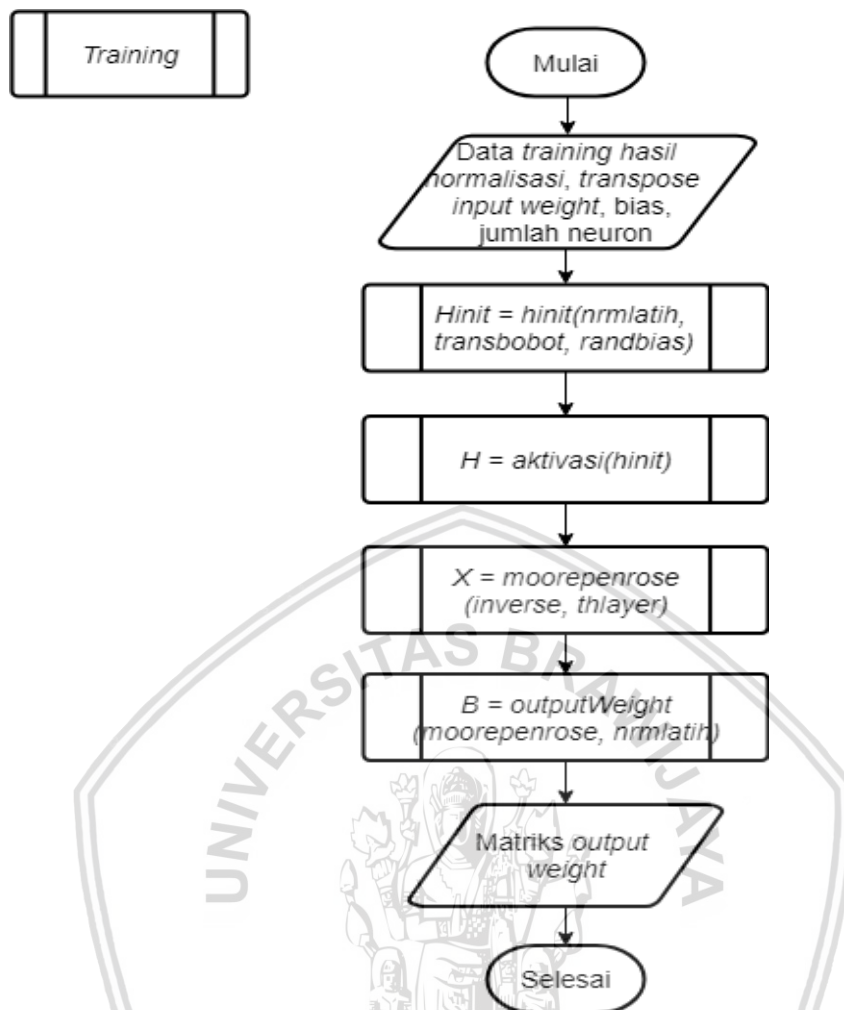
**Gambar 4.6 Diagram Alir Inisialisasi Bias**

Berikut merupakan langkah-langkah proses inisialisasi bias ( $b$ ) berdasarkan Gambar 4.6.

1. Masukan berupa jumlah jumlah *neuron hidden*
2. Melakukan proses inisialisasi bias ( $b$ ) dengan rentang [0,1]
3. Menampilkan hasil inisialisasi bias ( $b$ ).

### 4.3.3 Proses *Training*

Proses *training* merupakan proses awal dari peramalan menggunakan metode *Extreme Learning Machine* (ELM). Proses *training* dilakukan untuk mendapatkan *input weight* ( $w$ ) dan *output weight* ( $\hat{\beta}$ ) yang kemudian akan digunakan untuk proses *testing*. Nilai *input weight* ( $w$ ) dan *output weight* ( $\hat{\beta}$ ) yang didapatkan harus mampu memiliki nilai *error* yang rendah untuk dapat menghasilkan hasil prediksi yang baik. Alur dari proses *training* dapat dilihat pada Gambar 4.7.



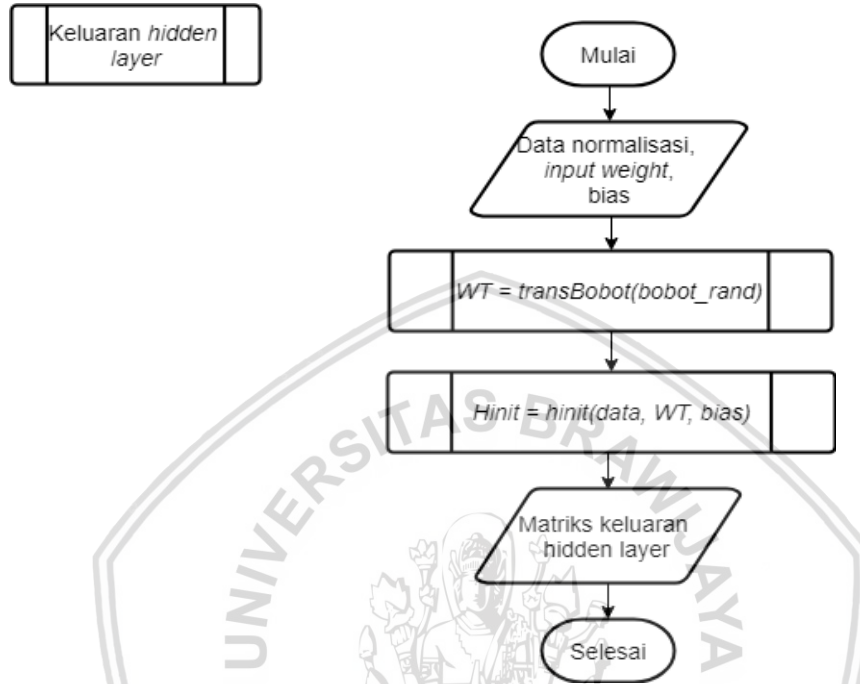
**Gambar 4.7 Diagram Alir Proses Training**

Berikut merupakan langkah-langkah proses *training* berdasarkan Gambar 4.7.

1. Sistem menerima masukan berupa data *training* yang sudah dinormalisasi, *input weight* ( $w$ ), bias, dan jumlah *neuron*.
2. Menghitung keluaran pada *hidden layer* ( $H_{init}$ ) menggunakan Persamaan 2.5 yang ditunjukkan pada Gambar 4.10.
3. Menghitung keluaran fungsi aktivasi ( $H$ ) menggunakan Persamaan 2.6. Diagram alir untuk proses perhitungan keluaran *hidden layer* dengan fungsi aktivasi ( $H$ ) ditunjukkan pada Gambar 4.11.
4. Menghitung matriks *Moore-Penrose pseudo inverse* ( $H^+$ ) dengan menggunakan Persamaan 2.7. Bentuk diagram alir dari proses tersebut ditunjukkan pada Gambar 4.12.
5. Menghitung matriks *output weight* ( $\hat{\beta}$ ) dengan mengalikan matriks hasil perhitungan *Moore-Penrose pseudo inverse* ( $H^+$ ) dengan matriks target. Rumus untuk melakukan proses tersebut dapat ditunjukkan pada Persamaan 2.8. Hasil *output weight* ( $\hat{\beta}$ ) yang didapatkan dari proses ini akan digunakan dalam proses *testing*. Diagram alir untuk menghitung *output weight* ( $\hat{\beta}$ ) ditunjukkan pada Gambar 4.13.

### 4.3.3.1 Proses Menghitung Keluaran *Hidden Layer*

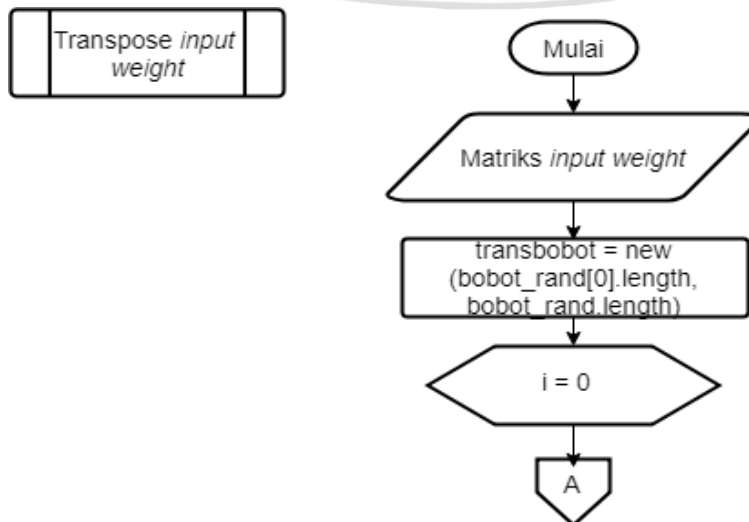
Proses perhitungan nilai keluaran *hidden layer* (*Hinit*) dilakukan dengan mengalikan nilai data hasil normalisasi ( $d'$ ) dengan hasil *transpose* bobot ( $w^T$ ). Kemudian tambahkan dengan nilai bias ( $b$ ). Bentuk diagram alir dari proses tersebut dapat dilihat pada Gambar 4.8.



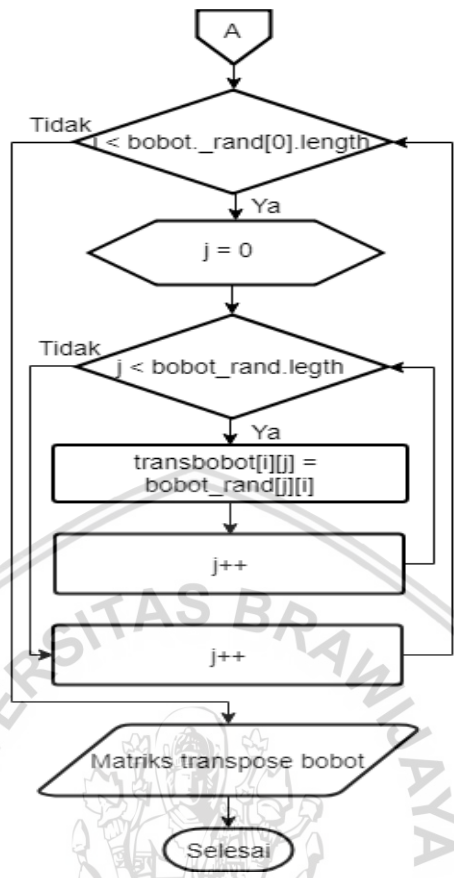
**Gambar 4.8 Diagram Alir Perhitungan Keluaran *Hidden Layer***

Berikut merupakan langkah-langkah perhitungan keluaran *hidden layer* berdasarkan Gambar 4.8.

1. Masukan berupa data hasil normalisasi ( $d'$ ), *input weight* ( $w$ ), dan bias ( $b$ )
2. Melakukan proses *transpose input weight* ( $w^T$ )
3. Melakukan proses perhitungan keluaran *hidden layer* (*Hinit*)
4. Hasil keluaran berupa matriks keluaran *hidden layer*.





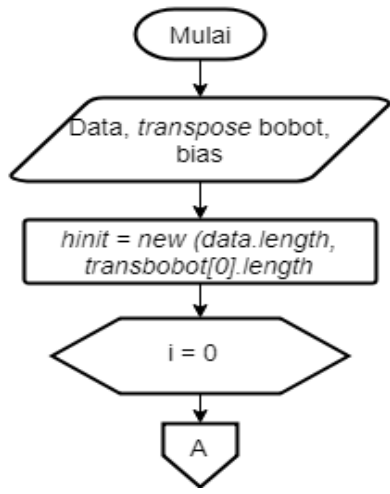


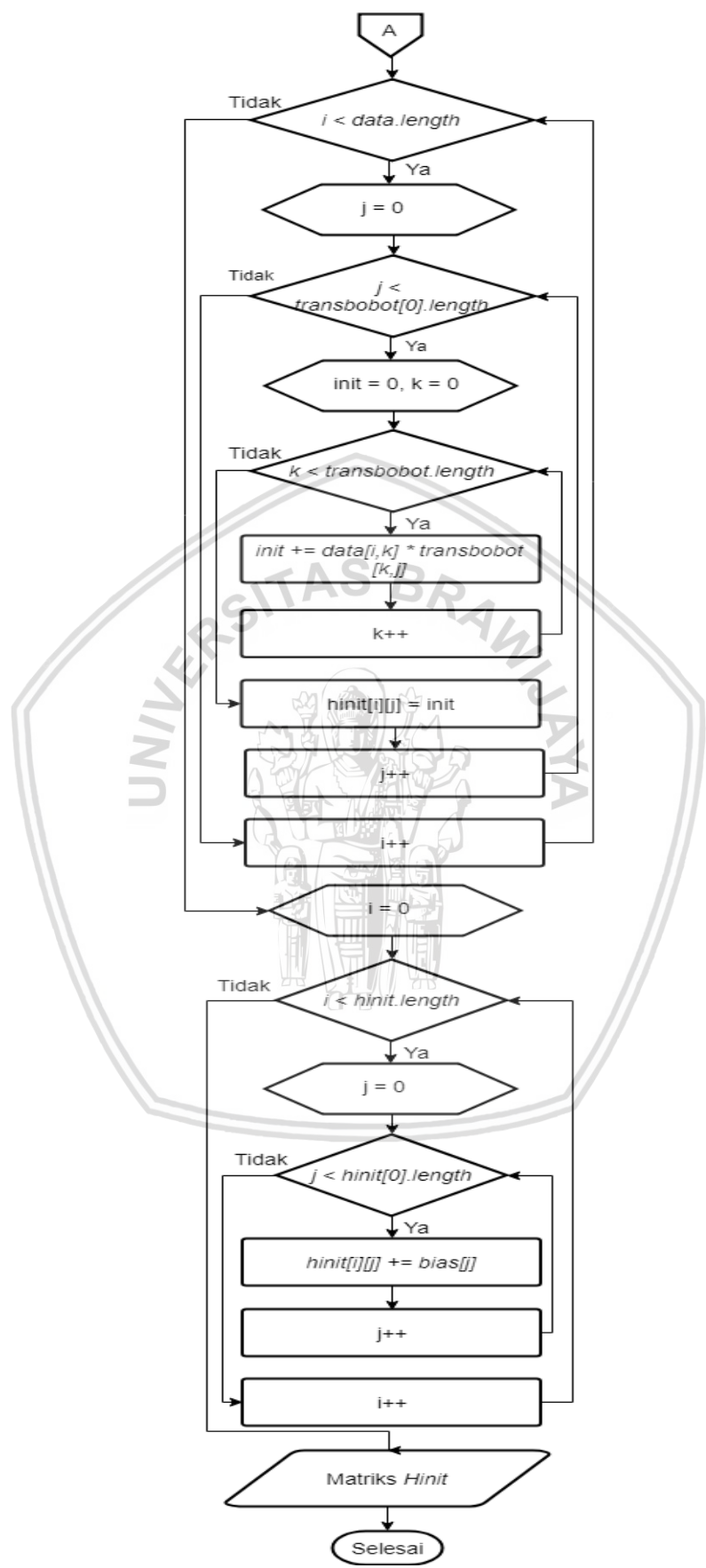
**Gambar 4.9 Diagram Alir Transpose Matriks *Input Weight***

Berikut merupakan langkah-langkah proses *tranpose input weight* ( $w^T$ ) berdasarkan Gambar 4.9.

1. Masukan data berupa matriks *input weight* ( $w$ )
2. Melakukan proses *transpose* dengan mengubah baris dan kolom suatu matriks dengan ordo baris  $x$  kolom menjadi suatu matriks baru dengan ordo kolom  $x$  baris
3. Hasil keluaran berupa matriks bobot hasil *transpose*.

Perhitungan <i>Hinit</i>
--------------------------





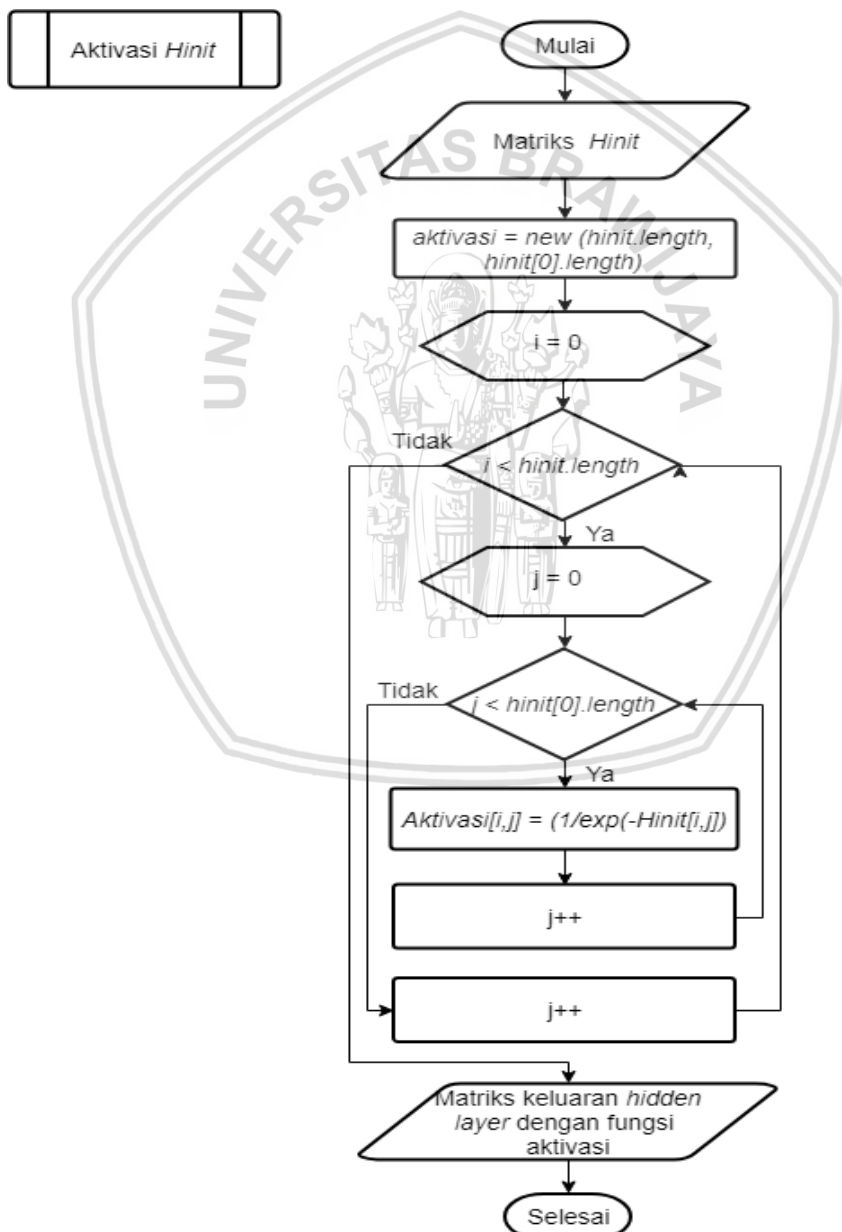
Gambar 4.10 Diagram Alir Perhitungan Keluaran Hidden Layer

Berikut merupakan langkah-langkah proses perhitungan keluaran *hidden layer* (*Hinit*) berdasarkan Gambar 4.10.

