PREDIKSI HARGA PASAR DAGING SAPI DI KOTA MALANG DENGAN MENGGUNAKAN METODE *EXTREME LEARNING MACHINE* (ELM)

SKRIPSI

Disusun oleh: Cusen Mosabeth NIM: 145150207111052



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

PENGESAHAN

PREDIKSI HARGA PASAR DAGING SAPI DI KOTA MALANGDENGAN MENGGUNAKAN METODE EXTREME LEARNING MACHINE (ELM)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer

> Disusun Oleh: Cusen Mosabeth NIM: 145150207111052

Skripsi ini telah diuji dan dinyatakan lulus pada 27 Juli 2018 Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Muhammad Tanzil Furgon S.Kom, M.CompSc

Randy Cahya Wihandika S.Kom, M.Kom

NIP: 19820930 200801 1 004

NIK: 201405 880206 1 001

Mengetahui, Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, didalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik disuatu 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 unsurunsur 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, 03 Agustus 2018

Cusen Mosabeth

NIM: 145150207111052



KATA PENGANTAR

Puji syukur penulis panjatkan atas kehadirat Allah SWT, karena dengan limpahkan Rahmat dan Karunia-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul "Prediksi Harga Pasar Daging Sapi Menggunakan Metode Extreme Learning Machine (ELM)".

Skripsi yang dibuat merupakan tugas akhir untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer pada Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya. Dalam penyusunan skripsi ini penulis mendapat banyak bantuan dari berbagai pihak, untuk itu dalam kesempatan ini penulis ingin menyampaikan ucapan terima kasih kepada pihak-pihak yang telah membantu selama penyusunan skripsi ini, diantaranya:

- 1. Bapak Muhammad Tanzil Furqon, S.Kom., M.CompSc selaku Dosen Pembimbing I yang telah meluangkan waktunya untuk membimbing, mengarahkan dan memberikan ilmu dalam penyusunan skripsi.
- 2. Bapak Randy Cahya Wihandika, S.ST., M.Kom selaku Dosen Pembimbing II yang telah meluangkan waktunya untuk membimbing dan memberikan ilmu dalam penyusunan 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. Bapak dan Ibu Dosen Fakultas Ilmu Komputer yang telah memberikan Ilmu selama proses perkuliahan.
- 7. Kedua orang tua penulis yang selalu memberikan doa dan motivasi kepada penulis dalam menyelesaikan skripsi ini.
- 8. Yulia Kurniawati, Vania Nuraini, Novirra Dwi Asri, Nuriya Fadilah, Eka Miyahil Uyun, Evilia Nurharsanti, Nadya Sylviani, Olivia Bonita, Revinda Bertananda, Irma Ramadanti F, Krishnanti Dewi, Yuniar Siska Fatmala, Nur Afdaliyah Anwar, Isradi Azhar, Nirmala Faizah Saraswati, Audia Refanda, BE'M' FILKOM, serta teman-teman dekat lainnya yang selalu memberikan dukungan, semangat dan menghibur setiap harinya.
- 9. Teman-teman Informatika 2014 yang telah membantu dan memberikan dukungan.
- 10. Serta semua pihak yang telah membantu secara langsung maupun tidak langsung sehingga penulis dapat menyelesaikan skripsi ini.

Penulis menyadari masih terdapat banyak kekurangan dari skripsi ini, untuk itu penulis mengharapkan kritik dan saran yang membangun. Semoga skripsi ini dapat bermanfaat bagi pihak yang membutuhkan.

Malang, 03 Agustus 2018 Penulis

cusenmosabeth@student.ub.ac.id



ABSTRAK

Cusen Mosabeth. Prediksi Harga Pasar Daging Sapi Di Kota Malang Dengan Menggunakan Metode Extreme Learning Machine (ELM)

Pembimbing: Muhammad Tanzil Furqon, S.Kom., M.CompSc dan Randy Cahya Wihandika, S.Kom., M.Kom

Daging sapi merupakan salah satu kebutuhan pokok yang keberadaannya sangat meningkat di Indonesia. Kebutuhan mengkonsumsi daging sapi sangat tajam sebanding dengan peningkatan jumlah penduduk dan kesadaran masyarakat pentingnya mengkonsumsi makanan bergizi sangat tinggi. Pada dasarnya kebutuhan protein hewani tidak dapat digantikan dengan protein lainnya. Memperkirakan permintaan konsumen dimasa datang dengan membuat perancanaan produksi suatu tantangan bagi suatu industri. Hal ini membuat prediksi memiliki peranan penting. Perancangan yang efektif dan efesien harus didukung oleh sistem prediksi yang akurat. ELM Merupakan jaringan saraf tiruan yang terdiri feed-forward dengan satu atau hidden layer-forwad neural. Oleh karena itu, pada penelitian ini penulis menggunakan metode Extreme Learning Machine (ELM). Hasil uji coba dalam penelitian menujukkan bahwa metode ELM memiliki error yang baik diukur dengan tingkat kesalahan Mean Absolute Percentage Error (MAPE) sebesar 0,344% dengan menggunakan perbandingan jumlah data training 90%: 10%, rentang input weight antara -1 dan 1, jumlah neuron pada hidden layer 7, lalu menggunakan fungsi aktivasi sigmoid biner, dan menggunakan jumlah fitur 3. Hasil tersebut membuktikan dengan menggunakan metode Extreme Learning Machine dapat memprediksi harga daging sapi dengan akurat dan tepat serta mendapatkan harga daging sapi dimasa yang akan datang.

Kata Kunci: Daging Sapi, Prediksi, Extreme Learning Machine.

ABSTRACT

Cusen Mosabeth. Prediksi Harga Pasar Daging Sapi Di Kota Malang Dengan Menggunakan Metode Extreme Learning Machine (ELM)

Pembimbing: Muhammad Tanzil Furqon, S.Kom., M.CompSc dan Randy Cahya Wihandika, S.Kom., M.Kom

Beef is one of the basic needs whose existence is greatly increased in Indonesia. The need to consume beef is very sharp in proportion to the increase in population and the awareness of the importance of consuming very high nutritious foods. Basically the need for animal protein cannot be replaced with other proteins. Estimating future consumer demand by making production plans a challenge for an industry. This makes predictions play an important role. Effective and efficient design must be supported by an accurate prediction system. ELM Is an artificial neural network consisting of feed-forward with one or hidden layer-forwad neural. Therefore, in this study the author uses the Extreme Learning Machine (ELM) method. The experimental results showed that the ELM method had a good error measured by the Mean Absolute Percentage Error (MAPE) error rate of 0.344% using the ratio of the training data 90%: 10%, the input weight range between -1 and 1, the number of neurons in the hidden layer 7, then use the binary sigmoid activation function, and use the number of features 3. The results are proved by using the method of Extreme Learning Machine can predict the price of beef with accurate and precise and get the price of beef in the future.

Keywords: Beef, Prediction, Extreme Learning Machine.



DAFTAR ISI

PENGESAHAN	i
PERNYATAAN ORISINALITAS	ii
KATA PENGANTAR	iv
ABSTRAK	vi
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL	
DAFTAR GAMBAR	xiv
DAFTAR KODE PROGRAM	xv
BAB 1 PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Rumusan masalah	2
1.3 Tujuan	2
1.4 Manfaat	2
1.5 Batasan masalah	
1.6 Sistematika pembahasan	
BAB 2 LANDASAN KEPUSTAKAAN	
2.1 Kajian Pustaka	5
2.2 Daging Sapi	
2.3 Prediksi	
2.4 Jaringan Saraf Tiruan (JST)	11
2.4.1 Faktor Bobot	12
2.4.2 Fungsi Aktivasi	12
2.5 Metode Extreme Learning Machine (ELM)	14
2.5.1 Normalisasi Data	15
2.5.2 Proses Denormalisasi Data	15
2.5.3 Proses <i>Training</i>	15
2.5.4 Proses Testing	17
2.6 Operasi Baris Elementer	18
2.7 Nilai Evaluasi	18
2.7.1 Mean Absolute Percentage Error (MAPE)	19

DAD 3 MIETUDULUGI	20
3.1 Studi Literatur	20
3.1.1 Tipe Penelitian	21
3.2 Pengumpulan Data	21
3.2.1 Teknik Pengumpulan Data	21
3.2.2 Partisipan Penelitian	21
3.3 Analisis Dan Kebutuhan Sistem	22
3.4 Perancangan Sistem	22
3.4.1 Strategi Penelitian	22
3.5 Implementasi Sistem	22
3.5.1 Implementasi Algoritma	22
3.6 Pengujian Sistem	23
3.7 Kesimpulan	23
BAB 4 PERANCANGAN	24
4.1 Deskripsi Masalah	
4.2 Perancangan Jaringan	
4.3 Siklus Metode Extreme Learning Machine (ELM)	26
4.3.1 Proses Extreme Learning Machine (ELM	
4.3.2 Proses Normalisasi	28
4.3.3 Proses <i>Training</i>	31
4.3.4 Proses Testing	47
4.3.5 Proses Denormalisasi	49
4.3.6 Proses Perhitungan Nilai Mean Absolute Percentage Error	
4.4 Perhitungan Manual	
4.4.1 Perhitungan Normalisasi Data	
4.4.2 Inisialisasi Input Weight	56
4.4.3 Inisialisasi Input Bias	57
4.4.4 Perhitungan Proses <i>Training</i>	57
4.4.5 Perhitungan Proses <i>Testing</i>	61
4.5 Perhitungan Nilai Mean Absolute Percentage Error (MAPE)	62
4.6 Perancangan Antar Muka	63
4.6.1 Perancangan Halaman Impor Data	63

	4.6.2 Perancangan Halaman Pembagian Data	64
	4.6.3 Perancangan Halaman <i>Training</i>	65
	4.6.4 Perancangan Halaman <i>Testing</i>	66
	4.6.5 Perancangan Halaman Hasil Evaluasi	67
	4.6.6 Perancangan Halaman K-Fold	68
	4.7 Perancangan Uji Coba dan Evaluasi	69
	4.7.1 Pengujian Input Jumlah Fitur	70
	4.7.2 Pengujian Fungsi Aktivasi	70
	4.7.3 Pengujian Jumlah Hidden Neuron	71
	4.7.4 Pengujian Perbandingan Data Training dan Data Testing	71
	4.7.5 Pengujian Perbandingan Data Menggunakan Cross Validation .	72
BAB 5	IMPLEMENTASI	
	5.1 Spesifikasi Sistem	73
	5.1.1 Spesifikasi Perangkat Keras	
	5.1.2 Spesifikasi Perangkat Lunak	
	5.2 Batasan Implementasi	73
	5.3 Implementasi Program	74
	5.3.1 Implementasi Proses Normalisasi Data	74
	5.3.2 Implementasi Proses Membuat Nilai Random Bobot dan Bias	75
	5.3.3 Implementasi Proses Training	77
	5.3.4 Implementasi Proses Evaluasi	
	5.4 Implementasi User Interface	85
BAB 6	PENGUJIAN DAN ANALISIS	88
	6.1 Hasil dan Analisis Uji Coba Variasi Fitur Data	88
	6.2 Hasil dan Analisis Uji Coba Jumlah Neuron	89
	6.3 Hasil dan Analisis Uji Coba Fungsi Aktivasi	91
	6.4 Hasil dan Analisis Uji Coba Jumlah Rasio Percentage Data Training of Data Testing	
	6.5 Hasil dan Analisis Uji Coba Jumlah Data Training dan Testing deng	_
BAB 7	PENUTUP	97
	7.1 Kesimpulan	97
	7.2 Saran	97

DAFTAR PUSTAKA......98



DAFTAR TABEL

Tabel 2.1 Kajian Pustaka	6
Tabel 4.1 Data Harga Daging Sapi	. 24
Tabel 4.2 Data <i>Training</i> dan Data <i>Testing</i>	. 55
Tabel 4.3 Nilai Maksimal dan Minimal	. 55
Tabel 4.4 Normalisasi Data	. 56
Tabel 4.5 Inisialisasi Input Weight	. 56
Tabel 4.6 Inisialisasi <i>Input Weight</i> Setelah di <i>Transpose</i>	. 57
Tabel 4.7 Inisialisasi Input Bias	
Tabel 4.8 Matriks Keluaran <i>Hidden Layer</i>	. 58
Tabel 4.9 Matriks Keluaran <i>Layer</i> Menggunakan Fungsi Aktivasi	. 58
Tabel 4.10 <i>Transpose</i> Matriks Keluaran <i>Hidden Layer</i> dengan Fungsi Aktivasi	. 59
Tabel 4.11 Perkalian Hasil <i>Transpose</i> dengan Keluaran <i>Hidden Lo</i> Menggunakan Fungsi Aktivasi	. 59
Tabel 4.12 Inverse Matriks	
Tabel 4.13 Matriks Moore-Penrose Generalized Inverse	
Tabel 4.14 Nilai Output Weight	. 60
Tabel 4.15 Matriks Keluaran Hidden Layer	. 61
Tabel 4.16 Matriks Keluaran Hidden Layer Menggunakan Fungsi Aktivasi	. 61
Tabel 4.17 Hasil Keluaran pada <i>Output Layer</i>	
Tabel 4.18 Denormalisasi Data	. 62
Tabel 4.19 Hasil Keluaran Nilai MAPE Proses Testing	. 62
Table 4.20 Rancangan Pengujian Fitur Data	. 70
Table 4.21 Rancangan Pengujian Fungsi Aktivasi	. 70
Table 4.22 Rancangan Pengujian Jumlah Neuron	. 71
Table 4.23 Pengujian Perbandingan Data <i>Training</i> dan Data <i>Testing</i>	. 72
Table 4.24 Pengujian Prediksi Harga Menggunakan K-Fold	. 72
Tabel 5.1 Lingkungan Perangkat Keras	. 73
Tabel 5.2 Lingkungan Perangkat Lunak	. 73
Tabel 6.1 Tabel Nilai MAPE Variasi Fitur Data	. 88
Tabel 6.2 Tabel Nilai MAPE Jumlah Neuron	. 90
Tabel 6.3 Tabel Nilai MAPE Variasi Fungsi Aktivasi	. 91

Tabel 6.4 Tabel Nilai MAPE Perbandingan Jumlah Data <i>Training</i> dan Data <i>Tes</i> i	ting
	. 93
Table 6.5 Tabel Nilai Prediksi Harga Dengan K-Fold	
Table 6.6 Tabel Nilai MAPE Perbandingan Cross Validation	. 95



DAFTAR GAMBAR

Gambar 2.1 Struktur Jaringan Saraf Tiruan	. 12
Gambar 2.2 Ilustrasi Fungsi Aktivasi Sigmoid	. 13
Gambar 2.3 Struktur Metode ELM	. 14
Gambar 3.1 Tahapan Penelitian	. 20
Gambar 4.1 Arsitektur Perancangan Sistem	. 26
Gambar 4.2 Diagram Alir Proses Extreme Learning Machine	. 27
Gambar 4.3 Diagram Alir Proses Normalisasi	. 29
Gambar 4.4 Diagram Alir Proses Input Weight dan Bias	. 31
Gambar 4.5 Diagram Alir Proses <i>Training</i>	. 32
Gambar 4.6 Diagram Alir Proses hidden layer (Hinit)	. 34
Gambar 4.7 Diagram Alir Keluaran Hidden Layer dengan Fungsi Aktivasi	. 35
Gambar 4.8 Diagram Alir Fungsi Aktivasi Sigmoid Biner	. 36
Gambar 4.9 Diagram Alir Fungsi Aktivasi Sigmoid Bipolar	. 37
Gambar 4.10 Diagram Alir Fungsi Fungsi Aktivasi Sin	. 38
Gambar 4.11 Diagram Alir Fungsi Aktivasi Radial Basis	. 39
Gambar 4.12 Diagram Alir Perkalian <i>Hidden Layer</i>	
Gambar 4.13 Diagram Alir Perhitungan <i>Inverse Matriks</i>	. 43
Gambar 4.14 Diagram Alir Moore-Penrose Generalized Inverse (H+)	. 45
Gambar 4.15 Diagram Alir Bobot Akhir	. 46
Gambar 4.16 Diagram Alir Proses Testing	. 47
Gambar 4.17 Diagram Alir <i>Output</i> Prediksi	. 49
Gambar 4.18 Diagram Alir Denormalisasi Data	. 50
Gambar 4.19 Diagram Alir Evaluasi Hasil	. 52
Gambar 4.20 Diagram Alir Proses Mencari Jarak	. 53
Gambar 4.21 Diagram Alir Proses Menjadi Persentasi	. 54
Gambar 4.22 Perancangan Halaman <i>Impor</i> Data	. 63
Gambar 4.23 Perancangan Halaman Bobot	. 65
Gambar 4.24 Perancangan Halaman Proses <i>Training</i>	. 66
Gambar 4.25 Perancangan Halaman Proses <i>Testing</i>	. 67
Gambar 4.26 Perancangan Halaman Hasil Evaluasi	. 68
Gambar 4.27 Perancangan Halaman K-Fold	. 69

Gambar 5.1 Implementasi Halaman <i>Load</i> Datasets85
Gambar 5.2 Implementasi Halaman Pembagian Data86
Gambar 5.3 Implementasi Halaman Proses <i>Training</i>
Gambar 5.4 Implementasi Halaman Proses <i>Testing</i> 87
Sambar 5.5 Implementasi Halaman Hasil <i>Evaluasi</i>
Sambar 6.1 Grafik Niai MAPE Variasi Fitur Data89
Gambar 6.2 Grafik Nilai MAPE Jumlah <i>Neuron</i> 90
Gambar 6.3 Grafik Nilai MAPE Pengujian Variasi Fungsi Aktivasi
Gambar 6.4 Grafik Nilai MAPE Pengujian Perbandingan Jumlah Data <i>Training</i> dar Data <i>Testing</i>
Gambar 6.5 Grafik Nilai MAPE Pengujian Perbandingan Harga Aktual dan Harga Prediksi Menggunakan <i>Cross Validation</i>
Gambar 6.6 Grafik Nilai MAPE Pengujian Perbandingan Jumlah Data <i>Training</i> dar Data <i>Testing</i> dengan <i>Cross Validation</i>



DAFTAR KODE PROGRAM

Kode Program 5.1 Proses Normalisasi Data	75
Kode Program 5.2 Proses Membuat Nilai <i>Random</i> Bobot dan Bias	77
Kode Program 5.3 Proses Perhitungan Hinit	77
Kode Program 5.4 Proses Perhitungan Hidden Layer	79
Kode Program 5.5 Proses Perhitungan <i>Inverse</i> Matriks Menggunakan OBE	80
${\sf Kode\ Program\ 5.6\ Proses\ Perhitungan\ Matriks\ } \textit{Moore-Penrose\ Pseudo\ Inverse}\ .$	82
Kode Program 5.7 Proses Perhitungan Output Weight	82
Kode Program 5.8 Proses Perhitungan <i>Output</i> Prediksi	83
Kode Program 5.9 Proses Denormalisasi Data	84
Kode Program 5.10 Proses Mencari Jarak MAPE	84
Kode Program 5.11 Proses Percentage MAPE	85



BAB 1 PENDAHULUAN

Bab ini menjelaskan tentang latar belakang dari penelitian, dari rumusan yang dibahas, tujuan serta manfaat yang ingin dicapai, batasan masalah yang ditentukan dan sistematika pembahasan dari penelitian tentang implementasi metode Extreme Learning Machine untuk menujukan tingkat kesalahan (Mean Absolute Percentage Error) MAPE yang paling baik.

1.1 Latar belakang

Di Indonesia, daging sapi merupakan salah satu yang banyak diminati untuk kebutuhan pokok sehari-hari oleh kalangan masyarakat. Populasi penduduk dan taraf hidup masyarakat di Kota Malang mengalami perubahan sangat meningkat setiap tahunnya. Hal ini dengan kesadaran masyarakat bahwa konsumsi makanan rumah tangga sehat secara bertahap mengalami perubahan ke konsumsi makanan yang berprotein. Produsen dituntut untuk membuat sebuah strategi penjualan yang dapat membantu konsumen dalam menentukan harga yang baik dan bersaing dengan produsen daging sapi lainnya.

Selama periode tahun 1987-1996 dengan nilai percepatan konsumsi daging sebesar 7,36% dalam setiap tahunnya (Ditjen Peternakan, 1997). Untuk kontribusi daging sapi (21,27%) masuk dalam peringkat kedua setelah daging unggas. Permintaan setiap konsumen terhadap daging sapi selalu berubah dan tak dapat diprediksi setiap waktu. Harga daging sapi yang berubah-ubah setiap waktu dan dipengaruhi oleh banyak faktor yang menjadikan daging sapi dapat diperkirakan dalam setiap bulan bahwa harga mengalami turun atau naik pada harga daging sapi tersebut.

Harga pasar merupakan yang tidak dapat diperkirakan mengalami kenaikan atau fluktuatif, ada kalanya harga naik dan mengalami harga turun. Hal ini terjadi karena, apabila harga pasar naik tajam dari harga pada bulan-bulan sebelumnya. Untuk menjaga harga yang stabil maka harga pasar daging sapi adalah tugas dari Dinas Perdagangan Kota Malang. Tindakan nyata ini agar menjaga harga daging sapi agar stabil dengan memantau harga pasaran komoditas daging sapi bulanannya. Tindakan preventif yang dilakukan untuk mencegah terjadinya anomali harga. Maka dari itu, perlu proyeksi kondisi pasar untuk melakukan suatu prediksi harga pasar daging sapi untuk periode yang datang dengan data pasar dari data perbulannya.

Banyak penelitian yang menggunakan metode untuk mendapatkan nilai prediksi yang akurat. Pada metode yang dapat memprediksi yaitu Support Vector Machine, Backprogation dan Salah satu metode yang digunakan dalam penelitian ini merupakan metode Extreme Learning Machine (ELM). ELM diperkenalkan pertama kali oleh Huang pada tahun 2004 (Huang, et al., 2004). ELM merupakan jaringan saraf tiruan feed-forward dengan hidden layer atau lebih dikenal dengan single hidden layer feed-forward neural networks (SLFNs). Metode Extreme Learning Machine ini memiliki kelebihan dalam learning speed serta mempunyai

akurasi yang baik dibandingkan dengan metode konvensional seperti *Exponential Smooting* dan *Moving Averge*. ELM mampu menghasilkan nilai prediksi yang lebih efektif dan akurat.

Terdapat beberapa dari penelitian yang membahas mengenai tentang prediksi dengan menggunakan Extreme Learning Machine (ELM). Untuk penelitian sebelumnya oleh (Pangaribuan, 2016). Metode yang digunakan yaitu ELM. Pada ELM merupakan jaringan saraf tiruan yang terdiri feed-forward dengan satu atau lebih pada hidden layer yang biasanya dengan sering digunakan dengan kata single hiden layer feed-forward neural telah membuktikan bahwa dengan metode ELM ini memberikan hasil akurasi prediksi yang baik dengan melihat tingkat kesalahan Mean Square Error (MSE) dengan hasil akurasi sebesar 0,4036 dan pada hasil yang menggunakan backpropagation sebesar 0,9425 dari hasi ini dapat disimpulkan hasil keselahan yang mendekati 0 ialah hasil dari metode menggunakan ELM yang disebut dengan hasil paling baik. Penelitian kedua dilakukan oleh oleh (Giusti, et al., 2018). Prediksi penjualan untuk dapat memprioritaskan pembelian bahan baku tertentu yang memiliki dengan tingkatan keminatan yang tinggi sehingga sisa bahan baku dapat berkurang. Penelitian ini melakukan pengujian dan menghasilkan suatu tingkat error yang baik atau terkecil dengan memasukkan 2 fitur data yaitu pada data historis dan fitur data sisa penjualan dan mendapatkan nilai error sebesar 0,0171.

1.2 Rumusan masalah

Berdasarkan permasalahan ini, maka dapat merumuskan masalah sebagai berikut:

- 1. Bagaimana memprediksi harga daging sapi dengan menggunakan metode Extreme Learning Machine (ELM)?
- 2. Bagaimana tingkat kesalahan prediksi yang didapatkan pada hasil prediksi harga daging sapi di Kota Malang yang diukur dengan perhitungan *Mean Absolute Percentage Error* (MAPE)?

1.3 Tujuan

- 1. Mendapatkan prediksi harga pasar daging sapi dengan Menerapkan metode *Extreme Learning Machine* (ELM) untuk memperoleh hasil prediksi dengan data yang didapatkan dari Dinas Perdagangan Kota Malang.
- 2. Menguji tingkat error untuk hasil prediksi harga daging sapi menggunakan metode *Extreme Learning Machine* (ELM) yang diukur dengan perhitungan *Mean Absolute Percentage Error* (MAPE).

1.4 Manfaat

Bagi Penulis

1. Agar dapat memahami tentang implementasi metode *Extreme Learning Machine* (ELM) dalam memprediksi indeks harga pasar daging sapi.

2. Dapat menerapkan ilmu yang telah diberikan dalam masa perkuliahan dalam prediksi yang digunakan metode *Extreme Learning Machine* (ELM).

Bagi Pengguna

- 1. Membantu dalam melakukan prediksi harga pasar daging sapi.
- 2. Mendapatkan hasil yang baik dalam memprediksi harga daging sapi yang akurat.

Bagi mahasiswa

- 1. Mahasiswa mengetahui cara menerapkan algoritma *Extreme Learning Machine* untuk menyelesaikan permasalahan tentang prediksi harga daging sapi.
- 2. Mengetahui perbedaan dan kelebihan algoritma *Extreme Learning Machine* dengan algoritma lainnya.

1.5 Batasan masalah

 Data harga pasar daging sapi yang digunakan, didapatkan dari Bidang Pertenakan dan Kesehatan Hewan Dinas Pertanian dan Ketahanan Pangan Kota Malang. Dalam penelitian ini ialah jumlah penjualan setiap bulan dari rentang waktu tahun 2012 sampai 2017.

TAS BA

- 2. Fungsi aktivasi yang digunakan dalam peneitian berikut ialah fungsi aktivasi sigmoid biner, fungsi aktivasi sigmoid bipolar, fungsi sin, dan fungsi radial basis.
- 3. Hasil akhir prediksi harga daging sapi di Kota Malang ini tidak dibandingkan dengan metode prediksi lainnya.
- 4. Perhitungan nilai kesalahan prediksi pada penelitian ini menggunakan *Mean Absolute Percentage Error* (MAPE).

1.6 Sistematika pembahasan

Sistematika pembahasan bagian ini menjelaskan tentang struktur skripsi dan gambaran singkat dari masing-masng bab. Pada penelitian ini terdiri dari Bab Pendahuluan, Bab Landasan Kepustakaan, Bab Metodologi, Bab Perancangan, Bab Hasil dan Pembahasan, dan Bab Penutup Sebagai berikut:

BAB 1 PENDAHULUAN

Bab ini menjelaskan tentang latar belakang dari penelitian, rumusan masalah yang dibahas, tujuan serta manfaat yang ingin dicapai dalam penelitian, batasan masalah yang ditentukan dan membahas sistematika dari penelitian tentang implementasi algoritma *Extreme Learning Machine* untuk mencari harga pasar daging sapi di Kota Malang.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini menjelaskan tentang kajian pustaka yang menjelaskan tentang penelitian yang telah dilakukan sebelumnya dan dasar teori yang diperlukan untuk mendukung penelitian ini.

