

**PREDIKSI INDEKS HARGA KONSUMEN (IHK) KELOMPOK
PERUMAHAN, AIR, LISTRIK, GAS, DAN BAHAN BAKAR
MENGUNAKAN METODE *EXTREME LEARNING MACHINE***

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Irma Ramadanti Fitriyani

NIM: 145150201111104



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

PENGESAHAN

PREDIKSI INDEKS HARGA KONSUMEN (IHK) KELOMPOK PERUMAHAN, AIR,
LISTRIK, GAS, DAN BAHAN BAKAR MENGGUNAKAN METODE *EXTREME LEARNING*
MACHINE

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

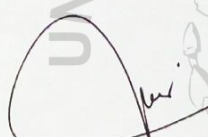
Disusun Oleh :
Irma Ramadanti Fitriyani
NIM: 145150201111104

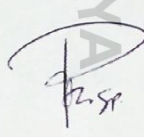
Skripsi ini telah diuji dan dinyatakan lulus pada
13 Juli 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II


Budi Darma Setiawan S.Kom., M.Cs
NIP: 19841015 201404 1 002


Rizal Setya Perdana S.Kom., M.Kom
NIK: 2016039101181001

Mengetahui
Ketua Jurusan Teknik Informatika



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



PERNYATAAN ORISINALITAS

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

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

Malang, 22 Juli 2018



METERAI
TEMPEL
SE 060AFF 169804379
6000
ENAM RIBU RUPIAH

mahe

Irma Ramadanti Fitriyani

NIM: 145150201111104

KATA PENGANTAR

Puji syukur kehadiran Tuhan YME yang telah melimpahkan rahmat-Nya sehingga laporan skripsi yang berjudul “Prediksi Indeks Harga Konsumen (IHK) Kelompok Perumahan, Air, Listrik, Gas, dan Bahan Bakar Menggunakan Metode *Extreme Learning Machine*” dapat terselesaikan.

Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Bapak Budi Darma Setiawan, S.Kom., M.Cs. dan Bapak Rizal Setya Perdana, S.Kom., M.Kom. selaku dosen pembimbing skripsi yang telah membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
2. Agus Wahyu Widodo, S.T, M.Cs. selaku Ketua Program Studi Teknik Informatika.
3. Bapak Tri Astoto Kurniawan, S.T., M.T, Ph.D. selaku Ketua Jurusan Teknik Informatika.
4. Kedua orang tua, Adik serta keluarga besar yang telah senantiasa mendoakan, memberi nasehat, memberi kasih sayang, dan memberikan semangat kepada penulis demi terselesaikannya skripsi ini.
5. Revinda Bertananda, Krishnanti Dewi, Novirra Dwi Asri, Vania Nuraini Latifah, dan Yulia Kurniawati yang telah menjadi teman penulis selama berada di perantauan dan semoga akan tetap berlanjut. Terima kasih karena selalu ada serta senantiasa memberikan semangat, doa, dan dukungan kepada penulis sehingga terselesaikannya skripsi ini.
6. Keluarga Advokesma BEMTIK Periode 2015/2016 yang telah memberikan kesan keluarga kepada penulis selama di perantauan. Terima kasih atas segala suka dan duka yang telah dialami bersama penulis.
7. Teman-teman DISPLAY FILKOM 2017/2018 khususnya divisi PSDM yang telah memberikan banyak kenangan menyenangkan, pelajaran, kesan, dan semangat kepada penulis.
8. Audia Refanda, Cusen Mosabeth, Nadzir Muhammad, Iqbal Pratama, Kakak Christy Makrina, Kakak Sindy Erika, dan Ivan Fadilla yang telah menjadi teman berdiskusi dan memberikan semangat untuk penulis demi terselesaikannya skripsi ini.
9. Teman-teman PASUS SMAN 1 Cibinong angkatan 26 yang selalu menjadi penyemangat untuk penulis. Terima kasih atas pertemanan yang mengesankan selama bertahun-tahun dan tidak pernah berhenti.
10. Teman-teman Informatika angkatan 2014, seluruh dosen, dan civitas akademika Fakultas Ilmu Komputer yang telah banyak memberi bantuan dan

dukungan selama penulis menempuh studi dan selama penyelesaian skripsi ini.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih sangat banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 22 Juli 2018

Penulis

irmarfitriyani@gmail.com



ABSTRAK

Irma Ramadanti Fitriyani. Prediksi Indeks Harga Konsumen (IHK) Kelompok Perumahan, Air, Listrik, Gas, dan Bahan Bakar Menggunakan Metode *Extreme Learning Machine* (ELM).