1. Masukan berupa matriks *input* hasil normalisasi, matriks bobot hasil *transpose* ( $w^T$ ), serta matriks bias ( $b$ ).
2. Melakukan perhitungan keluaran *hidden layer* (*Hinit*) dengan mengalikan baris *input* dengan nilai kolom *transpose* bobot ( $w^T$ ), kemudian ditambahkan dengan nilai bias ( $b$ ).
3. Hasil keluaran berupa matriks keluaran *hidden layer* (*Hinit*).

#### 4.3.3.2 Proses Menghitung Aktivasi Keluaran *Hidden Layer*

Proses perhitungan aktivasi dilakukan dengan fungsi aktivasi *sigmoid biner*. Diagram alir proses perhitungan aktivasi ditunjukkan pada Gambar 4.11.



Gambar 4.11 Diagram Alir Aktivasi Keluaran *Hidden Layer*

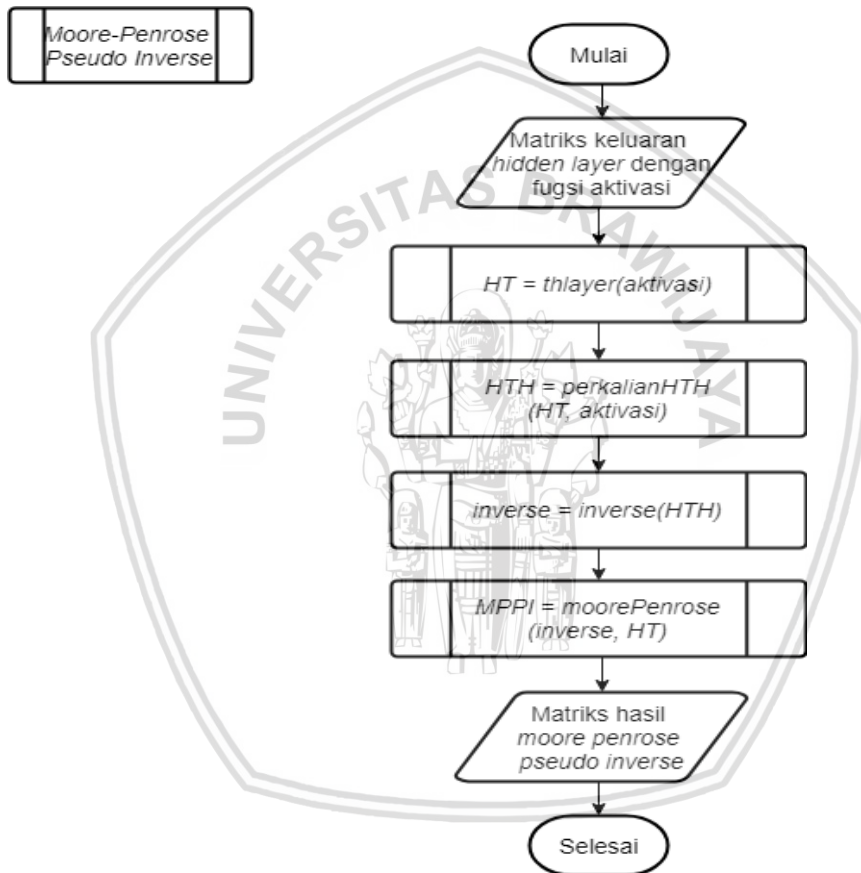


Berikut merupakan langkah-langkah untuk melakukan fungsi aktivasi terhadap hasil keluaran *hidden layer* ( $H$ ) berdasarkan Gambar 4.11.

1. Masukan berupa matriks keluaran *hidden layer* ( $H_{init}$ )
2. Melakukan proses perhitungan fungsi aktivasi dengan menggunakan Persamaan 2.6
3. Hasil keluaran berupa matriks keluaran *hidden layer* dengan fungsi aktivasi ( $H$ ).

#### 4.3.3.3 Proses Menghitung Moore-Penrose Pseudo Inverse

Diagram alir proses perhitungan matriks Moore-Penrose Pseudo Inverse ( $H^+$ ) ditunjukkan pada Gambar 4.12.



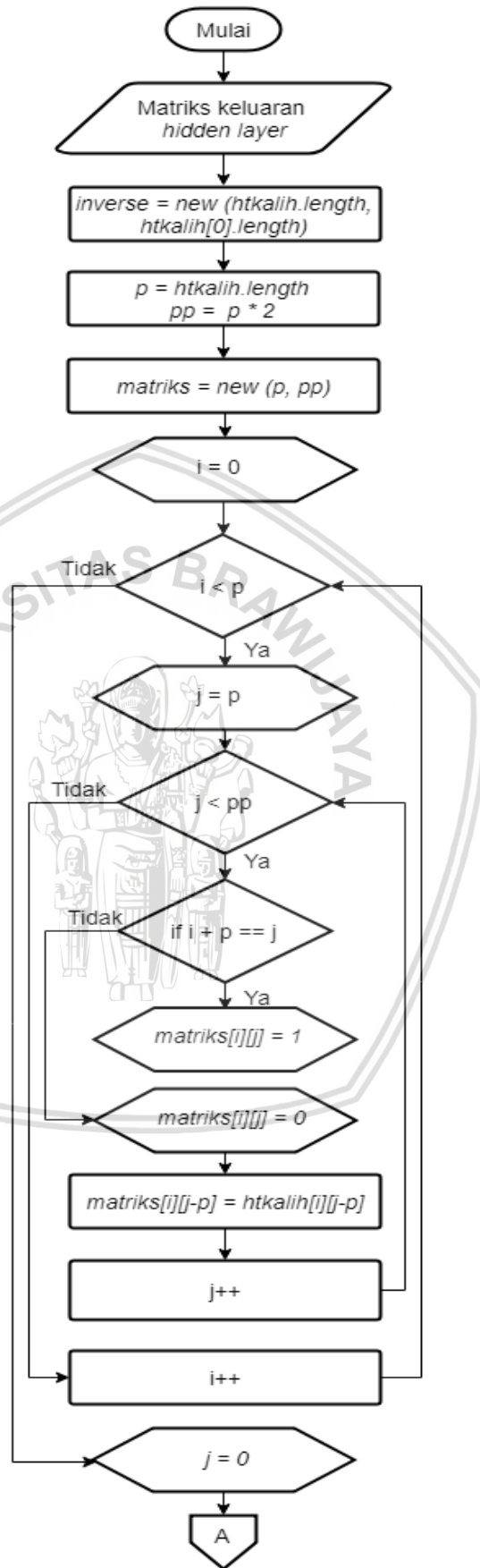
**Gambar 4.12 Diagram Alir Perhitungan Moore-Penrose Pseudo Inverse**

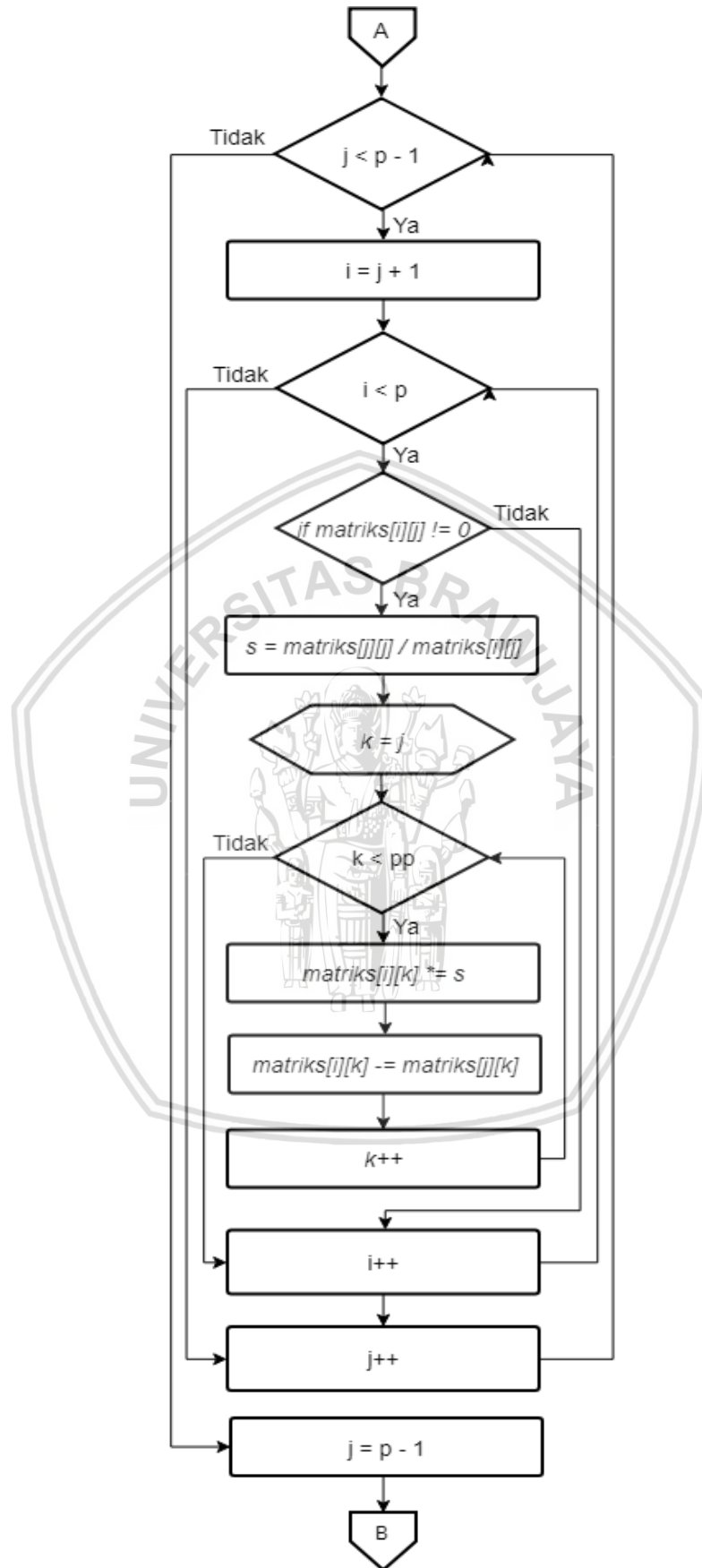
Berikut merupakan langkah-langkah untuk melakukan proses perhitungan Moore-Penrose Pseudo Inverse ( $H^+$ ) berdasarkan Gambar 4.12.

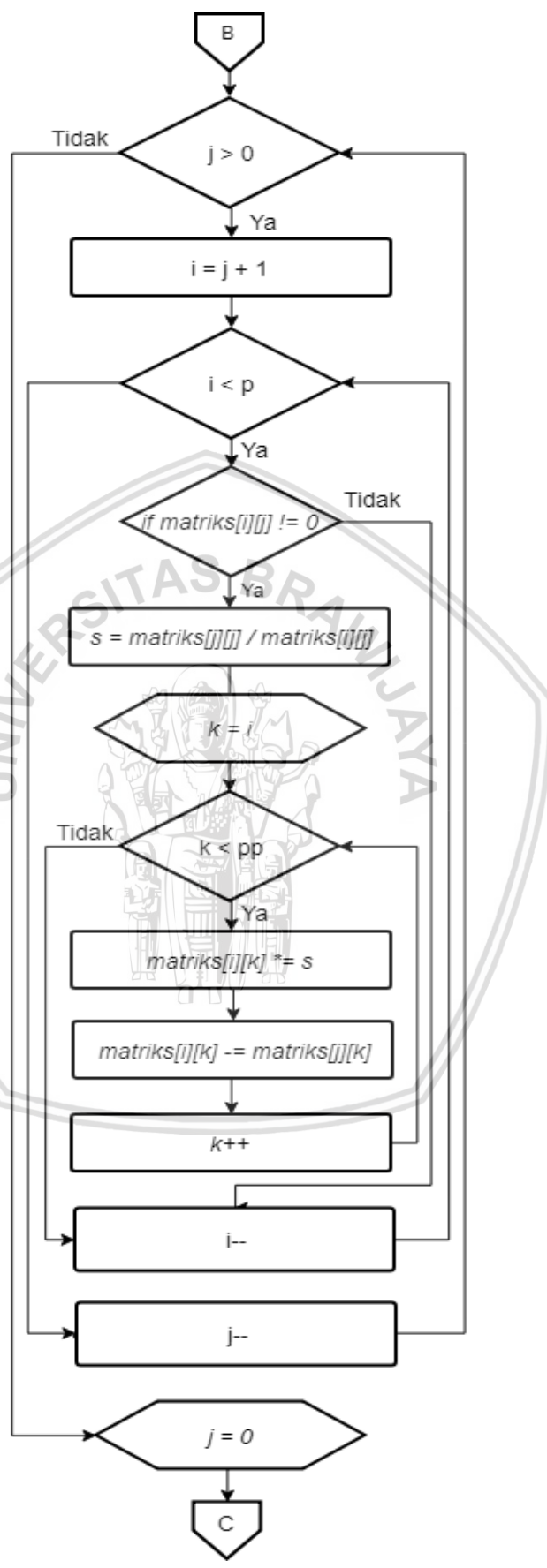
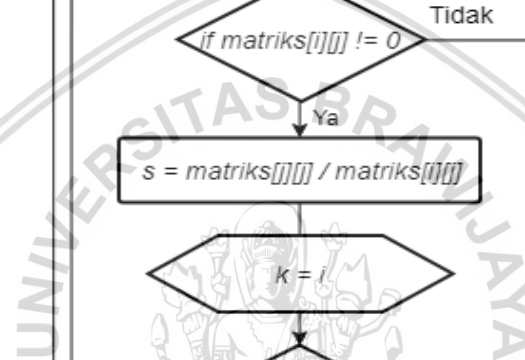
1. Masukan berupa matriks keluaran *hidden layer* yang sudah di aktivasi ( $H$ )
2. Melakukan *transpose* matriks hasil aktivasi ( $H^T$ )
3. Mengalikan hasil *transpose* dengan matriks hasil aktivasi awal
4. Mencari nilai *inverse* dari hasil perhitungan sebelumnya
5. Mengalikan hasil *inverse* dengan matriks *transpose* hasil aktivasi
6. Hasil keluaran berupa matriks hasil perhitungan Moore-Penrose Pseudo Inverse ( $H^+$ ).