BAB 3 METODOLOGI

Bab ini menjelaskan yang berisi tentang langkah-langkah yang dilakukan dalam penelitian yang meliputi studi literatur, pengumpulan data, analisis dan kebutuhan sistem, perancangan sistem, implementasi sistem, pengujian sistem, serta kesimpulan hasil dari penelitian yang telah dilakukan.

BAB 4 PERANCANGAN

Bab ini menjelaskan yang berisi tentang perancangan yang dilakukan dalam penelitian ini. Untuk bab ini membahas tentang siklus metode *Extreme Learning Machine* (ELM), perhitungan manual, perancangan antarmuka dan perancangan pengujian.

BAB 5 IMPLEMENTASI

Bab ini menjelaskan tentang pengujian sistem yang telah diimplementasikan dan dilakukan analisis dari hasil yang telah didapatkan dari pengujian tersebut.

BAB 6 PENGUJIAN DAN ANALISIS

Bab ini menjelaskan tentang tentang pengujian sistem yang telah diimplementasikan dan dilakukan analisis dari hasil pengujian penelitian yang telah dilakukan.

BAB 7 PENUTUP

Bab ini menjelaskan yang berisi tentang kesimpulan dari hasil implementasi dan pengujian sistem yang telah diakukan serta saran untuk dapat menghasilkan sistem yang lebih baik lagi dari rumusan masalah pada bab pendahuluan.

BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini dijelaskan kajian pustaka dan dasar teori untuk mendukung penelitian ini. Dasar teori ini didapatkan dari beberapa referensi yang relevan dengan topik yang diangkat dalam penelitian ini. Hal ini menjelaskan kajian pustaka dan metode *Extreme Learning Machine* (ELM).

2.1 Kajian Pustaka

Kajian pertama adalah penelitian yang telah dilakukan sebelumnya oleh (Siwi, et al., 2016). Dengan menggunakan metode *Extreme Learning Machine* ini untuk melakukan prediksi suatu produksi gula pasir dalam menangani dalam masalah pengambilan keputusan untuk melakukan produksi oleh karyawan. Telah didapatkan dari penelitian ini dengan menggunakan metode *Extreme Learning Machine* ini mendapatkan suatu nilai *error* MAPE yang kecil, hal ini dapat dikatakan baik yaitu sebesar 3,66%.

Kajian kedua adalah penelitian membahas tentang prediksi dengan menggunakan metode ELM oleh (Giusti, et al., 2018). Berdasarkan dari penelitian ini melakukan prediksi penjualan untuk dapat memprioritaskan pembelian bahan baku tertentu yang memiliki dengan tingkatan keminatannya yang tinggi sehingga sisa bahan baku dapat berkurang. Penelitian ini melakukan pengujian dan menghasilkan suatu tingkat *error* yang baik atau terkecil dengan memasukkan 2 fitur data yaitu pada data historis dan fitur data sisa penjualan dan mendapatkan nilai *error* sebesar 0,0171%.

Kajian ketiga adalah penelitian lainnya masih berhubungan dengan topik prediksi oleh (Pangaribuan, 2016). Metode yang digunakan yaitu ELM, pada ELM merupakan jaringan saraf tiruan yang terdiri *feed-forward* dengan satu atau lebih pada hidden layer yang biasanya dengan sering digunakan dengan kata *single hiden layer feed-forward neural*. Telah membuktikan bahwa dengan metode ELM ini memberikan hasil akurasi prediksi yang baik dengan melihat tingkat kesalahan *Mean Square Error* (MSE) dengan hasil akurasi sebesar 0,4036 dan pada hasil yang menggunakan backpropagation sebesar 0,9425. dari hasil ELM dan *backpropagation* ini dapat disimpulkan hasil keselahan yang mendekati 0 ialah hasil yang disebut dengan hasil paling baik.

Kajian keempat adalah penelitian ini masih menggunakan metode yang sama dengan yang sebelumnya oleh (Fardani, et al., 2015). Dalam penelitian ini memiliki tujuan untuk merancang dan membangun suatu sistem pendukung keputusan dalam meramalkan banyaknya pasien yang berkunjung ke poli Gigi RSU DR. Wahidin Sudiro Husodo Mojokerto. Dengan mengambil prediksi ini mengharapkan dapat menemukan suatu kemudahan dalam kerja pada RSU sendiri. Penelitian ini menggunakan data berupa jumlah kunjungan pasien untuk setiap harinya. Nilai MSE yang didapatkan pada penelitian ini dengan menggunakan hidden layer sejumlah 7 didapatkan nilai terkecil yaitu 0,027.

Kajian kelima adalah penelitian yang lainnya yaitu masih topik prediksi, yaitu oleh (Chandar, et al., 2016). Hal ini dapat memprediksi harga emas yang akurat karena menggunakan metode *Extreme Learning Machine* dan memiliki tujuan dalam memprediksi harga emas masa depan. Dengan jumlah data 1 januari 2000 sampai 31 April 2014 mendapatkan nilai akurasi untuk data pelatihan mendapatkan 93,83% dan untuk data pengujian mendapatkan 93,82%. Penelitian ini membandingkan ketiga model yaitu *Feed forward networks without feedback, Feed-forward backpropagation networks and* ELM learning model yang mencapai sekitar 3% peningkatan dalam pengujian dan efesiensi pengujian dibanding lainnya.

Kajian keenam adalah penelitian selanjutnya iyalah dengan judul topik predksi oleh (Yaseen, et al., 2016). Memprediksi arus aliran air pada studi kasus daerah semi kering dilraq. Pada penelitian ini melakukan perbandingan antara metode Extreme Learning Machine (ELM), Support Vector Regression (SVR), dan Generalized Regression Neural Network (GRNN). Hasil yang didapatkan adalah nilai Root Mean Squared Error (RMSE) dan MAE (Mean Absolute Error) terkecil diperoleh metode ELM sebesar 87,906 untuk RMSE dan 71,544 untuk MAE.

Kajian ketujuh adalah penelitian penulis untuk melakukan suatu prediksi yang bertujuan memprediksi harga daging sapi diKota Malang dengan menggunakan metode Extreme Learning Machine (ELM). Dalam prediksi ini menggunakan 5 fitur yaitu fitur harga daging sapi yang terdiri dari data harga setiap bulan dalam 1 tahun dengan hasil data yang didapatkan dari tahun 2012-2017. Karena itu, penelitian ini sebelumnya menunjukan dengan metode Extreme Learning Machine (ELM) untuk mendapatkan tingkat kesalahan yang kecil, maka sangat cocok dalam memprediksi harga daging sapi yang membantu dalam mempersiapkan apabila terjadi kenaikan harga dalam waktu dekat.

Tabel 2.1 Kajian Pustaka

No	Judul	Nama Penulis	Masukkan	Metode	Hasil
1.	Prediksi Produksi Gula Pasir Menggunak an Extreme Learning Machine (ELM) pada PG Candi Baru Sidoarjo	Siwi, et al., (2016)	Data yang digunakan merupakan gula pasir dan jumlah hidden layer Nilai Bobot Fitur yang digunakan Fungsi Aktivasi Pembagian Data Training dan Testing	Extreme Learning Machine (ELM)	Pada Metode Extreme Learning Machine (ELM) didapatkan nilai Error MAPE yang baik yaitu dengan menghasilkan sebesar 3.66%

2.	Prediksi Penjualan Mi Menggunak an Metode Extreme Learning Machine (ELM) diKober Mie Setan Cabang Soekarno Hatta	Giusti et al., (2018)	 Data penjualan dan sisa para pembeli hidden layer Nilai Bobot Fitur yang digunakan Fungsi Aktivasi Pembagian Data Training dan Testing 	Extreme Learning Machine (ELM)	Metode ini menggunakan ELM yang berhasil mendapatkan nilai Error MAPE yang baik dengan nilai terkecil yaitu 0.0171. Hal ini didapatkan MSE yang terkecil ini pada jumlah fitur sebanyak 7.
3.	Mendiagnos is Penyakit Diabetes Melitus Dengan Menggunak an Metode Extreme Learning Machine	Pangaribuan ,(2016)	 Data yang digunakan berupa data jumah pasian yang terkena Diabetes Nilai Bobot Nilai Bias Fungsi Aktivasi 	Extreme Learning Machine (ELM)	Metode ini digunakan yaitu Extreme Learning Machine yang dimana pada dataset terdiri dari 769 populasi wanita. Telah mendapatkan hasil tingkat kesalahan Mean Square Error (MSE) untuk ELM sebesar 0.4036 dan pada pengujian Backproagatio n sebesar 0.9425.
4.	Sistem Pendukung Keputusan Prediksi Jumlah	Fardani, et al.,(2015)	Data Jumlah Kunjungan PasienNilai bobot	Extreme Learning Machine (ELM)	Metode ini menggunakan Extreme Learning Machine (ELM)

	Kunjungan Pasien Menggunak an Metode Extreme Learning Machine (Studi Kasus : Poli Gigi RSU DR. Wahidin Sudiro Husodo Mojokerto).		Nilai biasFungsi aktivasi		yang telah mendapatkan suatu nilai error MSE yang kecil yaitu 0.027 dengan menggunakan hidden layer sejumlah 7.
5.	Forecasting Gold Prices Based on Extreme Learning Machine	Chandar, et al.,(2016)	 Data harga emas dari data mulai 1 januari 2000 sampai 31 april 2014 hidden layer Nilai Bobot Fitur yang digunakan Fungsi Aktivasi 	Extreme Learning Machine (ELM)	Dalam metode ini menggunakan metode ELM yang melakukan prediksi harga emas untuk masa depan. Dengan menggunakan metode ini mendapatkan akurasi dan akurat yang baik yaitu 97,5% dengan data latih dan 93,85% pengujian.
6.	Stream- Flow Forecasting Using Extreme Learning Machine: A Case Study In A Semi-	(Yaseen, et al., 2016)	 Data arus aliran air. Input Layer Pattern Layer Summation Layer Ouput Layer 	Extreme Learning Machine (ELM)	prediksi arus aliran air pada studi kasus daerah semi kering dilraq. Penelitian ini melakukan perbandingan antara metode ELM, SVR, dan

7	Arid Region In Iraq	(Mosabeth,	AS BRA	Extreme	GRNN (Generalized Regression Neural Network). Hasil adalah nilai RMSE (Root Mean Squared Error) dan MAE (Mean Absolute Error) terkecil diperoleh metode ELM sebesar 87,906 untuk RMSE dan 71,544 untuk MAE. Metode ini
	Harga Daging Sapi Di Kota Malang Menggunak an Metode Extreme Learning Machine	(Mosabeth, 2018)	 Data yang digunakan ini yaitu harga setiap bulan dalam 5 tahun belakang yang telah dinormalisas ikan, nilai bobot, nilai bias dan fungsi aktivasi. Nilai Bobot Nilai Error/bias Fungsi Aktivasi Nilai Error/MSE Nilai Denormalisa i 	Learning Machine (ELM)	menggunakan metode Extreme Learning Machine yang menghasilkan prediksi harga daging sapi kedepan yang lebih akurant dan mendapatkan nilai error MSE yang kecil.

2.2 Daging Sapi

Perkembangan untuk daging sapi saat ini sangat meningkat tajam sebanding dengan peningkatan jumlah penduduk, dalam kesadaran masyarakat pentingnya konsumsi makanan bergizi tinggi. Daging sapi merupakan salah satu dari sekian banyak sumber makanan yang mengandung protein hewani. Semakin meningkatnya kesadaran masyarakat untuk mengkonsumsi makanan bergizi, maka mengakibatkan meningkatnya permintaan pangan berprotein hewani. Hal ini, pada dasarnya kebutuhan protein hewani tidak dapat digantikan dengan protein lainnya.

Data konsumsi daging sapi penduduk Indonesia di tahun 2016 sebesar Rp.460.639.000 kg/kapita/tahun dan mengalami peningkatan 48,68% bila dibandingkan dengan tahun 2015, rata-rata yang dihasilkan mengeluarkan per kapita sebulan penduduk Indonesia sebesar 11,68%. Sedangkan dalam segi produksi daging sapi secara nasional menunjukkan bahwa produksi daging sapi masih fluktuasi. Tahun 2009 produksi sebesar 409.308 ton, tahun 2010 sebesar 436.450 ton, tahun 2011 sebesar 485.335 ton, tahun 2012 sebesar 508.905 ton, tahun 2013 sebesar 504.819 ton, tahun 2014 sebesar 497.669 ton, tahun 2015 sebesar 506.661 ton, dan tahun 2016 sebesar 524.109 ton (Direktorat Jendral Peternakan, 2011).

Data statistic menunjukkan dari produksi daging sapi di Kota Malang dari tahun 2009 sampai 2016 mengalami kenaikan. Produksi daging sapi tertinggi terjadi tahun 2009 sebesar 409.308 kg/tahun dan tertinggi tahun 2016 yaitu 524.109 kg/tahun (BPS Kota Malang, 2009).

2.3 Prediksi

Dalam menyelesaikan permasalahan dimasa yang datang yang tidak dapat dipastikan orang biasanya menyelesaikannya dengan model pendekatan. Model pendekatan yang dikatakan yang digunakan adalah dengan melakukan pendekatan yang sesuai dengan suatu perilaku ataupun pola data yang aktual, dengan begitu untuk melakukan prediksi. Prediksi memerlukan untuk memperkirakan apa yang terjadi dimasa yang datang, maka dengan dapat memperkirakan para pengambil keputusan bisa membuat perencanaan untuk masa depan. Prediksi merupakan suatu prediksi membuat perkiraan suatu nilai pada periode tertentu yang terjadi dimasa mendatang dengan menggunakan penjelasan secara matematik dan statistik. Data yang digunakan untuk melakukan prediksi ialah data yang televan yang dapat dijelaskan secara statistik maupun matematik, dan biasanya berupa data yang digunakan lalu maupun saat ini (Mendome, et al., 2016)

Prediksi yang ini dibedakan secara umum dari beberapa segi, tergantung dalam sudut pandangnya antara lain:

1. Prediksi berdasarkan metode

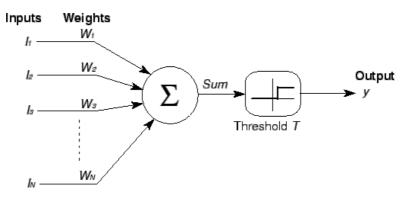
- Prediksi kualitatif merupakan prediksi yang didasarkan atas data kualitatif (non-numerik) yang terjadi dimasa lalu. Prediksi seperti ini merupakan prediksi yang bersifat subyektif yaitu hasil bergantung pada opini, pengalaman, dan pengetahuan, dan pengetahuan orang yang mengambil keputusan.
- 2. Prediksi kuantitatif merupakan prediksi yang didasarkan pada data kuantitatif (numerik) masa lalu. Prediksi biasanya merupakan prediksi yang bersifat obyektif. Karena, hasil prediksinya bergantung pada metode yang digunakan. Metode prediksi kuantitatif dibedakan menjadi dua jenis, yaitu:
 - 1. Metode Time Series ialah metode yang digunakan untuk analisis variable data dengan variasi waktu, yang merupakan deret waktu.
 - 2. Metode Causal ialah metode yang berdasarka analisis pola hubungan sebab akibat antara variable yang diprediksikan dengan variabel yang mempengaruhi.

2. Prediksi berdasarkan jangka waktu

- a. Prediksi jangka pendek merupakan prediksi yang dilakukan untuk jangka yang kurang dalam satu tahun. Prediksi seperti ini yang sering digunakan dalam prediksi permintaan konsumen, prediksi curah hujan, penjadwalan kerja, tingkat produktivitas.
- b. Prediksi jangka menengah merupakan peralaman yang dilakukan untuk jangka waktu rentang tiga bulan sampai dengan tiga tahun. Prediksi seperti ini memerlukan untuk perencaan anggaran modal dan perencanaan pembelian produk baru.
- c. Prediksi jangka panjang merupakan prediksi yang dilakukan dalam jangka waktu rentang tiga tahun sampai lebih dari tiga tahun. Untuk prediksi seperti ini memerlukan untuk perencaan anggaran modal dan perencanaan pembelian produk baru.

2.4 Jaringan Saraf Tiruan (JST)

Jaringan Saraf Tiruan (JST) suatu metode pengelompokan dan pemisah data yang pada prinsip kerjanya sama dengan *neural network* pada manusia, seperti yang sudah diketahui bahwa otak manusia selalu mensimulasikan proses pembelajaran. Hal ini JST bekerja seperti otak manusia yang memiliki *neuron* yang selalu berhubungan satu dengan yang lainnya. *Neuron* berikut menstransfer informasi-informasi yang diterima oleh satu *neuron* ke *neuron* yang lainnya. Pada informasi yang dibawa oleh *neuron* ini disimpan pada suatu nilai tertentu yang disebut dengan bobot. Untuk struktur pada node atau *neuron* di jaringan saraf tiruan ini ditunjukkan pada Gambar 2.1



Gambar 2.1 Struktur Jaringan Saraf Tiruan

Sumber: (Humaini, 2015)

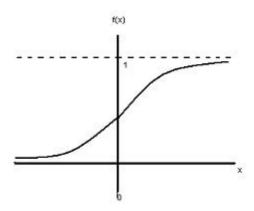
pada struktur jaringan saraf tiruan ini nilai masukan sebanyak n masukan tersebut yaitu nilai $X_i = X_1, X_2, \ldots, X_n$. Maka masukan tersebut diproses dengan bobot $W_i = W_1, W_2, \ldots, W_n$. Semua yang dimasukkan ini diolah *neuron* menjadi $X_i^1 = X_i, W_i, l = 1, 2, \ldots, n$. Lalu *neuron* menjumlahkan semua sinyal masukan dan dibandingkan dengan suatu nilai ambang (threshold) tertentu melalui fungsi pada aktivasi pada setiap *neuron*. Untuk dapat meneruskan informasi yang didapatkan, maka nilai pada *input* ini harus dapat melewati nilai ambang tertentu sehingga nilai pada *input* dapat diaktifkan dan dapat dilanjutkan menjadi output yang dikirimkan kepada *neuron* yang lain. (Sihuhaji, 2009).

2.4.1 Faktor Bobot

Bobot merupakan nilai tingkat atau kepentingan dari hubungn antara suatu neuron dengan neuron lainnya. Jadi, besar bobot suatu koneksi, bisa menandakan semakin pentingnya hubungan kedua neuron tersebut. Bobot dapat berupa bilangan real maupun integer, lalu untuk bobot tersebut dapat diinisialisasi untuk berada didalam nilai tertentu. Jaringan Saraf Tiruan, suatu jaringan otomatis memperbaiki bobot secara terus-menerus, karena adanya kemampuan untuk belajar atau pelatihan. Setiap suatu masalah baru atau data baru, jaringan dapat belajar lagi dengan menambahkan masalah/data baru tersebut (Puspitaningrum, 2006).

2.4.2 Fungsi Aktivasi

Fungsi Aktivasi merupakan suatu proses *training* dan proses *testing* untuk digunakan pada *Extreme Learning Machine* (ELM). Berikut beberapa fungsi aktivasi yang digunakan dalam metode jaringan saraf tiruan (Singh & Balasundaram, 2007) yaitu:



Gambar 2.2 Ilustrasi Fungsi Aktivasi Sigmoid

1. Fungsi Aktivasi sigmoid Biner

Fungsi aktivasi sigmoid merupakan suatu nilai yang memiliki rentang keluaran 0 sampai 1. Persamaan 2.1 merupakan rumus aktivasi sigmoid. Lalu dijeaskan pada Gambar 2.2 merupakan ilustrasi fungsi aktivasi sigmoid.

$$H = \frac{1}{1 + \exp(-H_{init})} \tag{2.1}$$

keterangan:

H = fungsi aktivasi sigmoid

 $\frac{1}{1 + exp^{(-Hinit)}} = \text{Eksponensial dengan pangkat minus data x}$

2. Fungsi Aktivasi sigmoid Bipolar

Fungsi Aktivasi Biner untuk memproses suatu *output* jaringan, sehingga sesuai dengan nilai target. Dalam jaringan saraf tiruan memiliki dua fungsi aktivasi yaitu fungsi sigmoid biner dan fungsi sigmoid *bipolar*. Untuk fungsi-fungsi aktivasi ini harus bersifat *continue*, *direfensibel*, dan tidak menurun secara monoton. Lalu, untuk fungsi aktivasi juga diharapkan mendekati nilai-nilai maksimum dan minimum dengan baik (Puspaningrum, 2006). Untuk menghitung fungsi aktivasi sigmoid biner dapat menggunakan Persamaan 2.9 (Jahuri et al., 2016).

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

Keterangan

H= Fungsi aktivasi Sigmoid

 H_{init} = nilai output jaringan pada data ke-j sebelum dilakukan proses aktivasi

3. Fungsi Aktivasi sin

Fungsi ini ialah rentang nilai untuk aktivasi ini yaitu antara -1 sampai 1. Persamaan untuk aktivasi sin ditunjukkan pada Persamaan (2.3).

$$H = Sin(H_{init}) (2.3)$$

Keterangan

H = merupakan fungsi aktivasi sin

 $\sin x = \sin \text{data } x$

4. Fungsi Aktivasi Radial Basis

Fungsi ini merupakan suatu jaringan lapisan tunggal yang dimana untuk digunakan mengkonversi nilai masukkan bernilai kontinyu menjadi suatu nilai keluaran biner (0 sampai 1). Berikut persamaan yang menggambarkan untuk fungsi aktivasi Radial Basis:

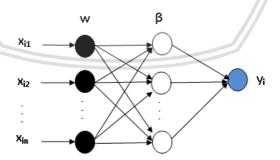
$$H = exp(-((H_{init})^2)$$
 (2.4)

Dalam penelitian ini penulis hanya menggunakan satu fungsi saja aktivasi yaitu fungsi aktivasi sigmoid ((Singh & Balasundaram, 2007).

2.5 Metode Extreme Learning Machine (ELM)

Extreme Learning Machine (ELM) yang dikenalkan oleh (Huang, et al., 2004). Metode ELM ini metode yang dikenal baru dari Jaringan Saraf Tiruan. ELM ini suatu metode pengembangan dari jaringan saraf tiruan feed-forward sederhana dengan menggunakan satu hidden layer atau biasanya dikenal dengan Single Hidden Layer Feed-forward Neural Networks (SLFNs).

Pada Layer Feed-forward parameter yang digunakan ditentukan secara manual seperti input weight dan bias. Dalam input weight dan bias ini selalu dibangkitkan secara acak dalam suatu rentang tertentu. Dengan nilai yang diacak tersebut, hal ini untuk menghindari hasil prediksi yang tidak stabil. Secara struktur ini umumnya ELM ditunjukkan pada Gambar 2.3.



Gambar 2.3 Struktur Metode ELM

Sumber: (Abadi & Soeprijanto, 2014)

Dalam struktur Extreme Learning Machine ini bahwa dijelaskan suatu proses yang mana data dimasukan diolah dengan nilai input weight dan bias kemudian diproses didalam setiap neuron dan pada tahap terakhir adalah output layer yang merupakan hasil dari proses ELM.

Maka setiap langkah perhitungan dengan menggunakan metode ELM ialah pertama adalah melakukan normalisasi data kemudian data yang dilakukan suatu proses untuk langkah proses *training* dan proses *testing*.

2.5.1 Normalisasi Data

Normalisasi data merupakan suatu nilai yang range nilainya input tidak sama, yaitu bernilai puluhan hingga ribuan. Untuk input ini dilakukan suatu proses ke nilai output yang kecil sehingga data yang digunakan harus disesuaikan agar dapat diproses untuk didapatkan nilai normalisasi yang baik atau kecil. Maka untuk penelitian ini, data yang mau digunakan dapat disesuaikan dengan cara dinormalisasikan datanya. Maka hal ini merupakan proses normalisasi data menggunakan metode Min-Max Normalization (Jain & Bhandare, 2011).

$$y' = \frac{x - min}{max - min} \tag{2.5}$$

Keterangan:

d' = nilai dari hasil normalisasi data

d = nilai asli data

min = nilai minimal pada dataset fitur x

max = nilai maksimal pada dataset fitur x

2.5.2 Proses Denormalisasi Data

Proses Denormalisasi Data ini yang membangkitkan nilai yang sudah dinormalisasi menjadi nilai asli. Dari persamaan ini merupakan untuk proses denormalisasi data:

$$d = d'(max - min) + min \tag{2.6}$$

Keterangan:

d' =nilai hasil dari prediksi sebelum didenormalisasi

d = nilai asli dari yang didapatkan setelah didenormalisasi

min = nilai minimum dari dataset fitur x

max = nilai maksimal dai dataset fitur x

2.5.3 Proses *Training*

Proses *training* ini bahwa hal ini harus dilalui sebelum melakukan proses prediksi. Proses ini bertujuan untuk mendapatkan nilai *output wight*. Berikut ini langkah-langkah dalam proses *training* yaitu (Huang, et al., 2006):

1. Langkah yang pertama ialah menginisialisasi *input weight* dan *bias*. Nilai ini diinisiliasasi secara acak atau *random* dengan *range* antara -1 hingga 1. Lalu untuk nilai *random* pada matriks bias *b* dengan rentang [0,1] dalam ukuran 1 x (jumlah banyak *hidden neuron*).

2. Keluaran di hidden layer dihitung menggunakan fungsi aktivasi. Pada langkah pertama ini dihitung keluaran hidden layer (H_{init}), lalu nilai H_{init} didapatkan kemudian dihitung menggunakan fungsi aktivasi sigmoid yang ditunjukkan pada Persamaan 2.1 merupakan persamaan untuk menghitung suatu keluaran hidden layer:

$$H_{init \, ij} = \left(\sum_{k=1}^{n} W_{jk} \, . \, X_{ik}\right) + b_{j} \tag{2.7}$$

3. Menghitung nilai matriks keluaran pada hidden layer dengan menggunakan Persamaan 2.8. Perhitungan b (ones(i_{train} ,1),:) memperbanyak matriks untuk bias sebanyak jumlah data latih.

$$H = 1 + exp(-(-x_{train} W^T + b(ones(i_{train}, 1, :))))$$
 (2.8)

Keterangan:

H = Matriks keluaran hidden layer

 $X_{train} = Matriks input$ pada data latih yang telah dinormalisasikan

 W^T = Matriks *transpose* dari bobot

 $i_{train} = \text{jumlah data latih}$

b = matriks bias

4. Menghitung pada output weight. Hal ini untuk mendapat output weight ini, hal yang pertama yang harus dilakukan transpose matriks hasil keluaran hidden layer dengan fungsi aktivasi. Lalu jika setelah ditranspose, selanjutnya matriks tersebut dikalikan dengan matriks hasil keluaran hidden layer seperti fungsi aktivasi biasanya yang disebut matriks H. setelah itu nilai inverse dihitung dari matriks H tersebut. Jika sudah mendapatkan maka dapat menghitung matriks Moore-Penrose Generalized Inverse dari hasil keluaran hidden layer dengan fungsi aktivasi. Berikut ini merupakan persamaan untuk melakukan perhitungan nilai output weight:

$$\hat{\beta} = H^+ t \tag{2.9}$$

$$H^{+} = (H^{T}H)^{-1}H^{T} {(2.10)}$$

Keterangan:

 $\hat{\beta}$ = Matriks Ouput Weight.