Pembimbing: Budi Darma Setiawan, S.Kom., M.Cs dan Rizal Setya Perdana S.Kom, M.Kom.

Indeks Harga Konsumen (IHK) merupakan salah satu indikator untuk mengukur tingkat inflasi di Indonesia. Pada tahun 2017 Inflasi yang terjadi di Indonesia menurut kelompok pengeluaran secara umum adalah sebesar 3,61%. Kelompok perumahan, air, listrik, gas, dan bahan bakar menjadi penyumbang inflasi terbesar sebanyak 5,14%. Maka dari itu prediksi perlu dilakukan untuk mengantisipasi serta mengurangi laju inflasi domestik. Prediksi yang dilakukan pada penelitian ini menggunakan metode *Extreme Learning Machine* (ELM) dengan inialisasi bobot menggunakan *Nguyen-Widrow*. Tahapan metode ELM yang dilakukan yaitu proses normalisasi data, proses pelatihan untuk mendapatkan nilai bobot keluaran (*output weight*) yang nantinya akan digunakan pada proses pengujian untuk mendapatkan nilai keluaran *output layer* yang setelah di denormalisasi akan menjadi hasil prediksi dalam bentuk aktual. Selanjutnya adalah melakukan evaluasi hasil prediksi menggunakan *Root Mean Squared Error* (RMSE). Data yang digunakan dalam penelitian ini adalah 84 data Indeks Harga Konsumen kelompok perumahan perumahan, air, listrik, gas, dan bahan bakar dalam bentuk *time-series* periode Januari 2011 s.d. Desember 2017, diperoleh dari *website* resmi Bank Indonesia (www.bi.go.id). Hasil pengujian yang dilakukan pada penelitian ini adalah didapatkan nilai RMSE minimum sebesar 0,65 dengan jumlah fitur = 7, jumlah data latih 30 dan data uji 11, jumlah *hidden neuron* = 7, dan fungsi aktivasi yang cocok adalah fungsi sigmoid biner.

Kata kunci: prediksi, indeks harga konsumen, *Extreme Learning Machine* (ELM), *Nguyen Widrow*, *Root Mean Squared Error* (RMSE).

ABSTRACT

Irma Ramadanti Fitriyani. Prediction of Consumer Price Index (CPI) of Housing, Water, Electricity, Gas, and Fuel Using Extreme Learning Machine (ELM).

Adviser: Budi Darma Setiawan, S.Kom., M.Cs and Rizal Setya Perdana S.Kom., M.Kom.

The Consumer Price Index (CPI) is one of the indicators to measure the inflation rate in Indonesia. In 2017 Inflation occurring in Indonesia by expenditure group in general is 3.61%. The housing, water, electricity, gas, and fuel group accounted for the largest inflation of 5.14%. Therefore, the prediction needs to be done to anticipate and reduce domestic inflation rate. Prediction done in this research using method of Extreme Learning Machine (ELM) with initialization of weight using Nguyen-Widrow. Stages of ELM method performed is the process of data normalization, the training process to get the output weight value which will be used in the testing process to get the output value from the output layer, which after the denormalization will be the result prediction in the actual value. Next is to evaluate the prediction result using Root Mean Squared Error (RMSE). The data that used in this research are 84 time series data of Consumer Price Index of housing, water, electricity, gas and fuel in January 2011 until December 2017, the data obtained from the official website of Bank Indonesia (www.bi.go.id). The result of the testing in this research is the minimum RMSE is 0,65 with the number of features = 7, the amount of training data = 30 and testing data = 11, the number of hidden neurons = 7, and the suitable activation function is the binary sigmoid function.

Keywords: prediction, consumer price index, Extreme Learning Machine (ELM), Nguyen Widrow, Root Mean Squared Error (RMSE).