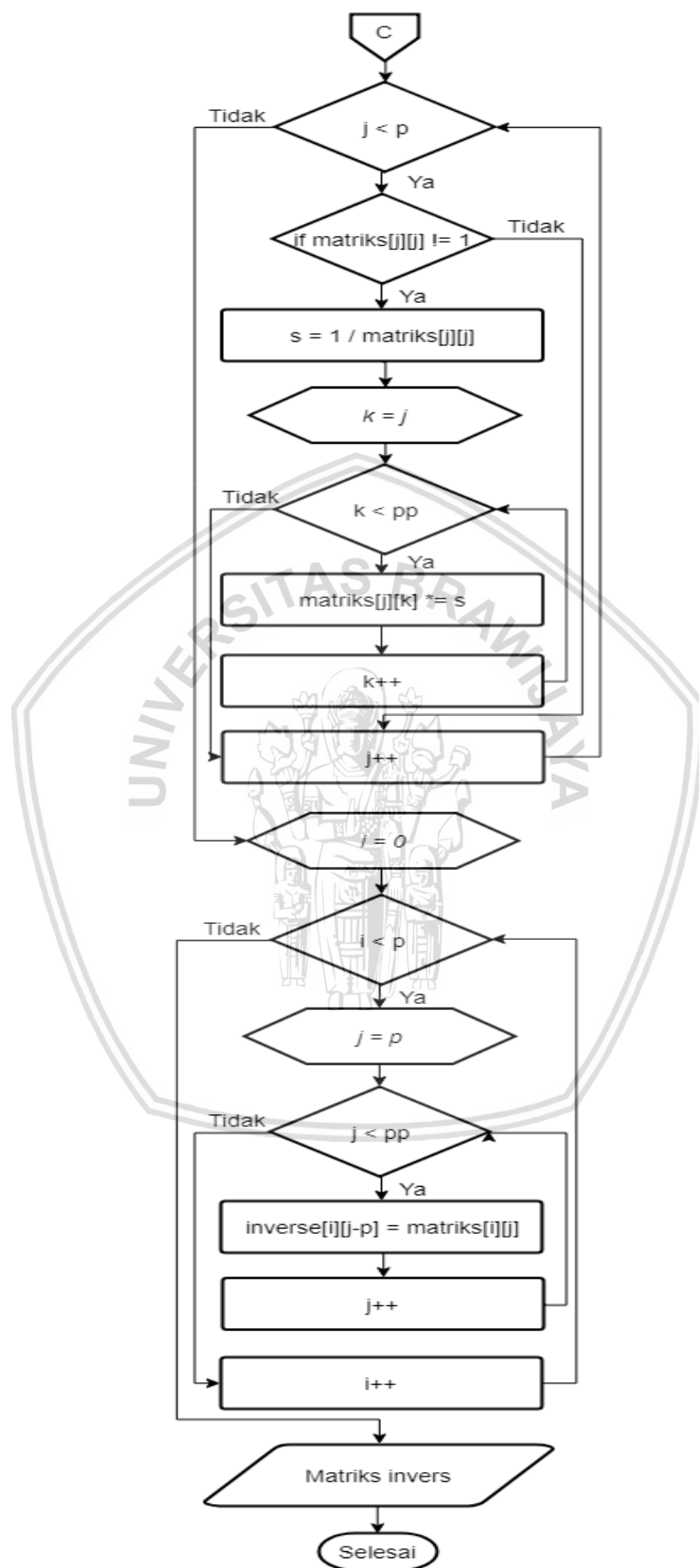


Inverse Matriks









Gambar 4.13 Diagram Alir Inverse Matriks



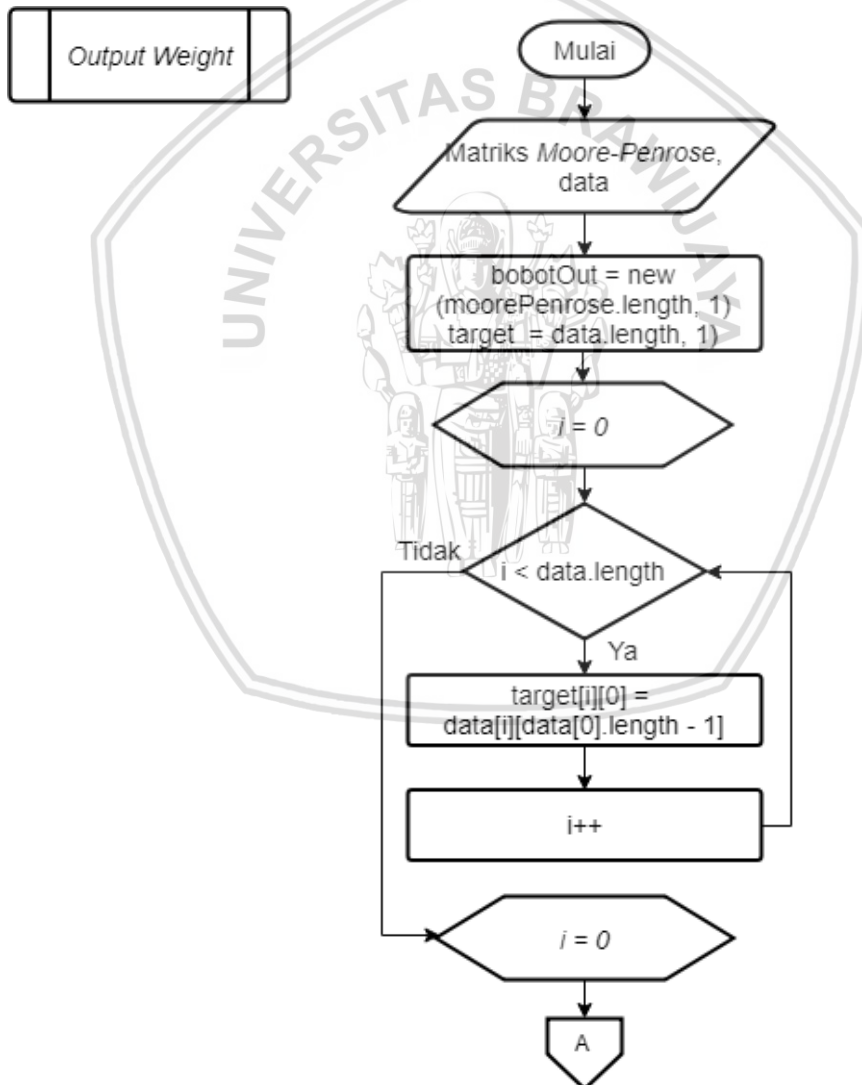


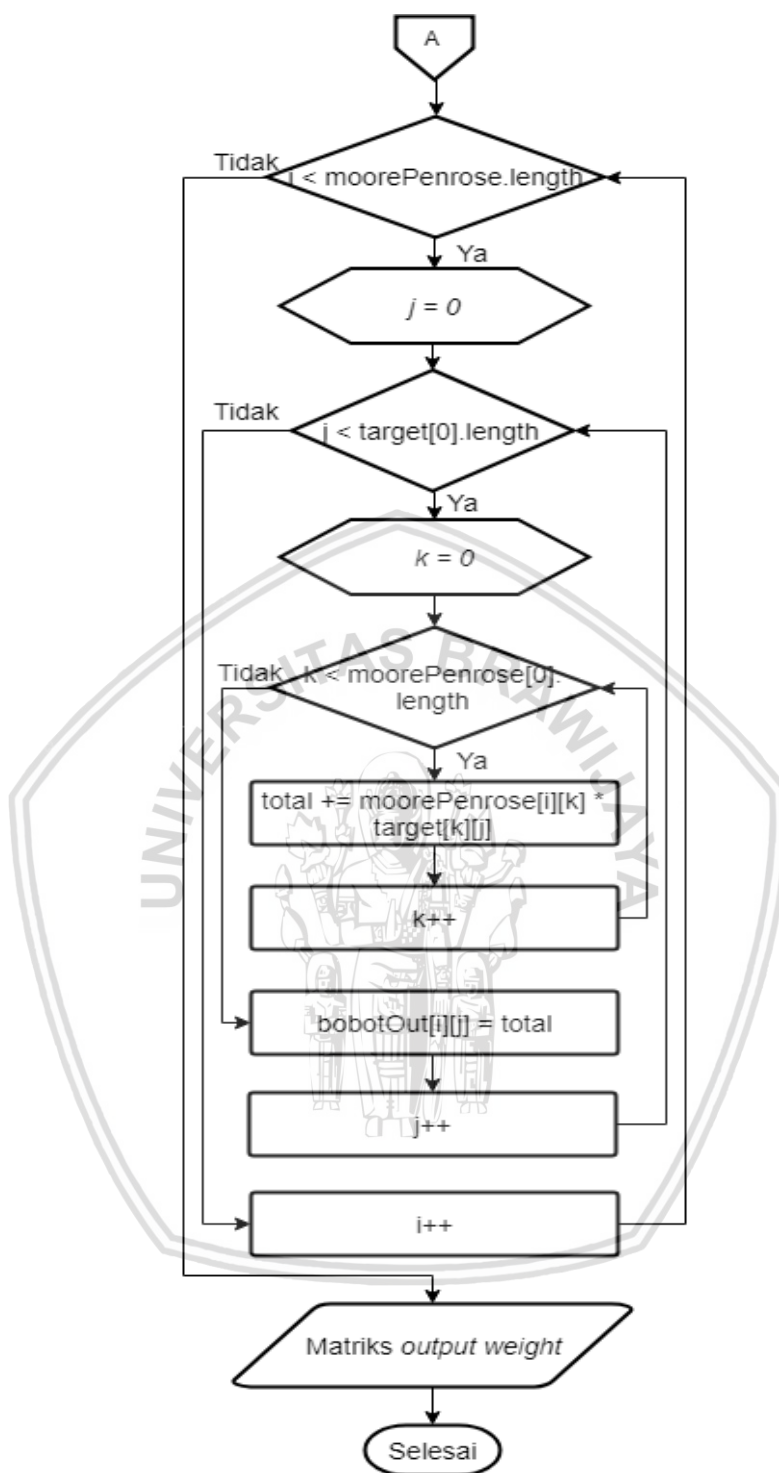
Berikut merupakan langkah-langkah proses perhitungan *inverse* matriks berdasarkan Gambar 4.13.

1. Inisialisasi matriks identitas sesuai dengan bentuk matriks ( $H/I$ ). Matriks identitas digunakan sebagai operator perhitungan *inverse* matriks dengan menggunakan operasi baris elementer (OBE).
2. Melakukan proses OBE dengan mengubah nilai pada matriks ( $H$ ) menjadi matriks identitas ( $I$ ) dan nilai matriks identitas menjadi sebuah matriks baru.
3. Nilai matriks baru yang merupakan perubahan dari matriks identitas merupakan hasil dari *inverse* matriks ( $H$ ).

#### 4.3.3.4 Proses Perhitungan Matriks *Output Weight*

Diagram alir proses perhitungan untuk menghasilkan matriks *output weight* ( $\hat{\beta}$ ) ditunjukkan pada Gambar 4.14.





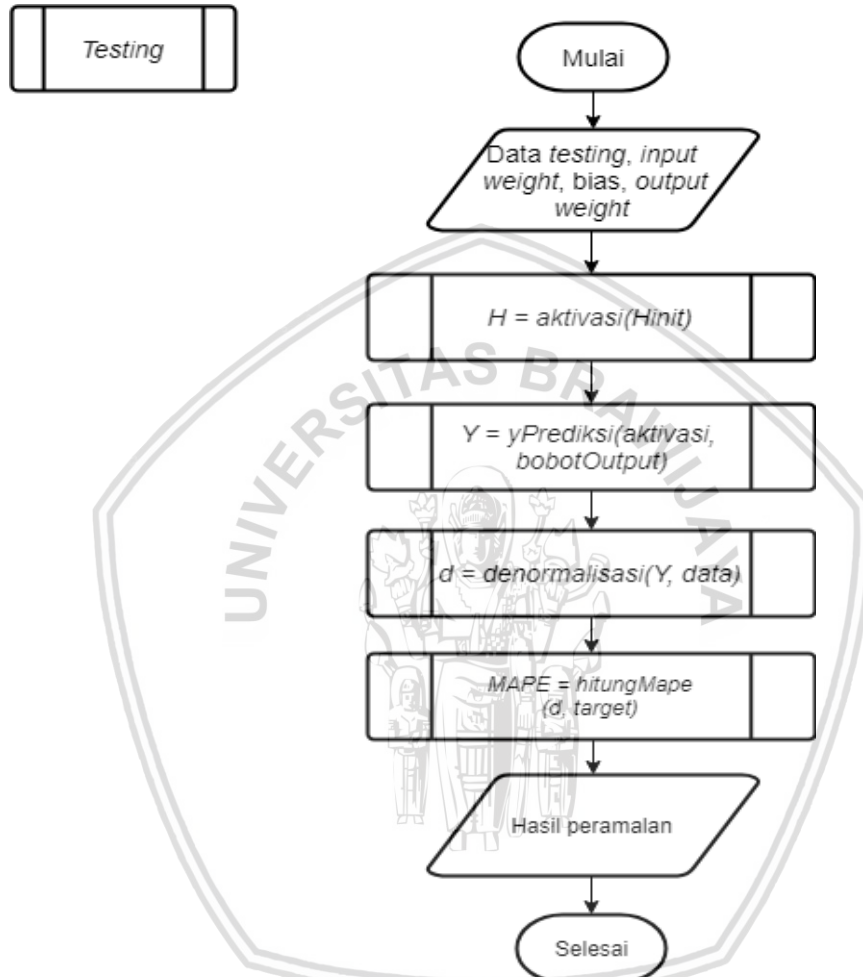
Gambar 4.14 Diagram Alir Output Weight

Berikut merupakan langkah-langkah proses perhitungan matriks *output weight* ( $\hat{\beta}$ ) berdasarkan Gambar 4.14.

1. Masukan berupa matriks hasil perhitungan *Moore-Penrose pseudo inverse* ( $H^+$ ) dan matriks target yang sudah dinormalisasi
2. Melakukan perkalian terhadap kedua matriks masukan tersebut
3. Hasil keluaran berupa nilai matriks *output weight* ( $\hat{\beta}$ ).

### 4.3.4 Proses Testing

Proses *testing* merupakan proses untuk mengevaluasi metode ELM berdasarkan hasil dari proses *training*. Beberapa parameter yang digunakan dalam proses *testing* didapatkan dari proses *training*. Parameter tersebut terdiri dari *input weight* ( $w$ ), *output weight* ( $\hat{\beta}$ ), dan bias ( $b$ ). Diagram alir dari proses *testing* ditunjukkan pada Gambar 4.15.



Gambar 4.15 Diagram Alir Proses Testing

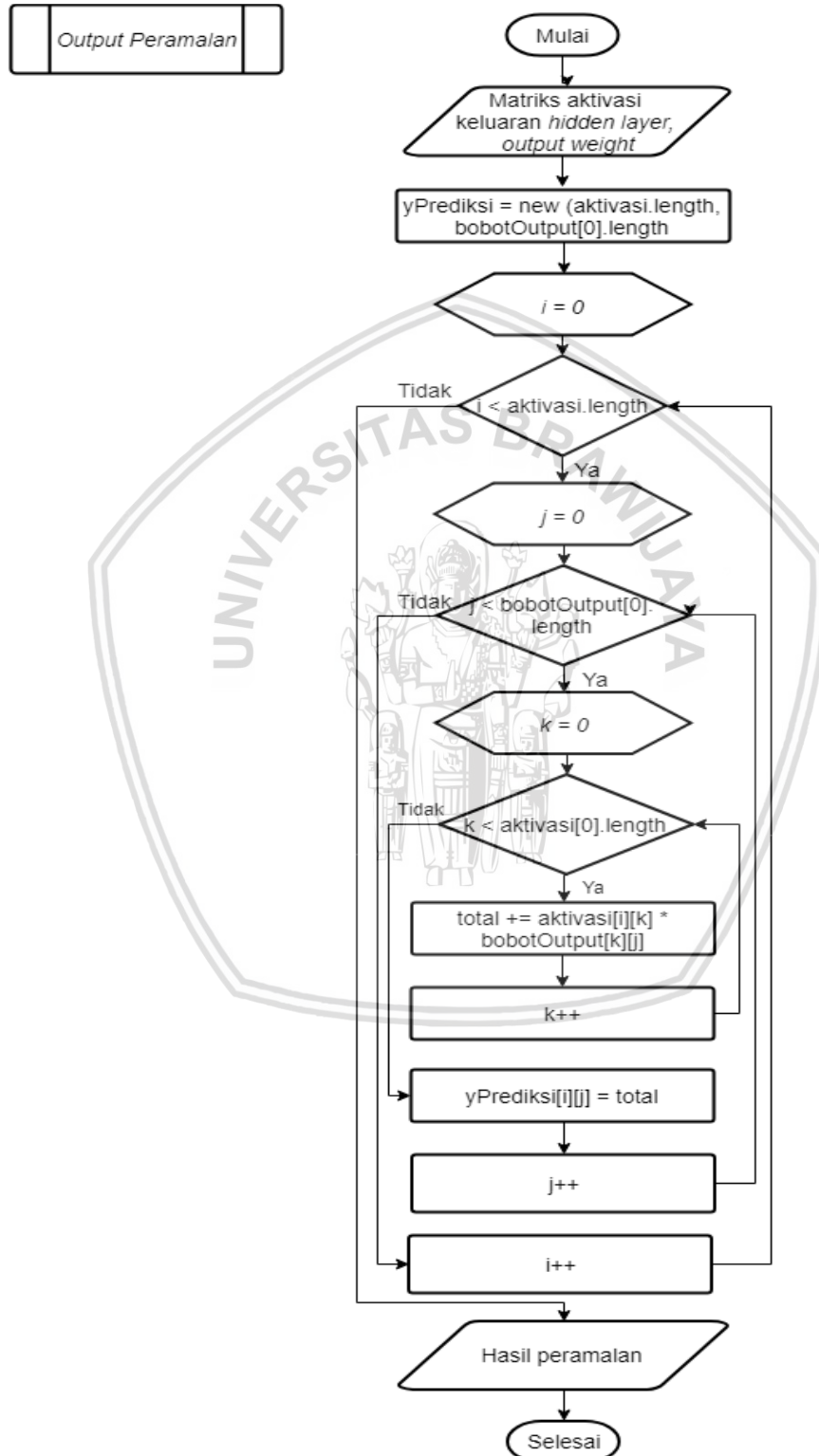
Berikut merupakan langkah-langkah proses *testing* berdasarkan Gambar 4.15.

1. Masukan berupa data *testing*, *input weight* ( $w$ ), *output weight* ( $\hat{\beta}$ ), dan bias ( $b$ )
2. Melakukan proses perhitungan untuk menghasilkan matriks keluaran *hidden layer* dengan fungsi aktivasi ( $H$ )
3. Hitung hasil peramalan ( $\hat{Y}$ )
4. Melakukan proses denormalisasi terhadap hasil peramalan ( $d$ )
5. Menghitung nilai tingkat kesalahan dengan *means absolute percentage error* (MAPE)
6. Hasil keluaran berupa hasil peramalan ( $\hat{Y}$ ).



### 4.3.4.1 Proses Perhitungan *Output* Peramalan

*Output* permalan ( $\hat{Y}$ ) merupakan hasil peramalan dengan menggunakan metode ELM. Diagram alir dari proses perhitungan *output* peramalan ( $\hat{Y}$ ) ditunjukkan pada Gambar 4.16.



Gambar 4.16 Diagram Alir *Output* Peramalan

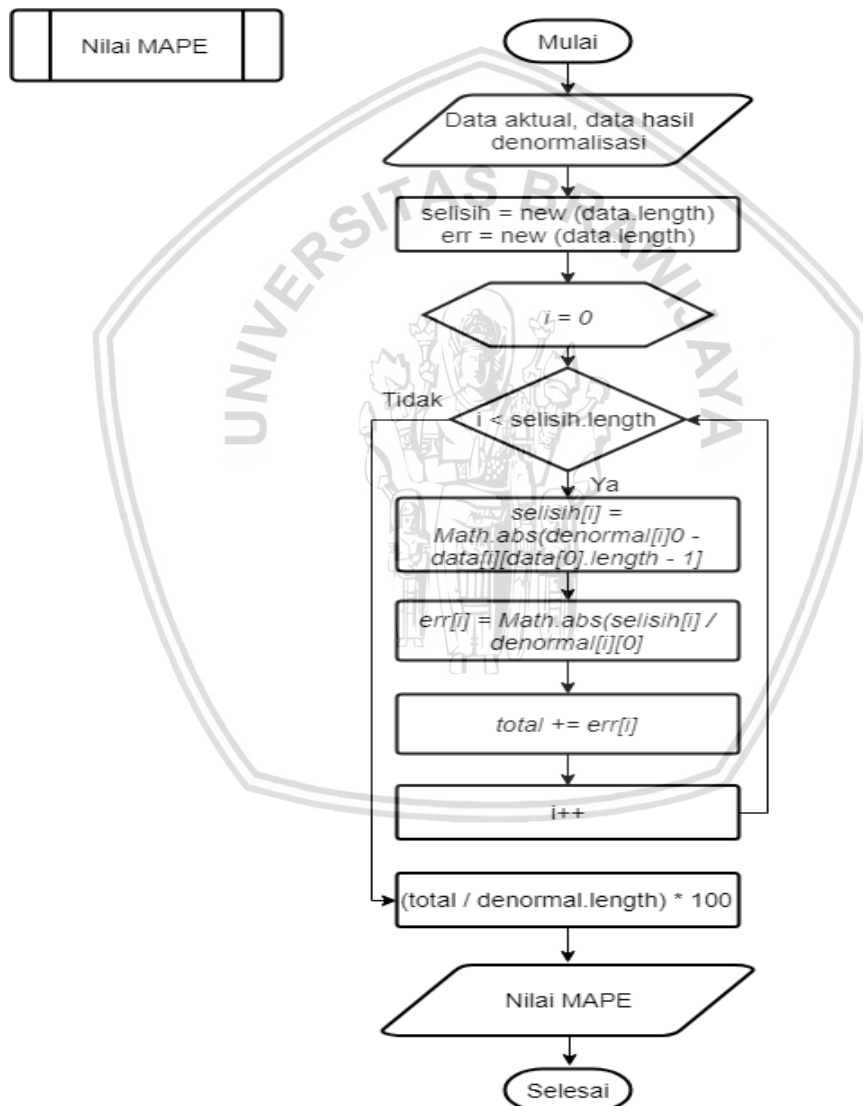




1. Masukan berupa data *output* peramalan ( $\hat{Y}$ ) serta nilai *max* dan *min* dari data.
2. Melakukan perhitungan denormalisasi dengan menggunakan Persamaan 2.11.
3. Hasil keluaran berupa hasil ramalan yang sudah di denormalisasi.

#### 4.3.4.3 Proses Perhitungan Mean Absolute Percentage Error

Mean Absolute Percentage Error (MAPE) merupakan suatu algoritme evaluasi yang digunakan untuk mengukur tingkat kesalahan dari hasil peramalan dengan menggunakan Persamaan 2.12. Diagram alir dari proses perhitungan MAPE ditunjukkan pada Gambar 4.18.



Gambar 4.18 Diagram Alir Mean Absolute Percentage Error

Berikut merupakan langkah-langkah proses evaluasi menggunakan MAPE berdasarkan Gambar 4.18.

1. Masukan berupa data aktual dan data hasil normalisasi.



2. Melakukan proses perhitungan nilai MAPE dengan menggunakan Persamaan 2.12.
3. Hasil keluaran berupa nilai MAPE yang menyatakan nilai *error* dari sistem peramalan.

#### 4.4 Perhitungan Manual

Perhitungan manual merupakan sebuah contoh perhitungan yang berdasarkan pada perancangan sistem. Perhitungan manual dilakukan untuk mengetahui kebenaran dari perhitungan sistem. Tahapan yang sama seperti pada perancangan sistem, yaitu dimulai dari proses normalisasi, proses *training*, serta proses *testing*.

Contoh proses perhitungan manual ini dilakukan dengan melibatkan data historis harga patokan batu bara dengan merek dagang Gunung Bayan I. Data historis tersebut kemudian disusun menjadi tiga parameter untuk melakukan prediksi harga bulan berikutnya. Parameter tersebut terbagi menjadi, harga satu bulan terakhir, harga dua bulan terakhir, dan harga tiga bulan terakhir.