 H^+ = Matriks Moore-Penrose Generalized Inverse dari Matriks H.

t = Matriks Target.

H = Matriks keluaran hidden layer

5. Menghitung hasil keluaran dengan menggunakan Persamaan 2.11.

$$\hat{Y} = H \hat{\beta} \tag{2.11}$$

Keterangan:

 $\hat{Y} = Matriks hasil predisi.$

H = Matriks keluaran hidden layer.

 $\hat{\beta}$ = Matriks Ouput Weight.

2.5.4 Proses Testing

Proses *testing* ini untuk betujuan menghasilkan nilai prediksi dan untuk mengevaluasi metode ELM dari hasil *proses training* yang telah dilakukan sebelumnya. Maka proses *testing* ini dilakukan menggunakan *input weight*, *bias* dan *ouput weight* yang dihasilkan dari proses *training* tersebut. Hal ini merupakan langkah-langkah yang dilakukan proses *testing* sebagai berikut (Huang, et al., 2006):

- 1. Langkah awal yang dilakukan ialah menginisialisasikan *inputan weight, bias* dan $\hat{\beta}$ yang telah didapatkan dari proses *training*.
- 2. Menghitung nilai matriks keluaran pada hidden layer dengan menggunakan Persamaan 2.12. Perhitungan b (ones(i_{train} ,1),:) memperbanyak matriks untuk bias sebanyak jumlah data latih.

$$H = 1 + exp(-(-x_{train} W^T + b(ones(i_{train}, 1,:))))$$
 (2.12)

Keterangan:

H = Matriks keluaran hidden layer

 $X_{train} = Matriks input$ pada data latih yang telah di normalisasikan

 W^T = Matriks *transpose* dari bobot

 $i_{train} =$ jumlah data latih

b = matriks bias

3. Nilai yang dihasilkan *output weight* yang didapatkan pada *proses training* ini digunakan untuk menghitung keluaran *output layer* yang merupakan hasil dari prediksi. Berikut ini merupakan Persamaan 2.13 untuk menghitung nilai output layer:

$$\widehat{Y} = H \widehat{\beta} \tag{2.13}$$

Keterangan:

y = Output layer yang merupakan hasil prediksi.

H = nilai output weight didapatkan dari proses training.

 β = Keluaran di *hidden layer* dihitung dengan fungsi aktivasi.

- 4. Denormalisasi hasil prediksi ini menggunakan Persamaan 2.6
- 5. Pada langkah terakhir ialah untuk menghitung nilai evaluasi untuk semua ouput layer yang menggunakan Persamaan 2.14. Nilai error ini didapatkan dari nilai kesalahan dari hasil prediksi yang didapatkan.

2.6 Operasi Baris Elementer

Operasi Baris Elementer (OBE) merupakan proses operasi yang digunakan pada baris yang memiliki pada suatu matriks. Operasi-operasi yang dapat dilakukan pada matriks seperti (Darmayasa, 2015):

- 1. Mengembalikan suatu baris dengan nilai bilangan yang tidak nol, operasi ini disimbolkan dengan $kR_i \to R_i$, yang artinya baris ke-1 berubah setelah melakukan dikalikan k.
- 2. Menambahkan perkalian dari baris pada baris lain, operasi ini dapat disimbolkan dengan $R_i + kR_j \rightarrow R_i$, yang artinya dari baris ke-I berubah menjadi hasil dari penjumlahan $R_i + kR_j$.
- 3. Menukar antara dua pasang baris, operasi ini disimbolkan dengan $R_i \leftrightarrow R_j$, yang artinya dari baris ke-i dan baris ke-j bertukar posisi.

OBE yang biasa digunakan untuk menentukan *inverse* suatu matriks, untuk menentukan nilai *inverse* matriks A, dapat menggunakan sejumlah OBE pada matriks A dan OBE pada matriks I (matriks identitas). Perhitungan *inverse* matriks ini menggunakan konsep [A][I] yang dilakukan menjadi [I][A^{-1}], yang artinya untuk OBE ini digunakan mengubah A menjadi matriks I (Darmayasa, 2015). Berikut adalah contoh dari penggunaan OBE untuk menentukan *inverse* matriks (Siwi, et al., 2016):

Contoh matriks $A = \begin{pmatrix} 1 & 2 \\ 2 & 5 \end{pmatrix}$

Bentuk awal : $\begin{bmatrix} A|I \end{bmatrix} = \begin{pmatrix} 1 & 2|1 & 0 \\ 2 & 5|0 & 1 \end{pmatrix}$

1.
$$\frac{1}{2}R_2 + (-R_1) \to R_2 \begin{pmatrix} 1 & 2 & 1 & 0 \\ 0 & 0.5 & -1 & 0.5 \end{pmatrix}$$

2.
$$\frac{0.5}{2}R_1 + (-R_2) \rightarrow R_1 \begin{pmatrix} 0.25 & 0 & | 1.25 & -0.5 \\ 0 & 0.5 & | -1 & 0.5 \end{pmatrix}$$

3.
$$\frac{1}{0.25}R_1 \to R_1 \begin{pmatrix} 1 & 0 \mid 5 & -2 \\ 0 & 0 \mid -1 & 0.5 \end{pmatrix}$$

4.
$$\frac{1}{0.5}R_2 \to R_2 \begin{pmatrix} 1 & 0 \mid 5 & -2 \\ 0 & 1 \mid -2 & 1 \end{pmatrix}$$

Bentuk akhir : $[I|A^{-1}] = \begin{pmatrix} 1 & 0|1 & 2\\ 0 & 1|2 & 5 \end{pmatrix}$

2.7 Nilai Evaluasi

Nilai Evaluasi berbagai macam jenis persamaan yang dapat digunakan untuk menghitung dari nilai evaluasi, tentunya dalam mengukur hasil dari prediksi, seperti: Mean Absolute Percentage Error (MAPE), Mean Square Error (MSE), Mean Absolute Error (MAE), Root Mean Square Error (RMSE), dan lain-lain. Mean Absolute Percentage Error (MAPE) adalah indikator yang tidak jarang digunakan dalam model prediksi. Nilai MAPE dapat dijadikan persentase dengan dikalikan

100 persen lalu menghasilkan nilai dalam bentuk persentase. MAPE dapat digunakan untuk mengukur validasi keakuratan pengujian (Worasucheep, 2015). MAPE digunakan jika ukuran atau besar variabel prediksi dianggap sangat penting dalam melakukan evaluasi pada hasil prediksi (Siwi, 2016). Lalu, dengan hasil presentase orang awam lebih dapat memahami dibandingkan nilai kesalahan kuadrat (Worasucheep, 2015). Dalam kemapuan prediksi sangat baik, untuk hasil nilai MAPE kurang dari 10% dan mempunyai kemampuan prediksi baik jika nilai MAPE kurang dari 20% (S dan W, 2015). Untuk melihat perhitungan MAPE ditunjukkan di Persamaan 2.14 yang digunakan untuk menghitung nilai MAPE (Jahuri et al., 2016).

2.7.1 Mean Absolute Percentage Error (MAPE)

Mean Absolute Percentage Error atau MAPE biasanya digunakan untuk melakukan suatu evaluasi hasil prediksi dengan membandingkan dari nilai ramalan dengan nilai aktual. Berikut ini merupakan persamaan untuk melakukan perhitungan nilai error dari untuk melakukan prediksi (Makridakis & Hibon, 1995):

$$MAPE = \frac{1}{n} \sum_{i=1}^{n} = \frac{\hat{y}_i - y_i}{\hat{y}_i} \times 100\%$$
 (2.14)

Keterangan:

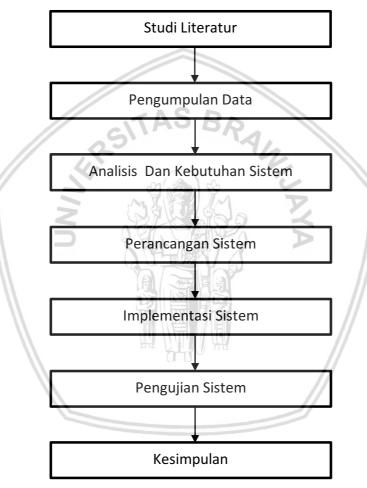
 $\hat{y}_i = N$ ilai hasil ramalan

 $y_i = Nilai aktual$

 $n \; = \; \mathsf{Banyak} \; \mathsf{data}$

BAB 3 METODOLOGI

Bab metodologi ini merupakan penjelasan tentang tahap penelitian, studi literature, teknik pengumpulan data, analisis dan kebutuhan sistem, perancangan sistem yang terdiri dari gambaran umum sistem dan perancangan sistem, implementasi sistem, implementasi dan pengujian, dan analisis sistem dalam mengimplementasikan metode pada prediksi harga daging sapi di Kota Malang. Tahapan penelitian yang digunakan ditunjukkan pada Gambar 3.1.



Gambar 3.1 Tahapan Penelitian

3.1 Studi Literatur

Untuk mempelajari beberapa literatur yang relevan dengan prediksi harga Daging Sapi di Kota Malang. Beberapa literatur tersebut diantaranya:

- 1. Prediksi
- 2. Metode Extreme Learning Machine (ELM).
- 3. Metode Min-Max Normalization untuk normalisasi data.

4. MAPE untuk menghitung nilai error.

Literatur diperoleh dari jurnal, e-book, buku, penelitian sebelumnya dan artikel-artikel dari internet yang dipandang layak dan berhubungan dengan tema penelitian.

3.1.1 Tipe Penelitian

Pada penelitian ini merupakan tipe penelitian non-implementatif. Tipe penelitian non-implementatif memfokuskan investigasi terhadap fenomena atau situasi tertentu, atau analisis terhadap hubungan antar fenomena yang sedang dikaji untuk kemudian menghasilkan hasil investigasi atau hasil analisis ilmiah sebagai produk atau artefak utamanya. Pendekatan analitik merupakan sebuah kegiatan penelitian nonimplementatif yang dilakukan untuk menjelaskan derajat hubungan antar elemen dalam objek penelitian dengan fenomena atau situasi tertentu yang sedang diteliti. Produk atau artefak utamanya yang dihasilkan adalah hasil analisis.

3.2 Pengumpulan Data AS BA

Pengumpulan data yang didapatkan dari Bidang Peternakan dan Kesehatan Hewan Kota Malang dan Dinas Perdagangan Kota Malang. Kegiatan ini dilakukan dengan cara meminta izin kepada petugas pekerja di Bidang Peternakan untuk meminta dan mengetahui data tentang harga daging sapi selama beberapa tahun belakang. Data ini digunakan adalah data yang disimpan dalam setiap bulan dari 6 tahun sebelumnya. Data ini digunakan sebagai fitur yang diimplementasikan untuk digunakan dalam metode *Extreme Learning Machine* (ELM). Berikut data yang kami dapatkan yang digunakan dalam penelitian ini.

3.2.1 Teknik Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah data sekunder. Data sekunder adalah sumber data penelitian yang diperoleh penelitian secara tidak langsung, namu melalui situs web resmi dinas perdagangan yang digunakan dalam penelitian adalah harga pasar daging sapi. Data yang digunakan adalah harga pasar yang diambil dari situs web Sistem Informasi Ketersediaan dan Perkembangan Harga Bahan Pokok di Jawa Timur. Data tersebut digunakan sebagai data training dan data uji. Pengumpulan data mempunyai tujuan seperti mendapatkan datadata yang mempunyai keterkaitan dengan topik dari penelitian. Pengumpulan data ini agar mendapatkan hasil yang akurat dan dapat diandalkan.

3.2.2 Partisipan Penelitian

Partisipan yang terlibat dalam penelitian ini adalah Bidang Perternakan dan Kesehatan Hewan Dinas Pertanian dan Ketahanan Pangan Kota Malang dan Dinas Perdagangan Kota Malang. Alasan pemilihan partisipan penelitian ini karena data skripsi yang didapatkan dari instansi tersebut. Sedangkan data harga pasar daging sapi yang didapatkan dari Website Dinas Perdagangan selalu diperbaharui setiap harinya sehingga data yang dihasilkan data yang valid.

3.3 Analisis Dan Kebutuhan Sistem

Analisis kebutuhan dalam penelitian ini berguna untuk mengetahui dan menganalisis dalam melakukan kebutuhan yang diperlukan untuk memprediksi harga dagi sapi menggunakan metode ELM di Kota Malang. Berikut ini kebutuhan yang digunakan dalam memprediksi harga daging sapi menggunakan algoritme ELM.

3.4 Perancangan Sistem

Dalam tahap perancangan untuk melakukan mengidentifikasi komponen yang dibutuhkan pada saat melakukan proses implementasi metode ELM untuk penelitian prediksi harga daging sapi menggunakan ELM di Kota Malang. Langkahlangkah yang dilakukan dalam perancangan sistem adalah sebagai berikut:

1. Perancangan User Interface (Antarmuka Pengguna)

Dalam melakukan perancangan *user interfac*e ini mempermudah pengguna dalam menggunakan sistem yang ingin digunakan.

2. Perancangan Pengujian

Dalam perancangan sistem yang dilakukan meliputi proses uji coba rentang, input weight, perbandingan jumlah data training dan data testing, jumlah hidden neuron pada hidden layer dan uji coba berdasarkan jenis analisis data.

3.4.1 Strategi Penelitian

Penelitian ini termasuk penelitian yang menggunakan metode eksperimen. Penelitian eksperimen adalah suatu penelitian yang berusaha mencari pengaruh variabel tertentu terhadap variabel yang lain dalam kondisi yang terkontrol secara ketat dan umumnya dilakukan dilaboratorium. Metode eksperimen yang digunakan dalam penelitian ini yaitu Extreme Learning Machine (ELM)

3.5 Implementasi Sistem

Implementais metode ELM merupakan agar dapat memprediksi harga daging sapi menggunakan metode ELM di Kota Malang agar sesuai dengan perancangan yang telah diidenfitikasikan sebelumnya.

3.5.1 Implementasi Algoritma

Proses implementasi algoritme yang dilakukan sebelum program dikerjakan, pada tahap ini dilakukan rancangan secara rinci bagaimana proses pengerjaan dari program yang dilakukan, agar dapat menjadi acuan saat dikerjakan. Langkah pertama yang dilakukan didalam implementasi algoritme yaitu memasukkan harga pasar daging sapi dengan data setiap bulan dalam satu tahun, kemudian harga daging sapi tersebut dilakukan proses normalisasi data. Lalu mencari nilai random bobot, hidden neuron, setelah itu melakukan proses training dan testing. Setelah melakukan testing maka didapatkan hasil harga daging sapi yang menggunakan metode ELM pada proses denormalisasi. Setelah

mendapatkan harga daging sapi asli dan harga daging sapi yang sudah diproses dengan perhitungan ELM, maka mencari nilai *Mean Absolute Percentage Error* (MAPE) untuk mengetahuan hasil tingkat *error* dan mendapatkan hasil prediksi dengan menggunakan *Extreme Learning Machine* (ELM).

3.6 Pengujian Sistem

Dalam tahap uji coba merupakan suatu tujuan agar dapat mengetahui tingkat keberhasilan dalam penerapan metode ELM untuk prediksi harga daging sapi di Kota Malang. Beberapa uji coba yang dilakukan untuk mengevaluasi sistem ini antara lain:

- 1. Uji coba variasi fitur data
- 2. Uji coba range hidden neuron
- 3. Uji coba jumlah fungsi aktivasi
- 4. Uji coba variasi jumlah data training dan data testing

Untuk mengevaluasi hasil prediksi harga daging sapi, digunakan ukuran hasil prediksi. Hasil untuk harga yang terbaik adalah yang memberikan nilai kesalahan prediksi yang terkecil. Nilai dari evaluasi yang digunakan adalah nilai MAPE (*Mean Absolute Percentage Error*).

3.7 Kesimpulan

Setelah melakukan tahapan uji coba yang telah dilakukan, maka hasil uji coba tersebut dievaluasi dengan tujuan untuk mengetahui parameter yang memiliki nilai terbaik dalam prediksi harga. Dalam parameter yang digunakan untuk pengujian adalah pengujian jumlah fitur pengaruh fitur data yang digunakan, hidden neuron, fungsi aktivasi, dan pengaruh persentase perbandingan jumlah data training dan data testing.

BAB 4 PERANCANGAN

Bab ini menjelaskan formulasi permasalahan, arsitektur perancangan sistem, diagram alir sistem, siklus penyelesaian prediksi harga pasar daging sapi menggunakan *Extreme Learning Machine* (ELM), perhitungan manual, perancangan *user interface* serta perancangan skenario pengujian.

4.1 Deskripsi Masalah

Permasalahan yang ingin diselesaikan adalah memprediksikan harga pasar daging sapi di Kota Malang dengan menggunakan variabel unit yaitu jumlah harga daging sapi dalam data harga setiap bulan. Kemudian tahap selanjutnya dilakukan evaluasi nilai *error* yang berdasarkan dari hasil prediksi agar memperoleh kualitas yang baik. Untuk data yang dideskripsikan ini data didapatkan dari instansi Dinas Perternakan dan Kesehatan Hewan Dinas Pertanian dan Ketahanan Pangan Kota Malang dan Dinas Perdagangan Kota Malang dengan data bulanan dari tahun 2012 sampai tahun 2017.

Dalam mengatasi masalah tersebut dapat dilakukan perhitungan dengan menggunakan metode Extreme Learning Machine (ELM). Selain data harga pasar daging sapi, perhitungan ini juga memasukkan berupa parameter jumlah neuron pada hidden layer, fungsi aktivasi, serta perbandingan antara jumlah data training dan data testing. Sebelum memasuki proses ELM, hal yang harus dilakukan adalah proses normalisasi menggunakan Min-Max Normalization dari inisialisasi data training dan data testing. Selanjutnya diakukan proses perhitungan prediksi dengan metode ELM. Kemudian masuk kedalam proses perhitungan nilai error yang dievaluasi menggunakan metode Mean Absolute Percentage Error (MAPE). Perhitungan ini memperoleh keluaran berupa hasil prediksi yang telah di denormalisasi dan nilai Mean Absolute Percentage Error (MAPE).

Dalam perhitungan manual, data yang digunakan untuk penelitian ini menggunakan 12 sampel data. Data harga pasar daging sapi yang digunakan adalah jumlah harga daging sapi yang terjadi setahun dalam 12 bulan, data yang digunakan untuk dalam perhitungan manual ini yaitu data harga 2015. Data yang digunakan yaitu dapat dilihat pada Tabel 4.1.

Data Harga Konsumen tahun 2017 **Tahun** No **Bulan X1 X2 X3 X4 X5** 1 111,412 111,433 111,451 111,433 111,432 111,612 Januari 2 Februari 111,433 111,451 111,433 111,432 111,612 113,078 2017 3 111,451 111,433 111,432 111,612 113,078 113,412 Maret 4 111,433 111,432 111,612 113,078 113,412 112,04 April 5 Mei 111,432 111,612 113,078 113,412 112,04 111,438

Tabel 4.1 Data Harga Daging Sapi

Tabel 4.1 Data Harga Daging Sapi (lanjutan)

Data Harga Konsumen tahun 2017								
Tahun	No	Bulan	X1	X2	Х3	X4	X5	Т
	6	Juni	111,612	113,078	113,412	112,04	111,438	111,006
	7	Juli	113,078	113,412	112,04	111,438	111,006	111000
	8	Agustus	113,412	112,04	111,438	111,006	111,000	111,032
2017	9	September	112,04	111,438	111,006	111,000	111,032	111,000
	10	Oktober	111,438	111,006	111,000	111,032	111,000	111,609
	11	November	111,006	111,000	111,032	111,000	111,609	111,326
	12	Desember	111,000	111,032	111,000	111,609	111,326	112,103

Nilai pada fitur 5 didapatkan dari dari nilai target 1 bulan sebelumnya, untuk nilai fitur 4 didapatkan dari nilai target 2 bulan sebelumnya, nilai fitur 3 didapatkan dari nilai target 3 bulan sebelumnya, sedangkan nilai fitur 2 didapatkan dari nilai target 4 bulan sebelumnya, dan nilai fitur 1 didapatkan dari nilai target 5 bulan sebelumnya. Keterangan dari setiap parameter data ditunjukkan pada Tabel 4.1:

1. Bulan : Menerangkan bulan yang sesuai dihasilkan pada (string) harga daging sapi itu terjadi.

2. Fitur (X1) Sampai (X5) : Menjelaskan fitur pertama (X1) merupakan harga (Double)

daging sapi lima bulan sebelumnya pada parameter ke hari T (target) atau bulan pertama. Menjelaskan fitur kedua (X2) yang merupakan harga pasar daging sapi empat bulan sebelumnya pada parameter ke hari T (target) atau bulan pertama. Menjelaskan fitur ketiga (X3) yang merupakan harga pasar daging sapi tiga bulan sebelumnya pada parameter ke hari T (target) atau bulan pertama. Menjelaskan fitur keempat (X4) merupakan harga pasar daging sapi dua bulan sebelumnya pada parameter ke hari T (target) atau bulan pertama, Menjelaskan fitur kelima (X5) merupakan harga pasar daging sapi satu bulan sebelumnya pada parameter ke hari T (target) atau bulan pertama.

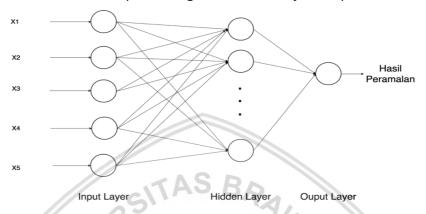
3. Target : Menjelaskan harga daging sapi perbulan perbulan (Double)

Target (T).

4.2 Perancangan Jaringan

Arsitektur untuk perancangan sistem adalah gambaran secara umum dari proses sistem prediksi harga pasar daging sapi menggunakan Extreme Learning

Machine (ELM). Neuron pada input layer yang begitu banyak disesuaikan dengan masukkan untuk sistem yang terdiri dari 5 input data yaitu Fitur X1, Fitur X2, Fitur X3, Fitur X4, Fitur X5 dan target (T). Setiap neuron untuk input layer terhubung oleh hidden neuron yang ada pada hidden layer. Neuron ini kemudian setiap neuron pada hidden layer terhubung dengan output layer yang dihubungkan oleh output weight. Semua hidden neuron yang terhubung pada satu output. Berikut ini gambaran untuk arsitektur perancangan sistem ditunjukkan pada Gambar 4.1.

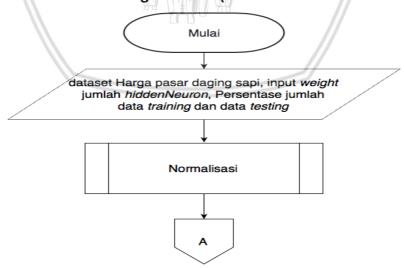


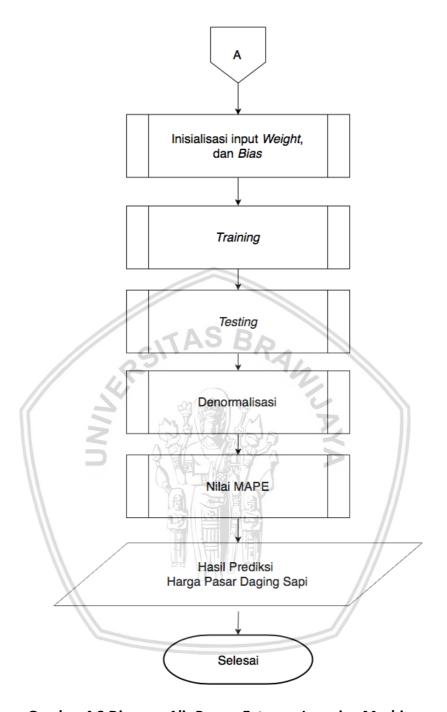
Gambar 4.1 Arsitektur Perancangan Sistem

4.3 Siklus Metode Extreme Learning Machine (ELM)

Dalam siklus menggunakan metode Extreme Learning Machine (ELM) sangat dibutuhkan alur kerja algoritma. Alur kerja ini menjelaskan proses normalisasi data menggunakan rumus Min-Max Normalization, proses algoritma Extreme Learning Machine (ELM), proses perhitungan Mean Absolute Percentage Error (MAPE) lalu proses denormalisasi data.

4.3.1 Proses Extreme Learning Machine (ELM)





Gambar 4.2 Diagram Alir Proses Extreme Learning Machine

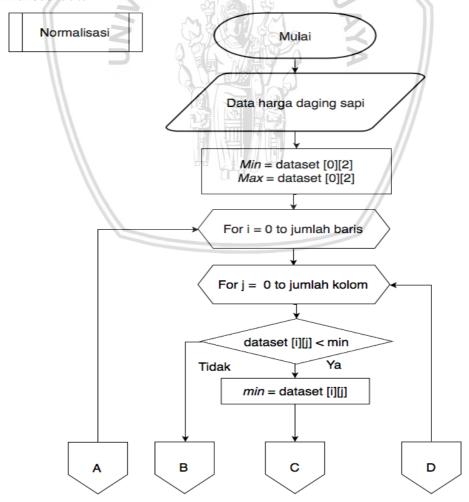
Proses metode *Extreme Learning Machine* ini ditunjukan pada Gambar 4.2 memiliki tahapan-tahapan sebagai berikut:

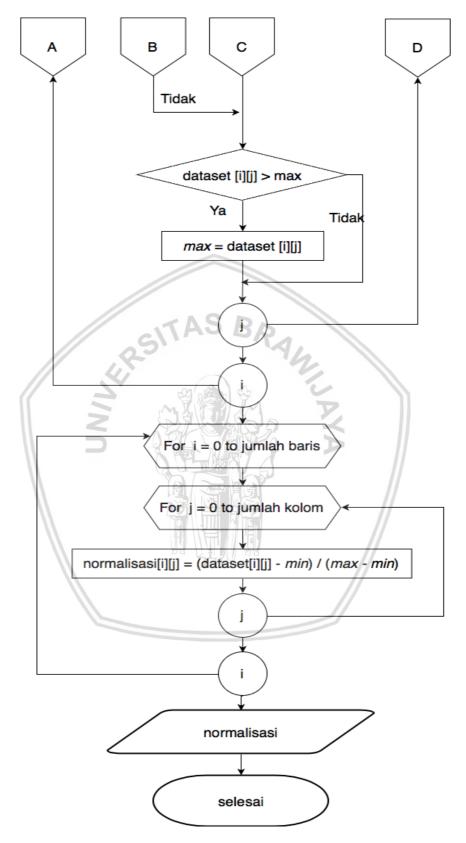
- 1. Memasukkan ke sistem berupa data harga pasar daging sapi, input weight, jumlah hidden neuron, persentase data training dan data testing.
- 2. Melakukan perhitungan yang diproses tahap normalisasi data yang ditunjukkan pada Persamaan 2.5. untuk melihat diagram alir proses normalisasi dapat dilihat pada Gambar 4.3.
- 3. Inisialisasi input weight dan input bias yang dapat dilihat pada Gambar 4.4.