DAFTAR ISI

| | |
|--|------|
| PENGESAHAN | ii |
| PERNYATAAN ORISINALITAS | iii |
| KATA PENGANTAR..... | iv |
| ABSTRAK..... | vi |
| ABSTRACT | vii |
| DAFTAR ISI..... | viii |
| DAFTAR TABEL..... | xii |
| DAFTAR GAMBAR..... | xiv |
| DAFTAR KODE PROGRAM | xvi |
| DAFTAR LAMPIRAN | xvii |
| BAB 1 PENDAHULUAN..... | 1 |
| 1.1 Latar belakang..... | 1 |
| 1.2 Rumusan masalah..... | 3 |
| 1.3 Tujuan | 3 |
| 1.4 Manfaat..... | 4 |
| 1.5 Batasan masalah | 4 |
| 1.6 Sistematika pembahasan..... | 4 |
| BAB 2 LANDASAN KEPUSTAKAAN | 6 |
| 2.1 Kajian Pustaka | 6 |
| 2.2 Prediksi..... | 11 |
| 2.3 Indeks Harga Konsumen | 11 |
| 2.4 Jaringan Syaraf Tiruan..... | 11 |
| 2.5 Normalisasi dan Denormalisasi Data | 12 |
| 2.6 Fungsi Aktivasi..... | 12 |
| 2.7 Nguyen-Widrow..... | 13 |
| 2.8 <i>Extreme Learning Machine</i> | 13 |
| 2.8.1 Proses Pelatihan | 14 |
| 2.8.2 Proses Pengujian | 15 |
| 2.8.3 Root Mean Squared Error (RMSE) | 16 |
| BAB 3 METODOLOGI | 17 |

| | |
|---|-----------|
| 3.1 Studi Literatur | 17 |
| 3.2 Pengumpulan Data | 18 |
| 3.3 Perancangan Sistem..... | 18 |
| 3.3.1 Mekanisme Penyelesaian Masalah | 18 |
| 3.3.2 Perancangan Antarmuka Pengguna..... | 19 |
| 3.3.3 Perancangan Pengujian..... | 19 |
| 3.4 Implementasi | 19 |
| 3.5 Pengujian dan Analisis | 20 |
| 3.6 Kesimpulan dan Saran | 21 |
| BAB 4 PERANCANGAN..... | 22 |
| 4.1 Formulasi Permasalahan..... | 22 |
| 4.2 Tahapan <i>Extreme Learning Machine</i> | 23 |
| 4.3 Normalisasi Data | 24 |
| 4.4 Pelatihan | 27 |
| 4.4.1 Membuat Nilai <i>Random</i> untuk Bobot Masukan | 28 |
| 4.4.2 Inisialisasi Bobot Masukan Menggunakan <i>Nguyen-Widrow</i> | 29 |
| 4.4.3 Membuat Nilai <i>Random</i> untuk Bias | 32 |
| 4.4.4 Menghitung Matriks Keluaran pada <i>Hidden Layer</i> | 33 |
| 4.4.5 Menghitung Bobot Keluaran..... | 39 |
| 4.5 Pengujian | 41 |
| 4.5.1 Denormalisasi Data | 42 |
| 4.5.2 Menghitung Nilai Evaluasi..... | 43 |
| 4.6 Contoh Perhitungan Manualisasi..... | 44 |
| 4.7 Perancangan <i>User Interface</i> | 55 |
| 4.8 Perancangan Pengujian | 59 |
| 4.8.1 Rancangan Pengujian Jumlah Fitur | 59 |
| 4.8.2 Rancangan Pengujian Jumlah Data Latih | 59 |
| 4.8.3 Rancangan Pengujian Jumlah <i>Hidden Neuron</i> | 60 |
| 4.8.4 Rancangan Pengujian Fungsi Aktivasi | 60 |
| 4.8.5 Rancangan Pengujian Perbandingan Metode..... | 61 |
| BAB 5 IMPLEMENTASI | 62 |
| 5.1 Implementasi Kode Program | 62 |