Langkah-langkah penyelesaian prediksi harga patokan batu bara dengan menggunakan metode *Extreme Learning Machine* (ELM), terdiri dari:

##### 1. Normalisasi

Dalam proses prediksi menggunakan metode *Extreme Learning Machine* (ELM), data akan dibagi menjadi dua yaitu data *training* dan data *testing*. Untuk kasus ini, perbandingan data yang digunakan adalah sebesar 70%:30% dari 28 data, sehingga didapatkan data *training* sebanyak 20 data dan data *testing* sebanyak 8 data. Sebelum diproses, data-data tersebut akan dinormalisasi terlebih dahulu menggunakan Persamaan 2.10. Tujuan dari proses normalisasi data untuk menyamakan rentang data dengan fungsi aktivasi yang digunakan, yaitu fungsi aktivasi *sigmoid biner*. Bentuk penyajian data yang digunakan dalam proses prediksi dapat dilihat pada Tabel 4.2 dan Tabel 4.3.

**Tabel 4.2 Representasi Data Training**

Tanggal	Harga tiga bulan terakhir	Harga dua bulan terakhir	Harga satu bulan terakhir	Target
Juni 2015	72,60	69,04	65,36	63,75
Juli 2015	69,04	65,36	63,75	63,28
Agustus 2015	65,36	63,75	63,28	63,26
September 2015	63,75	63,28	63,26	62,25
Oktober 2015	63,28	63,26	62,25	61,36
November 2015	63,26	62,25	61,36	58,16
Desember 2015	62,25	61,36	58,16	57,16
Januari 2016	61,36	58,16	57,16	56,82
Februari 2016	58,16	57,16	56,82	54,36

Maret 2016	57,16	56,82	54,36	55,11
April 2016	56,82	54,36	55,11	55,87
Mei 2016	54,36	55,11	55,87	54,66
Juni 2016	55,11	55,87	54,66	55,32
Juli 2016	55,87	54,66	55,32	56,61
Agustus 2016	54,66	55,32	56,61	62,42
September 2016	55,32	56,61	62,42	68,45
Oktober 2016	56,61	62,42	68,45	74,01
November 2016	62,42	68,45	74,01	91,15
Desember 2016	68,45	74,01	91,15	109,35
Januari 2017	74,01	91,15	109,35	92,60

**Tabel 4.3 Representasi Data Testing**

Tanggal	Harga tiga bulan terakhir	Harga dua bulan terakhir	Harga satu bulan terakhir	Target
Februari 2017	91,15	109,35	92,6	89,45
Maret 2017	109,35	92,6	89,45	87,91
April 2017	92,6	89,45	87,91	88,57
Mei 2017	89,45	87,91	88,57	89,98
Juni 2017	87,91	88,57	89,98	80,94
Juli 2017	88,57	89,98	80,94	84,72
Agustus 2017	89,98	80,94	84,72	90,15
September 2017	80,94	84,72	90,15	98,88

Langkah-langkah dalam proses perhitungan normalisasi berdasarkan Persamaan 2.14 adalah sebagai berikut.

**Langkah 1:** Mencari nilai *maximum* dan nilai *minimum* seperti pada Tabel 4.4.

**Tabel 4.4 Nilai Maximum dan Minimum**

<b>MAX</b>	137,02
<b>MIN</b>	54,36

**Langkah 2:** Menghitung nilai normalisasi harga patokan batu bara dengan *Min-Max Normalization*. Berikut contoh perhitungan normalisasi serta hasil normalisasi data *training* dan data *testing* yang ditunjukkan pada Tabel 4.5 dan Tabel 4.6.

$$\begin{aligned}
 d'_{1,1} &= \frac{(d-d_{min})}{(d_{max}-d_{min})} \\
 &= \frac{72,60-54,36}{137,02-54,36} = 0,220663
 \end{aligned}$$





**Tabel 4.5 Normalisasi Data Training**

NO	X1	X2	X3	Y
1	0,220663	0,177595	0,133075	0,113598
2	0,177595	0,133075	0,113598	0,107912
3	0,133075	0,113598	0,107912	0,10767
4	0,113598	0,107912	0,10767	0,095451
5	0,107912	0,10767	0,095451	0,084684
6	0,10767	0,095451	0,084684	0,045971
7	0,095451	0,084684	0,045971	0,033874
8	0,084684	0,045971	0,033874	0,02976
9	0,045971	0,033874	0,02976	0
10	0,033874	0,02976	0	0,009073
11	0,02976	0	0,009073	0,018268
12	0	0,009073	0,018268	0,003629
13	0,009073	0,018268	0,003629	0,011614
14	0,018268	0,003629	0,011614	0,02722
15	0,003629	0,011614	0,02722	0,097508
16	0,011614	0,02722	0,097508	0,170457
17	0,02722	0,097508	0,170457	0,237721
18	0,097508	0,170457	0,237721	0,445076
19	0,170457	0,237721	0,445076	0,665255
20	0,237721	0,445076	0,665255	0,462618

**Tabel 4.6 Normalisasi Data Testing**

NO	X1	X2	X3	Y
1	0,445076	0,665255	0,462618	0,42451
2	0,665255	0,462618	0,42451	0,40588
3	0,462618	0,42451	0,40588	0,413864
4	0,42451	0,40588	0,413864	0,430922
5	0,40588	0,413864	0,430922	0,321558
6	0,413864	0,430922	0,321558	0,367288
7	0,430922	0,321558	0,367288	0,432978
8	0,321558	0,367288	0,432978	0,538592

2. Menentukan *input weight* ( $w$ ), bias ( $b$ ), dan jumlah *hidden neuron*

Jumlah *hidden neuron* yang digunakan dalam proses manualisasi adalah sebanyak 3. Sedangkan penentuan nilai *input weight* ( $w$ ) dan bias ( $b$ ) didapat secara acak yang dengan ukuran matriks [(jumlah *hidden neuron*) x (jumlah *input neuron*)] untuk matriks *input weight* ( $w$ ) dan ukuran [1 x (jumlah *hidden neuron*) untuk matriks bias ( $b$ ). Rentang yang digunakan untuk inisialisasi *input weight* ( $w$ ) secara acak adalah [-1, 1] dan untuk bias ( $b$ ) adalah [0, 1]. Hasil inisialisasi nilai *input weight* ( $w$ ) dan bias ( $b$ ) ditunjukkan pada Tabel 4.7 dan Tabel 4.8.



**Tabel 4.7 Matriks *Input Weight***

	1	2	3
1	0,333	-0,2834	-0,6496
2	-0,5867	0,3902	-0,593
3	-0,3327	0,9457	0,6068

**Tabel 4.8 Matriks Bias**

1	2	3
0,712	0,1202	0,1178

### 3. Proses *training*

Proses *training* dalam metode ELM bertujuan untuk mendapatkan nilai *output weight* ( $\hat{\beta}$ ) terbaik yang nantinya akan digunakan pada proses *testing*. Berikut merupakan langkah-langkah dalam proses *training*.

**Langkah 1:** Menghitung matriks keluaran *hidden layer* (*Hinit*) dengan menggunakan fungsi aktivasi. Fungsi aktivasi yang digunakan adalah fungsi aktivasi *sigmoid biner* dengan persamaan yang ditunjukkan pada Persamaan 2.2. Keluaran *hidden layer* (*Hinit*) didapatkan dengan melakukan perkalian antara data hasil normalisasi dengan *input weight* yang sudah di *transpose* ( $w^T$ ), kemudian ditambahkan dengan nilai bias (*b*). Bentuk persamaan yang digunakan berdasarkan pada Persamaan 2.5. Hasil keluaran *hidden layer* dengan fungsi aktivasi (*H*) akan memiliki ordo 20x3. Berikut merupakan contoh perhitungan manualnya.

**Langkah 1.1:** Menghitung keluaran *hidden layer*.

$$\begin{aligned}
 H_{init_{1,1}} &= (X_{1,1} * W^T) + b \\
 H_{init_{1,1}} &= (0,220663 * 0,333) + (0,177595 * -0,2834) \\
 &\quad + (0,133075 * -0,6496) + 0,712 \\
 &= 0,648705
 \end{aligned}$$

Berikut merupakan hasil perhitungan keluaran *hidden layer* (*Hinit*) untuk seluruh data yang ditunjukkan pada Tabel 4.9.

**Tabel 4.9 Matriks Keluaran *Hidden Layer* (*Hinit*) Proses *Training***

No	1	2	3
1	0,648705	-0,01888	0,293087
2	0,659632	0,000567	0,253495
3	0,654021	0,022459	0,246436
4	0,649303	0,031811	0,247392
5	0,655416	0,042298	0,241641
6	0,665792	0,044057	0,223633
7	0,689923	0,069981	0,194025
8	0,705167	0,068367	0,153655
9	0,698376	0,088798	0,152598



10	0,714846	0,111939	0,134675
11	0,716016	0,097359	0,113404
12	0,697562	0,112908	0,137465
13	0,707487	0,119853	0,134259
14	0,70951	0,104012	0,122202
15	0,692235	0,106461	0,144093
16	0,644812	0,066185	0,198846
17	0,582701	0,041196	0,304391
18	0,541739	-0,01146	0,39081
19	0,412271	-0,15098	0,555974
20	0,232877	-0,2401	0,863296

Langkah 1.2: Menghitung fungsi aktivasi ( $H$ )

$$H_{1,1} = \frac{1}{1 + \exp^{-H_{init_{1,1}}}}$$

$$H_{1,1} = \frac{1}{1 + \exp^{-0,648705}} = 0,656719$$

Berikut merupakan hasil perhitungan fungsi aktivasi yang ditunjukkan pada Tabel 4.10.

**Tabel 4.10 Matriks Hasil Fungsi Aktivasi**

No	1	2	3
1	0,656719	0,49528	0,572752
2	0,659178	0,500142	0,563036
3	0,657916	0,505614	0,561299
4	0,656853	0,507952	0,561535
5	0,65823	0,510573	0,560118
6	0,66056	0,511013	0,555676
7	0,66595	0,517488	0,548355
8	0,669332	0,517085	0,538338
9	0,667828	0,522185	0,538076
10	0,671471	0,527956	0,533618
11	0,671729	0,524321	0,528321
12	0,667647	0,528197	0,534312
13	0,669846	0,529927	0,533514
14	0,670293	0,525979	0,530513
15	0,666464	0,52659	0,535961
16	0,65584	0,51654	0,549548
17	0,641689	0,510298	0,575515
18	0,632217	0,497134	0,596478
19	0,601632	0,462327	0,63552
20	0,557957	0,440262	0,703349



**Langkah 2:** Menghitung keluaran matriks *Moore-Penrose pseudo inverse* ( $H^+$ ) dengan menggunakan Persamaan 2.11. Sebelum melakukan proses perhitungan *Moore-Penrose pseudo inverse* ( $H^+$ ), terdapat beberapa tahapan yang perlu dilakukan terlebih dahulu sebagai berikut.

**Langkah 2.1:** Mencari matriks *transpose* dari matriks  $H$ . Hasil *transpose* akan memiliki ordo  $3 \times 20$  yang ditunjukkan pada Tabel 4.11.

**Tabel 4.11 Transpose Matriks Keluaran Hidden Layer Proses Training**

$H^T$	1	2	...	19	20
1	0,656719	0,659178	...	0,601632	0,557957
2	0,49528	0,500142	---	0,462327	0,440262
3	0,572752	0,563036	...	0,63552	0,703349

**Langkah 2.2:** Mengalikan matriks *transpose hidden layer* ( $H^T$ ) dengan matriks  $H$ . Hasil dari perkalian tersebut akan menghasilkan sebuah matriks baru dengan ordo  $3 \times 3$ .

$$\begin{aligned}
 H^T H_{1,1} &= (0,656719 * 0,656719) + (0,659178 * 0,659178) \\
 &\quad + (0,657916 * 0,657916) + \dots + (0,557957 * 0,557957) \\
 &= 8,542073
 \end{aligned}$$

Berikut hasil perhitungan dari proses perkalian di atas yang ditunjukkan pada Tabel 4.12.

**Tabel 4.12 Perkalian Matriks**

$H^T * H$	1	2	3
1	8,542073	6,656481	7,32771
2	6,656481	5,188125	5,709858
3	7,32771	5,709858	6,368256

**Langkah 2.3:** Menghitung matriks *inverse* dari hasil perkalian  $H^T H$ . Proses perhitungan *inverse* menggunakan metode OBE (Operasi Baris Elementer). Bentuk *inverse* yang dihasilkan akan memiliki ordo  $3 \times 3$ . Berikut merupakan langkah-langkah dari proses perhitungan *inverse* menggunakan metode OBE beserta hasil perhitungan *inverse* yang ditunjukkan pada Tabel 4.13.

1. Buat matriks identitas ( $I$ ) disebelah kiri matriks yang akan dicari nilai inversnya.

$$[H|I] = \begin{bmatrix} 8,542073 & 6,656481 & 7,32771 \\ 6,656481 & 5,188125 & 5,709858 \\ 7,32771 & 5,709858 & 6,368256 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2.  $R_1 = R_1 / (8,542073)$

$$[H|I] = \begin{bmatrix} 1 & 0,779258 & 0,857837 \\ 6,656481 & 5,188125 & 5,709858 \\ 7,32771 & 5,709858 & 6,368256 \end{bmatrix} \begin{bmatrix} 0,117068 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3.  $R_2 = R_2 - (6,656481 * R_1)$



$$[H|I] = \begin{bmatrix} 1 & 0,779258 & 0,857837 \\ 0 & 0,001007 & -0,00032 \\ 7,32771 & 5,709858 & 6,368256 \end{bmatrix} \begin{bmatrix} 0,117068 & 0 & 0 \\ -0,77926 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

4.  $R_3 = R_3 - (7,32771 * R_1)$

$$[H|I] = \begin{bmatrix} 1 & 0,779258 & 0,857837 \\ 0 & 0,001007 & -0,00032 \\ 0 & -0,00032 & 0,082272 \end{bmatrix} \begin{bmatrix} 0,117068 & 0 & 0 \\ -0,77926 & 1 & 0 \\ -0,85784 & 0 & 1 \end{bmatrix}$$

5.  $R_2 = R_2 / (0,001007)$

$$[H|I] = \begin{bmatrix} 1 & 0,779258 & 0,857837 \\ 0 & 1 & -0,31898 \\ 0 & -0,00032 & 0,082272 \end{bmatrix} \begin{bmatrix} 0,117068 & 0 & 0 \\ -774,16 & 993,4578 & 0 \\ -0,85784 & 0 & 1 \end{bmatrix}$$

6.  $R_1 = R_1 - (0,779258 * R_2)$

$$[H|I] = \begin{bmatrix} 1 & 0 & 1,106402 \\ 0 & 1 & -0,31898 \\ 0 & -0,00032 & 0,082272 \end{bmatrix} \begin{bmatrix} 603,3878 & -774,16 & 0 \\ -774,16 & 993,4578 & 0 \\ -0,85784 & 0 & 1 \end{bmatrix}$$

7.  $R_3 = R_3 - (-0,00032 * R_2)$

$$[H|I] = \begin{bmatrix} 1 & 0 & 1,106402 \\ 0 & 1 & -0,31898 \\ 0 & 0 & 0,08217 \end{bmatrix} \begin{bmatrix} 603,3878 & -774,16 & 0 \\ -774,16 & 993,4578 & 0 \\ -1,1064 & 0,318976 & 1 \end{bmatrix}$$

8.  $R_3 = R_3 / (0,08217)$

$$[H|I] = \begin{bmatrix} 1 & 0 & 1,106402 \\ 0 & 1 & -0,31898 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 603,3878 & -774,16 & 0 \\ -774,16 & 993,4578 & 0 \\ -13,4649 & 3,881926 & 12,16996 \end{bmatrix}$$

9.  $R_2 = R_2 - (-0,31898 * R_3)$

$$[H|I] = \begin{bmatrix} 1 & 0 & 1,106402 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 603,3878 & -774,16 & 0 \\ -778,455 & 994,696 & 3,881926 \\ -13,4649 & 3,881926 & 12,16996 \end{bmatrix}$$

10.  $R_1 = R_1 - (1,106402 * R_3)$

$$[H|I] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 618,2854 & -778,455 & 13,4649 \\ -778,455 & 994,696 & 3,881926 \\ -13,4649 & 3,881926 & 12,16996 \end{bmatrix}$$

Tabel 4.13 Matriks Invers

$(H^T * H)^{-1}$	1	2	3
1	618,2854	-778,455	-13,4649
2	-778,455	994,696	3,881926
3	-13,4649	3,881926	12,16996

**Langkah 2.4:** Menghitung nilai matriks *Moore-Penrose pseudo inverse* ( $H^+$ ) dengan mengalikan matriks *inverse* dengan matriks *transpose hidden layer* ( $H^T$ ). Bentuk matriks yang akan dihasilkan adalah matriks dengan ordo 3x20. Berikut merupakan langkah-langkah proses perhitungan serta tabel hasil



perhitungan *Moore-Penrose pseudo inverse* ( $H^+$ ) yang ditunjukkan pada Tabel 4.14.

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

$$H^+_{1,1} = (618,2854 * 0,656719) + (-778,455 * 0,49528) + (-13,4649 * 0,572752) = 12,77386$$

**Tabel 4.14 Matriks Moore-Penrose Pseudo Inverse**

$H^+$	1	2	...	19	20
1	12,77386	10,64078	...	3,522512	-7,21786
2	-16,3491	-13,4656	...	-6,00204	6,312418
3	0,050378	-0,0821	...	1,428077	2,755966

**Langkah 3:** Menghitung nilai matriks *output weight* ( $\hat{\beta}$ ). Nilai *output weight* ( $\hat{\beta}$ ) didapatkan dengan mengalikan nilai matriks *Moore-Penrose pseudo inverse* ( $H^+$ ) dengan matriks target ( $T$ ). Bentuk matriks yang dihasilkan berupa matriks *output weight* ( $\hat{\beta}$ ) dengan ordo 3x1. Berikut merupakan langkah-langkah proses perhitungan serta tabel hasil perhitungan *output weight* yang ditunjukkan pada Tabel 4.15.

$$\beta = H^+ \cdot T$$

$$\beta_{1,1} = (12,77386 * 0,113598) + (10,64078 * 0,107912) + \dots + (-7,21789 * 0,462618) = -2.15725$$

**Tabel 4.15 Matriks Output Weight**

$\beta$	1
1	-2,15725
2	0,320308
3	2,459021

4. Proses *Testing*

Proses *testing* merupakan proses evaluasi dari metode ELM dengan menggunakan beberapa parameter yang didapatkan dari proses *training*, yaitu nilai *input weight* ( $w$ ), bias ( $b$ ), serta *output weight* ( $\hat{\beta}$ ). Data *testing* ditunjukkan pada Tabel 4.14. Berikut merupakan langkah-langkah dari proses *testing*.

**Langkah 1:** Menghitung keluaran *hidden layer* dengan fungsi aktivasi ( $H$ ). Proses pencarian matriks tersebut sama dengan yang dilakukan pada proses *training*. Bentuk matriks yang akan dihasilkan dari proses ini adalah matriks dengan ordo 8x3. Berikut merupakan contoh perhitungan manualnya.

**Langkah 1.1:** Menghitung matriks keluaran *hidden layer* ( $H_{init}$ ).



$$H_{init_{1,1}} = (X_{1,1} * W^T) + b$$

$$\begin{aligned}
 H_{init_{1,1}} &= (0,445076 * 0,333) + (0,665255 * -0,2834) \\
 &\quad + (0,462618 * -0,6496) + 0,712 \\
 &= 0,37116
 \end{aligned}$$

Berikut merupakan hasil perhitungan keluaran *hidden layer* (*Hinit*) untuk seluruh data yang ditunjukkan pada Tabel 4.16.