- 4. Melakukan proses perhitungan *training* dengan data yang telah melakukan proses normalisasi data. Melihat proses diagram alir proses *training* pada Gambar 4.5.
- 5. Melakukan proses perhitungan *testing* untuk menentukan hasil prediksi. Diagram alir proses *testing* ditunjukkan pada Gambar 4.14.
- 6. Melakukan proses perhitungan denormalisasi data dengan Persamaan 2.6. Diagram alir proses normalisasi ditunjukkan pada Gambar 4.3.
- 7. Melakukan perhitungan untuk nilai *Mean Absolute Percentage Error* (MAPE) yang ditunjukkan pada Persamaan 2.14. Diagram alir proses nilai MAPE ditunjukkan pada Gambar 4.17.
- 8. Untuk keluaran hasil prediksi harga daging sapi yang telah didenormalisasi.

4.3.2 Proses Normalisasi

Normalisasi adalah proses yang dimana nilai data yang diubah ke skala tertentu. Pada penelitian ini dilakukan normalisasi data menggunakan metode *Min-Max Normalization,* maka merubah data menjadi nilai lebih kecil yaitu nilai *range* antara [0-1]. Pada Gambar 4.3 yang menunjukkan diagram proses normalisasi data.

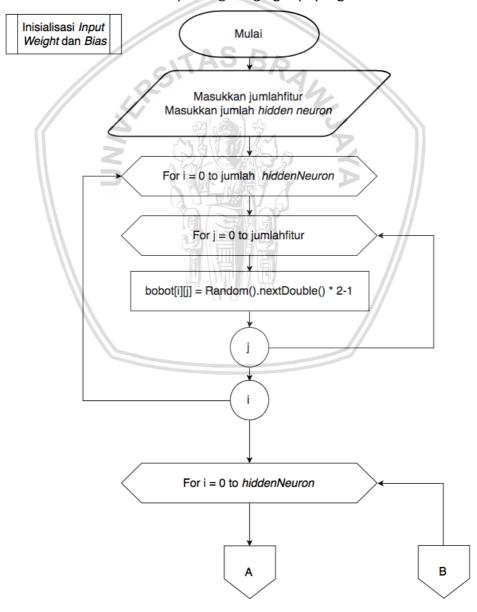


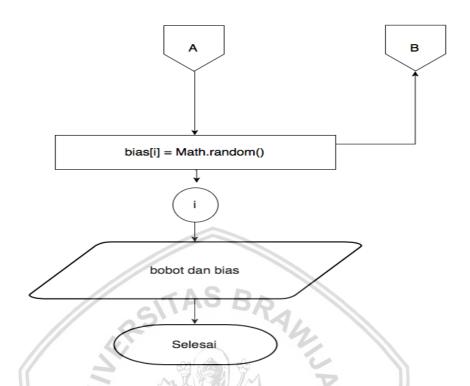


Gambar 4.3 Diagram Alir Proses Normalisasi

Berdasarkan Gambar 4.3 menunjukkan langkah-langkah untuk proses normalisasi harga daging sapi menggunakan perhitungan metode *Min-Max Normalization* yang dijelaskan sebagai berikut:

- 1. Masukkan pada program berupa data harga daging sapi yang di normalisasi.
- 2. Mencari nilai maksimum atau nilai tertinggi dan nilai minimum atau nilai terendah dari semua data yang sudah dimasukkan.
- 3. Melakukan proses perulangan sebanyak jumlah pada baris.
- 4. Melakukan proses perulangan sebanyak jumlah pada kolom.
- 5. Melakukan proses menghitung nilai normalisasi untuk setiap data yang masukan dengan rumus yang ditunjukkan pada Persamaan 2.5.
- 6. Keluaran nilai data ini berupa harga daging sapi yang telah dinormalisasi.





Gambar 4.4 Diagram Alir Proses Input Weight dan Bias

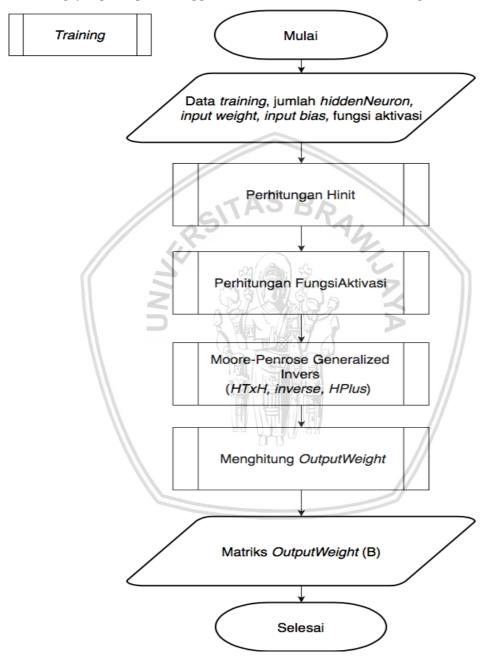
Proses inisialisasi *input weight* ditunjukkan pada Gambar 4.4 memiliki tahapan-tahapan sebagai berikut:

- 1. Masukkan pada program berupa inputan hidden neuron dan jumlah fitur.
- 2. Melakukan perulangan sebanyak proses jumlah hidden neuron.
- 3. Melakukan perulangan sebanyak proses jumlah jumlah Fitur.
- 4. Menghitung nilai *input weight* dengan menggunakan rumus *random* yaitu *range* antara -1 dan 1.
- 5. Melakukan perulangan sebanyak jumlah hidden neuron dan jumlah fitur.
- 6. Melakukan perulangan sebanyak proses jumlah hidden neuron.
- 7. Menghitung nilai *input* bias dengan menggunakan rumus *random* antara 0 sampai 1.
- 8. Keluaran data ini mendapatkan nilai weight dan bias.

4.3.3 Proses Training

Proses training adalah proses pelatihan yang dilakukan untuk memprosws nilai input weight dan mendapatkan hasil output weight yang atau bobot akhir dari proses training. Nilai yang diperoleh tersebut digunakan untuk proses testing. Langkah yang pertama dilakukan yaitu masukkan nilai input weight dan bias pada proses yang sudah diinisialisasi secara random. Kemudian untuk menghitung keluaran pada hidden layer dengan menggunakan fungsi aktivasi (H_{init}) yang

dilanjutkan melakukan perhitungan matriks *Moonre-Penrose Generalized* (H⁺) dari hasil keluaran pada *hidden layer* yang menggunakan fungsi aktivasi. Untuk hasil perkalian antara H+ dan juga target digunakan untuk mencari nilai *output weight*. Lalu nilai *output weight* dibawa untuk menuju ke proses *testing* yang dilakukan untuk perhitungan kembali. Gambar 4.4 ini menujukkan untuk diagram alir proses *training* yang dengan menggunakan metode *Extreme Learning Machine* (ELM).

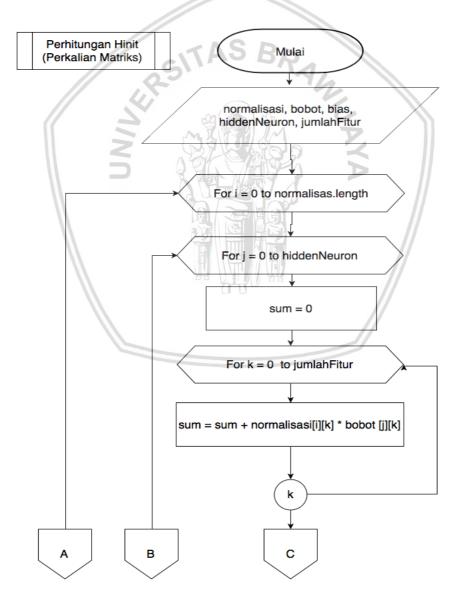


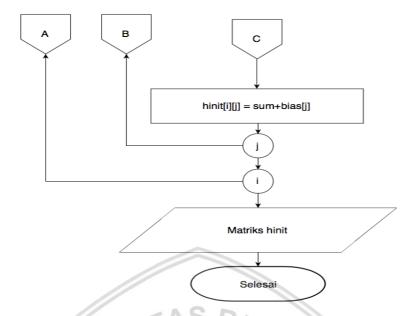
Gambar 4.5 Diagram Alir Proses Training

Prosses *training* ini dapat dilihat pada Gambar 4.5 memiliki tahapan-tahapan sebagai berikut:

1. Memasukkan yang telah diterima sistem berupa data *training*, jumlah *hidden neuron*, *input weight*, *input* bias dan fungsi aktivasi.

- 2. Melakukan proses perhitungan keluaran oleh *hidden layer* dengan Persamaan 2.7.
- 3. Melakukan proses perhitungan fungsi aktivasi ketika mendapatkan nilai dari proses *hidden layer* dengan Persamaan 2.1 sampai Persamaan 2.4.
- 4. Menghitung pada matriks *Moore-Penrose Generalized Inverse* yang terdiri dari proses (HTxH, *Inverse*, dan nilai H⁺) dari data yang diinputkan nilai hasil keluaran oleh *hidden layer* menggunakan pada fungsi aktivasi. Diagram alir untuk menghitung *output weight* dapat ditunjukkan pada Gambar 4.11.
- 5. Melakukan proses perhitungan *output weight* yang sesuai dengan Persamaan 2.9 diagram alir untuk proses perhitungan *output weight* yang ditunjukkan pada Gambar 4.13.
- 6. Hasil dari keluaran sistem yaitu berupa matriks output weight.

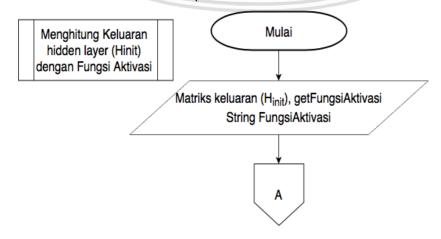


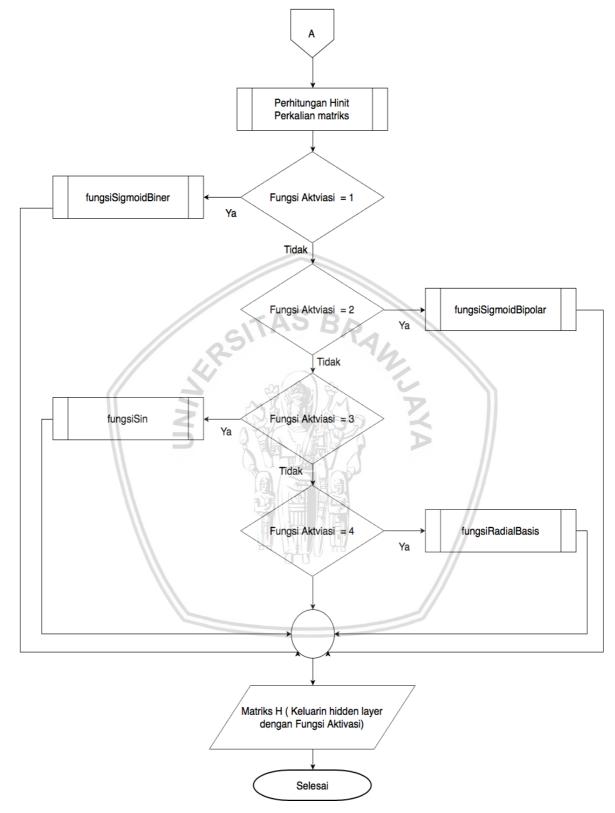


Gambar 4.6 Diagram Alir Proses hidden layer (Hinit)

Berdasarkan Gambar 4.6 dapat dilihat langkah-langkah proses dalam melakukan perhitungan *hidden layer* yang diuraikan sebagai berikut:

- 1. Sistem menerima masukkan pada program berupa *input* normalisasi, bobot, bias, *hidden neuron*, dan jumlah fitur.
- 2. Melakukan perulangan sebanyak proses jumlah panjang normalisasi.
- 3. Melakukan perulangan sebanyak proses jumlah hidden neuron.
- 4. Melakukan perkalian kedua matriks dengan cara mengalikan bilangan yang terdapat pada kolom matriks normalisasi dengan bilangan-bilangan yang terdapat pada matriks *hidden neuron* sebanyak jumlah baris matriks normalisasi.
- 5. Menghitung nilai hinit dengan hasil *sum* yang didapatkan ditambah bias.
- 6. Keluaran data ini mendapatkan matriks Hinit.

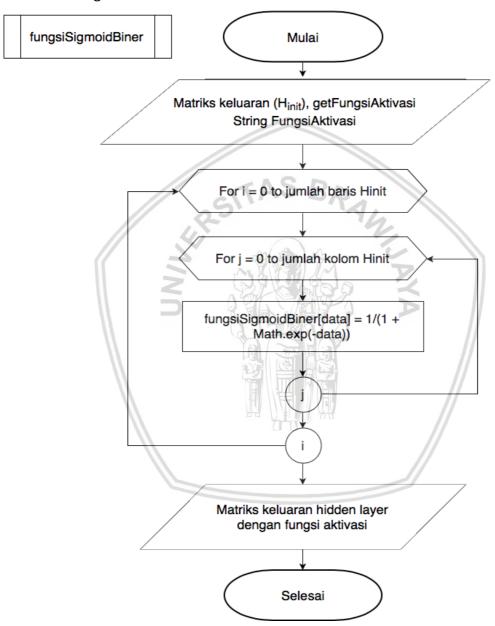




Gambar 4.7 Diagram Alir Keluaran Hidden Layer dengan Fungsi Aktivasi

Berdasarkan Gambar 4.7 dapat dilihat langkah-langkah dalam melakukan perhitungan keluaran *hidden layer* oleh fungsi aktivasi yang diuraikan sebagai berikut:

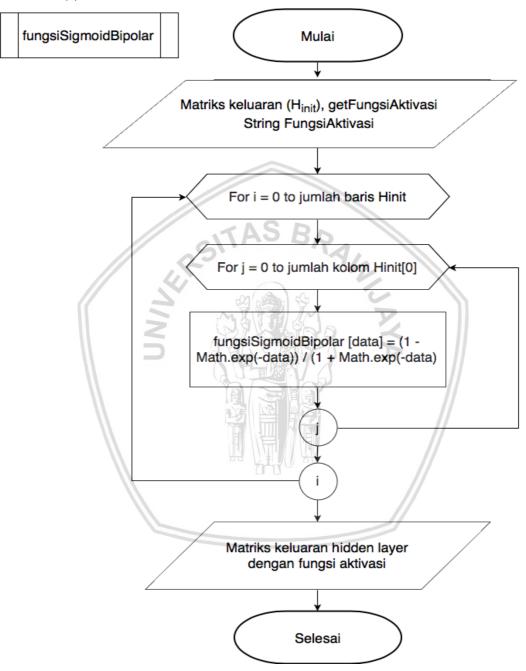
- 1. Masukkan yang terima oleh sistem berupa data yang sudah dilakukan matriks keluaran hidden layer (Hinit).
- 2. Melakukan proses perhitungan fungsi aktivasi menggunakan rumus yang dipilih sesuai *input* yang ditunjukkan oleh Persamaan 2.1 sampai 2.4.
- 3. Hasil keluaran berupa matriks H yaitu keluaran *hidden layer* yang sudah di fungsi aktivasi.



Gambar 4.8 Diagram Alir Fungsi Aktivasi Sigmoid Biner

Berdasarkan Gambar 4.7 dapat dilihat langkah-langkah dalam melakukan perhitungan menggunakan fungsi aktivasi sigmoid biner yang diuraikan sebagai berikut:

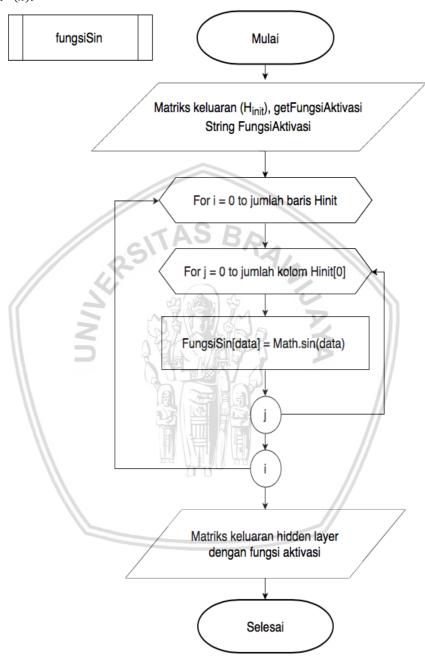
- 2. Melakukan proses perhitungan fungsi aktivasi sigmoid biner menggunakan rumus yang ditunjukkan pada Persamaan 2.1
- 3. Keluaran sistem yaitu matriks keluaran hidden layer yang telah diaktivasi $(H_{(x)})$



Gambar 4.9 Diagram Alir Fungsi Aktivasi Sigmoid Bipolar

Berdasarkan Gambar 4.8 dapat dilihat langkah-langkah dalam melakukan perhitungan menggunakan fungsi aktivasi sigmoid bipolar yang diuraikan sebagai berikut:

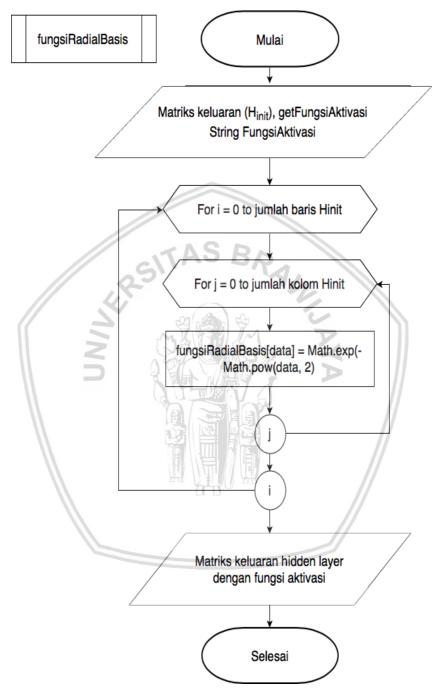
- 2. Melakukan proses perhitungan fungsi aktivasi sigmoid bipolar menggunakan rumus yang ditunjukkan pada Persamaan 2.2
- 3. Keluaran sistem yaitu matriks keluaran hidden layer yang telah diaktivasi $(H_{(x)})$



Gambar 4.10 Diagram Alir Fungsi Fungsi Aktivasi Sin

Berdasarkan Gambar 4.10 dapat dilihat langkah-langkah dalam melakukan perhitungan menggunakan fungsi aktivasi sin yang diuraikan sebagai berikut:

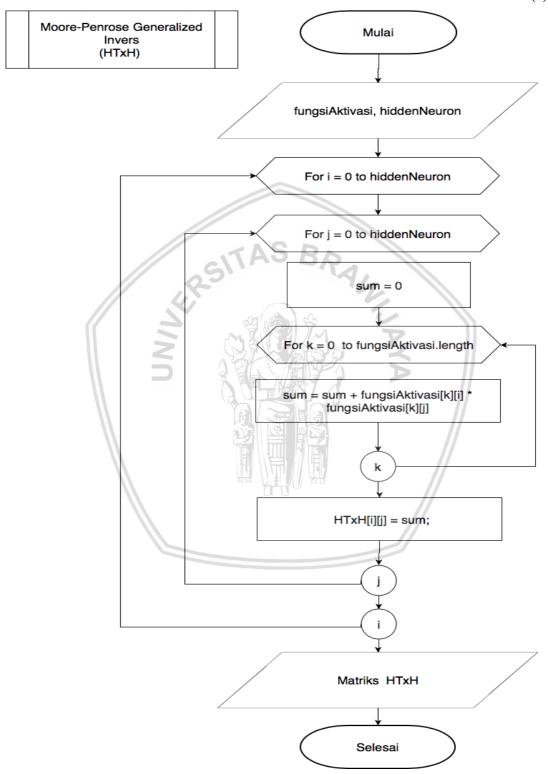
- 2. Melakukan proses perhitungan fungsi aktivasi sin menggunakan rumus yang ditunjukkan pada Persamaan 2.3
- 3. Keluaran sistem yaitu matriks keluaran hidden layer yang telah diaktivasi $(H_{(x)})$



Gambar 4.11 Diagram Alir Fungsi Aktivasi Radial Basis

Berdasarkan Gambar 4.11 dapat dilihat langkah-langkah dalam melakukan perhitungan menggunakan fungsi aktivasi radial basis yang diuraikan sebagai berikut:

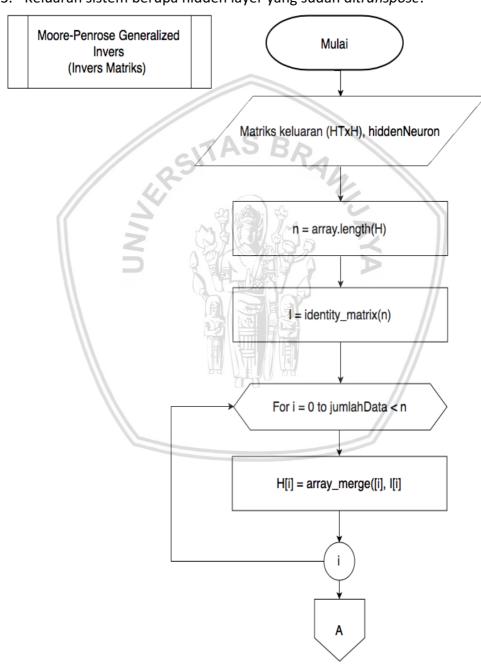
- 2. Melakukan proses perhitungan fungsi aktivasi radial basismenggunakan rumus yang ditunjukkan pada Persamaan 2.4
- 3. Keluaran sistem yaitu matriks keluaran hidden layer yang telah diaktivasi $(H_{(x)})$

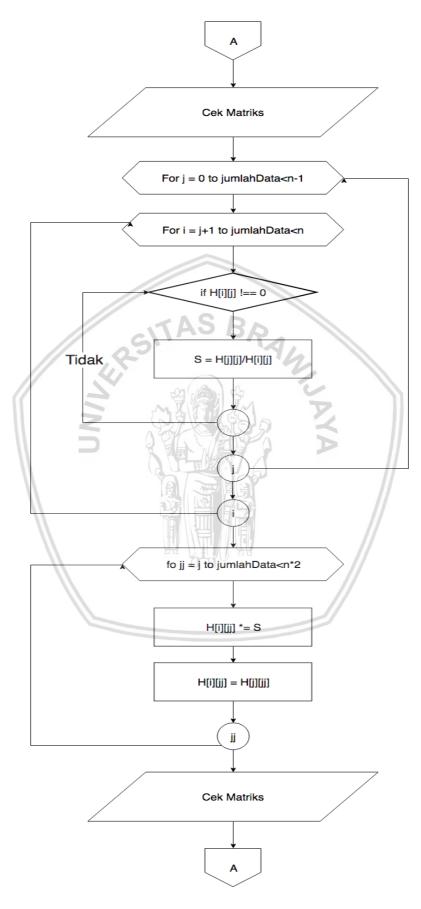


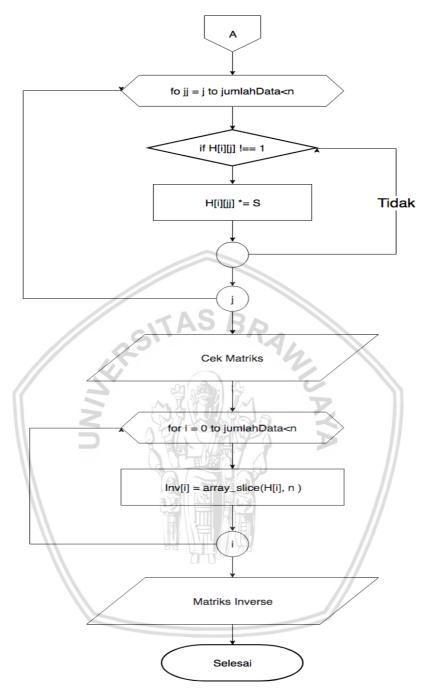
Gambar 4.12 Diagram Alir Perkalian Hidden Layer

Berdasarkan Gambar 4.12 dapat dilihat langkah-langkah dalam melakukan perhitungan perkalian matriks yang diuraikan sebagai berikut:

- 1. Masukkan yang diterima dalam sistem yaitu matriks keluaran *hidden neuron* dan *fungsi Aktivasi*.
- 2. Melakukan proses perhitungan untuk mendapatkan nilai *transpose* dengan mengalikan fungsi aktivasinya sendiri dengan mengubah kolom menjadi baris dan baris menjadi kolom.
- 3. Keluaran sistem berupa hidden layer yang sudah ditranspose.





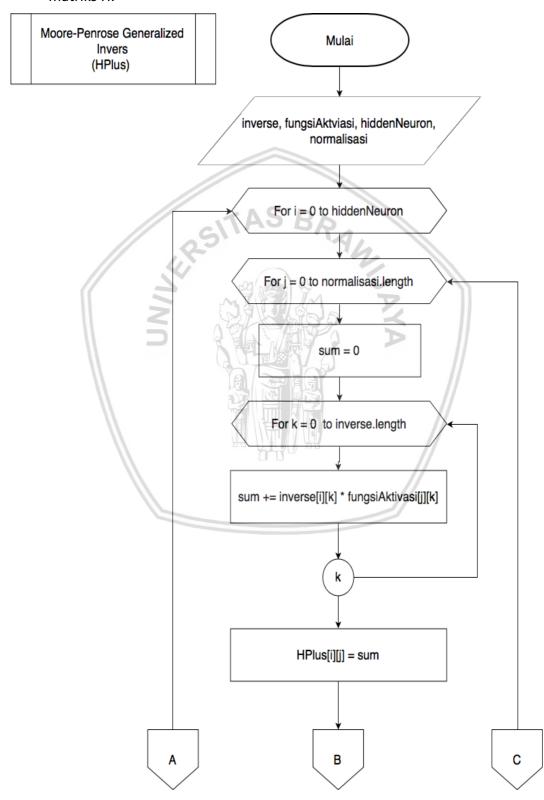


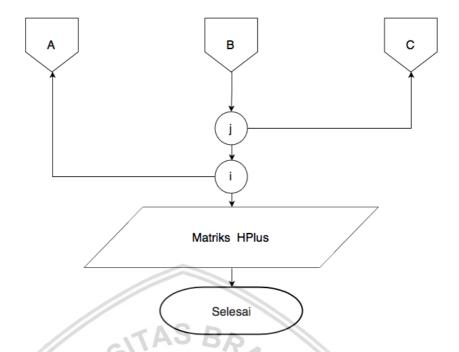
Gambar 4.13 Diagram Alir Perhitungan Inverse Matriks

Berdasarkan Gambar 4.13 dapat dilihat langkah-langkah dalam melakukan perhitungan inverse matriks yang diuraikan sebagai berikut:

- 1. Inisialisasi bentuk matriks identitas sesuai dengan bentuk pada matriks (H|I). Perhitungan *inverse*, matriks identitas digunakan dalam suatu operasi baris elementer (OBE).
- 2. Cek kondisi dimana matriks tidak bernilai nol atau kosong.

- 3. Untuk proses OBE ini mengubah matriks H menjadi matriks identitas, sehingga matriks identitas (I) menjadi matriks yang baru.
- 4. Setelah matriks H berubah menjadi matriks identitas, maka diperoleh matriks baru didapatkan perubahan matriks identitas yang merupakan *inverse* dari matriks H.