| | |
|---|-----------|
| 5.1.1 Implementasi Proses Normalisasi Data | 62 |
| 5.1.2 Implementasi Proses Membuat Nilai Random untuk Bobot Masukan | 63 |
| 5.1.3 Implementasi Proses Inialisasi Bobot Masukan Menggunakan <i>Nguyen-Widrow</i> | 63 |
| 5.1.4 Implementasi Proses Membuat Nilai Random untuk Bias | 64 |
| 5.1.5 Implementasi Proses Memperbanyak Matriks Bias | 65 |
| 5.1.6 Implementasi Proses Transpose Matriks | 65 |
| 5.1.7 Implementasi Proses Menghitung Perkalian Matriks | 65 |
| 5.1.8 Implementasi Proses Menghitung Penjumlahan Matriks..... | 66 |
| 5.1.9 Implementasi Proses Menghitung H dengan Fungsi Aktivasi | 67 |
| 5.1.10 Implementasi Proses Menghitung <i>Inverse</i> Matriks | 68 |
| 5.1.11 Implementasi Proses Denormalisasi Data | 68 |
| 5.1.12 Implementasi Proses Perhitungan Nilai Evaluasi..... | 69 |
| 5.2 Implementasi <i>User Interface</i> | 69 |
| 5.2.1 Implementasi <i>Interface</i> Tampilan Halaman Utama | 69 |
| 5.2.2 Implementasi <i>Interface</i> Tampilan Menu Tab Data | 70 |
| 5.2.3 Implementasi <i>Interface</i> Tampilan Menu Tab Data Normalisasi . | 71 |
| 5.2.4 Implementasi <i>Interface</i> Tampilan Menu Tab Pembagian Data .. | 71 |
| 5.2.5 Implementasi <i>Interface</i> Tampilan Menu Tab Proses Pelatihan .. | 72 |
| 5.2.6 Implementasi <i>Interface</i> Tampilan Menu Tab Proses Pengujian . | 72 |
| 5.2.7 Implementasi <i>Interface</i> Tampilan Menu Tab Hasil Evaluasi | 73 |
| BAB 6 PENGUJIAN DAN ANALISIS..... | 74 |
| 6.1 Pengujian Jumlah Fitur..... | 74 |
| 6.2 Pengujian Jumlah Data Latih..... | 75 |
| 6.3 Pengujian Jumlah <i>Hidden Neuron</i> | 77 |
| 6.4 Pengujian Fungsi Aktivasi..... | 78 |
| 6.5 Pengujian Perbandingan Metode | 79 |
| BAB 7 PENUTUP | 80 |
| 7.1 Kesimpulan..... | 80 |
| 7.2 Saran | 80 |
| DAFTAR PUSTAKA..... | 82 |