**Tabel 4.16 Matriks Keluaran *Hidden Layer* (*Hinit*) Proses Testing**

No	1	2	3
1	0,37116	-0,15568	0,879572
2	0,526662	-0,34133	0,59156
3	0,482086	-0,22626	0,611634
4	0,46949	-0,21591	0,611538
5	0,449942	-0,21198	0,635638
6	0,518809	-0,14515	0,582752
7	0,525777	-0,22495	0,5014
8	0,433727	-0,1819	0,620893

**Langkah 1.2:** Menghitung fungsi aktivasi (*H*)

$$H_{1,1} = \frac{1}{1 + \exp^{-H_{init_{1,1}}}}$$

$$H_{1,1} = \frac{1}{1 + \exp^{-0,37116}} = 0,591739$$

Berikut merupakan hasil perhitungan fungsi aktivasi yang ditunjukkan pada Tabel 4.17.

**Tabel 4.17 Matriks Hasil Fungsi Aktivasi**

No	1	2	3
1	0,591739	0,461159	0,706733
2	0,628704	0,415487	0,643723
3	0,61824	0,443675	0,648313
4	0,615263	0,446232	0,648292
5	0,610625	0,447203	0,653767
6	0,626869	0,463775	0,6417
7	0,628498	0,443998	0,622788
8	0,606763	0,45465	0,650422

**Langkah 2:** Menghitung nilai hasil peramalan ( $\hat{Y}$ ). Untuk mendapatkan hasil peramalan, maka dilakukan perkalian antara hasil keluaran *hidden layer* dengan fungsi aktivasi (*H*) dengan nilai *output weight* ( $\hat{\beta}$ ) yang didapatkan dari proses *training*. Bentuk matriks yang akan dihasilkan adalah matriks dengan ordo 8x1. Berikut merupakan langkah-langkah proses perhitungan serta tabel hasil perhitungan peramalan yang ditunjukkan pada Tabel 4.18.



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

$$\begin{aligned} \hat{Y}_{1,1} &= (0,591739 * -2,15725) + (0,461159 * 0,320308) + (0,706733 \\ &\quad * 2,459021 \\ &= 0.609056 \end{aligned}$$

**Tabel 4.18 Matriks Nilai Hasil Peramalan**

No	$\hat{Y}$
1	0,609056
2	0,35974
3	0,40263
4	0,409819
5	0,433598
6	0,374192
7	0,317839
8	0,436089

**Langkah 3:** Melakukan denormalisasi data. Denormalisasi dilakukan untuk mengembalikan nilai hasil ramalan menjadi nilai asli. Berikut merupakan langkah-langkah proses denormalisasi berdasarkan Persamaan 2.11, serta hasil denormalisasi yang ditunjukkan pada Tabel 4.19.

$$d = d'(max - min) + min$$

$$\begin{aligned} d_{1,1} &= 0,609056(137,02 - 54,36) + 54,36 \\ &= 104,7046 \end{aligned}$$

**Tabel 4.19 Matriks Denormalisasi Data Testing**

No	Y
1	104,7046
2	84,09613
3	87,64141
4	88,23562
5	90,2012
6	85,29072
7	80,63259
8	90,4071

**Langkah 4:** Evaluasi nilai tingkat kesalahan (*error*) dengan menggunakan *Mean Absolute Percentage Error* (MAPE). Berikut contoh perhitungan nilai MAPE dari hasil *testing* serta tabel hasil perhitungan yang ditunjukkan pada Tabel 4.20.

$$MAPE = \sum_{i=1}^n \left| \frac{y_i - t_i}{y_i} \right| \times 100$$

$$\begin{aligned} MAPE &= \frac{1}{8} \left( \left( abs \left( \frac{104,7046 - 89,45}{104,7046} \right) \right) + \dots + \left( abs \left( \frac{90,4071 - 98,88}{90,4071} \right) \right) \right) \times 100 \\ &= 6.78157 \% \end{aligned}$$





**Tabel 4.20 Nilai MAPE Proses Testing**

	Y target	y prediksi
1	89,45	104,7046
2	87,91	84,09613
3	88,57	87,64141
4	89,98	88,23562
5	80,94	90,2012
6	84,72	85,29072
7	90,15	80,63259
8	98,88	90,4071
MAPE (%)		6,78157

#### 4.5 Perancangan User Interface

User interface yang digunakan dalam penelitian ini terdiri dari 3 tampilan, yang terdiri dari tampilan untuk menampilkan keseluruhan data, tampilan hasil pembentukan fitur, dan tampilan hasil normalisasi data.

##### 4.5.1 Perancangan Halaman Awal

Gambar 4.19 menunjukkan rancangan tampilan awal sistem yang berisi informasi seluruh data harga diikuti dengan beberapa *textfield* yang harus di masukan oleh pengguna untuk melakukan proses prediksi. Proses prediksi dimulai dengan pengguna memasukkan jumlah fitur yang akan dibentuk, jumlah *k-fold*, jumlah *neuron*, serta fungsi aktivasi yang akan digunakan. Setelah semua masukan tersebut terisi, pilih tombol “prediksi” untuk memulai proses prediksi.

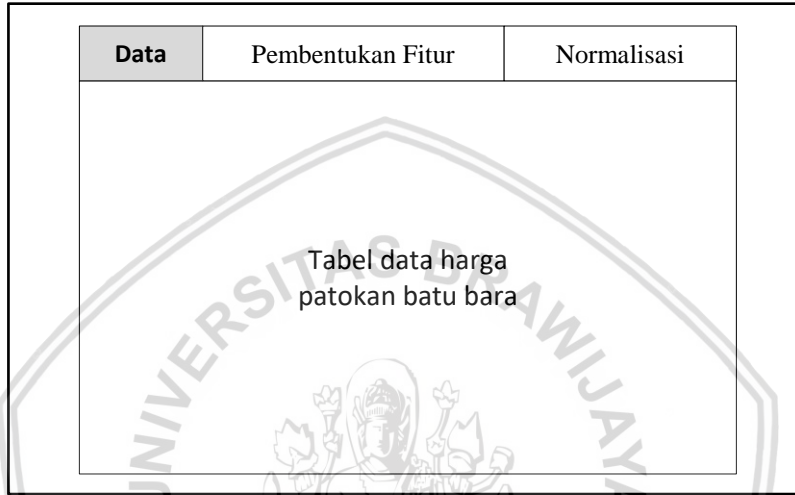
**PREDIKSI HARGA PATOKAN BATU BARA  
UNTUK MEREK DAGANG GUNUNG BAYAN I  
DENGAN METODE EXTREME LEARNING MACHINE (ELM)**

<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">                 Jumlah Fitur <input style="width: 60%;" type="text"/> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">                 Jumlah <i>K-fold</i> <input style="width: 60%;" type="text"/> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">                 Jumlah <i>Neuron</i> <input style="width: 60%;" type="text"/> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">                 Fungsi Aktivasi <input style="width: 60%;" type="text"/> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px; text-align: center;">                 PREDIKSI             </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">                 Nilai MAPE untuk setiap <i>k-fold</i> </div>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">Data</th> <th style="width: 40%;">Pembentukan Fitur</th> <th style="width: 40%;">Normalisasi</th> </tr> </thead> <tbody> <tr> <td colspan="3" style="text-align: center; height: 150px; vertical-align: middle;">                     Tabel data harga patokan batu bara                 </td> </tr> </tbody> </table> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <div style="text-align: center;">                 Rata-Rata MAPE Latih  <input style="width: 80%;" type="text"/> </div> <div style="text-align: center;">                 Rata-Rata MAPE Uji  <input style="width: 80%;" type="text"/> </div> </div>	Data	Pembentukan Fitur	Normalisasi	Tabel data harga patokan batu bara		
Data	Pembentukan Fitur	Normalisasi					
Tabel data harga patokan batu bara							

**Gambar 4.19 Perancangan Halaman Data**

**4.5.2 Perancangan Tabel Data, Pembentukan Fitur, dan Normalisasi**

Terdapat tiga tabel yang tersusun dalam sebuah *tabbed panel* pada halaman sistem. Ketiga tabel tersebut adalah tabel data, tabel pembentukan fitur, dan tabel normalisasi. Tabel data berisi seluruh data historis harga patokan batu bara setiap bulan. Tabel pembentukan fitur berisi data historis harga patokan batu bara yang sudah telah disusun menjadi beberapa fitur sesuai dengan masukan pengguna. Tabel normalisasi berisi hasil normalisasi dari data hasil pembentukan fitur. Bentuk perancangan tabel berdasarkan penjelasan diatas ditunjukkan pada Gambar 4.20.



**Gambar 4.20 Perancangan Tabel Data, Pembentukan Fitur, dan Normalisasi**

**4.5.3 Perancangan Halaman Masukan Pengguna**

Untuk melakukan proses prediksi, pengguna harus memasukkan beberapa parameter yang ditunjukkan pada Gambar 4.21. Masukan tersebut terdiri dari jumlah fitur, jumlah *k-fold*, jumlah *neuron*, serta memilih jenis fungsi aktivasi yang akan digunakan. Terdapat tiga jenis fungsi aktivasi yang dapat dipilih oleh pengguna. Fungsi aktivasi tersebut adalah fungsi *sigmoid biner*, fungsi *sigmoid bipolar*, dan fungsi *sin*.

Jumlah Fitur	<input style="width: 40%;" type="text"/>
Jumlah <i>K-fold</i>	<input style="width: 40%;" type="text"/>

Jumlah <i>Neuron</i>	<input style="width: 40%;" type="text"/>
Fungsi Aktivasi	<input style="width: 40%;" type="text"/>

**Gambar 4.21 Perancangan Halaman Masukan Pengguna**

## 4.6 Perancangan Uji Coba dan Evaluasi

Proses pengujian yang dilakukan dalam penelitian ini bertujuan untuk mengetahui hasil dari kinerja ELM dalam memecahkan suatu permasalahan prediksi. Skenario pengujian yang akan dilakukan terdiri dari:

1. Pengujian jumlah *neuron*
2. Pengujian jumlah fitur data
3. Pengujian fungsi aktivasi

### 4.6.1 Pengujian Jumlah *Neuron*

Pengujian jumlah *neuron* pada *hidden layer* digunakan untuk mengetahui jumlah *hidden layer* terbaik untuk digunakan dalam proses ELM. Jumlah *neuron* terbaik didapatkan dengan melihat hasil nilai MAPE. Jumlah *neuron* dengan MAPE minimal merupakan jumlah *neuron* terbaik. Banyaknya jumlah *neuron* yang diuji dimulai dari 3 hingga 10. Tabel rancangan pengujian ditunjukkan pada Tabel 4.21.

**Tabel 4.21 Pengujian Jumlah *Neuron***

Jumlah <i>Neuron</i>		Nilai MAPE Percobaan Ke-i										Rata-Rata MAPE (%)	
		1	2	3	4	5	6	7	8	9	10		
3	Latih												
	Uji												
4	Latih												
	Uji												
5	Latih												
	Uji												
6	Latih												
	Uji												
7	Latih												
	Uji												
8	Latih												
	Uji												
9	Latih												
	Uji												
10	Latih												
	Uji												

### 4.6.2 Pengujian Jumlah Fitur Data

Pengujian jumlah fitur data dilakukan untuk mengetahui variasi fitur data yang tepat sebagai masukkan dalam proses prediksi. Jumlah fitur dengan nilai MAPE minimal merupakan jumlah fitur terbaik. Proses pengujian dilakukan sebanyak 10 kali percobaan untuk masing-masing uji coba jumlah fitur. Tabel rancangan pengujian ditunjukkan pada Tabel 4.22.



**Tabel 4.22 Pengujian Jumlah Fitur Data**

Jumlah Fitur Data		Nilai MAPE Percobaan Ke-i										Rata-Rata MAPE (%)
		1	2	3	4	5	6	7	8	9	10	
1	Latih											
	Uji											
2	Latih											
	Uji											
3	Latih											
	Uji											
4	Latih											
	Uji											
5	Latih											
	Uji											
6	Latih											
	Uji											
7	Latih											
	Uji											
8	Latih											
	Uji											
9	Latih											
	Uji											
10	Latih											
	Uji											

**4.6.3 Pengujian Fungsi Aktivasi**

Pengujian fungsi aktivasi dilakukan untuk mengetahui fungsi aktivasi terbaik berdasarkan nilai MAPE yang dihasilkan. Fungsi aktivasi yang akan diuji terdiri dari fungsi aktivasi *sigmoid biner*, *sigmoid bipolar*, dan fungsi *sin*. Proses pengujian dilakukan sebanyak 10 kali percobaan untuk masing-masing fungsi aktivasi. Tabel rancangan pengujian ditunjukkan pada Tabel 4.23.

**Tabel 4.23 Pengujian Fungsi Aktivasi**

Fungsi Aktivasi		Nilai MAPE Percobaan Ke-i										Rata-Rata MAPE (%)
		1	2	3	4	5	6	7	8	9	10	
Sigmoid Biner	Latih											
	Uji											
Sigmoid Bipolar	Latih											
	Uji											
Sin	Latih											
	Uji											



## 4.7 Implementasi Sistem

Proses implementasi sistem dilakukan berdasarkan hasil perancangan yang dilakukan sebelumnya. Proses implementasi dilakukan menggunakan Bahasa pemrograman Java dengan Netbeans IDE 7.2.

### 4.7.1 Implementasi Normalisasi Data

Normalisasi data merupakan sebuah proses untuk memperkecil ukuran data yang mewakili data asli tanpa menghilangkan karakteristik data itu sendiri. Metode yang digunakan untuk melakukan normalisasi adalah normalisasi *min max*. Proses normalisasi pada Kode Program 4.3 dilakukan dengan bantuan parameter nilai maksimal dan minimal dari keseluruhan data yang didapatkan dengan bantuan fungsi *getmax()* dan fungsi *getmin()* yang ditunjukkan pada Kode Program 4.1 dan Kode Program 4.2.

No	Kode Program
1	<code>public double getmax(double[][] data){</code>
2	<code>    double max = 0;</code>
3	<code>    for(int i = 0; i &lt; data.length; i++){</code>
4	<code>        if(data[i][0] &gt; max){</code>
5	<code>            max = data[i][0];</code>
6	<code>        }</code>
7	<code>    }</code>
8	<code>    return max;</code>
9	<code>}</code>

**Kode Program 4.1 Proses *getmax()***

Penjelasan kode program fungsi *getmax()* data berdasarkan Kode Program 4.1 adalah sebagai berikut.

- Baris 2 : Inisialisasi nilai maksimal berupa 0
- Baris 3 s.d. 7 : Proses perulangan untuk mendapatkan nilai maksimal dari data
- Baris 8 : Mengembalikan nilai maksimal dari data

No	Kode Program
1	<code>public double getmin(double[][] data){</code>
2	<code>    double min = data[0][0];</code>
3	<code>    for(int i = 0; i &lt; data.length; i++){</code>
4	<code>        if(data[i][0] &lt; min){</code>
5	<code>            min = data[i][0];</code>
6	<code>        }</code>
7	<code>    }</code>
8	<code>    return min;</code>
9	<code>}</code>

**Kode Program 4.2 Proses *getmin()***

Penjelasan kode program fungsi *getmin()* data berdasarkan Kode Program 4.2 adalah sebagai berikut.

- Baris 2 : Inisialisasi nilai minimal
- Baris 3 s.d. 7 : Proses perulangan untuk mendapatkan nilai minimal dari

Baris 8 : Mengembalikan nilai minimal dari data

No	Kode Program
1	public double[][] normalisasi(double[][] data){
2	double normalisasi[][] = new
3	double[data.length][data[0].length];
4	for(int i = 0; i < data[0].length; i++){
5	for(int j = 0; j < data.length; j++){
6	normalisasi[j][i] = (data[j][i] -
7	getmin(data)) / (getmax(data) - getmin(data));
8	}
9	}

#### Kode Program 4.3 Proses Normalisasi Data

Penjelasan kode program fungsi *normalisasi* data berdasarkan Kode Program 4.3 adalah sebagai berikut.

- Baris 1 : Fungsi *normalisasi* menerima masukan parameter berupa matriks data
- Baris 2 : Proses pembuatan matriks baru dengan nama matriks *normalisasi*
- Baris 3 s.d. 7 : Proses perulangan untuk melakukan proses normalisasi data dengan Persamaan 2.10
- Baris 8 : Mengembalikan nilai matriks *normalisasi*

#### 4.7.2 Implementasi Inisialisasi *Input Weight*

Inisialisasi *input weight* dilakukan dengan pembentukan matriks dengan *ordo hidden neuron x input layer*. Nilai pada matriks *input weight* dilakukan secara acak dengan rentang -1 sampai 1. Implementasi proses inisialisasi *input weight* ditunjukkan pada Tabel 4.4.

No	Kode Program
1	public double[][] randbobot(int fitur, int hidden)
2	{
3	double bobot_rand[][] = new
4	double[hidden][fitur];
5	for (int i = 0; i < hidden; i++) {
6	for (int j = 0; j < fitur; j++) {
7	bobot_rand[i][j] = (double)
8	Math.random() * (1 - (-1)) + (-1);
9	}
10	}
11	return bobot_rand;
12	}

#### Kode Program 4.4 Proses Inisialisasi *Input Weight*



Penjelasan kode program fungsi inialisasi *input weight* berdasarkan Kode Program 4.4 adalah sebagai berikut.

- Baris 1 : Fungsi *randbobot* menerima masukan parameter berupa jumlah fitur dan jumlah *neuron hidden*
- Baris 2 : Proses pembuatan matriks baru dengan nama matriks *bobot\_rand*
- Baris 3 s.d. 7 : Proses perulangan untuk mengisi matriks *bobot\_rand* dengan nilai acak dari rentang -1 samapai 1
- Baris 8 : Mengembalikan nilai matriks *bobot\_rand*

### 4.7.3 Implementasi Inialisasi Bias

Inialisasi bias dilakukan dengan pembentukan matriks dengan *ordo* 1 x *hidden neuron*. Nilai pada matriks bias dilakukan secara acak dengan rentang 0 sampai 1. Implementasi proses inialisasi bias ditunjukkan pada Kode Program 4.5.

No	Kode Program
1	<code>public double[] randbias(int hidden) {</code>
2	<code>double bias_rand[] = new double[hidden];</code>
3	<code>for (int i = 0; i &lt; hidden; i++) {</code>
4	<code>    bias_rand[i] = (double) Math.random();</code>
5	<code>    }</code>
6	<code>return bias_rand;</code>
7	<code>}</code>

**Kode Program 4.5 Proses Inialisasi Bias**

Penjelasan kode program fungsi inialisasi bias berdasarkan Kode Program 4.5 adalah sebagai berikut.

- Baris 1 : Fungsi *randbias* menerima masukan parameter berupa jumlah *neuron hidden*
- Baris 2 : Proses pembuatan matriks baru dengan nama matriks *bias\_rand*
- Baris 3 s.d. 5 : Proses perulangan untuk mengisi matriks *bias\_rand* dengan nilai acak dari rentang 0 samapai 1
- Baris 6 : Mengembalikan nilai matriks *bias\_rand*

### 4.7.4 Implementasi Transpose Input Weight

Proses *transpose* dilakukan untuk mengubah ukuran suatu matriks. Proses yang dilakukan adalah mengubah posisi baris dan kolom pada matriks awal menjadi kolom dan baris pada matriks baru. Implementasi proses *transpose input weight* ditunjukkan pada Kode Program 4.6.