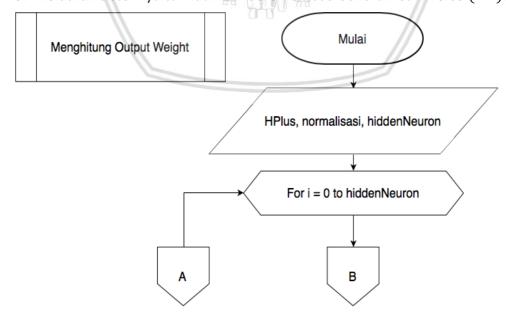


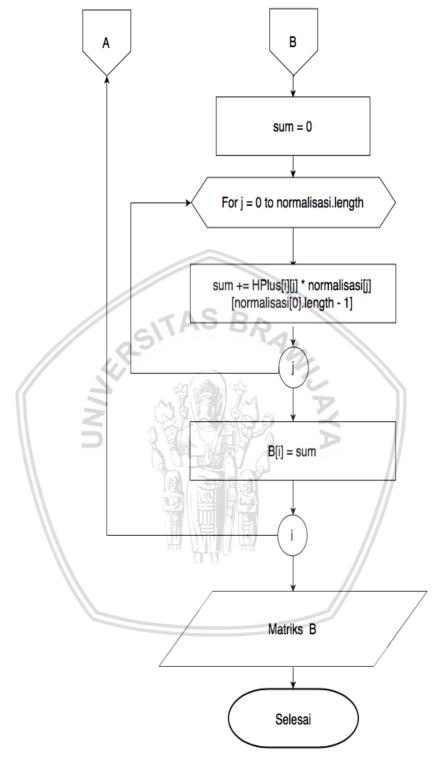


Gambar 4.14 Diagram Alir Moore-Penrose Generalized Inverse (H+)

Berdasarkan Gambar 4.14 dapat dilihat langkah-langkah dalam melakukan perhitungan proses *Moore-penrose Generalized Inverseh* (H+) yang di uraikan sebagai berikut:

- 1. Masukan yang diterima dalam sistem berupa matriks keluaran *inverse*, fungsi aktivasi, *hidden neuron*, normalisasi.
- 2. Melakukan proses perhitungan perkalian matriks dari hasil *inverse* dan fungsi aktivasi keluaran *hidden layer* yang di*transpose*.
- 3. Keluaran sistem yaitu matriks *Moore-Penrose Generalized Inverse* (\mathbf{H}^+) .





Gambar 4.15 Diagram Alir Bobot Akhir

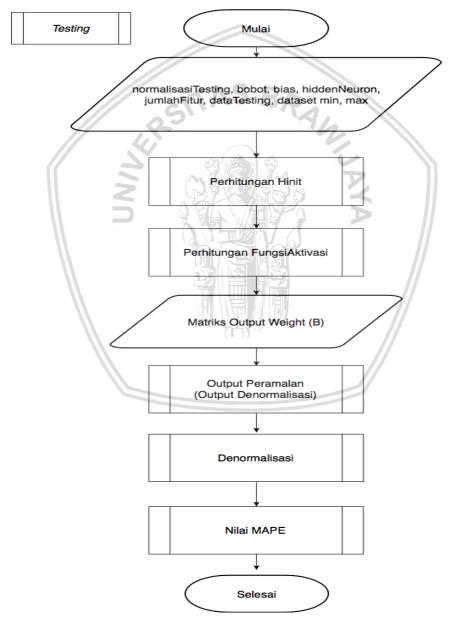
Berdasarkan Gambar 4.15 dapat dilihat langkah-langkah dalam melakukan perhitungan menggunakan diagram *output weight* yang diuraikan sebagai berikut:

1. Masukkan yang diterima dalam sistem yaitu matriks keluaran HTxH, inverse, Moore-Penrose Generalized Inverse (H^+) dan hasil matriks target.

- 2. Melakukan perkalian matriks dari hasil *Moore-Penrose Generalized Inverse* (H^+) dengan hasil matriks normalisasi target yang tunjukkan pada Persamaan 2.10.
- 3. Keluaran berupa matriks output weight.

4.3.4 Proses Testing

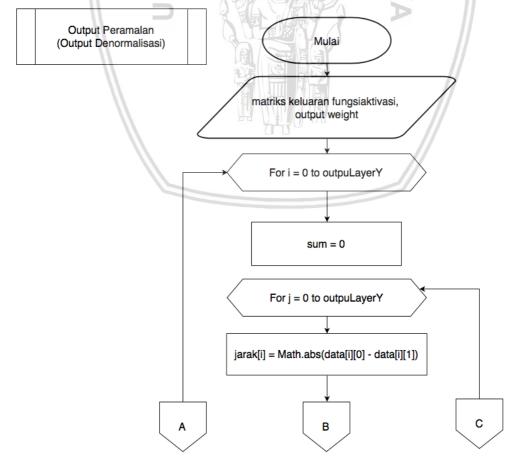
Proses testing adalah proses pelatihan yang dilakukan berdasarkan nilai input weight, dan output weight yang telah diperoleh dari perhitungan yang telah dilakukan diproses training. dalam proses training dilakukan sama seperti tahapan pada proses training. Gambar 4.14 menunjukkan diagram alir pada proses testing yang menggunakan metode Extreme Learning Machine (ELM).

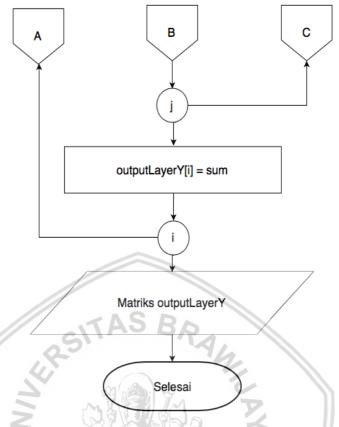


Gambar 4.16 Diagram Alir Proses Testing

Berdasarkan Gambar 4.16 dapat dilihat langkah-langkah dalam melakukan pencarian proses *testing* yang diuraikan sebagai berikut:

- 1. Masukkan data harga daging sapi yang telah dinormalisasi *testing*, bobot, bias, *hidden neuron*, jumlah fitur, data *testing*, dataset *min* dan *max*.
- 2. Inisialisasi input weigth dan input bias yang didapatkan dari data training.
- 3. Menghitung keluaran *hidden layer* yang sesuai dengan tahap-tahap diproses *training* ditunjukkan pada Gambar 4.5
- 4. Menghitung keluaran untuk *hidden layer* dengan fungsi aktivasi. Urutan perhitungan sesuai tahap-tahapan diproses *training* yang ditunjukkan pada Gambar 4.6.
- 5. Menghitung *output* prediksi dengan *output weight* yang didapatkan pada proses *training*. Untuk menghitung *output* prediksi ditunjukkan pada Gambar 4.15.
- 6. Hasil yang didapatkan dari proses perhitungan *output* prediksi di denormalisasi yang ditunjukkan pada proses Gambar 4.17. Proses denormalisasi mendapatkan hasil Prediksi menggunakan ELM yang ditunjukkan pada proses Gambar 4.18
- 7. Evaluasi adalah hasil berupa perhitungan tingkat kesalahan menggunakan MAPE. Diagam alir pada MAPE ditunjukkan pada proses Gambar 4.19



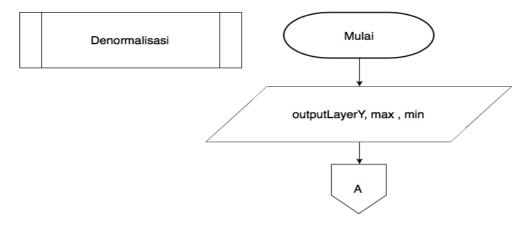


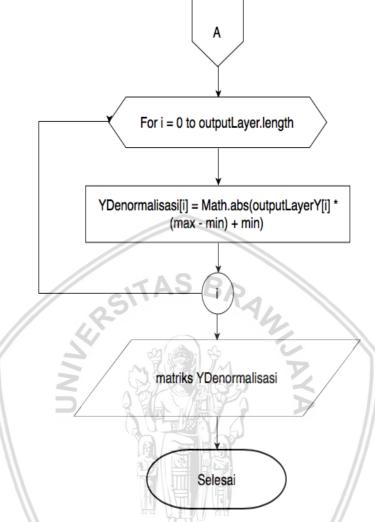
Gambar 4.17 Diagram Alir Output Prediksi

Berdasarkan Gambar 4.17 dapat dilihat langkah-langkah dalam melakukan perhitungan mencari nilai *output* prediksi yang diuraikan sebagai berikut:

- 1. Masukkan yang diterima sistem berupa matriks keluaran hidden layer dengan fungsi aktivasi dan matriks nilai output weight.
- 2. Hal ini agar mendapatkan *output* prediksi, matriks keluaran *output weight* dikalikan dengan matriks hidden layer yang sudah di fungsi aktivasi.
- 3. Hasil untuk keluaran dalam sistem berupa keluaran pada *output layer* (y). Untuk keluaran tersebut merupakan *output* hasil proses *testing*.

4.3.5 Proses Denormalisasi



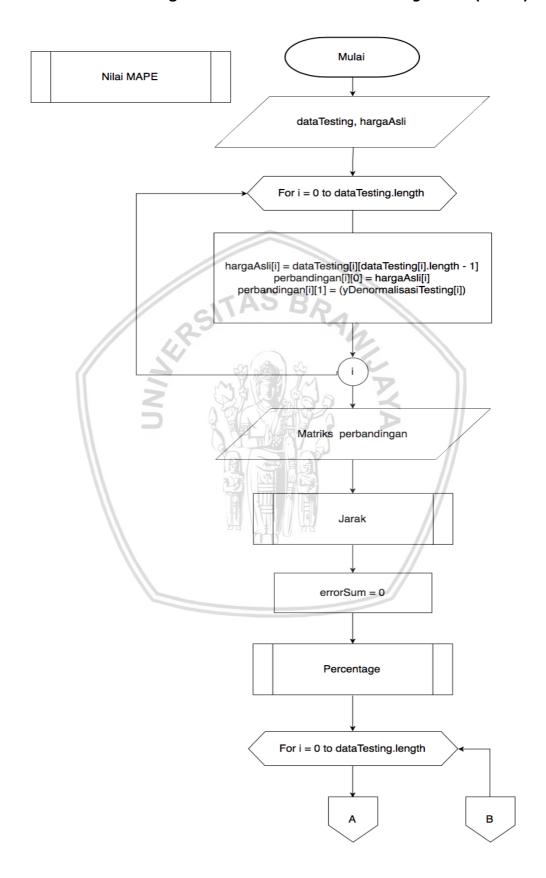


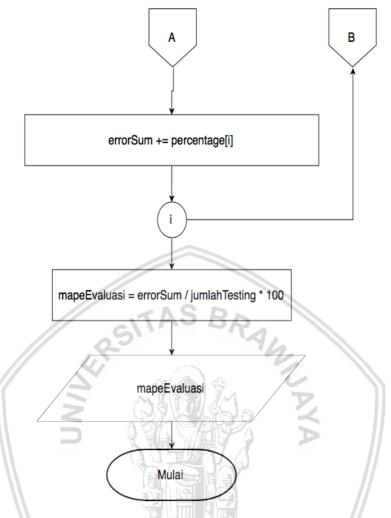
Gambar 4.18 Diagram Alir Denormalisasi Data

Berdasarkan Gambar 4.18 dapat dilihat langkah-langkah dalam melakukan perhitungan untuk proses denormalisasi yang diuraikan sebagai berikut:

- 1. Masukkan pada sistem merupakan matriks keluaran untuk *output layer* dengan nilai *maksimum* (*max*), dan nilai *minimum* (*min*).
- 2. Sistem melakukan perhitungan dengan proses denormalisasi yang ditunjukkan pada Persamaan 2.6.
- 3. Hasil denormalisasi berupa data hasil prediksi.

4.3.6 Proses Perhitungan Nilai Mean Absolute Percentage Error (MAPE)

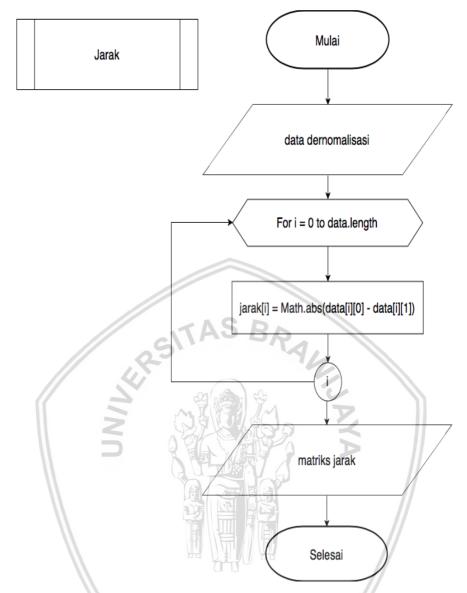




Gambar 4.19 Diagram Alir Evaluasi Hasil

Berdasarkan Gambar 4.19 dapat dilihat langkah-langkah dalam melakukan perhitungan mendapatkan hasil MAPE yang diuraikan sebagai berikut:

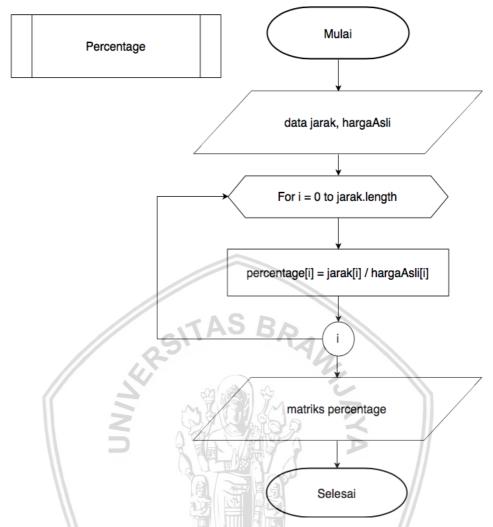
- 1. Masukkan pada sistem merupakan data hasil prediksi yang telah di denormalisasi.
- 2. Menghitung nilai MAPE yang ditunjukkan pada Persamaan 2.14
- 3. Hasil berupa nilai keluaran MAPE yaitu nilai error.



Gambar 4.20 Diagram Alir Proses Mencari Jarak

Berdasarkan Gambar 4.20 dapat dilihat langkah-langkah dalam melakukan perhitungan jarak MAPE yang diuraikan sebagai berikut:

- 1. Masukkan pada sistem merupakan data hasil prediksi yang telah di denormalisasi.
- 2. Menghitung nilai MAPE yang ditunjukkan pada Persamaan 2.14
- 3. Hasil berupa nilai keluaran MAPE yaitu mendapatkan jarak harga aktual dengan harga yang sudah di denormalisasi.



Gambar 4.21 Diagram Alir Proses Menjadi Persentasi

Berdasarkan Gambar 4.21 dapat dilihat langkah-langkah dalam melakukan perhitungan persen (%) yang diuraikan sebagai berikut:

- 1. Masukkan pada sistem merupakan data hasil prediksi yang telah di denormalisasi.
- 2. Menghitung nilai MAPE yang ditunjukkan pada Persamaan 2.14
- 3. Hasil berupa nilai keluaran MAPE yaitu nilai error dengan satuan persen (%).

4.4 Perhitungan Manual

Perhitungan manual adalah contoh perhitungan dari suatu perancangan sistem yang dibuat agar dapat mengetahui tujuan dalam membuktikan dari perhitungan manual terdapat dari langkah-langkah proses perhitungan yaitu normalisasi, proses *training* dan proses *testing*.

Perhitungan manual dalam penelitian ini menggunakan 12 sampel data dengan pembagian data training 70% dan data testing sebanyak 30%. Sehingga didapatkan 9 data training dan 3 data testing. Jumlah neuron pada hidden layer

yang digunakan untuk perhitungan manualisasi ini sebanyak 2 fungsi namun pada proses ini menggunakan fungsi aktivasi sigmoid. Berikut untuk data yang digunakan ditunjukkan pada Tabel 4.2

Tabel 4.2 Data Training dan Data Testing

Data Harga Pasar Konsumen tahun 2017								
Tahun	No	Bulan	X1	X2	Х3	X4	X5	Т
2017	1	Januari	111,412	111,433	111,451	111,433	111,432	111,612
	2	Februari	111,433	111,451	111,433	111,432	111,612	113,078
	3	Maret	111,451	111,433	111,432	111,612	113,078	113,412
	4	April	111,433	111,432	111,612	113,078	113,412	112,040
	5	Mei	111,432	111,612	113,078	113,412	112,040	111,438
	6	Juni	111,612	113,078	113,412	112,040	111,438	111,006
	7	Juli	113,078	113,412	112,040	111,438	111,006	111,000
	8	Agustus	113,412	112,040	111,438	111,006	111,000	111,032
	9	September	112,040	111,438	111,006	111,000	111,032	111,000
	10	Oktober	111,438	111,006	111,000	111,032	111,000	111,609
	11	November	111,006	111,000	111,032	111,000	111,609	111,326
	12	Desember	111,000	111,032	111,000	111,609	111,326	112,103

4.4.1 Perhitungan Normalisasi Data

Data berikut merupakan data sekunder yang dilihat pada Tabel 4.2 dinormalisasi terlebih dahulu dengan menggunakan *Min-Max Normalization* untuk proses pengolahan data. Berikut ini merupakan langkah-langkah manualisasi untuk data yang dihitung berdasarkan Persamaan 2.5

Langkah 1: menentukan nilai *minimal* dan *maksimal* dari data keseluruhan. Berikut hasil nilai minimal dan maksimal yang ditunjukkan pada Tabel 4.3

Tabel 4.3 Nilai Maksimal dan Minimal

MAX	113,412
MIN	111,000

Langkah 2: menghitung nilai untuk dinormalisasi data dengan menggunakan metode dengan perhitungan *Min-Max Normalization* untuk menghitung keperluan pengolahan data. Contoh perhitungan dalam normalisasi data, yaitu sebagai berikut:

$$x'_{1,1} = \frac{x_{1,1} - min}{max - min} = \frac{111.412 - 111.000}{113.412 - 111.000} = 0,170812604$$

Data yang telah dinormalisasi dimasukkan sebagai data inputan terdiri dari X1,X2,X3,X4,X5 dan target dengan membentuk suatu pola data. Berikut ini hasil pola perhitungan dengan data secara keseluruhan dapat ditunjukkan pada Tabel 4.4

Tabel 4.4 Normalisasi Data

Normalisasi data						
X1	X2	Х3	X4	X5	Т	
0,170812604	0,170812604	0,170812604	0,179519071	0,179104478	0,253731343	
0,179519071	0,186981758	0,179519071	0,179104478	0,253731343	0,861525705	
0,186981758	0,179519071	0,179104478	0,253731343	53731343 0,861525705		
0,179519071 0,179104478		0,253731343	0,861525705 1		0,431177446	
0,179104478	0,253731343	0,861525705	1	0,431177446	0,18159204	
0,253731343	0,861525705	SITAT	0,431177446	0,18159204	0,002487562	
0,861525705	1	0,431177446	0,18159204	0,002487562	0	
1	0,431177446	0,18159204	0,002487562	0	0,013266998	
0,431177446	0,18159204	0,002487562	0	0,013266998	0	
0,18159204	0,002487562	2 6 0	0,013266998	0	0,252487562	
0,002487562	0	0,013266998	0	0,252487562	0,135157546	
1	1		0,921037124	0,843444227	0,810087809	

4.4.2 Inisialisasi Input Weight

Langkah yang dilakukan setelah *input* data dan normalisasi ialah inisialisasi *input weight*. Jumlah yang digunakan pada *hidden neuron* dalam manualisasi yaitu 5. Untuk penentuan nilai *input weight* yang didapatkan secara *random* dalam bentuk matriks yang berukuran matriks disesuikan dengan banyaknya *input layer* dan *hidden layer* seperti(jumlah *hidden neuron* x *input layer*). Untuk kasus ini, yang digunakan untuk *input layer* sebanyak 5 sehingga *input weight* memiliki ordo 2x5 sebanyak 10 nilai. Nilai untuk *input weight* memiliki *range* [-1,1] yang ditunjukkan pada Tabel 4.5

Tabel 4.5 Inisialisasi Input Weight

Bobot Awal Random						
X1 X2 X3 X4						
0,065137274	-0,52512742	0,193255482	-0,115656972	0,001256011		
-0,624566185	0,738883239	0,30180876	0,253123725	0,351273513		

Langkah yang dilakukan setelah *input* data, normalisasi ialah inisialisasi *input* weight dan setelah mendapatkan nilai *input* weight maka melakukan *transpose input* weight yang gunakan untuk melakukan perhitungan pada data *training*. Bentuk matriks yang di *transpose* memiliki *input* weight dengan ordo 5x2 sebanyak 10 nilai.

Tabel 4.6 Inisialisasi Input Weight Setelah di Transpose

Transpose Bobot Awal			
X1	0,065137274	-0,624566185	
X2	-0,52512742	0,738883239	
Х3	0,193255482	0,30180876	
X4	-0,115656972	0,253123725	
X5	0,001256011	0,351273513	

4.4.3 Inisialisasi Input Bias

Langkah yang dilakukan setelah mendapatkan *input weight* lalu mencari input bias yang didapatkan dari nilai acak . Jumlah yang digunakan pada *hidden neuron* dalam manualisasi yaitu 2. Untuk penentuan nilai *input weight* yang didapatkan secara *random* dalam bentuk matriks yang berukuran matriks disesuikan dengan banyaknya *hidden neuron* seperti (1 x *hidden neuron*). Untuk kasus ini, yang digunakan untuk *input layer* sebanyak 2 sehingga *input weight* memiliki ordo 1x2 sebanyak 2 nilai. Nilai untuk *input weight* memiliki *range* [0,1] yang ditunjukkan pada Tabel 4.7

Tabel 4.7 Inisialisasi Input Bias

Bi	as T
0,310069605	0,532381179

4.4.4 Perhitungan Proses Training

Langkah 1: Perhitungan ini mejelaskan proses training yang dilakukan untuk memperoleh bobot terbaik yang bertujuan untuk digunakan pada proses testing. Terdapat 9 data yang digunakan untuk proses training dengan 2 neuron pada hidden layer dan fungsi aktviasi sigmoid. Berikut ini merupakan langkah-langkah dalam perhitungan proses training menggunakan metode ELM.

Langkah 1.1: melakukan perhitungan nilai keluaran untuk hidden layer menggunakan fungsi aktivasi. Keluaran pada hidden layer dapat dilakukan perhitungan dengan persamaan 2.2, kemudian untuk melakukan perhitungan fungsi aktivasi menggunakan fungsi aktivasi Sigmoid Biner. Hasil untuk keluaran hidden layer dengan fungsi aktivasi ini memiliki ordo 9x2 Berikut ini contoh perhitungan keluaran pada hidden layer dan hasilnya.

$$H_{init\ 1,1} = (N * w^T) + b$$

$$H_{init 1,1} = ((0,170812604 * 0,065137274 + 0,170812604 * 0,52512742 + 0,170812604 * 0,193255482 + 0,179519071 * 0,115656972 + 0,179104478 * 0,001256011)) + 0,310069605 = 0,243970287$$

Tabel 4.8 Matriks Keluaran Hidden Layer

H init		
0,243970287	0,711815908	
0,237870792	0,747062446	
0,234327909	0,969154212	
0,178359846	1,198521384	
0,239873854	1,272597361	
0,017801089	1,48521454	
-0,096612235	0,910156794	
0,183589732	0,281840512	
0,243293767	0,402668754	

Langkah 1.2: menghitung fungsi aktivasi

$$H(x)_{1,1} = \frac{1}{1 + e^{-x}} = \frac{1}{1 + e^{-0.243970287}} = 0.560691831$$

Berikut ini merupakan hasil keluaran *hidden layer* menggunakan fungsi aktivasi sigmoid biner secara perolehan perhitungan keseluruhan dapat dilihat pada Tabel 4.6

Tabel 4.9 Matriks Keluaran Layer Menggunakan Fungsi Aktivasi

Fungsi Hic	lden Layer
0,560691831	0,670802285
0,559188873	0,678538284
0,558315382	0,724950882
0,544472127	0,768261642
0,559682562	0,781187049
0,504450155	0,815358915
0,475865711	0,713032247
0,545768951	0,569997392
0,560525186	0,599328688

Langkah 2: Mencari keluaran hasil matriks *Moore-Penrose Generalized Inverse* dari Persamaan 2.7 yaitu $H^+ = (H^T H)^{-1} H^T$

Langkah 2.1: mencari matriks *transpose* dari hasil matriks H(x). hasil *transpose* memiliki ordo 5x8 yang ditunjukkan pada Tabel 4.7

Tabel 4.10 Transpose Matriks Keluaran Hidden Layer dengan Fungsi Aktivasi

Transpose Hidden Layer (HT)				
1 2 3 9				9
0,560691831 0,559188873 0,558315382 0,544472127 0,56052518				0,560525186
0,670802285 0,678538284 0,724950882 0,768261642 0,599328688				

Langkah 2.2: Proses mengalikan matriks *transpose* dengan matriks H(x). hasil dari perkalian matriks ini menghasilkan matriks ordo 2x2. Berikut ini contoh perhitungan dan hasil proses perhitungan yang ditunjukkan pada Tabel 4.8.

$$(H^T H)_{1,1} = (0,560691831 * 0,560691831 + 0,559188873 * 0,559188873 + + ..._{(n)} * ..._{(n)} + 0,560525186 * 0,560525186) = 2,641448424$$

Tabel 4.11 Perkalian Hasil *Transpose* dengan Keluaran *Hidden Layer*Menggunakan Fungsi Aktivasi

Н	T*H
2,641448424	3,413450698
3,413450698	4,493739893

Langkah 2.3: Menghitung matriks *inverse* yang dihasilkan dari perhitungan $(H^T H)$. Hasil dari *inverse* matriks ini memiliki ordo 2x2 menggunakan perhitungan yaitu OBE. Berikut merupakan langkah-langkah perhitungan mendapatkan *inverse* matriks menggunakan OBE. Hasil perhitungan *inverse* yang ditunjukkan Tabel 4.9.