No	Kode Program
1	<code>public double[][] transbobot(double[][] bobot_rand)</code>
2	<code>{</code> <code>    double[][] transbobot = new</code>



```

3   double[bobot_rand[0].length][bobot_rand.length];
4       for (int i = 0; i < bobot_rand[0].length; i++)
5       {
6           for (int j = 0; j < bobot_rand.length; j++)
7           {
8               transbobot[i][j] = bobot_rand[j][i];
9           }
10      }
11      return transbobot;
12  }

```

**Kode Program 4.6 Proses Transpose Input Weight**

Penjelasan kode program fungsi inialisasi *transpose input weight* berdasarkan Kode Program 4.6 adalah sebagai berikut.

- Baris 1 : Fungsi *transbobot* menerima masukan parameter berupa hasil acak bobot
- Baris 2 : Proses pembuatan matriks baru dengan nama matriks *transbobot*
- Baris 3 s.d. 7 : Proses perulangan untuk melakukan *transpose* matriks dengan mengubah posisi baris dan kolom matriks masukan menjadi kolom dan baris baru hasil *transpose*
- Baris 8 : Mengembalikan nilai matriks *transbobot*

### 4.7.5 Implementasi Hitung Keluaran *Hidden Layer (Hinit)*

Implementasi proses perhitungan keluaran *hidden layer (Hinit)* ditunjukkan pada Kode Program 4.7.

No	Kode Program
1	public double[][] Hinit(double[][] data, double[][] transbobot, double[] bias) {
2	double[][] hinit = new double[data.length][transbobot[0].length];
3	for (int i = 0; i < data.length; i++) {
4	for (int j = 0; j < transbobot[0].length; j++) {
5	double init = 0;
6	for (int k = 0; k < transbobot.length; k++) {
7	init += data[i][k] * transbobot[k][j];
8	}
9	hinit[i][j] = init;
10	}
11	}
12	for (int i = 0; i < hinit.length; i++) {
13	for (int j = 0; j < hinit[0].length; j++) {
14	hinit[i][j] += bias[j];
15	}
16	}
17	return hinit;



18	}
----	---

**Kode Program 4.7 Proses Hitung Keluaran *Hidden Layer***

Penjelasan kode program fungsi hitung keluaran *hidden layer* berdasarkan Kode Program 4.7 adalah sebagai berikut.

- Baris 1 : Fungsi *hinit* menerima masukan parameter berupa data, *transbobot*, dan bias
- Baris 2 : Proses pembuatan matriks baru dengan nama matriks *hinit*
- Baris 3 s.d. 11 : Proses perulangan untuk menjumlahkan hasil dari perkalian input data dengan masing-masing *input weight* yang disimpan dalam variabel *init*. Kemudian, hasil nilai dari variabel *init* dimasukkan ke dalam matriks *hinit*
- Baris 12 s.d. 16 : Proses perulangan untuk menjumlahkan nilai dalam matriks *hinit* dengan nilai bias
- Baris 17 : Mengembalikan nilai matriks *hinit*

**4.7.6 Implementasi Hitung Aktivasi Keluaran *Hidden Layer* (H)**

Proses aktivasi keluaran *hidden layer* dilakukan dengan menggunakan fungsi aktivasi *sigmoid biner*. Implementasi proses aktivasi ditunjukkan pada Kode Program 4.8.

No	Kode Program
1	<code>public double[][] aktivasiHinit(double[][] hinit,</code>
2	<code>String aktif) {</code>
3	<code>double[][] aktivasi = new</code>
4	<code>double[hinit.length][hinit[0].length];</code>
5	<code>if (aktif == "Sigmoid Biner") {</code>
6	<code>for (int i = 0; i &lt; hinit.length; i++) {</code>
7	<code>for (int j = 0; j &lt; hinit[0].length;</code>
8	<code>j++) {</code>
9	<code>aktivasi[i][j] = 1 / (1 +</code>
10	<code>Math.exp(-hinit[i][j]));</code>
11	<code>}</code>
12	<code>}</code>
13	<code>return aktivasi;</code>
14	<code>}</code>

**Kode Program 4.8 Proses Aktivasi Keluaran *Hidden Layer***

Penjelasan kode program fungsi hitung aktivasi keluaran *hidden layer* berdasarkan Kode Program 4.8 adalah sebagai berikut.

- Baris 1 : Fungsi *aktivasiHinit* menerima masukan parameter berupa matriks *hinit* dan fungsi aktivasi yang digunakan
- Baris 2 : Proses pembuatan matriks baru dengan nama matriks *aktivasi*
- Baris 3 : Proses pengecekan fungsi aktivasi yang digunakan
- Baris 4 s.d. 8 : Proses perulangan untuk melakukan proses perhitungan



#### 4.7.7 Implementasi Hitung *Inverse* Matriks

Proses perhitungan nilai *inverse* dilakukan dengan menggunakan metode Operasi Baris Elementer (OBE). Implementasi proses perhitungan *inverse* ditunjukkan pada Kode Program 4.9.

No	Kode Program
1	<code>public double[][] inverse(double[][] htkalih) {</code>
2	<code>    double[][] inverse = new</code>
	<code>double[htkalih.length][htkalih[0].length];</code>
3	<code>    int p = htkalih.length;</code>
4	<code>    int pp = p * 2;</code>
5	<code>    double[][] matriks = new double[p][pp];</code>
6	
7	<code>    //buat matriks identitas</code>
8	<code>    for (int i = 0; i &lt; p; i++) {</code>
9	<code>        for (int j = p; j &lt; pp; j++) {</code>
10	<code>            if (i + p == j) {</code>
11	<code>                matriks[i][j] = 1;</code>
12	<code>            } else {</code>
13	<code>                matriks[i][j] = 0;</code>
14	<code>            }</code>
15	<code>            matriks[i][j - p] = htkalih[i][j - p];</code>
16	<code>        }</code>
17	<code>    }</code>
18	
19	<code>    // proses OBE</code>
20	<code>    for (int j = 0; j &lt; p - 1; j++) {</code>
21	<code>        for (int i = j + 1; i &lt; p; i++) {</code>
22	<code>            if (matriks[i][j] != 0) {</code>
23	<code>                double s = matriks[j][j] /</code>
	<code>matriks[i][j];</code>
24	<code>                for (int k = j; k &lt; pp; k++) {</code>
25	<code>                    matriks[i][k] *= s;</code>
26	<code>                    matriks[i][k] -= matriks[j][k];</code>
27	<code>                }</code>
28	<code>            }</code>
29	<code>        }</code>
30	<code>    }</code>
31	
32	<code>    for (int j = p - 1; j &gt; 0; j--) {</code>
33	<code>        for (int i = j - 1; i &gt;= 0; i--) {</code>
34	<code>            if (matriks[i][j] != 0) {</code>
35	<code>                double s = matriks[j][j] /</code>
	<code>matriks[i][j];</code>
36	<code>                for (int k = i; k &lt; pp; k++) {</code>
37	<code>                    matriks[i][k] *= s;</code>
38	<code>                    matriks[i][k] -= matriks[j][k];</code>
39	<code>                }</code>
40	<code>            }</code>

```

41     }
42     }
43
44     for (int j = 0; j < p; j++) {
45         if (matriks[j][j] != 1) {
46             double s = 1 / matriks[j][j];
47             for (int k = j; k < pp; k++) {
48                 matriks[j][k] *= s;
49             }
50         }
51     }
52
53     for (int i = 0; i < p; i++) {
54         for (int j = p; j < pp; j++) {
55             inverse[i][j - p] = matriks[i][j];
56         }
57     }
58
59     return inverse;
60 }

```

**Kode Program 4.9 Proses Hitung Inverse Matriks**

Penjelasan kode program fungsi hitung *nilai inverse* berdasarkan Kode Program 4.9 adalah sebagai berikut.

- Baris 1 : Fungsi *inverse* menerima masukan parameter berupa matriks *htkalih*. *htkalih* merupakan matriks hasil perkalian matriks *transpose H* dengan matriks *H*
- Baris 2 : Proses pembuatan matriks baru dengan nama matriks *inverse* untuk menampung hasil *inverse*
- Baris 3 : Inisialisasi nilai *p* dengan panjang matriks *htkalih*.
- Baris 4 : Inisialisasi nilai *pp* dengan 2 kali nilai *p*
- Baris 5 : Proses pembuatan matriks baru dengan nama matriks
- Baris 8 s.d. 17 : Proses perulangan untuk pembuatan matriks identitas dan penggabungan dengan matriks *htkalih*
- Baris 20 s.d. 42 : Proses perhitungan OBE untuk mengubah nilai matriks *htkalih* menjadi 0
- Baris 44 s.d. 51 : Proses perhitungan OBE untuk mengubah nilai matriks *htkalih* menjadi 1
- Baris 53 s.d. 57 : Proses mengambil nilai matriks identitas yang sudah diubah menjadi matriks *inverse*
- Baris 59 : Mengembalikan nilai matriks *inverse*

**4.7.8 Implementasi Hitung Output Weight**

Implementasi proses hitung *output weight* ditunjukkan pada Kode Program 4.10.

No	Kode Program
1	public double[][] bobotOutput(double[][] moorePenrose, double[][] data) {

```

2      double[][]      bobotOut      =      new
double[moorePenrose.length][1];
3      double[][] target = new double[data.length][1];
4      for (int i = 0; i < data.length; i++) {
5          target[i][0] = data[i][data[0].length - 1];
6      }
7      for (int i = 0; i < moorePenrose.length; i++) {
8          for (int j = 0; j < target[0].length; j++)
9          {
10             double total = 0;
11             for (int k = 0; k <
moorePenrose[0].length; k++) {
12                 total += moorePenrose[i][k] *
target[k][j];
13             }
14             bobotOut[i][j] = total;
15         }
16     }
17     return bobotOut;
}

```

**Kode Program 4.10 Proses Hitung Output Weight**

Penjelasan kode program fungsi *bobotOutput* berdasarkan Kode Program 4.10 adalah sebagai berikut.

- Baris 1 : Fungsi *bobotOutput* menerima masukan parameter berupa matriks *moorePenrose* dan matriks *data*
- Baris 2 : Proses pembuatan matriks baru dengan nama matriks *bobotOut*
- Baris 3 : Proses pembuatan matriks baru dengan nama matriks *target*
- Baris 4 s.d. 6 : Proses perulangan untuk mengambil nilai target dari matriks *data*
- Baris 7 s.d. 15 : Proses perulangan untuk mengalikan nilai pada matriks *moorePenrose* dengan matriks *target*
- Baris 16 : Mengembalikan nilai matriks *bobotOut*

#### 4.7.9 Implementasi Hitung Output Peramalan

*Output* peramalan merupakan hasil peramalan yang didapatkan dengan menggunakan metode ELM. Implementasi proses hitung *output* peramalan ditunjukkan pada Kode Program 4.11.

No	Kode Program
1	public double[][] yPrediksi(double[][] aktivasi, double[][] bobotOutput) {
2	double[][] yPrediksi = new double[aktivasi.length][bobotOutput[0].length];
3	for (int i = 0; i < aktivasi.length; i++) {
4	for (int j = 0; j < bobotOutput[0].length; j++) {
5	double total = 0;



6	for (int k = 0; k < aktivasi[0].length;
7	k++) {
8	total += aktivasi[i][k] *
9	bobotOutput[k][j];
10	}
11	yPrediksi[i][j] = total;
12	}
13	return yPrediksi;
	}

**Kode Program 4.11 Proses Hitung Output Peramalan**

Penjelasan kode program fungsi *yPrediksi* berdasarkan Kode Program 4.11 adalah sebagai berikut.

- Baris 1 : Fungsi *yPrediksi* menerima masukan parameter berupa matriks hasil aktivasi dan *bobotOutput*
- Baris 2 : Proses pembuatan matriks baru dengan nama matriks *yPrediksi*
- Baris 3 s.d. 11 : Proses perulangan untuk mendapatkan hasil prediksi dengan mengalikan matriks aktivasi dengan matriks *bobotOutput*
- Baris 12 : Mengembalikan nilai matriks *yPrediksi*

#### 4.7.10 Implementasi Denormalisasi Data

Denormalisasi merupakan proses untuk mengembalikan nilai hasil prediksi menjadi nilai asli. Implementasi proses denormalisasi data ditunjukkan pada Kode Program 4.12.

No	Kode Program
1	public double[][] denormalisasi(double[][]
2	yPrediksi, double[][] data) {
3	double[][] denormal = new
4	double[yPrediksi.length][yPrediksi[0].length];
5	for (int i = 0; i < yPrediksi.length; i++) {
6	for (int j = 0; j < yPrediksi[0].length;
7	j++) {
8	denormal[i][j] = (yPrediksi[i][j] * (getmax(data) - getmin(data))) + getmin(data);
9	}
10	}
11	return denormal;
12	}

**Kode Program 4.12 Proses Denormalisasi Data**

Penjelasan kode program fungsi *denormalisasi* berdasarkan Kode Program 4.12 adalah sebagai berikut.

- Baris 1 : Fungsi *denormalisasi* menerima masukan parameter berupa matriks *yPrediksi* data
- Baris 2 : Proses pembuatan matriks baru dengan nama matriks

- Baris 3 s.d. 7 : Proses perulangan untuk melakukan proses denormalisasi berdasarkan Persamaan 2.15
- Baris 12 : Mengembalikan nilai matriks *denormal*

#### 4.7.11 Implementasi Hitung Mean Absolute Percentage Error

Perhitungan nilai MAPE dilakukan untuk mengukur tingkat kesalahan dari hasil peramalan. Implementasi proses hitung MAPE ditunjukkan pada Kode Program 4.13.

No	Kode Program
1	<code>public double mape(double[][] denormal, double[][] data) {</code>
2	<code>double[] selisih = new double[data.length];</code>
3	<code>double[] err = new double[data.length];</code>
4	<code>double total = 0;</code>
5	<code>for (int i = 0; i &lt; selisih.length; i++) {</code>
6	<code>selisih[i] = Math.abs(denormal[i][0] -</code>
7	<code>data[i][data[0].length - 1]);</code>
8	<code>err[i] = Math.abs(selisih[i] /</code>
9	<code>denormal[i][0]);</code>
10	<code>total += err[i];</code>
11	<code>}</code>
12	<code>return (total / denormal.length) * 100;</code>
13	<code>}</code>

**Kode Program 4.13 Proses Hitung Nilai MAPE**

Penjelasan kode program fungsi MAPE berdasarkan Kode Program 4.13 adalah sebagai berikut.

- Baris 1 : Fungsi *mape* menerima masukan parameter berupa matriks *denormal* dan data
- Baris 2 : Proses pembuatan matriks baru dengan nama matriks *selisih* yang akan diisi dengan hasil pengurangan nilai *denormal* dengan nilai target
- Baris 3 : Proses pembuatan matriks baru dengan nama matriks *err* yang akan diisi dengan hasil pembagian nilai *selisih* dengan nilai *denormal*
- Baris 4 : Proses inisialisasi nilai total berupa 0
- Baris 5 s.d. 9 : Proses perulangan untuk mendapatkan nilai total *error* dari hasil prediksi
- Baris 12 : Mengembalikan hasil perhitungan MAPE

#### 4.8 Implementasi Antarmuka

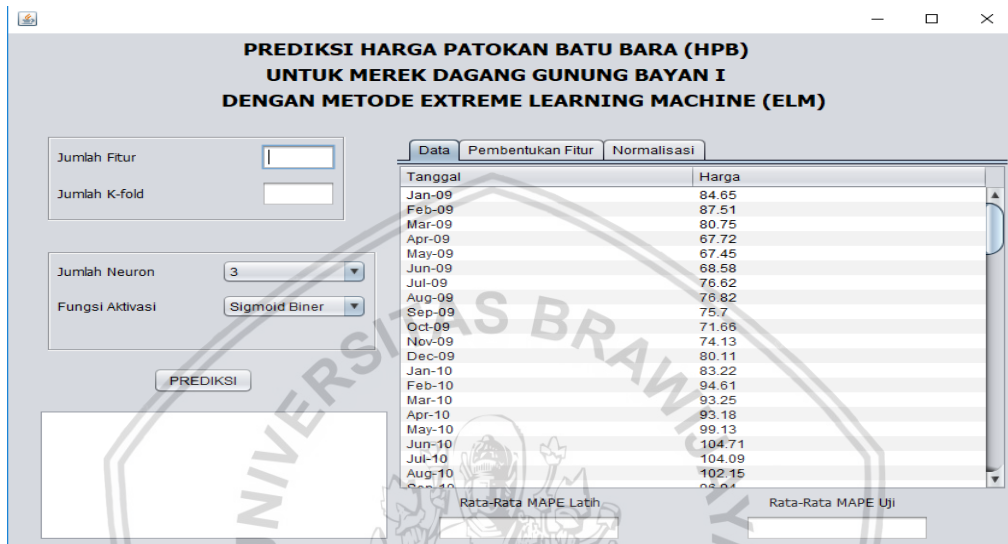
Implementasi antarmuka merupakan hasil implementasi dari perancangan antarmuka yang telah dibuat pada sub bab sebelumnya. Sesuai dengan pembahasan berdasarkan skenario perancangan, dibuat implementasi



antarmuka yang terdiri dari implementasi halaman awal hingga tampilan halaman ketika proses prediksi dimulai.

#### 4.8.1 Implementasi Halaman Awal

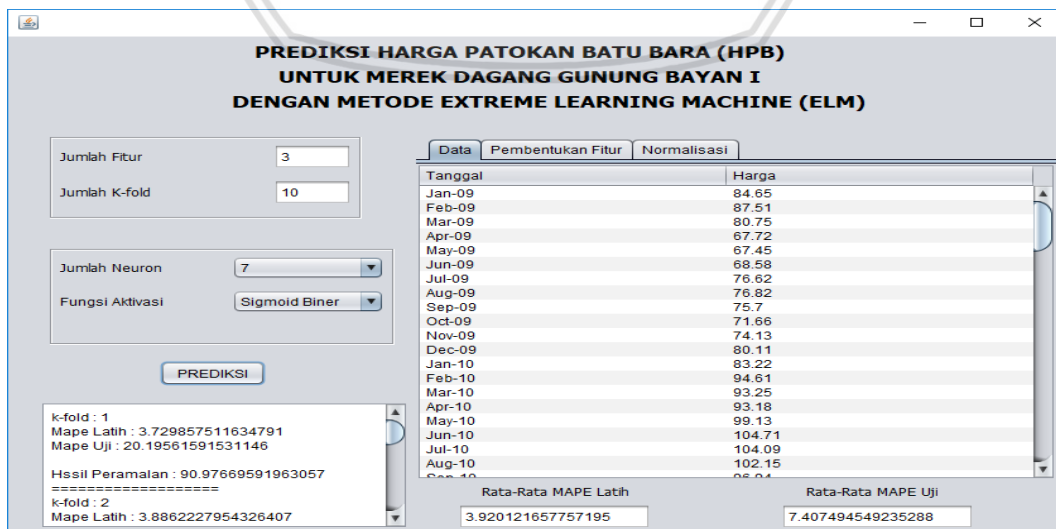
Halaman awal berisi tampilan untuk menampilkan seluruh dataset yang akan digunakan untuk melakukan proses prediksi dengan menggunakan ELM. Dataset terdiri dari daftar tanggal dan daftar harga. Bentuk implementasi halaman awal ditunjukkan pada Gambar 4.22.