1. Bentuk Matriks Identitas (I)

$$[H|I] = \begin{bmatrix} 2,641448424 & 3,413450698 \\ 3,413450698 & 4,493739893 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

2. Tukar baris 2 (R2) dan baris (R1)

$$[H|I] = \begin{bmatrix} 2,641448424 & 3,413450698 \\ 3,413450698 & 4,493739893 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

3. R1: 2,641448424 -> R1

$$[H|I] = \begin{bmatrix} 1 & 1,394285094 \\ 0,91854066429 & 1,704184924 \end{bmatrix} \begin{bmatrix} 1,517935077 & 0 \\ 0 & 1 \end{bmatrix}$$

4. R2- (3,413450698 * R1) -> R2

$$[H|I] = \begin{bmatrix} 1 & 1,394285094122719 \\ 0 & 0,839290403785639 \end{bmatrix} \qquad \begin{bmatrix} 1,51793507715284 & 0 \\ -1,51385827568929 & 1 \end{bmatrix}$$

5. R2: 4.493739893 -> R2

$$[H|I] = \begin{bmatrix} 1 & 1,394285094 \\ 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1,517935077 & 0 \\ -1,803735952 & 1,1914827 \end{bmatrix}$$

6. R1 3,413450698* R2 -> R1

$$[H|I] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 20,67772286 & -15,70044463 \\ -15,70044463 & 12,14359942 \end{bmatrix}$$

Tabel 4.12 Inverse Matriks

Inverse HT*H-1		
20,67772286	-15,70044463	
-15,70044463	12,14359942	

Langkah 2.4: Menghitung matriks *Moore-Penrose Pseudo Inverse* dengan mengalikan matriks *inverse* dengan matriks yang telah di*transpose* dari H(x). hasil dari perkalian matriks ini memiliki ordo 5x8. Berikut ini contoh dari perhitungan dan hasil yang ditunjukkan pada Tabel 4.10

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

 $H_{1,1}^+ = ((20,67772286 * 0,560691831) + (-15,70044463 * 0,670802285)))$
 $= 1,061936162$

Tabel 4.13 Matriks Moore-Penrose Generalized Inverse

\\	5 2	H		
1	2	3	••••	9
1,061936162	0,909399783	0,162639548	-0,803605609	2,180657578
-0,657156814	-0,539616828	0,037713371	0,781007132	-1,522487146

Langkah 3: Menghitung *output weight* yang mengalikan matriks *Moore-Penrose Pseudo Inverse* dengan matriks target. Hasil dari perhitungan ini digunakan untuk proses *testing* yang bertujuan menentukan *output* yang dihasilkan. Pada kasus ini *output weight* berupa matriks dengan ordo 5x1. Berikut contoh perhitungan dan hasil perhitungan ditunjukkan pada Tabel 4.11

$$\beta = H^+T$$

 $\beta_{1,1} = (1,061936162*0,253731343) + (0,909399783*0,861525705) + + ..._{(n)} * ..._{(n)} + 2,180657578*0,013266998) = 0,768490096$

Tabel 4.14 Nilai Output Weight

β
0,768490096
-0,147129654

4.4.5 Perhitungan Proses Testing

Proses testing dilakukan ini merupakan berdasarkan input weight dan output weight yang didapatkan dari proses training.

Langkah 1: Menghitung keluaran pada hidden layer dengan fungsi aktivasi. Untuk proses ini pencarian dalam output yang dikeluarkan hidden layer yang diproses testing sama dengan yang dilakukan pada proses training. Hasil untuk keluaran hidden layer ini dengan fungsi aktivasi berupa matriks yaitu ordo 4x5. Berikut merupakan contoh perhitungan manual dan hasil dari perhitungan keluaran hidden layer yang ditunjukkan pada Tabel 4.12

Langkah 1.1: Menghitung keluaran hidden layer

$$H_{1,1} = N * w^T$$

 $H_{init\ 1,1} = ((0.181592039*0.065137274 + 0.00248756218* - 0.52512742 + 0*0.193255482 + 0.0132669983* - 0.115656972 + 0*0.001256011) + 0.310069605$

 $H_{init 1,1} = 0.319057307$

Tabel 4.15 Matriks Keluaran Hidden Layer

(H)	init
0,319057307	0,424161142
0,313112685	0,623523821
0,274070553	0,6535718

Langkah 1.2: Menghitung fungsi aktivasi

$$H(x)_{1,1} = \frac{1}{1 + e^{-x}}$$

$$H(x)_{1,1} = \frac{1}{1 + e^{-0.319057307}} = 0.579094494$$

Tabel 4.16 Matriks Keluaran Hidden Layer Menggunakan Fungsi Aktivasi

Fungsi <i>Hidden Layer</i> (H(x)		
0,579094494	0,604478545	
0,57764485	0,651019562	
0,568091946	0,657814908	

Langkah 2: Menghitung nilai hasil prediksi (*0*). Langkah selanjutnya adalah menghitung nilai hasil dari prediksi dengan *output weight* yang didapatkan pada proses *training*. hasil yang didapatkan dari matriks *output* prediksi ini memiliki matriks ordo 4x1. Berikut merupakan contoh perhitungan dan hasil seluruh perhitungan ditunjukkan pada Tabel 4.18

$$O = H(x).\beta$$

$$O_{1,1} = (0.579094494 * 0.768490096) + (0.604478545 * -0.147129654)$$

 $O_{1,1} = 0.356091664$

Tabel 4.17 Hasil Keluaran pada Output Layer

Hasil Prediksi (y)
0,356091664
0,348130063
0,339788954

Perhitungan denormalisasi data merupakan proses *testing* yang digunakan dalam persaman yang sama dengan proses *trainng*. Berikut ini contoh dari manual dan hasil perhtungan yang ditunjukkan pada Tabel 4.19.

$$x_{1,1} = ABS(x^{1}(x_{max} - x_{min})) + x_{min}$$

$$x_{1,1} = (0.356091664 * (113.412 - 111.000) + 111.000) = 111.859$$

Tabel 4.18 Denormalisasi Data

YC	enormalisasi (Hasil P	rediksi)
	111.859	~
9	111.840	D
	111.820	

4.5 Perhitungan Nilai Mean Absolute Percentage Error (MAPE)

Dalam tahapan terkahir ini pada proses *testing* ialah menghitung tingkat *error* atau dapat disebut *error rate*. Proses mendapatkan *error rate* pada proses *testing* menggunakan persamaan yang sama pada proses *training*. Berikut ini adalah contoh dari perhitungan nilai MAPE yang diproses untuk ditahap *testing* dan hasil perhitungan yang ditunjukkan pada Tabel 4.20.

$$\begin{aligned} \mathit{MAPE} &= \frac{1}{n} \sum_{i=1}^{n} |\frac{\hat{y}_i - y_i}{\hat{y}_i}| \times 100\% \\ \mathit{MAPE} &= \left(abs\left(\frac{111.609 - 111.859}{111.609} \times 100\%\right) + abs\left(\frac{111.326 - 111.840}{111.326} \times 100\%\right) + abs\left(\frac{112.103 - 111.820}{112.103} \times 100\%\right)\right) = 0.312719319 \end{aligned}$$

Tabel 4.19 Hasil Keluaran Nilai MAPE Proses Testing

MAPE								
У	Yi	у-уі	y-yi/y					
111.609	111.859	250	0,223900485					

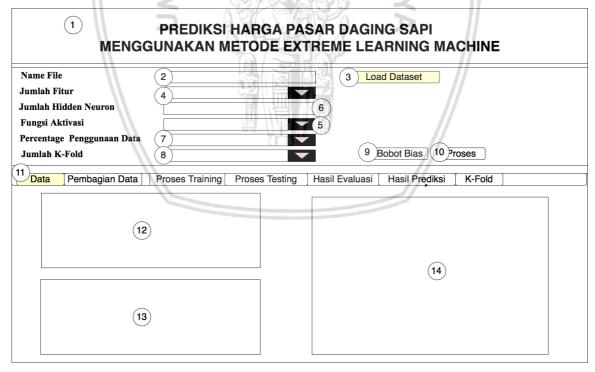
111.326	111.840	514	0,461428338
112.103	111.820	283	0,252829133
		Sum	0,938157957
		MAPE	0,312719319

4.6 Perancangan Antar Muka

Perancangan antar muka, aplikasi untuk penelitian ini dibuat 6 halaman menu utama. Hal ini sangat membutuhkan untuk menggambarkan agar dapat diimplementasikan ke sistem prediksi harga pasar daging sapi dengan menggunakan ELM yang dibangun. Menu yang digunakan antara lain halaman impor data, halaman bobot, halaman normalisasi, halaman training, halaman testing, halaman evaluasi.

4.6.1 Perancangan Halaman Impor Data

Halaman *impor* data ini adalah halama awal yang ketika aplikasi dijalankan. Halan ini digunakan pengguna untuk melakukan *entry* data dari *storage* berupa *file* dengan *file* ekstensi.xls. berikut perancangan halaman *impor* data yang ditunjukkan pada Gambar 4.22



Gambar 4.22 Perancangan Halaman Impor Data

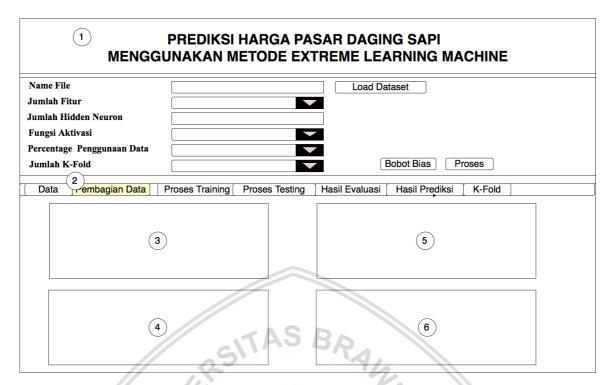
Berikut keterangan mengenai rancangan antarmuka halaman *impor* data yang dijelaskan sebagai berikut:

1. Header Program.

- 2. Tampilan untuk memilih data yang ingin digunakan dengan melakukan memasukkan data secara *impor*.
- 3. Nama file yang telah dimasukkan ditampilkan.
- 4. Melakukan pengambilan data yang ingin digunakan.
- 5. Memasukkan jumlah fitur yang ingin digunakan.
- 6. Memasukkan jumlah hidden neuron yang ingin digunakan.
- 7. Memilih perhitungan aktivasi sigmoid yang ingin digunakan.
- 8. Memilih perbandingan data yang ingin digunakan yaitu data *training* dan *testing*.
- 9. Untuk menampilkan ketahap semua proses-proses data yang ingin ditampilkan.
- 10. Menu bar berwarna kuning menandakan bahwa sedang aktif digunakan yaitu halaman Menu.
- 11. Textbox untuk menampilkan data
- 12. Textbox untuk menampilkan dataset yang digunakan
- 13. Textbox untuk menampilkan data sesuai fitur
- 14. Textbox untuk menampilkan data yang sudah dinormalisasi sesuai fitur

4.6.2 Perancangan Halaman Pembagian Data

Halaman bobot merupakan yang menampilkan *input weight*. Nilai yang didapatkan dari *input weight* berupa nilai *random* yang terbentuk matriks (jumlah *hidden neuron* x jumlah *input* fitur) dan bias yang terbentuk (1 x *hidden neuron*). Berikut ini perancangan untuk halaman bobot yang ditunjukkan pada Gambar 4.23.



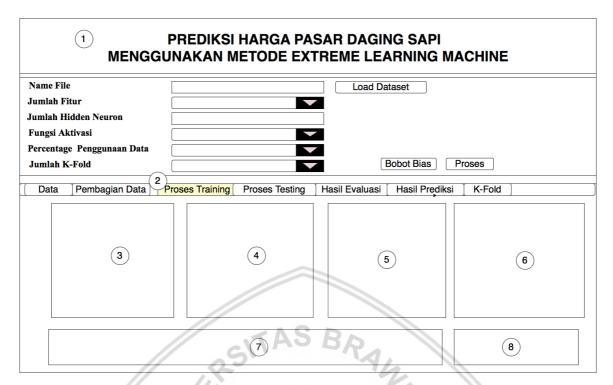
Gambar 4.23 Perancangan Halaman Bobot

Berikut keterangan mengenai rancangan antarmuka halaman bobot yang dijelaskan sebagai berikut:

- 1. Header program.
- 2. Menu pilihan pembagian Data yang menampilkan data.
- 3. Textbox yang menampilkan data training sudah normalisasi.
- 4. Textbox untuk menampilkan hasil bobot secara diacak.
- 5. Textbox untuk menampilkan data testing yang sudah dinormalisasi.
- 6. Textbox untuk menampilkan hasil bias secara diacak.

4.6.3 Perancangan Halaman Training

Halaman *training* merupakan tampilan hasil dari setiap perhitungan yang sudah diproses *training*. Dimulai dari perhitungan keluaran *hidden layer* dengan fungsi aktivasi hingga perhitungan nilai MAPE. Berikut merupakan perancangan halaman *training* yang ditunjukkan pada Gambar 4.24.



Gambar 4.24 Perancangan Halaman Proses Training

Berikut keterangan mengenai rancangan antarmuka halaman proses training yang dijelaskan sebagai berikut:

- 1. Header Program.
- 2. Menu bar berwarna kuning menandakan bahwa sedang aktif digunakan yaitu Menu Normalisasi.
- 3. Textbox untuk menampilkan hasil setiap proses perhitungan pada H_{init} .
- 4. Textbox untuk menampilkan hasil setiap perhitungan pada fungsi aktivasi.
- 5. *Textbox* untuk menampilkan hasil setiap perhitungan pada proses perkalian *transpose*.
- 6. Textbox untuk menampilkan hasil setiap perhitungan pada proses matriks OBE.
- 7. Textbox untuk menampilkan hasil setiap perhitungan pada bobot akhir.
- 8. Textbox untuk menampilkan hasil setiap perhitungan pada MAPE training.

4.6.4 Perancangan Halaman Testing

Halaman testing merupakan tampilan dari hasil setiap perhitungan yang ada pada proses testing. Dimulai dari perhitungan keluaran hidden layer dengan fungsi aktivasi sampai hingga perhitungan nilai MAPE. Berikut merupakan perancangan halaman testing yang ditunjukkan pada Gambar 4.25



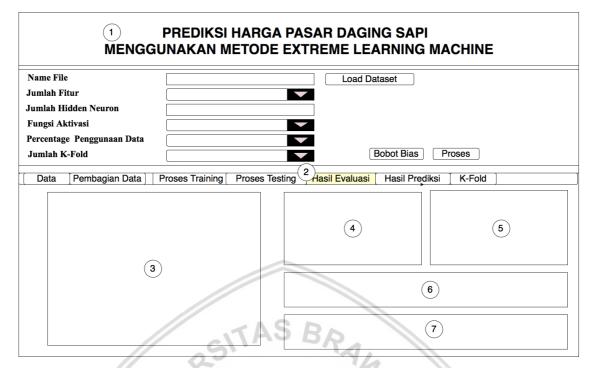
Gambar 4.25 Perancangan Halaman Proses Testing

Berikut keterangan mengenai rancangan antarmuka halaman proses testing yang dijelaskan sebagai berikut:

- 1. Header Program.
- 2. Menu *bar* berwarna kuning menandakan bahwa sedang dipilih yaitu menu *training*.
- 3. *Textbox* untuk menampilkan hasil dari perhitungan yang telah diproses *hidden layer*.
- 4. *Textbox* untuk menampilkan hasil setiap proses perhitungan pada fungsi aktivasi bobot.
- 5. *Textbox* untuk menampilkan hasil setiap perhitungan hasil prediksi *output* prediksi.
- 6. *Textbox* untuk menampilkan hasil setiap perhitungan hasil prediksi denormalisasi.

4.6.5 Perancangan Halaman Hasil Evaluasi

Halaman evaluasi merupakan hasil prediksi antara data aktual dan data ramalan dari proses *training* dan proses *testing*. Berikut merupakan perancangan halaman hasil evaluasi yang ditunjukkan pada Gambar 4.26



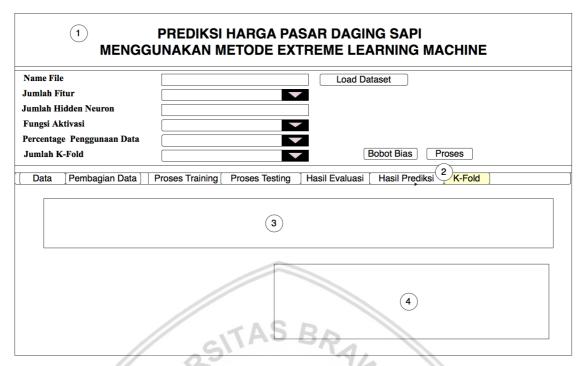
Gambar 4.26 Perancangan Halaman Hasil Evaluasi

Berikut keterangan mengenai rancangan antarmuka halaman hasil evaluasi yang dijelaskan sebagai berikut:

- 1. *Header* Program.
- 2. Menu *bar* berwarna kuning menandakan bahwa sedang aktif digunakan yaitu menu *testing*.
- 3. *TextBox* untuk menampilkan hasil dari perhitungan yang telah diproses pada tahap *testing*.
- 4. *Textbox* untuk menampilkan harga aktual dan harga setelah dilakukan denormalisasi.
- 5. *Textbox* untuk menampilkan jarak harga dari harga aktual dan harga denormalisasi.
- 6. Textbox untuk menampilkan jumlah error dari keduanya.
- 7. Textbox untuk menampilkan perhitungan hasil prediksi MAPE testing.

4.6.6 Perancangan Halaman K-Fold

Halaman $cross\ validation\ merupakan\ hasil\ prediksi data ramalan dari proses <math>training\ dan\ proses\ testing\$. Berikut merupakan perancangan halaman k- $fold\$ yang ditunjukkan pada Gambar 4.24



Gambar 4.27 Perancangan Halaman K-Fold

Berikut keterangan mengenai rancangan antarmuka halaman k-fold yang dijelaskan sebagai berikut:

- 1. Header Program.
- 2. Menu *bar* berwarna kuning menandakan bahwa sedang aktif digunakan yaitu menu Evaluasi.
- 3. Menampilkan data aktual, data hasil prediksi dan hasil tingkat kesalahan atau MAPE terhadap data dari proses *training* dan *testing*.
- 4. Textbox untuk menampilkan perhitungan hasil prediksi MAPE hasil K-Fold.

4.7 Perancangan Uji Coba dan Evaluasi

Pengujian dan pembahasan untuk penelitian ini adalah untuk mengethaui kinerja metode *Extreme Learning Machine* (ELM) dalam pemecahan suatu permasalahan. Skenario pengujian yang dilakukan antara lain sebagai berikut:

- 1. Pengujian range input weight.
- 2. Pengujian jumlah neuron.
- 3. Pengujian Fungsi Aktivasi.
- 4. Pengujian untuk perbandingan data training dan data testing.
- 5. Pengujian untuk perbandingan data menggunakan Cross Validation.

4.7.1 Pengujian Input Jumlah Fitur

Pengujian berdasarkan analisis data yang digunakan untuk mengetahui jenis data terbaik yang cocok digunakan untuk menghasilkan nilai kesalahan yang minimal pada prediksi harga pasar daging sapi. Terdapat 5 jenis analisis, yaitu jumlah fitur 1 sampai 5. Inisialisasi parameter yang digunakan pada pengujian ini adalah input weight dengan range [-1,1], jumlah neuron yaitu 5, fungsi aktivasi sigmoid biner, serta perbandingan jumlah data training dan data testing sebesar 90%:10%. Input weight dan bias dilakukan untuk mendapatkan nilai secara acak dengan 10 kali percobaan. Tabel 4.20 merupakan table rancangan pengujian fitur untuk mendapatkan hasil error terbaik. Berikut perancangan pengujian jumlah fitur dengan nilai MAPE yang ditunjukkan Tabel 4.20

Nilai MAPE (%) Percobaan ke-i Rata-Rata **Jumlah Fitur** Nilai 5 1 2 3 6 8 9 10 **MAPE (%)** 1 2 3 4 5

Table 4.20 Rancangan Pengujian Fitur Data

4.7.2 Pengujian Fungsi Aktivasi

Pengujian fungsi aktivasi digunakan untuk mengetahui fungsi aktivasi yang cocok untuk perhitungan dalam metode Extreme Learning Machine (ELM). Hal ini mendapatkan berupa hasil nilai keselahan yang baik atau minimal. Pada pengujian fungsi aktivasi ini terdapat 4 fungsi yang dilakukan pengujian yaitu fungsi aktivasi sigmoid biner, fungsi aktivasi sigmoid bipolar, fungsi aktivasi sin, fungsi aktivasi radial basis. Inisialisasi parameter untuk pengujian ini input weight dengan range [-1,1], untuk jumlah neuron yaitu7, jumlah fitur yaitu 3 serta dengan perbandingan jumlah data training dan testing sebesar 90%:10%. Didapatkan input weight dan bias ini dilakukan secara random dengan 10 kali percobaan. Berikut perancangan pengujian range input weight dengan nilai MAPE yang ditunjukkan Tabel 4.21.

Nilai MAPE (%) Percobaan ke-i Rata-Rata **Fungsi Aktivasi** Nilai 2 3 4 5 6 7 8 9 **MAPE (%)**

10

Table 4.21 Rancangan Pengujian Fungsi Aktivasi

1

Fungsi Sigmoid Biner						
Fungsi Sigmoid Bipolar						
Fungsi Sin						
Fungsi Radial Basis						

4.7.3 Pengujian Jumlah Hidden Neuron

Pengujian untuk jumlah *neuron* pada *hidden layer* digunakan untuk mengetahui nilai terbaik jumlah *hidden neuron* dalam *Extreme Learning Machine* (ELM) untuk menghasilkan nilai kesalahan yang kecil atau minimal. Banyak jumlah *neuron* pada *hidden layer* ini yang diuji coba dimulai dari angka 2 dan berhenti pada titik menunjukkan hasil konvergen. Inisialisasi untuk parameter yang digunakan pada pengujian ini ialah *input weight* dengan *range* [-1,1] dan perbandingan jumlah data *training* dan *testing* yang digunakan sebesar 90%:10%. *Input weight* yang dilakukan secara *random* dengan 10 kali percobaan. Berikut perancangan pengujian jumlah *neuron* pada *hidden layer* dengan nilai MAPE ditunjukkan pada Tabel 4.22.

Table 4.22 Rancangan Pengujian Jumlah Neuron

	N		_	144		112/1			4		
		1	Nil	ai MA	PE (%)	Perc	obaan	ke-i			Rata-Rata
Jumlah						大	7				Nilai
Neuron	1	2	3	4	5	6	7	8	9	10	MAPE (%)
1										//	
2						V W					
3					- G - D				//		
4											
5											
6											
7											
8											
9											
10											

4.7.4 Pengujian Perbandingan Data Training dan Data Testing

Pengujian perbandingan untuk jumlah data *training* dan data *testing* hasil terbaik dengan 6 jenis perbandingan yang terdiri dari 90%:10%, 80%:20%, 70%:30%, 60%:40%, 50%:50%, 40%:60%. Inisialisasi parameter ini digunakan pada

pengujian jumlah *range input weight* [-1;1]. *Input weight* ini dilakukan secara *random* 10 kali dalam percobaan. Berikut perancangan pengujian perbandingan jumlah data *training* dan data *testing* menggunakan nilai MAPE ditunjukkan pada Tabel 4.23.

Table 4.23 Pengujian Perbandingan Data Training dan Data Testing

Perbandingan jumlah		Vila	i M	APE	(%						
Data Training dan											Rata-Rata
Data Testing	1	2	3	4	5	6	7	8	9	10	Nilai MAPE (%)
Training: Testing											
90% : 10%											
80% : 20%											
70% : 30%											
60% : 40%			1	A	S	B	R				
50% : 50%	8	7,							1		
40% : 60%			8	23 6/		8					

4.7.5 Pengujian Perbandingan Data Menggunakan Cross Validation

Pengujian perbandingan *cross validation* ini dilakukan setelah mendapatkan pengujian fitur terbaik, *hidden neuron* terbaik, fungsi aktivasi terbaik. Setelah mendapatkan hasil terbaik dari proses tersebut maka proses *cross validation* ini untuk mendapatkan prediksi harga yang diinginkan. Inisialisasi parameter ini digunakan pada pengujian jumlah *range input weight* [-1;1]. *Input weight* ini dilakukan secara *random* 10 kali dalam percobaan. Berikut perancangan pengujian perbandingan jumlah data *training* dan data *testing* menggunakan nilai MAPE ditunjukkan pada Tabel 4.24.

Table 4.24 Pengujian Prediksi Harga Menggunakan K-Fold

Perbandingan jumlah		Nilai MAPE (%) Percobaan ke-i								Rata-Rata	
Data Training dan											Nilai
Data Testing											MAPE (%)
K-Fold	1	2	3	4	5	6	7	8	9	10	
K-1											
K-2											
K-3											

BAB 5 IMPLEMENTASI

Bab ini menjelaskan tentang hasil implementasi dari sistem yang telah dirancang sebelumnya, serta spesifikasi dan batasan implementasi dari sistem. Pada bab ini juga menjelaskan antarmuka dari sistem untuk prediksi harga menggunakan Metode *Extreme Learning Machine* (ELM).

5.1 Spesifikasi Sistem

Spesifikasi sistem menjelaskan suatu spesifikasi yang digunakan dalam membangun sistem yang dibuat. Spesifikasi dan membangun sistem terdiri dari spesifikasi perangkat keras dan perangkat lunak.

5.1.1 Spesifikasi Perangkat Keras

Perangkat keras digunakan dalam implementasi sistem bertujuan untuk menemukan hasil prediksi harga daging sapi ditunjukkan pada Tabel 5.1

Tabel 5.1 Lingkungan Perangkat Keras

Komponen	Spesifikasi
Prosesor	Intel Iris Grapghics 6100 1536 MB 2,7 GHz Intel Core i5
Memori	8 GB
Hardisk	128 GB

5.1.2 Spesifikasi Perangkat Lunak

Tabel 5.2 Lingkungan Perangkat Lunak

Nama	Spesifikasi
Sistem Operasi	macOS High Sierra
Bahasa Pemrograman	Java
Tools Pemrograman	NetBeans IDE

5.2 Batasan Implementasi

Dalam implementasi sistem ini menemukan prediksi harga sebagai penempatan harga daging sapi yang stabi dipasar dan tidak terjadinya perbedaan antara pedagang satu dengan yang lainnya terdapat beberapa batasan, sebagai berikut:

- 1. Input yang diterima dalam sistem berupa file.
- 2. Algoritme yang dapat dipilih adalah Extreme Learning Machine (ELM)

3. *Output* dari sistem ini adalah nilai jumlah fitur, jumlah *hidden neuron*, fungsi aktivasi, *percentage* perbandingan data, dan hasil prediksi harga untuk bulan selanjutnya.

5.3 Implementasi Program

Berdasarkan perancangan yang telah dilakukan pada bab 4, maka dibahas mengenai implementasi sistem sesuai dengan perancangan yang telah dibuat. Sistem diimplementasikan menggunakan bahasa Pemrograman Java dengan NetBeans IDE.

5.3.1 Implementasi Proses Normalisasi Data

Proses ini diawali dengan melakukan normalisasi terhadap seluruh data yang diolah untuk standarisasi data. Langkah awal adalah mencari nilai *maximum* dan *minimum* data pada setiap parameter. Proses normalisasi menggunakan *min-max normalization*. Proses normalisasi data dapat dilihat pada Kode Program 5.1.

```
Normalisasi Data
    public Dataset(int jumlahFitur, String pathFile)
2
            try {
3
4
              this.work = new File(pathFile);
             Workbook workbook = Workbook.getWorkbook(work);
5
              this.datasetSheet = workbook.getSheet(0);
6
              this.dataset = new
8
              String[datasetSheet.getRows()][datasetSheet.getColumns()];
9
10
                 min = Double.parseDouble(datasetSheet.getCell(2,
                 0).getContents().replace(",", ""));
11
                 max = Double.parseDouble(datasetSheet.getCell(2,
                 0).getContents().replace(",", ""));
13
14
15
                 for (int i = 0; i < dataset.length; i++) {
16
                     for (int j = 0; j < dataset[0].length; <math>j++) {
17
                         Cell cell = datasetSheet.getCell(j, i);
                         dataset[i][j] = cell.getContents().replace(",",
18
                         "");
19
20
21
                     if (Double.parseDouble(dataset[i][2]) < min) {</pre>
22
                         min = Double.parseDouble(dataset[i][2]);
23
                     if (Double.parseDouble(dataset[i][2]) > max) {
24
25
                         max = Double.parseDouble(dataset[i][2]);
```

```
26
27
28
                 this.fitur = new Double[datasetSheet.getRows() -
                 jumlahFitur][jumlahFitur + 1];
29
30
                 for (int i = 0; i < fitur.length; i++) {
                     int counter = i;
31
32
                     for (int j = 0; j < fitur[0].length; <math>j++) {
                         fitur[i][j] =
33
34
                  Double.parseDouble(dataset[counter][2]);
35
                         counter++;
36
37
38
                 this.normalisasi = new
                Double[fitur.length][fitur[0].length];
39
                 for (int i = 0; i < fitur.length; i++) {
40
                     for (int j = 0; j < fitur[0].length; j++) {</pre>
41
                         normalisasi[i][j] = (fitur[i][j] - min) / (max
42
43
                           min);
44
45
             } catch (IOException | BiffException ex) {
46
47
            Logger.getLogger(Dataset.class.getName()).log(Level.SEVERE,
48
           null, ex);
49
50
```

Kode Program 5.1 Proses Normalisasi Data

Penjelasan dari kode Program 5.1 adalah sebagai berikut.

- 1. Baris ke-3 sampai ke-8 merupakan proses *file* dari excel yang dimasukkan ke dalam java.
- 2. Baris ke-10 sampai ke-30 merupakan proses mencari nilai *minimum* dan *maximum* dari seluruh data.
- 3. Baris ke-31 sampai ke-37 merupakan pembagian kolom berdasarkan jumlah fitur yang ingin ditampilkan.
- 4. Baris ke-38 sampai ke-50 merupakan proses perhitungan normalisasi menggunakan *min-max normalization*.

5.3.2 Implementasi Proses Membuat Nilai Random Bobot dan Bias

Proses ini diawali dengan memasukkan nilai hidden neuron dan jumlah fitur yang diolah untuk mendapatkan nilai bobot dan bias. Langkah awal adalah mencari nilai hidden neuron untuk mengetahui panjang kolom dan baris untuk

yang dimasukkan dari *hidden neuron* dan jumlah fitur. Proses ini dapat dilihat pada Kode Program 5.2

```
Bobot dan Bias
    private void hitungBobotBias() {
2
            try {
3
                 int hiddenNeuron =
                 Integer.parseInt(textFieldHiddenNeuron.getText());
4
5
                 int jumlahFitur =
6
                 comboBoxJumlahFitur.getSelectedIndex();
7
                 List<String> bobotHeader = new ArrayList<>();
8
9
                 for (int i = 1; i \le jumlahFitur; i++) {
10
                     bobotHeader.add("X" + i);
11
12
13
                 bobot = new Double[hiddenNeuron][jumlahFitur];
                 for (int i = 0; i < bobot.length; i++) {</pre>
14
                    for (int j = 0; j < bobot[i].length; j++) {</pre>
15
                         bobot[i][j] = new Random().nextDouble() * 2 - 1;
16
17
18
19
                 tabelBobot.setModel(new DefaultTabelModel(bobot,
                 bobotHeader.toArray());
20
                 this.bias = new Double[hiddenNeuron];
21
22
                 Double[][] biasPrint = new Double[1][hiddenNeuron];
                 List<String> biasHeader = new ArrayList<>();
23
                 for (int i = 0; i < hiddenNeuron; i++) {</pre>
24
                     this.bias[i] = Math.random();
25
                    biasPrint[0][i] = this.bias[i];
26
27
                     biasHeader.add("B" + (i + 1));
28
29
30
                 tabelBias.setModel(new DefaultTabelModel(biasPrint,
31
                 biasHeader.toArray()));
32
            } catch (NumberFormatException e) {
                 JOptionPane.showMessageDialog(null, "Belum memasukkan
33
                 hidden neuron");
34
35
            }
        }
```

Kode Program 5.2 Proses Membuat Nilai Random Bobot dan Bias

Penjelasan dari Kode Program 5.2 adalah sebagai berikut.

- 1. Baris ke-3 sampai ke-4 merupakan proses untuk inisialisasi *hidden neuron* dengan masukkan nilai dari *text*.
- 2. Baris ke-5 sampai ke-6 merupakan proses untuk inisialisasi *jumlah fitur* dengan masukkan nilai dari *text*.
- 3. Baris ke-7 sampai ke-12 merupakan mencetak nilai sesuai jumlah fitur dengan bobot header X.
- 4. Baris ke-13 sampai ke-23 merupakan perhitungan untuk mendapatkan nilai bobot.
- 5. Baris ke-24 sampai ke-36 merupakan perhitungan untuk mendapatkan nilai bias.

5.3.3 Implementasi Proses Training

5.3.3.1 Implenetasi Proses Perhitungan H_{init}

Proses ini untuk mendapatkan nilai merupakan proses perhitungan perkalian data normalisasi dengan bobot yang didapat setelah itu dilakukan penambahan dengan bias yang didapatkan. Proses perhitungan $H_{\rm init}$ dapat dilihat pada Kode Program 5.3

```
Perhitungan H<sub>init</sub>
    public static Double[][] getHInit(Double[][] normalisasi, Double[][]
1
    bobot, Double[] bias, int hiddenNeuron, int jumlahFitur)
2
             Double[][] hInit = new
3
             Double[normalisasi.length][hiddenNeuron];
4
             for (int i = 0; i < hInit.length; i++) {
5
                 for (int j = 0; j < hInit[i].length; <math>j++) {
6
                     double sum = 0;
7
8
                     for (int k = 0; k < jumlahFitur; k++) {
9
                          sum += normalisasi[i][k] * bobot[j][k];
10
                          // calculate += "("+normalisasiTraining[i][k]+"
                            * "+bobot[j][k]+") +";
12
13
                     hInit[i][j] = sum + bias[j];
14
                 }
15
16
             return hInit; }
17
```

Kode Program 5.3 Proses Perhitungan H_{init}

Penjelasan dari Kode Program 5.3 adalah sebagai berikut.

- 4. Baris ke-2 sampai ke-7 merupakan proses inisialisasi dengan mendapatkan nilai dari data normalisasi yang diproses dengan bobot dan *bias*.
- 5. Baris ke-8 sampai ke-16 merupakan proses perhitungan H_{init}.

5.3.3.2 Implementasi Proses Perhitungan Hidden Layer dengan Fungsi Aktivasi

Proses ini merupakan proses perhitungan keluaran hidden layer dengan fungsi aktivasi dilakukan berdasarkan perhitungan yang telah dijelaskan pada bab sebelumnya. Perhitungan keluaran hidden layer. Dengan fungsi aktivasi pada proses training. Sama dengan yang dilakukan pada proses testing. Proses mencari keluaran hidden layer dengan fungsi aktivasi dapat dilihat pada Kode Program 5.4

```
Perhitungan Hidden Layer
               static Double[][] getFungsiAktivasi(Double[][] hInit,
      public
1
      String fungsiAktivasi) {
2
               Double[][] fungsiAktivasiValue = new
3
               Double[hInit.length][hInit[0].length];
4
               for (int i = 0; i < fungsiAktivasiValue.length; i++) {</pre>
5
                   for (int j = 0; j < fungsiAktivasiValue[i].length;</pre>
6
                        switch (fungsiAktivasi) {
8
                         case "Fungsi Sigmoid Biner":
9
                         fungsiAktivasiValue[i][j] =
10
                         FungsiAktivasi.fungsiSigmoidBiner(hInit[i][j]);
11
                             break;
12
                         case "Fungsi Sigmoid Bipolar":
13
                         fungsiAktivasiValue[i][j] =
14
                         FungsiAktivasi.fungsiSigmoidBiner(hInit[i][j]);
1.5
                                break;
16
                         case "Fungsi Sin":
17
                         fungsiAktivasiValue[i][j] =
18
                         FungsiAktivasi.fungsiSin(hInit[i][j]);
19
                                break;
20
                         case "Fungsi Hard Limit":
21
                         fungsiAktivasiValue[i][j] =
22
                         FungsiAktivasi.fungsiHardLimit(hInit[i][j]);
23
                                break;
24
                            default:
25
                                fungsiAktivasiValue[i][j] = null;
26
                                break:
27
28
29
```

30	}
31	return fungsiAktivasiValue;
32	}

Kode Program 5.4 Proses Perhitungan Hidden Layer

Penjelasan dari Kode Program 5.4 adalah sebagai berikut.

- 4. Baris ke-3 sampai ke-6 merupakan merupakan proses inisialisasi dengan data yang didapatkn dari $H_{\rm init}$.
- 5. Baris ke-7 sampai ke-32 merupakan proses perhitungan fungsi aktivasi yang diproses dengan fungsi sesuai pilihan tersebut.

5.3.3.3 Implementasi Proses Perhitungan Inverse Matriks Menggunakan OBE

Proses ini dimulai dengan membuat matriks identitas dan menggabungkan dengan matriks A, kemudian menghitung *inverse* pada matriks A dengan OBE. Kode Program pada proses ini dapat dilihat pada Kode Program 5.5.

```
Inverse Matriks
1
    public static Double[][] inverseOBE(Double[][] matrix) {
2
            int length = matrix.length;
            int doubleLength = length * 2;
3
            Double[][] inverse = new
4
5
            Double[matrix.length][matrix[0].length];
            Double[][] matriks = new Double[length][doubleLength];
6
            for (int i = 0; i < length; i++) {
7
                 for (int j = length; j < doubleLength; j++) {
8
9
                     if (i + length == j) {
                         matriks[i][j] = 1.0;
10
                     } else {
11
                         matriks[i][j] = 0.0;
12
13
                     matriks[i][j - length] = matrix[i][j - length];
14
15
16
            }
17
18
            for (int j = 0; j < length - 1; j++) {
                 for (int i = j + 1; i < length; i++) {
20
                     if (matriks[i][j] != 0) {
21
                         double value = matriks[j][j] / matriks[i][j];
22
                         for (int k = j; k < doubleLength; k++) {
23
                             matriks[i][k] *= value;
24
                             matriks[i][k] -= matriks[j][k];
25
```

```
26
27
28
             }
29
30
             for (int j = length - 1; j > 0; j--) {
                 for (int i = j - 1; i >= 0; i--) {
                     if (matriks[i][j] != 0) {
32
                          double value = matriks[j][j] / matriks[i][j];
33
34
                          for (int k = i; k < doubleLength; k++) {
35
                              matriks[i][k] *= value;
36
                              matriks[i][k] -= matriks[j][k];
37
38
39
41
             for (int j
42
43
                 if (matriks[j][j] != 1)
                     double value = 1 / matriks[j][j];
44
45
                     for (int k = j; k < doubleLength; k++)
                         matriks[j][k] *= value;
46
47
48
49
50
             for (int i = 0; i < length; i++) {
51
                 for (int j = length; j < doubleLength; j++) {</pre>
52
53
                     inverse[i][j - length] = matriks[i][j];
54
55
56
57
            return inverse;
58
        }
59
```

Kode Program 5.5 Proses Perhitungan Inverse Matriks Menggunakan OBE

Penjelasan dari Kode Program 5.5 adalah sebagai berikut.

- 1. Baria ke-2 merupakan proses untuk inisialisasi *p* dengan panjang baris matriks A.
- 2. Baris ke-3 merupakan proses untuk inisialisasi pp dengan 2 kali nilai p.

- 3. Baris ke-4 merupakan proses inisialisasi matriks *inverse* untuk menampung hasil *inverse* matriks.
- 4. Baris ke-5 merupakan proses pembuatan identitas dan penggabungan dengan matriks A.
- 5. Baris ke-6 sampai ke-15 merupakan proses pembuatan identitas dan penggabungan dengan matriks A.
- 6. Baris ke-16 sampai ke-26 merupakan proses perhitungan OBE langkah pertama untuk menghitung *Inverse* matriks.
- 7. Baris ke-27 sampai Ke-37 merupakan proses perhitungan OBE langkah kedua untuk menghitung *inverse* matriks.
- 8. Baris ke-38 sampai ke-45 merupakan proses perhitungan OBE langkah ketiga untuk menghitung *inverse* matriks.
- 9. Baris ke-46 sampai ke-50 merupakan proses memasukkan hasil *inverse* matriks ke matriks *inverse*.

5.3.3.4 Implementasi Perhitungan Matriks Moore-Penrose Pseudo Inverse (H+)

Proses perhitungan matriks *moore-penrose pseudo inverse* pada proses *training*. Proses ini dapat dilihat pada Kode Program 5.6

```
Perhitungan perhitungan matriks Moore-Penrose Pseudo Inverse
1
              static Double[][] getHPlus(Double[][]
      Double[][] fungsiAktivasi,
                                    int hiddenNeuron, Double[][]
2
      normalisasi) {
3
              Double[][] HPlus = new
4
              Double[hiddenNeuron][normalisasi.length];
5
               for (int i = 0; i < HPlus.length; i++) {
6
                   for (int j = 0; j < HPlus[i].length; <math>j++) {
7
                       double sum = 0;
8
                               String calculate = "";
9
                       for (int k = 0; k < inverse.length; k++) {
10
                           sum += inverse[i][k] *
11
                           fungsiAktivasi[j][k];
12
                   calculate += "(" + inverse[i][k] + " * " +
13
                   fungsiAktivasi[j][k] + ") + ";
14
15
                               calculate += " = " + sum;
16
                               System.out.println(calculate);
17
                       HPlus[i][j] = sum;
18
                   }
19
              }
```

20	return HPlus;
21	}

Kode Program 5.6 Proses Perhitungan Matriks *Moore-Penrose Pseudo Inverse*

Penjelasan dari Kode Program 5.6 adalah sebagai berikut.

- 1. Baris ke-3 sampai ke-4 merupakan inisialisasi H_+ dengan didapatkan dari nilai niali hidden neuron.
- 2. Baris ke-5 sampai ke-21 merupakan proses perhitungan *moore-penrose* pseudo inverse yang dimulai dari perkalian matriks, kemudian inverse dan dilanjutkan dengan perkalian, lalu mendapatkan matriks inverse.

5.3.3.5 Implementasi Proses Perhitungan Output Weight

Proses perhitungan *output weight* merupakan perkalian antara matriks H_+ atau matriks *moore-penrose pseudo inverse* dengan matriks target dari hasil normalisasi dan jumlah *hidden neuron*. Proses perhitungan *output weight* dapat dilihat pada Kode Program 5.7

```
Perhitungan B (Bobot Akhir)
1
      public static Double[] getBobotAkhir(Double[][] HPlus, Double[][]
      normalisasi, int hiddenNeuron) {
2
               Double[] B = new Double[hiddenNeuron];
3
               for (int i = 0; i < B.length; i++) {
4
                 double sum = 0;
5
6
                    for (int j = 0; j < normalisasi.length; <math>j++) {
7
                        sum += HPlus[i][j] *
8
                        normalisasi[j][normalisasi[0].length - 1];
9
10
                   B[i] = sum;
11
12
               return
13
14
```

Kode Program 5.7 Proses Perhitungan Output Weight

Penjelasan dari Kode Program 5.7 adalah sebagai berikut.

- 1. Baris ke-1 sampai ke-7 merupakan proses untuk mendapatkan nilai matriks target yang sudah dinormalisasi.
- 2. Baris ke-8 sampai ke-14 merupakan proses perhitungan untuk mendapatkan nilai output weight yang merupakan perkalian matriks H_+ atau Moore-Penrose Pseudo Inverse dengan matriks implementasi proses testing.

5.3.3.6 Implementasi Proses Output Prediksi

Setelah mendapatkan nilai dari *output weight*, maka yang dilakukan pada proses pencarian nilai *output* prediksi yaitu mengalikan *output weight* dengan matriks fungsi aktivasi. Proses perhitungan *output* prediksi dapat dilihat pada kode program 5.8

```
Perhitungan Output Layer Y
1
    public static Double[]
                                getOutputLayerY(Double[]
    fungsiAktivasi) {
2
             Double[] outputLayerY = new Double[fungsiAktivasi.length];
3
             for (int i = 0; i < fungsiAktivasi.length; i++) {</pre>
4
                 double sum = 0;
5
6
                 for (int j = 0; j < fungsiAktivasi[i].length; j++) {</pre>
7
                      sum += fungsiAktivasi[i][j] * B[j];
8
9
10
11
             return outputLayerY;
12
13
```

Kode Program 5.8 Proses Perhitungan Output Prediksi

Penjelasan dari Kode Program 5.8 adalah sebagai berikut.

- 1. Baris 1-7 merupakan proses untuk mendapatkan nilai matriks keluaran *hidden layer*.
- 2. Baris 9-13 merupakan proses perhitungan mencari nilai *output* prediksi yang merupakan perkalian antara matriks keluaran *hidden layer* dengan matriks *output weight*.

5.3.3.7 Implementasi Proses *Y* Denormalisasi

Denormalisasi merupakan hasil prediksi yang bertujuan untuk mengetahui nilai prediksi serta untuk mengetahui perbandingan nilai aktual dengan nilai prediksi. Proses denormalisasi ini hasil pengembalian nilai dari proses normalisasi. Proses denormalisasi data yang dapat dilihat pada Kode Program 5.9

```
Perhitungan Y Denormalisasi

public static Double[] getYDenormalisasi(Double[] outputLayerY,

Double max, Double min) {

Double[] YDenormalisasi = new Double[outputLayerY.length];

for (int i = 0; i < YDenormalisasi.length; i++) {

YDenormalisasi[i] = Math.abs(outputLayerY[i] * (max - min) + min);
}</pre>
```

|--|

Kode Program 5.9 Proses Denormalisasi Data

Penjelasan dari Kode Program 5.9 adalah sebagai berikut.

- 1. Baris ke 1 sampai ke-4 merupakan proses mengambil jumlah data *training* atau *testing*.
- 2. Baris ke-5 sampai ke-8 merupakan proses denormalisasi.

5.3.4 Implementasi Proses Evaluasi

Proses perhitungan nilai MAPE merupakan proses untuk mengetahui nilai kesalahan yang didapatkan pada proses prediksi dengan metode *Extreme Learning Machine*. Proses perhitungan nilai MAPE dapat dilihat pada Kode Program 5.13

5.3.4.1 Implementasi Proses Mencari Jarak MAPE

Proses pembagi data jumlah *training* dan *testing*. Proses mencari jarak MAPE data yang dapat dilihat pada Kode Program 5.10

```
Perhitungan mencari jarak MAPE

1  public static Double[] getJarak(Double[][] data) {
2          Double[] jarak = new Double[data.length];
3          for (int i = 0; i < jarak.length; i++) {
4               jarak[i] = Math.abs(data[i][0] - data[i][1]);
5          }
6          return jarak;
7     }</pre>
```

Kode Program 5.10 Proses Mencari Jarak MAPE

Penjelasan dari Kode Program 5.10 adalah sebagai berikut.

1. Baris ke-3 sampai ke-7 merupakan proses untuk mendapatkan jarak dari harga data aktual dan hasil prediksi menggunakan ELM.

5.3.4.2 Implementasi Proses Percentage MAPE

Proses pembagi data jumlah *training* dan *testing*. Proses mencari rata-rata MAPE data yang dapat dilihat pada Kode Program 5.11

```
Perhitungan Percentage MAPE
   public
            static
                    Double[] getPercentage(Double[]
1
                                                         jarak,
                                                                  Double[]
   hargaAsli) {
           Double[] percentage = new Double[jarak.length];
3
            for (int i = 0; i < jarak.length; i++) {
4
                percentage[i] = jarak[i] / hargaAsli[i];
5
6
7
        return percentage;
8
```

9 | }

Kode Program 5.11 Proses Percentage MAPE

Penjelasan dari Kode Program 5.11 adalah sebagai berikut:

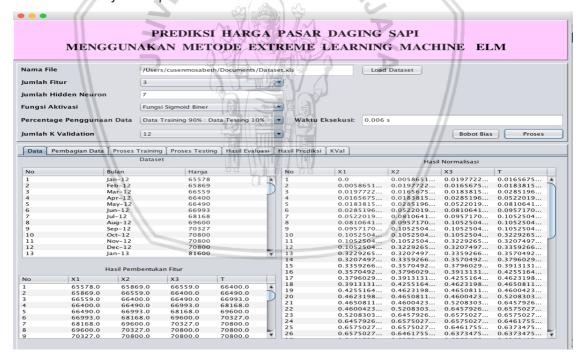
1. Baris ke-2 sampai ke-9 merupakan proses untuk mendapatkan nilai percentage MAPE.

5.4 Implementasi User Interface

Sub bab ini menjelasakan User Interface merupakan pembahasan mengenai hasil yang telah dilakukan pada bab 4 implementasi.

5.4.1.1 Implementasi Halaman Load Dataset

Halaman *load* dataset merupakan halaman yang digunakan untuk melakukan *input file* data ekstensi .xls, masukkan jumlah fitur, masukkan jumlah *hidden neuron*, pilih fungsi aktivasi yang digunakan, dan memilih penggunaaan data untuk data *training* dan data *testing*. Lalu terdapat tombol proses untuk memulai perhitungan pada sistem. Halaman data halaman yang menampilkan tabel dataset, hasil pembagian fitur dan hasil normalisasi. Implementasi halaman load data ditunjukkan pada Gambar 5.1.

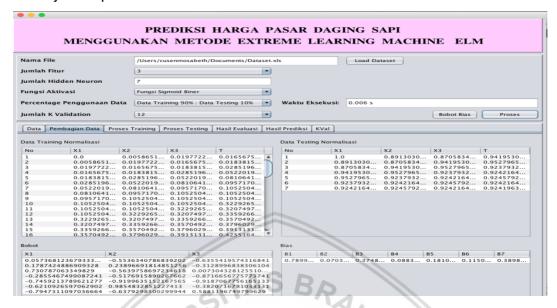


Gambar 5.1 Implementasi Halaman Load Datasets

5.4.1.2 Implementasi Halaman Pembagian Data

Halaman pembagian data merupakan halaman yang berisi tabel pembagian data training normalisasi dan data testing normalisasi, setelah itu

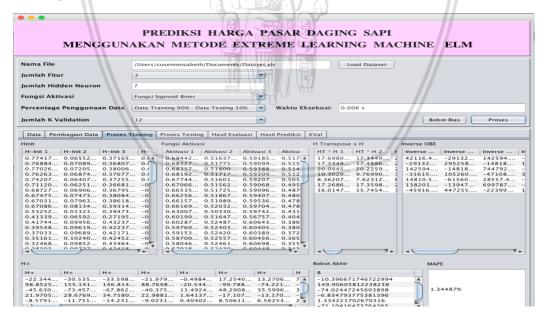
berisi tabel bobot dan tabel bias. Implementasi halaman pembagian data ditunjukkan pada Gambar 5.2.



Gambar 5.2 Implementasi Halaman Pembagian Data

5.4.1.3 Implementasi Halaman Proses Training

Halaman proses *training* merupakan hasil dari perhitungan yang berisi masing-masing proses pada proses *training*. Implementasi halaman *training* dapat dilihat pada Gambar 5.3.



Gambar 5.3 Implementasi Halaman Proses Training

5.4.1.4 Implementasi Halaman Proses Testing

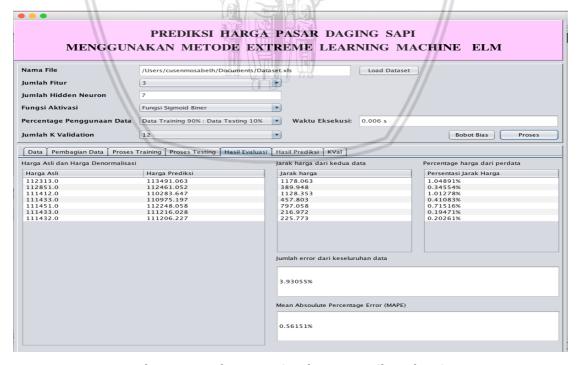
Halaman proses *testing* yang berisi perhitungan untuk masing-masing proses pada proses *testing*. Implementasi halaman *testing* dapat dilihat pada Gambar 5.4.



Gambar 5.4 Implementasi Halaman Proses Testing

5.4.1.5 Implementasi Halaman Hasil Evaluasi

Halaman hasil evaluasi berisi perhitungan untuk masing-masing proses pada hasil evaluasi. Implementasi halaman evaluasi dapat dilihat pada Gambar 5.5.



Gambar 5.5 Implementasi Halaman Hasil Evaluasi

BAB 6 PENGUJIAN DAN ANALISIS

Bab ini menjelaskan hasil pengujian dan analisis dari hasil uji coba yang telah dilakukan dalam prediksi harga prediksi daging sapi yang menggunakan metode *Extreme Learning Machine* untuk mengetahui dari sistem yang telah dibangun dan dilakukan dari hasil pengujian tersebut. Pada bab ini terdapat 4 macam pengujian yang dilakukan, yaitu:

- 1. Pengujian Variasi Fitur Data.
- 2. Pengujian Jumlah Neuron.
- 3. Pengujian Fungsi Aktivasi.
- 4. Pengujian Persentase Data Training dan Testing.
- 5. Pengujian data untuk perbandingan data menggunakan Cross Validation.

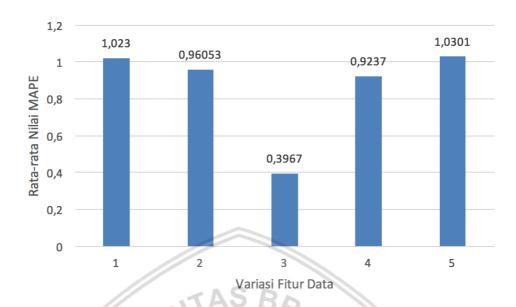
6.1 Hasil dan Analisis Uji Coba Variasi Fitur Data

Pengujian berdasarkan analisis data digunakan untuk mengetahui jenis data terbaik yang cocok untuk digunakan agar menghasilkan nilai kesalahan yang minimal pada memprediksi harga daging sapi. Terdapat satu jenis analisis data, yaitu berupa harga daging sapi. Namun untuk menentukan fitur maka ditentukan dari harga aktual satu bulan sebelumnya disetiap fitur (data harga daging sapi lengkap disertakan pada lampiran A). Inisialisasi parameter yang digunakan pada pengujian ini adalah jumlah hidden neuron 5, input weight dengan range [-1,1] (nilai input weight dan bias setiap kali percobaan disertakan pada Lampiran B), fungsi aktivasi sigmoid biner dan perbandingan data training dan testing sebesar 90%:10%. Pengujian dilakukan sebanyak 10 kali dalam percobaan. Hasil pengujian variasi fitur data dapat dilihat pada Tabel 6.1.