Gambar 4.22 Implementasi Halaman Awal

#### 4.8.2 Implementasi Halaman Masukan Pengguna

Proses prediksi dengan menggunakan algoritme ELM akan bisa dikerjakan apabila pengguna sudah memasukkan beberapa parameter yang dibutuhkan oleh sistem. Masukan tersebut terdiri dari jumlah fitur, jumlah *k-fold*, jumlah *neuron*, serta jenis fungsi aktivasi yang digunakan. Bentuk implementasi halaman masukan pengguna ditunjukkan pada Gambar 4.23.



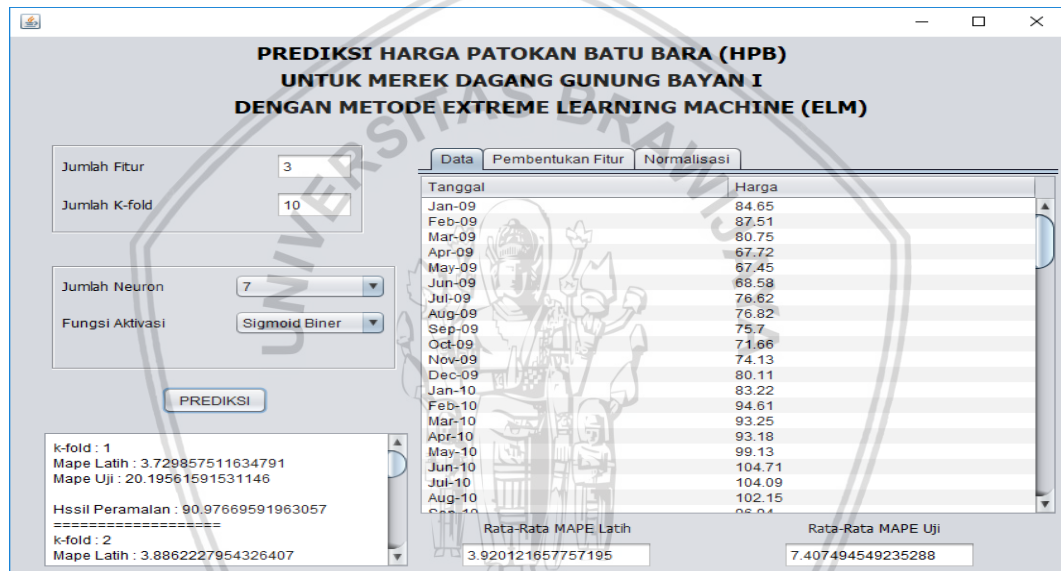
Gambar 4.23 Implementasi Halaman Masukan Pengguna



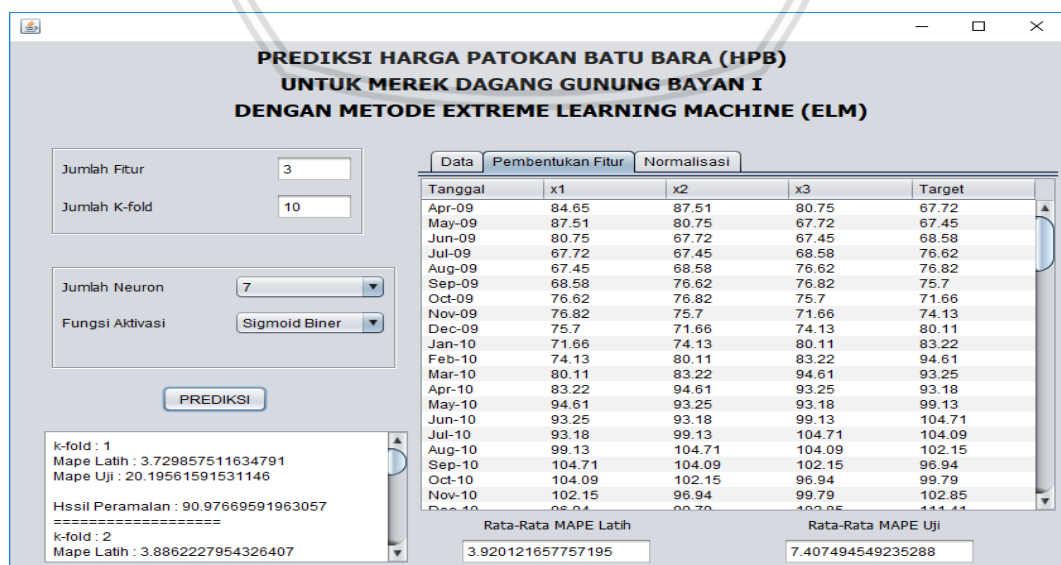
Setelah semua parameter dimasukkan oleh pengguna, pengguna dapat menekan tombol “Prediksi”. Hasil yang akan di tampilkan pada saat melakukan prediksi adalah hasil MAPE latih dan uji tiap *k-fold*, serta rata-rata MAPE untuk proses latih dan uji secara keseluruhan.

### 4.8.3 Implementasi Tabel Data, Tabel Pembentukan Fitur, dan Tabel Normalisasi

Berdasarkan perancangan yang sudah dibuat sebelumnya, tabel data akan berisi seluruh *dataset* historis harga patokan batu bara. Tabel pembentukan fitur akan dibentuk sesuai dengan jumlah masukan dari pengguna. Tabel normalisasi akan menormalisasi seluruh data hasil dari pembentukan fitur. Implementasi untuk masing-masing tabel secara berurutan ditunjukkan pada Gambar 4.24 sampai 4.26.



Gambar 4.24 Implementasi Tabel Data



Gambar 4.25 Implementasi Pembentukan Fitur





### PREDIKSI HARGA PATOKAN BATU BARA (HPB) UNTUK MEREK DAGANG GUNUNG BAYAN I DENGAN METODE EXTREME LEARNING MACHINE (ELM)

Jumlah Fitur:

Jumlah K-fold:

Jumlah Neuron:

Fungsi Aktivasi:

**PREDIKSI**

k-fold : 1  
Mape Latih : 3.729857511634791  
Mape Uji : 20.19561591531146

Hssil Peramalan : 90.97669591963057  
=====

k-fold : 2  
Mape Latih : 3.8862227954326407

Tanggal	x1	x2	x3	Target
Apr-09	0.366440842...	0.401040406...	0.319259617...	0.161625937...
May-09	0.401040406...	0.319259617...	0.161625937...	0.158359545...
Jun-09	0.319259617...	0.161625937...	0.158359545...	0.172030002...
Jul-09	0.161625937...	0.158359545...	0.172030002...	0.269295910...
Aug-09	0.158359545...	0.172030002...	0.269295910...	0.271715460...
Sep-09	0.172030002...	0.269295910...	0.271715460...	0.258165981...
Oct-09	0.269295910...	0.271715460...	0.258165981...	0.209291071...
Nov-09	0.271715460...	0.258165981...	0.209291071...	0.239172513...
Dec-09	0.258165981...	0.209291071...	0.239172513...	0.311517057...
Jan-10	0.209291071...	0.239172513...	0.311517057...	0.349141059...
Feb-10	0.239172513...	0.311517057...	0.349141059...	0.486934430...
Mar-10	0.311517057...	0.349141059...	0.486934430...	0.470481490...
Apr-10	0.349141059...	0.486934430...	0.470481490...	0.469634647...
May-10	0.486934430...	0.470481490...	0.469634647...	0.541616259...
Jun-10	0.470481490...	0.469634647...	0.541616259...	0.609121703...
Jul-10	0.469634647...	0.541616259...	0.609121703...	0.601621098...
Aug-10	0.541616259...	0.609121703...	0.601621098...	0.578151463...
Sep-10	0.609121703...	0.601621098...	0.578151463...	0.515122187...
Oct-10	0.601621098...	0.578151463...	0.515122187...	0.549600774...
Nov-10	0.578151463...	0.515122187...	0.549600774...	0.586619888...
Dec-10	0.515122187...	0.549600774...	0.586619888...	0.580476627...

Rata-Rata MAPE Latih:

Rata-Rata MAPE Uji:

Gambar 4.26 Implementasi Normalisasi



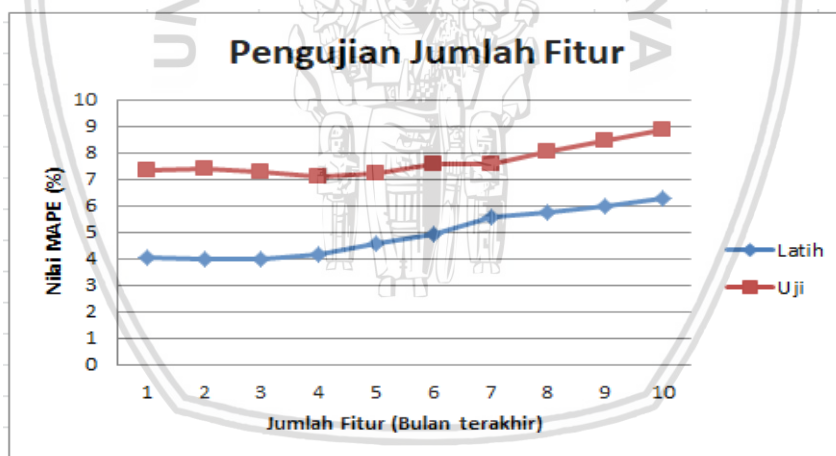
## BAB 5 HASIL DAN ANALISIS

Bab ini berisi pembahasan mengenai prosedur dan hasil pengujian prediksi nilai harga patokan batu bara untuk merek dagang Gunung Bayan I dengan menggunakan metode *Extreme Learning Machine* (ELM). Terdapat empat pengujian yang akan dilakukan pada penelitian ini. Pengujian tersebut terdiri dari pengujian jumlah fitur data, pengujian jumlah *neuron*, dan pengujian fungsi aktivasi. Proses pembagian data dalam proses *training* dan *testing* dilakukan dengan menggunakan evaluasi 10 *cross validation* yang telah dijelaskan pada Bab 2.

### 5.1 Pengujian Jumlah Fitur

Pengujian jumlah fitur pada metode ELM dilakukan untuk mengetahui banyaknya jumlah fitur terbaik yang harus dibentuk. Jumlah fitur merepresentasikan banyaknya data historis harga batu bara dalam kurun waktu beberapa bulan terakhir, yang kemudian akan digunakan untuk melakukan proses prediksi harga di masa yang akan datang.

Pengujian dilakukan sebanyak 10 kali dengan parameter jumlah *neuron* sebanyak 5, dan jenis fungsi aktivasi yang digunakan adalah fungsi *sigmoid biner*. Hasil pengujian jumlah fitur secara lengkap ditunjukkan pada Tabel 5.1.



Gambar 5.1 Grafik Pengujian Jumlah Fitur Terhadap Nilai MAPE (%)

Berdasarkan hasil pengujian pada Gambar 5.1 dan Tabel 5.1, nilai MAPE terkecil didapatkan dengan jumlah fitur sebanyak 3, yaitu dengan MAPE *training* sebesar 3,955183% dan MAPE *testing* sebesar 7,303268%. Penentuan jumlah fitur dalam penelitian prediksi harga patokan batu bara berpengaruh terhadap berapa jangka waktu harga di masa lalu yang memengaruhi nilai harga di masa yang akan datang.

Jumlah fitur yang banyak tidak selalu menghasilkan hasil prediksi yang baik. Hal ini dikarenakan dalam melakukan prediksi harga batu bara, harga masa lalu yang terlalu lama tidak dapat memengaruhi harga di masa yang akan datang. Hal ini dibuktikan dengan meningkatnya nilai *error* apabila jumlah fitur yang digunakan dinaikkan.

Tabel 5.1 Pengujian Jumlah Fitur Terhadap Nilai MAPE (%)

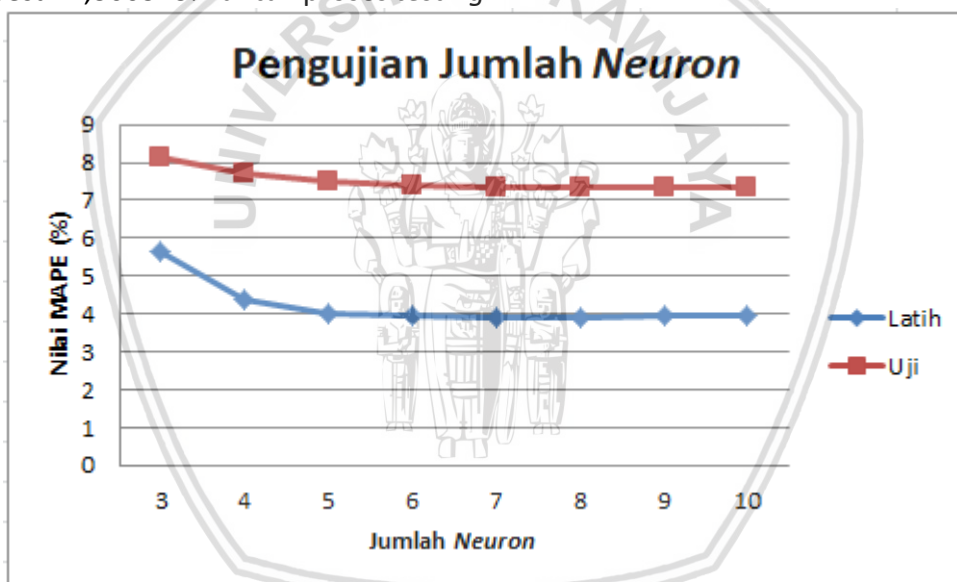
Fitur	Percobaan										Rata-Rata	
	1	2	3	4	5	6	7	8	9	10		
1	Latih	3,967798	4,014712	4,018211	4,037144	4,208997	3,965805	4,043196	4,125261	3,966911	4,085993	4,043403
	Uji	7,351603	7,451256	7,635928	7,385465	6,981823	7,333767	7,453269	6,983993	7,329593	7,347287	7,325399
2	Latih	3,971821	3,966867	3,973417	3,965351	3,963028	3,969753	3,98342	3,965911	3,96526	3,963123	3,968795
	Uji	7,452455	7,374686	7,489337	7,365866	7,361317	7,363409	7,204718	7,402892	7,300971	7,355228	7,367088
3	Latih	<b>3,920718</b>	<b>3,979877</b>	<b>3,948316</b>	<b>3,994107</b>	<b>3,955803</b>	<b>3,944472</b>	<b>3,914285</b>	<b>3,977591</b>	<b>3,965137</b>	<b>3,951524</b>	<b>3,955183</b>
	Uji	<b>7,274269</b>	<b>7,349498</b>	<b>7,560616</b>	<b>7,394933</b>	<b>7,258643</b>	<b>7,369928</b>	<b>7,078611</b>	<b>7,189336</b>	<b>7,231474</b>	<b>7,325369</b>	<b>7,303268</b>
4	Latih	4,133555	4,252043	4,089588	3,980538	4,284764	4,238013	4,21049	4,449057	4,150718	4,023824	4,181259
	Uji	6,857699	6,226871	7,546516	7,532883	7,886841	6,223994	6,757458	6,874451	7,139445	7,750599	7,079676
5	Latih	4,479361	4,419889	4,626239	4,937984	4,254989	4,73138	4,391388	4,301859	4,87861	4,78075	4,580245
	Uji	6,503723	7,488328	7,91239	6,914769	6,677242	7,85161	7,367511	6,950775	7,552177	7,039672	7,22582
6	Latih	4,864113	4,989294	4,794162	4,939431	5,551545	4,887504	4,735215	4,734107	4,954759	4,908078	4,935821
	Uji	8,056552	8,482468	7,272132	8,073279	6,193471	6,935386	7,79136	7,269794	7,783747	7,765218	7,562341
7	Latih	6,10836	5,61765	5,733165	5,289326	5,245096	5,730676	5,517819	4,713032	5,373717	6,300703	5,562954
	Uji	7,834313	6,725385	8,298213	6,563778	8,892071	7,095097	8,892125	7,592369	7,135328	6,587293	7,561597
8	Latih	4,808458	5,954697	4,82783	5,8512	6,444365	5,924458	5,64055	5,750284	6,566209	5,419986	5,718804
	Uji	8,387138	7,737162	7,763527	7,585416	7,100049	8,919825	8,929372	8,245061	7,668682	8,08558	8,042181
9	Latih	6,006544	6,431412	5,491988	6,35802	5,92382	6,106437	5,633071	5,636984	6,122234	6,369086	6,00796
	Uji	9,867161	7,373041	9,373848	7,196131	8,758553	9,216709	6,903933	9,442524	7,709942	8,799464	8,464131
10	Latih	5,999419	5,639205	6,214649	6,281251	6,59356	6,594558	6,51679	6,287546	5,875719	6,500611	6,250331
	Uji	6,960849	9,678901	8,931992	9,840854	7,792986	9,276879	8,835249	9,793984	8,776204	8,546176	8,843407

## 5.2 Pengujian Jumlah *Neuron*

Pengujian jumlah *neuron* pada ELM dilakukan untuk mengetahui jumlah *neuron* terbaik yang digunakan dalam prediksi harga patokan batu bara. *Neuron* merupakan jaringan saraf yang berfungsi untuk mentransformasikan informasi. Jumlah *neuron* menentukan banyaknya *neuron* yang harus diaktifkan untuk mendapatkan suatu hasil keluaran yang baik.

Proses pengujian jumlah *neuron* dilakukan sebanyak 10 kali dengan parameter jumlah fitur sebanyak 3, jumlah *k-fold* 10 dan jenis fungsi aktivasi yang digunakan adalah fungsi *sigmoid biner*. Pengujian jumlah *neuron* dilakukan mulai dari jumlah *neuron* sebanyak 3 hingga 10. Hasil dari pengujian tersebut ditunjukkan pada Tabel 5.2.

Nilai rata-rata MAPE terbaik untuk setiap proses *training* dan *testing* didapatkan ketika jumlah *neuron* yang digunakan adalah sebanyak 7 *neuron*. Nilai rata-rata MAPE tersebut adalah sebesar 3,926804% untuk proses *training* dan sebesar 7,360343% untuk proses *testing*.



Gambar 5.2 Grafik Pengujian Jumlah *Neuron* Terhadap Nilai MAPE (%)

Hasil yang ditunjukkan pada Gambar 5.2, didapatkan bahwa semakin banyak jumlah *neuron* maka tingkat kesalahan yang dihasilkan oleh proses ELM akan semakin kecil dan cukup stabil. Hal ini sesuai dengan hasil pengujian pada Tabel 5.2 yang menunjukkan nilai MAPE yang cukup stabil di 3,9% untuk proses *training* dan 7% untuk proses *testing*.

Nilai MAPE yang tinggi pada saat jumlah *neuron* yang digunakan sedikit terjadi karena *underfitting*. *Underfitting* adalah suatu kejadian ketika proses *training* tidak mampu mengenali pola pembelajaran dengan baik akibat jumlah *neuron* yang sedikit. Hal tersebut menyebabkan pola yang dikenali cukup terbatas dan menghasilkan nilai *error* yang tinggi ketika melakukan proses *testing* (Giusti, et al., 2017).