Tabel 6.1 Tabel Nilai MAPE Variasi Fitur Data

	Nilai MAPE (%) Percobaan ke-i														
Jumlah															
Fitur	1	2	3	4	5	6	7	8	9	10	MAPE (%)				
1	0,737	0,746	0,751	0,752	0,746	0,736	0,799	0,726	0,656	3,581	1,023				
2	0,947	0,967	0,992	0,956	0,961	0,965	0,952	0,941	0,969	0,955	0,960				
3	0,408	0,294	0,374	0,387	0,381	0,412	0,406	0,451	0,481	0,373	0,396				
4	0,814	0,324	1,161	1,405	0,443	1,174	1,261	0,675	1,213	0,767	0,923				
5	0,914	1,634	1,278	0,625	1,158	0,489	1,557	0,617	1,157	0,872	1,030				

Untuk mempermudah melakukan analisis jenis data terbaik untuk menghasilkan nilai kesalahan yang didapatkan dari 10 kali percobaan dan mendapatkan nilai kesalahan minimal berdasarkan Tabel 6.1 dipresentasikan melalui grafik pada Gambar 6.1



Gambar 6.1 Grafik Niai MAPE Variasi Fitur Data

Analisis variasi fitur data merupakan analisis yang menggunakan data faktor-faktor yang mempengaruhi harga daging sapi, salah satunya yaitu pada faktor harga daging sapi. Pada penelitian dengan menggunakan analisis harga daging sapi yang dijadikan sebuah fitur, setiap fitur yang digunakan mempunyai harga daging sapi. Setiap nilai fitur bernilai harga sebulan dari sebelum harga aktual tersebut (data lengkap disertakan pada Lampiran B).

Berdasarkan grafik pada Gambar 6.1 menjelaskan analisis dengan fitur pada X3 atau mengambil 3 bulan harga sebelumnya yaitu memiliki nilai MAPE minimal dengan hasil sebesar 0,3963%. Dengan grafik yang didapatkan disimpulkan bahwa semakin banyak dimasukkan jumlah fitur maka hasil prediksi yang dihasilkan cenderung baik, namun pada pengenalan pola ini mengalami kenaikan pada fitur 4 dan 5 dikarenakan *overfitting*. *Overfitting* dimana pengenalan pola terlalu banyak sehingga data terlalu bantuk dikenali sehingga mengalami kenaikan *error*. Selain itu, fitur bergantung pada objek yang digunakan untuk memprediksi dan menghasilkan pola data yang lebih baik.

6.2 Hasil dan Analisis Uji Coba Jumlah Neuron

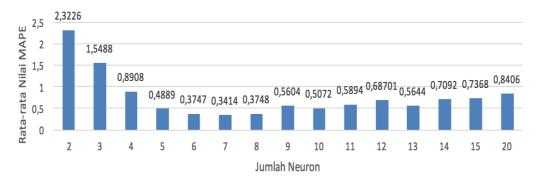
Pengujian jumlah *neuron* yang digunakan untuk mengetahui *neuron* terbaik dalam ELM untuk menghasilkan nilai kesalahan yang minimal. Banyaknya jumlah *neuron* yang diuji coba adalah dimulai dari angka 2 dan berhenti pada titik dimana menunjukkan hasil yang konvergen. Inisialisasi parameter yang digunakan pada pengujian ini adalah jumlah *fitur* dari hasil pengujian sebelumnya, yaitu *fitur* ke-3 (nilai input fitur setiap kali percobaan disertakan pada Lampiran B) dan perbandingan data *training* dan *testing* sebesar 90%:10%. Pengujian menggunakan fungsi aktivasi sigmoid *biner* dengan pengujian masing-masing

dilakukan sebanyak 10 kali percobaan. Hasil pengujian jumlah *neuron layer* dapat dilihat pada Tabel 6.2

Tabel 6.2 Tabel Nilai MAPE Jumlah Neuron

Jumlah	Nilai MAPE (%) Percobaan ke-i											
Neuron											Nilai MAPE (%)	
	1	2	3	4	5	6	7	8	9	10		
2	1,615	0,408	5,363	0,422	1,421	2,356	4,232	0,079	1,107	6,222	2,3226	
3	0,275	2,382	1,437	1,493	1,897	0,593	0,975	0,602	4,513	1,321	1,5488	
4	1,388	0,302	0,402	0,423	0,479	1,405	0,429	1,431	2,162	0,487	0,8908	
5	0,352	0,374	0,397	0,347	0,521	1,311	0,505	0,287	0,411	0,384	0,4889	
6	0,395	0,384	0,344	0,341	0,308	0,466	0,413	0,302	0,385	0,409	0,3747	
7	0,285	0,354	0,275	0,329	0,357	0,337	0,409	0,421	0,321	0,326	0,3414	
8	0,349	0,401	0,262	0,653	0,335	0,276	0,48	0,305	0,418	0,269	0,3748	
9	0,6	0,631	0,375	0,492	0,722	0,494	0,513	0,611	0,597	0,569	0,5604	
10	0,564	0,48	0,625	0,715	0,353	0,578	0,305	0,703	0,459	0,29	0,5072	
11	0,522	0,87	0,335	0,576	0,671	0,66	0,669	0,525	0,462	0,604	0,5894	
12	0,888	0,798	0,663	0,674	0,6	0,622	0,611	0,789	0,400	0,825	0,6870	
13	0,558	0,603	0,413	0,483	0,433	0,585	0,777	0,588	0,46	0,744	0,5644	
14	0,749	0,962	0,768	0,532	0,75	0,689	0,458	0,709	0,829	0,646	0,7092	
15	0,746	0,574	0,537	1,077	0,787	0,798	0,925	0,546	0,673	0,705	0,7368	
20	0,781	0,969	0,78	0,825	0,894	0,78	0,811	1,024	0,952	0,59	0,8406	

Untuk mempermudah melakukan analisis jumlah *neuron* terbaik untuk menghasilkan nilai kesalahan yang minimal didapatkan dari 10 kali percobaan dan mendapatkan nilai kesalahan minimal berdasarkan Tabel 6.2 dipresentasikan melalui grafik pada Gambar 6.2.



Gambar 6.2 Grafik Nilai MAPE Jumlah Neuron

Hidden neuron pada Extreme Learning Machine (ELM) yaitu dari node-node yang merupakan unit yang melakukan untuk proses perhitungan yang mengolah masukkan menjadi keluaran. Selain itu, proses hidden layer memiliki parameter yang menghubungkan antar layer (Shamsirband dkk, 2009). Parameter-parameter tersebut dibentuk berdasarkan banyaknya hidden neuron. Untuk setiap proses menghubungkan memiliki tingkat kekuatan yang berbeda (nilai yang berbeda) (Huang dkk, 2006). Perbedaan ini memungkinkan hasil yang dicapai dalam setiap masukkan oleh unit hidden neuron.

Berdasarkan grafik pada Gambar 6.2, semakin banyak jumlah hidden neuron yang digunakan semakin kecil niai rata-rata untuk MAPE yang diperoleh. Nilai hidden neuron yang semakin besar membentuk banyak penghubung (connector) dengan input layer dan output layer, untuk itu neuron terbaik pada input neuron 7 menghasilkan MAPE sebesar 0,3414%. Kondisi ini memungkinkan unit pemrosesan pada sistem yang melakukan proses pembobotan untuk mengenali data memiliki kemampuan yang baik dengan semakin banyaknya pertimbangan keputusan yang dilakukan oleh hidden node.

6.3 Hasil dan Analisis Uji Coba Fungsi Aktivasi

Pengujian fungsi aktivasi dilakukan untuk menstransformasikan suatu *input* menjadi suatu *output* tertentu. Untuk pengujian fungsi aktivasi dilakukan evaluasi berdasarkan nilai rata-rata MAPE. Komputasi ELM menghasilkan nilai yang berbeda setiap kali program dijalankan karena nilai *input weight* dan bias yang diinisialisasi secara *random*. Dengan demikian untuk memperoleh nilai rata-rata secara keseluruhan, maka dilakukan percobaan sebanyak 10 kali setiap fungsi aktivasi yang diuji.

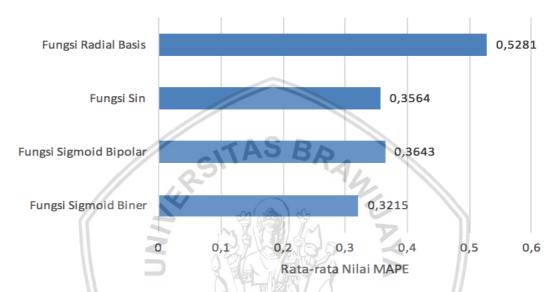
Pengujian aktivasi dilakukan dengan uji coba yang menggunakan fungsi aktivasi sigmoid biner, bipolar, sin, dan radial basis karena keempat fungsi tersebut yang lebih sering digunakan pada permasalahan *forecasting* menggunakan metode ELM. Inisialisasi parameter yang digunakan pada pengujian ini adalah jumlah fitur yang digunakan adalah 3, jumlah *hidden neuron* sebanyak 7, perbandingan data *training* dan *testing* sebesar 90%:10%. Hasil pengujian jumlah *neuron layer* dapat dilihat pada Tabel 6.3

Rata-Nilai MAPE (%) Percobaan ke-i Rata Nilai **Fungsi** 1 2 3 4 5 6 7 8 9 10 Aktivasi MAPE (%) Fungsi Sigmoid 0,33 0,27 0,34 0,32 0,34 0,35 0,31 0,26 0,32 0,33 0,321 4 8 3 Biner 0,33 0,65 0,31 0,29 0,31 0,35 0,40 0,25 0,33 Fungsi Sigmoid 0,37 0,364 **Bipolar** 2 6 8

Tabel 6.3 Tabel Nilai MAPE Variasi Fungsi Aktivasi

Fungsi Sin	0,36 3	0,48 8	0,31 2	0,31 8	0,28 7	0,32 9	0,37 4	0,29 2	0,35 5	0,44 6	0,356
Fungsi Radial	0,67	0,34	0,34	0,75	0,65	0,42	0,57	0,46	0,54	0,49	0,528
Basis	5	5	2	9	9	1	6	7	3	4	

Untuk mempermudah dalam menganalisis rasio perbandingan data terbaik untuk menghasilkan nilai MAPE minimal berdasarkan Tabel 6.4 dipresentasikan melalui grafik pada Tabel 6.3 ditunjukkan pada Gambar 6.3



Gambar 6.3 Grafik Nilai MAPE Pengujian Variasi Fungsi Aktivasi

Berdasarkan Tabel 6.3, rata-rata nilai MAPE terkecil yang didapatkan adalah menggunakan fungsi aktivasi sigmoid biner. Dari hasil pengujian yang telah dilakukan ini mendapatkan rata-rata nilai MAPE sebesar 0,321% menggunakan fungsi aktivasi sigmoid biner (Hasil *input weight* dan bias menggunakan fungsi aktivasi sigmoid biner ada pada Lampiran B.2). Hal ini menunjukkan bahwa dengan fungsi aktivasi sigmoid biner hasil prediksi yang didapatkan memiliki akurasi yang lebih baik. Karena, pada fungsi aktivasi sigmoid biner memiliki *range* [0-1] yang cocok sesuai *input* data yang diproses data yang dinormalisasi data menjadi *range* [0-1] sehingga ketika data ditransferkan ke fungsi aktivasi maka dapat dikenali dengan baik. Berdasarkan grafik pada Gambar 6.3 juga menunjukkan nilai MAPE yang cenderung rendah menggunakan fungsi aktivasi sigmoid biner. Karena menunjukkan hasil yang baik, maka untuk uji coba selanjutnya digunakan fungsi aktivasi sigmoid biner.

6.4 Hasil dan Analisis Uji Coba Jumlah Rasio Percentage Data Training dan Data Testing

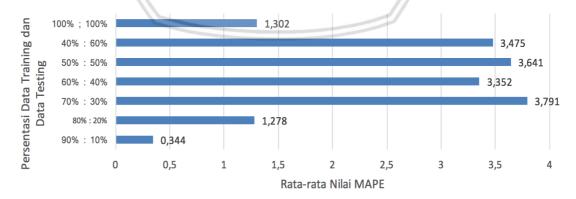
Pengujian perbandingan untuk data *training* dan *testing* ini bertujuan mengetahui pengaruh dari perbandingan jumlah data *training* dengan data *testing* terhadap nilai MAPE yang dihasilkan. Perbandingan jumlah data *training* dan jumlah data *testing* pada pengujian adalah 100%: 100%, 90%: 10%, 80%: 20%,

70%: 30%, 60%: 40%, 50%: 50%, 40%: 60%. Inisialisasi parameter yang digunakan secara *random* setiap kali percobaan dengan *range* data yang diambil yaitu [-1,1]. Pengujian ini menggunakan data fitur terbaik yaitu fitur ke-3 dengan percobaan sebanyak 10 kali untuk setiap rasio perbandingan, Pengujian ini menggunakan *hidden neuron* terbaik yaitu 7 dengan percobaan sebanyak 10 kali untuk setiap rasio perbandingan, pengujian ini menggunakan fungsi aktivasi terbaik yaitu fungsi sigmoid biner dengan percobaan sebanyak 10 kali percobaan untuk setiap rasio perbandingan.

Tabel 6.4 Tabel Nilai MAPE Perbandingan Jumlah Data *Training* dan Data *Testing*

Nilai MAPE (%) Percobaan ke-i											Rata-Rata
Data Training dan Data Testing											Nilai MAPE (%)
Training: Testing	1	2	3	4	S ⁵	6	7	8	9	10	
100% : 100%	1,356	1,279	1,335	1,374	1,276	1,236	1,351	1,257	1,306	1,259	1,302
90% : 10%	0,332	0,477	0,281	0,359	0,321	0,379	0,305	0,355	0,273	0,358	0,344
80% : 20%	1,232	1,174	1,435	1,388	1,209	1,294	1,365	1,134	1,415	1,136	1,278
70% : 30%	3,797	3,358	3,787	4,147	4,924	3,462	4,387	3,062	4,448	2,546	3,791
60% : 40%	3,424	2,916	3,244	4,727	3,8	2,428	5,124	3,285	2,848	1,726	3,352
50% : 50%	4,733	3,738	2,81	4,514	3,164	2,292	4,505	4,285	1,926	4,451	3,641
40% : 60%	4,731	4,145	2,895	3,353	1,984	2,27	4,421	4,359	1,797	4,797	3,475

Untuk mempermudah dalam menganalisis rasio perbandingan data terbaik untuk menghasilkan nilai MAPE minimal berdasarkan Tabel 6.4 dipresentasikan melalui grafik yang ditunjukkan pada Gambar 6.4



Gambar 6.4 Grafik Nilai MAPE Pengujian Perbandingan Jumlah Data *Training*dan Data *Testing*

Data *training* merupakan data yang digunakan untuk proses pembelajaran pada prediksi untuk penelitian ini. Data *training* pada proses *training* memberikan nilai *input weight* dan nilai *output weight* yang dilanjutkan pada proses *testing*. Data *testing* merupakan data uji yang digunakan untuk menghitung hasil prediksi dan kesalahan berdasarkan nilai MAPE.

Berdasarkan grafik pada Gambar 6.4, niai MAPE terkecil didapatkan pada perbandingan jumlah data *training* sebanyak 90% dengan 60 data dan data *testing* sebanyak 10% dengan 7 data yaitu rata-rata error 0,344%. Banyaknya jumlah data *training* dan testing pada ELM ini berpengaruh terhadap nilai *error* yang dihasilkan. Karena, disebabkan karena metode ELM merupakan metode pelatihan. Dengan ini semakin banyak jumlah data training yang digunakan maka semakin baik nilai *error* yang didapatkan. Disimpulkan dengan data *training* yang digunakan maka membantu dalam pengenalan pola yang diprediksi.

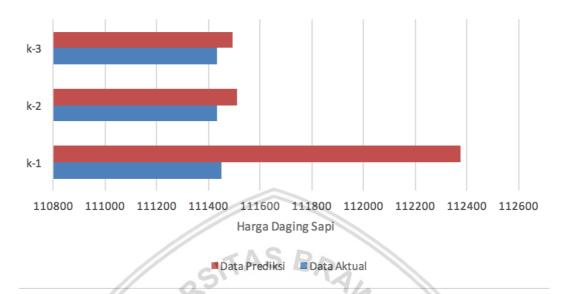
6.5 Hasil dan Analisis Uji Coba Jumlah Data *Training* dan *Testing* dengan *Cross Validation*

Pengujian ini merupakan K-fold yaitu salah satu metode *Cross Validation* yang melipat data sebanyak K dan mengulangi eksperimen sebanyak K juga. perbandingan untuk data *training* dan *testing* yang bertujuan mengetahui harga prediksi, karena proses training dengan menggunakan k-fold untuk memprediksi lebih akurat sehingga prediksi baik. Untuk proses ini data dibagi sesuai jumlah *input cross validation*. K-fold yang digunakan pada penelitian ini yaitu diambil 3 bulan . Inisialisasi parameter yang digunakan secara *random* setiap kali percobaan dengan *range* data yaitu [-1,1]. Pengujian ini menggunakan data fitur terbaik yaitu fitur ke-3 dengan percobaan sebanyak 10 kali untuk setiap rasio perbandingan, Pengujian ini menggunakan *hidden neuron* terbaik yaitu 7 dengan percobaan sebanyak 10 kali untuk setiap rasio perbandingan, pengujian ini menggunakan fungsi aktivasi terbaik yaitu fungsi sigmoid biner dengan percobaan sebanyak 10 kali percobaan untuk setiap rasio perbandingan. Proses tersebut digunakan karena sudah dibuktikan.

Table 6.5 Tabel Nilai Prediksi Harga Dengan K-Fold

Perbandingan jumlah Data	Harga Prediksi Percobaan ke-i												
Training dan Data Testing											Harga Prediksi		
K-Fold	1	2	3	4	5	6	7	8	9	10			
K-1	1122 27	1123 19	1125 69	1124 20	1124 15	1123 80	1124 88	1124 22	1123 88	1121 06	112373,4		
K-2	1112 88	1114 17	1116 79	1115 24	1115 71	1116 29	1115 70	1117 04	1113 62	1113 48	111509,2		
K-3	1112 80	1114 06	1116 54	1115 05	1115 56	1116 08	1115 53	1116 75	1113 59	1113 43	111493,9		

Untuk mempermudah dalam untuk menghasilkan harga prediksidipresentasikan melalui grafik pada Tabel 6.5 ditunjukkan pada data Gambar 6.5



Gambar 6.5 Grafik Nilai MAPE Pengujian Perbandingan Harga Aktual dan Harga Prediksi Menggunakan Cross Validation

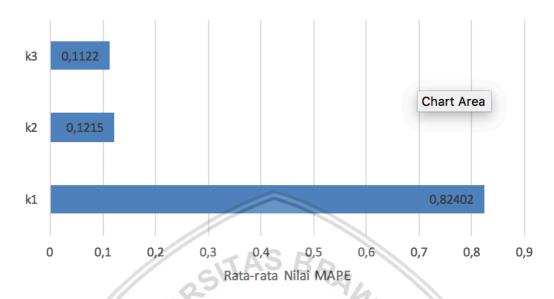
Data *training* merupakan data yang digunakan untuk proses pembelajaran pada prediksi untuk penelitian ini. Data *training* pada proses *training* memberikan nilai *input weight* dan nilai *output weight* yang dilanjutkan pada proses *testing*. Data *testing* merupakan data uji yang digunakan untuk menghitung hasil prediksi dan kesalahan berdasarkan nilai MAPE.

Berdasarkan grafik pada Gambar 6.5, mendapatkan harga prediksi setelah dilakukan proses pembagian data menggunakan pembagian k-fold yang didapatkan pada perbandingan jumlah data keseluruhan 69, Cross validation yang digunakan yaitu 3 maka jumlah *testing* yang digunakan adalah dari data ke 67 sampai 69.

Table 6.6 Tabel Nilai MAPE Perbandingan Cross Validation

Perbandingan jumlah Data		Harga Prediksi Percobaan ke-i												
Training dan														
Data Testing														
K-Fold														
	1	2	3	4	5	6	7	8	9	10				
K-1	0,696	0,778	1,01	0,869	0,86	0,834	0,93	0,871	0,841	0,558	0,824			
K-2	0,145	0,03	0,204	0,065	0,107	0,16	0,106	0,227	0,079	0,092	0,1215			
K-3	0,153	0,04	0,182	0,048	0,09	0,141	0,09	0,201	0,081	0,096	0,1122			

Untuk mempermudah dalam untuk menghasilkan harga prediksidipresentasikan melalui grafik pada Tabel 6.5 ditunjukkan pada data Gambar 6.5



Gambar 6.6 Grafik Nilai MAPE Pengujian Perbandingan Jumlah Data *Training* dan Data *Testing* dengan *Cross Validation*

Berdasarkan Tabel 6.3, rata-rata nilai MAPE Dari hasil pengujian yang telah dilakukan, didapatkan rata-rata nilai k1, k2 dan k3 MAPE rata-rata sebesar 0,352%. Dari hasil ini membuktikan bahwa dengan pembagian data dengan menggunakan cross validation ini sangat baik digunakan untuk memprediksi harga daging sapi yang mendekati harga aktual. Untuk banyaknya jumlah data training pada ELM ini berpengaruh terhadap nilai error yang dihasilkan. Hal ini tidak mempengaruhi karena metode ELM merupakan metode pelatihan. Dengan ini semakin banyak jumlah data training yang digunakan maka semakin baik nilai error yang didapatkan.

BAB 7 PENUTUP

Bab ini menjelaskan tentang kesimpulan dari hasil yang telah dilakukan implementasi dan pengujian sistem, hal ini menjawab dari rumusan masalah dan saran untuk dapat menghasilkan sistem yang lebih baik pada penelitian yang dilanjutkan oleh peneliti lainnya.

7.1 Kesimpulan

Berdasarkan hasil pengujian dan pembahasan dari prediksi harga daging sapi menggunakan *Extreme Learning Machine* (ELM), maka didapatkan kesimpulan sebagai berkut:

- 1. Metode Extreme Learning Machine (ELM) digunakan prediksi harga daging sapi yang menghasilkan nilai error terkecil dengan melibatkan jumlah hidden layer dan random input weight. Hidden layer pada ELM terdiri dari node-node yang berfungsi sebagai pemrosesan yang menghubungkan input layer dengan output layer. Banyaknya hidden node pada bagian hidden layer mempengaruhi hasil perhitungan dari suatu permasalahan hidden node mengolah semua masukkan yang menjadi keluaran. Pada hidden layer terdapat parameter input weight yang terhubung dengan input layer dan output weight yang terhubung pada output layer.
- 2. Perbandingan jumlah data training dan data testing serta penambahan jumlah neuron pada hidden layer berpengaruh terhadap output prediksi yang dihasilkan. Tingkat kesalahan yang berdasarkan penguji yang dilakukan dengan nilai MAPE yaitu sebesar 0,344% menggunakan fungsi aktviasi sigmoid biner, menggunakan 3 fitur, perbandingan jumlah data training dan testing yaitu 90%:10% serta hidden neuron sebanyak 7. Pada pengujian validasi sistem menggunakan k-fold cross validation, sistem estimasi produksi harga daging k-fold yakni k=3. Akurasi yang didapatkan selama 10 percobaan mendapatkan nilai rata-rata MAPE sebesar 0,352%.

7.2 Saran

Berdasarkan penelitian yang telah dilakukan, terdapat beberapa saran yang dapat dikembangkan untuk penelitian selanjutnya yaitu:

- Untuk penelitian selanjutnya, penelitian dapat menambahkan parameter lain yang merupakan faktor-faktor dalam harga daging sapi. Hal ini bertujuan agar hasil prediksi lebih obyektif, dan semakian banyak parameter faktor produksi yang digunakan menghasilkan hasil yang lebih baik, dan menambahkan fungsi aktivasi yang digunakan.
- 2. Pengembangan metode atau penggabungan metode lain dengan ELM seperti *Optimally Pruned metode Extreme Learning Machine* (OPELM) atau metode lainnya dapat digunakan pada penelitian selanjutnya untuk meningkatkan hasil yang didapatkan dari penelitian sebelumnya.

DAFTAR PUSTAKA

- Abadi, I. & Soeprijanto, A., 2014. Extreme learning machine approach to estimate hourly solar radiation on horizontal surface (PV) in Surabaya-East java.
- Chandar, S. K., Sumathi, M. & Sivanadam, S., 2016. Forecasting Gold Prices Based on Extreme Learning Machine, June.
- Ditjen Peternakan, 1997. Statistik Peternakan. [Online].
- Fardani, Wuryanto & Wediningsih, 2015. Sistem Pendukung Keputusan Prediksi Jymlah Kunjungan Pasien Menggunakan Metode Extreme Learning Machine (Studi Kasus : Poli Gigi RSU DR. Wahidin Sudiro Husodo Mojokerto.
- Giusti, A., Agus Wahyu Widodo & Sigit Adinugroho, 2018. Prediksi Penjualan Mi Menggunakan Metode Extreme Learning Machine (ELM) di Kober Mie Setan Cabang Soekarno Hatta.
- Huang, G., Zhu, Q. & Siew, C., 2004. Extreme Learning Machine: a New Learning Scheme of FeedForward neural networks.. Volume 2, pp. 985-990.
- Huang, G., Zhu, Q. & Siew, C., 2006. Application of extreme learning machine method for time series analysis. pp. 256-262.
- Humaini, Q., 2015. Jaringan Syaraf Tiruan Extreme Learning Machine (ELM) Untuk Memprediksi Kondisi Di Wilayah Malang.
- Ilham, N., 2001. Analisis Penawaran Dan Permintaan Daging Sapi Di Indonesia.
- Jain, Y. & Bhandare, S., 2011. Min max normalization based data perturbation method for privacy protection.. pp. 45-50.
- Makridakis, S. & Hibon, M., 1995. Evaluating Accuracy (Or Error) Measures.
- Mendome, K., Nainggolan, D. N. & Kekenusa, J., 2016. Penerapan Model ARIMA dalam Memprediksi Jumlah Tindak Kriminalitas di Wilayah POLRESTA Manado Provinsi Sulawesi Utara. *Jurnal MIPA UNSRAT Online*, Volume 5, p. 114.
- Mosabeth, C., 2018. Prediksi Harga Daging Sapi Di Kota Malang Dengan Menggunakan Metode Extreme Learning Machine. Agustus.
- Pangaribuan, J. J., 2016. Mendiagnosis Penyakit Diabetes Melitus Dengan Menggunakan Metode Extreme Learning Machine.
- Puspitaningrum, D., 2006. Pengantar Jaringan Saraf Tiruan.
- Singh, R. & Balasundaram, S., 2007. Application of extreme learning machine method for time series analysis.. pp. 256-262.
- Siwi, I. P., Imam Cholissodin & M. Tanzil Furqon, 2016. Prediksi Produksi Gula Pasir Menggunakan Extreme Learning Machine (ELM) Pada Pg Candi Baru Sidoarjo. Volume 8.

Ubay, M. S., 2012. Prediksi Harga Saham Dengan Menggunakan Jaringan Saraf Tiruan Metode *Extreme Learning Machine*.

Yaseen, Z. M. et al., 2016. Stream-flow forecasting using extreme learning machines: A case study in a semi-arid region in Iraq.