**Tabel 5.2 Pengujian Jumlah Neuron Terhadap Nilai MAPE (%)**

Neuron		Percobaan										Rata-Rata
		1	2	3	4	5	6	7	8	9	10	
3	Latih	5,91806	5,402264	5,621829	5,057313	4,783081	6,096101	5,94715	5,753015	5,261486	6,399111	5,623941
	Uji	8,494876	6,627647	8,6055	9,69686	6,761089	9,120281	8,839261	9,194789	7,527089	6,575438	8,144283
4	Latih	4,787993	4,526138	4,342601	4,204603	4,082286	4,314778	4,40623	4,464875	4,547519	4,279259	4,395628
	Uji	7,76875	7,673435	8,016321	6,892467	7,360905	8,126782	7,860613	8,384356	7,964791	7,236925	7,728534
5	Latih	4,013443	3,979877	4,136424	3,994107	4,052612	4,101318	3,933556	3,977591	3,965137	3,951524	4,010559
	Uji	8,006383	7,349498	7,72264	7,394933	7,679123	7,10234	7,818667	7,189336	7,231474	7,325369	7,481976
6	Latih	3,933061	3,996072	3,96651	3,915756	3,931376	3,922712	3,976233	3,942579	3,932749	3,918021	3,943507
	Uji	7,218953	8,294677	7,220166	7,304902	7,316115	7,245469	7,314751	7,163404	7,224261	7,608053	7,391075
7	Latih	<b>3,916647</b>	<b>3,937127</b>	<b>3,923503</b>	<b>3,929947</b>	<b>3,9312</b>	<b>3,923928</b>	<b>3,921874</b>	<b>3,932075</b>	<b>3,929011</b>	<b>3,922725</b>	<b>3,926804</b>
	Uji	<b>7,404651</b>	<b>7,313791</b>	<b>7,349051</b>	<b>7,262641</b>	<b>7,356093</b>	<b>7,489951</b>	<b>7,273953</b>	<b>7,365391</b>	<b>7,43755</b>	<b>7,350354</b>	<b>7,360343</b>
8	Latih	3,920967	3,928123	3,938311	3,916082	3,936199	3,927577	3,932907	3,929198	3,928128	3,937502	3,929499
	Uji	7,306865	7,302664	7,395789	7,330034	7,461323	7,303912	7,488248	7,36318	7,270147	7,340297	7,356246
9	Latih	3,932803	3,933926	3,930007	3,934782	3,931942	3,93731	3,931656	3,930218	3,934243	3,933568	3,933046
	Uji	7,315382	7,32627	7,423649	7,379525	7,388407	7,325681	7,360108	7,335357	7,288461	7,296949	7,343979
10	Latih	3,932027	3,935174	3,936618	3,93186	3,933502	3,936782	3,933645	3,931721	3,934577	3,934778	3,934068
	Uji	7,292925	7,354454	7,379183	7,307539	7,367326	7,360149	7,316393	7,211727	7,364057	7,305492	7,325924

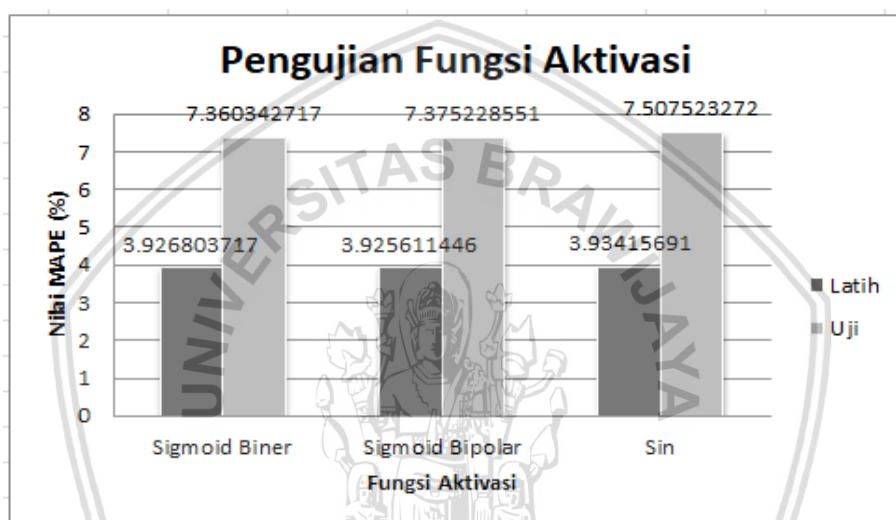
**Tabel 5.3 Pengujian Fungsi Aktivasi Terhadap Nilai MAPE (%)**

Fungsi Aktivasi		Percobaan										Rata-Rata
		1	2	3	4	5	6	7	8	9	10	
Sig Biner	Latih	<b>3,916647</b>	<b>3,937127</b>	<b>3,923503</b>	<b>3,929947</b>	<b>3,9312</b>	<b>3,923928</b>	<b>3,921874</b>	<b>3,932075</b>	<b>3,929011</b>	<b>3,922725</b>	<b>3,926804</b>
	Uji	<b>7,404651</b>	<b>7,313791</b>	<b>7,349051</b>	<b>7,262641</b>	<b>7,356093</b>	<b>7,489951</b>	<b>7,273953</b>	<b>7,365391</b>	<b>7,43755</b>	<b>7,350354</b>	<b>7,360343</b>
Sig Bipolar	Latih	3,93266	3,934266	3,918859	3,929461	3,922162	3,917595	3,930703	3,923936	3,921287	3,925185	3,925611
	Uji	7,304369	7,242863	7,370704	7,336045	7,478113	7,362299	7,306889	7,54234	7,364732	7,443931	7,375229
Sin	Latih	3,926297	3,967662	3,926761	3,927555	3,936946	3,925455	3,928921	3,933371	3,922401	3,9462	3,934157
	Uji	7,847482	7,27959	7,373168	7,431593	7,568773	7,835445	7,417778	7,574286	7,400617	7,3465	7,507523

### 5.3 Pengujian Fungsi Aktivasi

Pengujian fungsi aktivasi dilakukan untuk mendapatkan fungsi aktivasi terbaik yang sesuai untuk proses prediksi harga patokan batu bara dengan metode ELM. Penentuan fungsi aktivasi yang digunakan akan berpengaruh dalam proses perhitungan nilai keluaran *hidden layer*.

Proses pengujian dilakukan sebanyak 10 kali dengan membandingkan 3 jenis fungsi aktivasi, yaitu fungsi *sigmoid biner*, fungsi *sigmoid bipolar*, dan fungsi *sin*. Parameter yang digunakan untuk proses pengujian adalah jumlah fitur sebanyak 3, jumlah *neuron* sebanyak 7, dan jumlah *k-fold* sebanyak 10. Hasil rincian pengujian fungsi aktivasi ditunjukkan pada Tabel 5.3.



Gambar 5.3 Grafik Pengujian Fungsi Aktivasi Terhadap Nilai MAPE (%)

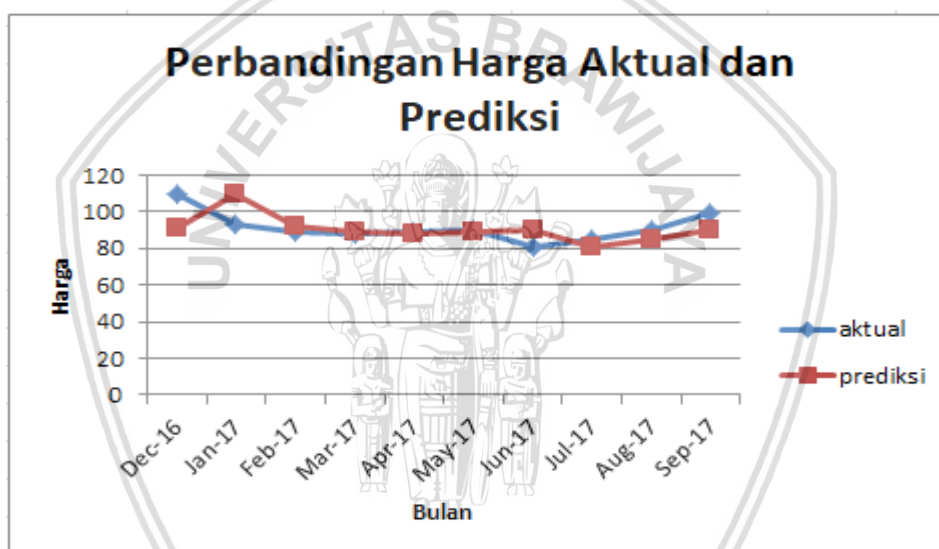
Gambar 5.3 berisi grafik rata-rata nilai MAPE untuk setiap fungsi aktivasi yang menunjukkan bahwa perbedaan nilai MAPE yang didapatkan oleh ketiga fungsi aktivasi tidak terlalu jauh. Akan tetapi, nilai MAPE terbaik didapatkan dengan menggunakan fungsi aktivasi *sigmoid biner* dengan nilai MAPE untuk proses *training* sebesar 3,926804% dan untuk proses *testing* sebesar 7,360343%. Berdasarkan hasil pengujian tersebut, maka fungsi aktivasi yang paling cocok adalah menggunakan fungsi *sigmoid biner*, karena rentang data yang digunakan sebagai masukan sistem cocok dengan rentang *output* yang dihasilkan oleh fungsi *sigmoid biner*, yaitu antara 0 sampai 1.

### 5.4 Perbandingan Harga Aktual dan Hasil Prediksi

Berdasarkan hasil pengujian yang dilakukan sebelumnya, didapatkan nilai parameter terbaik dari hasil pembelajaran dengan menggunakan algoritme ELM, yaitu jumlah fitur sebanyak 3, jumlah *neuron* sebanyak 7, dan jenis fungsi aktivasi terbaik adalah fungsi aktivasi *sigmoid biner*. Tabel 5.4 dan Gambar 5.4 menunjukkan perbandingan harga patokan batu bara aktual dengan harga hasil prediksi dengan menggunakan algoritme ELM.

**Tabel 5.4 Perbandingan Harga Aktual dan Hasil Prediksi**

Bulan	Aktual (US\$/ton)	Prediksi (US\$/ton)	MAPE (%)
Desember 2016	109,35	91,41349	19,62129
Januari 2017	92,6	109,3678	15,33155
Februari 2017	89,45	92,5729	3,373454
Maret 2017	87,91	89,23708	1,48714
April 2017	88,57	87,85416	0,814808
Mei 2017	89,98	88,70023	1,442806
Juni 2017	80,94	90,02906	10,09569
Juli 2017	84,72	80,80874	4,84015
Agustus 2017	90,15	84,71047	6,421318
September 2017	98,88	90,16908	9,660646
Rata-rata MAPE			7,308886



**Gambar 5.4 Grafik Perbandingan Harga Aktual dan Hasil Prediksi**

Nilai perbandingan harga aktual dan prediksi dari Desember 2016 hingga September 2017 menunjukkan nilai MAPE yang berbeda-beda setiap bulannya. Nilai MAPE terendah didapatkan pada bulan April 2017 sebesar 0.814808%, sedangkan nilai MAPE tertinggi didapatkan pada bulan Desember 2016 sebesar 19,62129%. Berdasarkan kriteria nilai MAPE yang ditunjukkan pada Tabel 2.2, hasil prediksi harga patokan batu bara tersebut masih tergolong baik karena nilai MAPE yang didapat < 20%.

## BAB 6 PENUTUP

### 6.1 Kesimpulan

Berdasarkan perancangan, implementasi, dan hasil pengujian dari penelitian dengan judul Prediksi Nilai Harga Patokan Batu Bara (HPB) untuk Merek Dagang Gunung Bayan I dengan Metode *Extreme Learning Machine* (ELM), maka diambil kesimpulan sebagai berikut:

1. Algoritme pembelajaran berbasis jaringan saraf tiruan berupa *Extreme Learning Machine* dapat digunakan untuk melakukan proses prediksi terhadap nilai harga patokan batu bara untuk merek dagang Gunung Bayan I. Proses implementasi algoritme tersebut terdiri dari:
  - a. Proses normalisasi data
  - b. Proses *training*
  - c. Proses *testing*
  - d. Proses denormalisasi data
  - e. Proses perhitungan nilai *Mean Absolute Percentage Error* (MAPE)
2. Penentuan parameter yang tepat untuk melakukan proses pembelajaran menggunakan algoritme *Extreme Learning Machine* (ELM) untuk proses prediksi harga patokan batu bara Gunung Bayan I akan mempengaruhi tingkat kesalahan yang dihasilkan pada hasil prediksi. Beberapa parameter yang digunakan untuk proses pembelajaran menggunakan ELM terdiri dari banyaknya jumlah fitur pembentukan data, jumlah *neuron*, serta fungsi aktivasi. Berdasarkan pengujian yang telah dilakukan, didapatkan nilai rata-rata MAPE minimum sebesar 3,926804% untuk proses *training* dan sebesar 7,360343% untuk proses *testing*. Hasil tersebut didapatkan dengan menggunakan jumlah fitur sebanyak 3, jumlah *neuron* sebanyak 7, serta fungsi aktivasi *sigmoid biner*. Berdasarkan kriteria nilai MAPE yang ditunjukkan pada Tabel 2.2, rata-rata nilai MAPE yang < 10% masuk dalam kategori sangat baik, sehingga metode ELM dapat dinyatakan sangat baik untuk melakukan prediksi harga patokan batu bara dengan merek dagang Gunung Bayan I.

### 6.2 Saran

Penelitian Prediksi Nilai Harga Patokan Batu Bara (HPB) untuk Merek Dagang Gunung Bayan I dengan Metode *Extreme Learning Machine* (ELM) ini, masih memiliki beberapa kekurangan. Beberapa saran yang dapat digunakan sebagai bahan pengembangan lebih lanjut adalah:

1. Menambahkan jumlah historis data yang digunakan untuk meningkatkan kinerja ELM agar dapat mengenali pola data lebih baik pada proses pembelajaran



2. Menambahkan parameter data lain seperti jumlah permintaan batu bara untuk menghasilkan nilai prediksi yang lebih akurat
3. Melakukan optimasi terhadap proses pencarian nilai bobot dan bias, seperti menggunakan metode optimasi algoritme genetika
4. Melakukan pengujian fungsi aktivasi lain yang belum diuji coba pada penelitian ini.



## DAFTAR PUSTAKA

- Agustin, M. & Prahasto, T., 2012. Penggunaan Jaringan Syaraf Tiruan Backpropagation Untuk Seleksi Penerimaan Mahasiswa Baru Pada Jurusan Teknik Komputer Di Politeknik Negeri Sriwijaya. *Jurnal Sistem Informasi Bisnis*, Volume 2 No 2, pp. 089-097.
- Anggraini, S. D., 2017. Prediksi Nilai Tukar Mata Uang Asing Menggunakan Extreme Learning Machine. *Jurnal Ilmiah Matematika*, pp. 110-115.
- Arifianty, A., M. & Irzal, M., 2015. Peramalan Indeks Harga Saham Gabungan (IHSG) dengan Metode Extreme Learning Machine (ELM). *Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Jakarta*.
- Atmojo, T. B., Pulungan, R. & Syahputra, H., 2013. Pengembangan Model Peramalan Permintaan Kebutuhan Reseller Menggunakan Extreme Learning Machine dalam Konteks Intelligent Warehouse Management System (IWMS). *Seminar Nasional Informatika*, pp. 258-263.
- Bahtiar, S. M., 2016. Peramalan Beban Dengan Menggunakan Metode Time Series Untuk Kebutuhan Tenaga Listrik Di Gardu Induk Sungai Raya. *Jurnal Teknik Elektro Universitas Tanjungpura*, Volume 2 No 1.
- Chandar, S. K., Sumathi, M. & Sivanadam, S., 2016. Forecasting Gold Prices Based on Extreme Learning Machine. *International Journal Of Computers Communications and Control*, pp. 372-380.
- Chang, P.-C., Wang, Y.-W. & Liu, C.-H., 2007. The Development of a Weighted Evolving Fuzzy Neural Network for PCB Sales Forecasting. *Elsevier, 32 (Expert System with Application)*, pp. 86-96.
- Dahria, M., 2008. Kecerdasan Buatan (Artificial Intelligence). *Jurnal SAINTIKOM Vol 5 No 2*.
- Fikriya, Z. A., Irawan, M. I. & S., 2017. Implementasi Extreme Learning Machine Untuk Pengenalan Objek Citra Digital. *Jurnal Sains dan Seni ITS*, Volume 6 No 1, pp. 18-23.
- Finkelman, R. B., 1993. *Trace and Minor Elements in Coal*. New York: Plenum Press.
- Fonseca-Delgado, R. & Gomez-Gil, P., 2013. *An Assessment of Ten-Fold and Monte Carlo Cross Validation for Time Series Forecasting*. Mexico City, Mexico, Institute of Electrical and Electronics Engineers, pp. 215-220.
- 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 Vol 2 No 8*, pp. 2972-2978.

- Huang, G.-B., Zhu, Q.-Y. & Siew, C.-K., 2006. Extreme Learning Machine: Theory and Applications. *Neurocomputing* 70, pp. 489-501.
- Pangaribuan, J. J., 2016. Mendiagnosis Penyakit Diabetes Melitus Dengan Menggunakan Metode Extreme Learning Machine. *Journal Information System Development*, Volume 1 No 2, pp. 32-40.
- Puspitaningrum, D., 2006. *Pengantar Jaringan Syaraf Tiruan*. Yogyakarta: Penerbit Andi.
- Rahayuningsih, E., 2006. *Penyusunan Anggaran Arus Kas (Cash Flow) Pada PT. PLN (Persero) Distribusi Jawa Tengah dan D.I. Yogyakarta*, Semarang: Fakultas Ekonomi, Universitas Negeri Semarang.
- Saluza, I., 2016. *Peramalan Saham Menggunakan Backpropagation dengan Hibridisasi Input*. s.l., Universitas Sriwijaya, pp. 701-705.
- Siang, J. J., 2004. *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab*. Yogyakarta: Penerbit Andi.
- Siwi, I. P., 2016. Peramalan Produksi Gula Pasir Menggunakan Extreme Learning Machine Pada PG Candi Baru Sidoarjo. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*.
- Stach, E., 1982. *Coal Petrology*. Gebru der Borntraeger, Berlin: s.n.
- Sundari, C., 2015. Pengaruh Harga Batubara Acuan (HBA) Terhadap Return Saham Dengan Profitabilitas Sebagai Variabel Intervening Pada Perusahaan Tambang Btubara Di Bei. *Jurnal Informasi dan Pengembangan Iptek*, Volume 11 No 2, pp. 150-162.
- Sutojo, T., Mulyanto, E. & Suhartono, V., 2011. *Kecerdasan Buatan*. Yogyakarta: Penerbit Andi.
- Syed, A. & Anggarwal, S., 2016. Wind Power Forecasting Using Extreme Learning Machine. *International Journal of Research and Scientific Innovation*, pp. 33-37.
- Umar, M. T. H., Salam, A. E. U., Abbas, A. Z. & Gunadin, I. C., 2014. Sistem Deteksi Kebocoran pada Jaringan Pipa Air PDAM Menggunakan Analisis Tekanan dengan Metode Extreme Learning Machine. *Jurnal Tugas Akhir Teknik Informatika*, pp. 1-9.
- Zhang, B. & Ma, J., 2011. Coal Price Index Forecast by a New Partial Least-Squares Regression. *Procedia Engineering*, pp. 5025-5029.