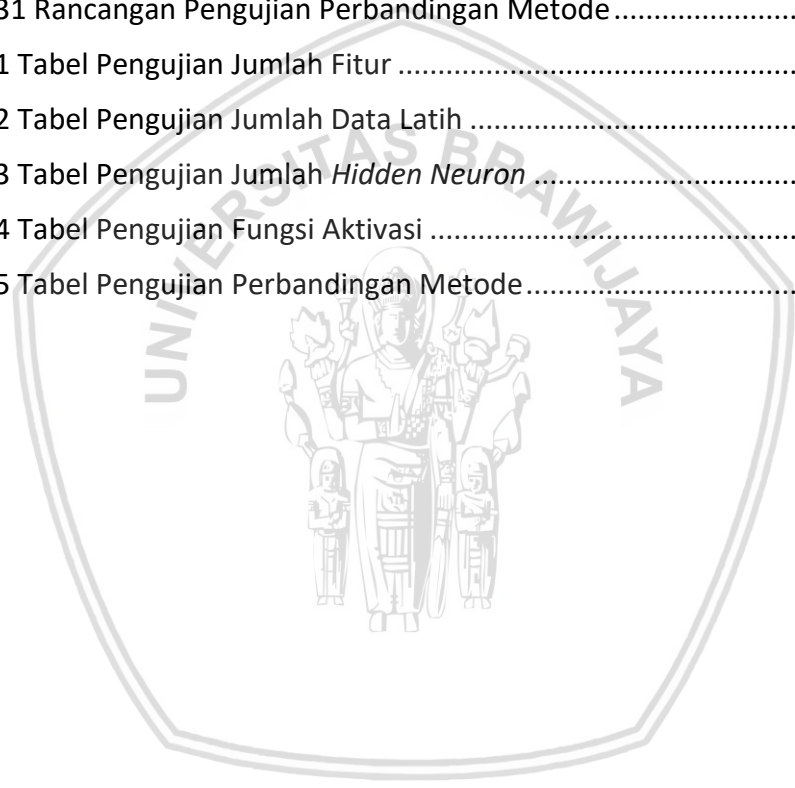
LAMPIRAN A DATA INDEKS HARGA KONSUMEN 85
LAMPIRAN B BOBOT MASUKAN DAN BIAS 88



DAFTAR TABEL

| | |
|--|----|
| Tabel 2.1 Kajian Pustaka | 8 |
| Tabel 3.1 Data Indeks Harga Konsumen Tahun 2014 | 18 |
| Tabel 3.2 Tabel Pengujian Jumlah Fitur | 20 |
| Tabel 3.3 Tabel Pengujian Jumlah Data Latih | 20 |
| Tabel 3.4 Tabel Pengujian Jumlah <i>Hidden Neuron</i> | 21 |
| Tabel 3.5 Tabel Pengujian Fungsi Aktivasi | 21 |
| Tabel 3.6 Tabel Pengujian Perbandingan Metode | 21 |
| Tabel 4.1 Data Indeks Harga Konsumen Kelompok Perumahan, Air, Gas, dan Bahan Bakar | 22 |
| Tabel 4.2 Data IHK Kelompok Perumahan, Air, Gas, dan Bahan Bakar | 44 |
| Tabel 4.3 Nilai Min dan Max | 45 |
| Tabel 4.4 Data Latih Normalisasi..... | 45 |
| Tabel 4.5 Data Uji Normalisasi | 46 |
| Tabel 4.6 Matriks Bobot Masukan | 46 |
| Tabel 4.7 Matriks Bias | 46 |
| Tabel 4.8 Nilai Gamma (γ)..... | 47 |
| Tabel 4.9 Matriks Bobot Masukan Menggunakan <i>Nguyen-Widrow</i> | 47 |
| Tabel 4.10 Matriks Bias Proses Pelatihan | 48 |
| Tabel 4.11 Matriks <i>Transpose</i> Bobot Masukan..... | 48 |
| Tabel 4.12 Hasil Perkalian Bobot Masukan <i>Transpose</i> dengan Data Latih..... | 48 |
| Tabel 4.13 Matriks Hinit..... | 49 |
| Tabel 4.14 Matriks Keluaran <i>Hidden Layer</i> (H) | 49 |
| Tabel 4.15 Matriks Keluaran <i>Hidden Layer Transpose</i> | 50 |
| Tabel 4.16 Matriks Hasil Perkalian <i>HT</i> dan <i>H</i> | 50 |
| Tabel 4.17 Matriks Hasil <i>Inverse</i> Perkalian <i>HT</i> dan <i>H</i> | 51 |
| Tabel 4.18 Hasil Perhitungan H+ | 51 |
| Tabel 4.19 Matriks Bobot Keluaran..... | 51 |
| Tabel 4.20 Matriks Bias Proses Pengujian..... | 52 |
| Tabel 4.21 Matriks <i>Transpose</i> Bobot Masukan Pada Proses Pengujian | 52 |
| Tabel 4.22 Hasil Perkalian Bobot Masukan <i>Transpose</i> dengan Data Uji | 52 |

| | |
|--|----|
| Tabel 4.23 Matriks Hinit | 53 |
| Tabel 4.24 Matriks Keluaran <i>Hidden Layer</i> (H) | 53 |
| Tabel 4.25 Matriks Keluaran <i>Output Layer</i> | 54 |
| Tabel 4.26 Matriks Denormalisasi Hasil Prediksi | 54 |
| Tabel 4.27 Rancangan Pengujian Jumlah Fitur | 59 |
| Tabel 4.28 Rancangan Pengujian Jumlah Data Latih | 59 |
| Tabel 4.29 Rancangan Pengujian Jumlah <i>Hidden Neuron</i> | 60 |
| Tabel 4.30 Rancangan Pengujian Fungsi Aktivasi | 61 |
| Tabel 4.31 Rancangan Pengujian Perbandingan Metode | 61 |
| Tabel 6.1 Tabel Pengujian Jumlah Fitur | 74 |
| Tabel 6.2 Tabel Pengujian Jumlah Data Latih | 75 |
| Tabel 6.3 Tabel Pengujian Jumlah <i>Hidden Neuron</i> | 77 |
| Tabel 6.4 Tabel Pengujian Fungsi Aktivasi | 78 |
| Tabel 6.5 Tabel Pengujian Perbandingan Metode | 79 |



DAFTAR GAMBAR

| | |
|--|----|
| Gambar 2.1 Struktur Metode ELM..... | 14 |
| Gambar 3.1 Diagram Alir Metode Penelitian..... | 17 |
| Gambar 3.2 Diagram Blok Mekanisme Penyelesaian Masalah..... | 19 |
| Gambar 4.1 Diagram Alir <i>Extreme Learning Machine</i> | 24 |
| Gambar 4.2 Diagram Alir Proses Normalisasi | 26 |
| Gambar 4.3 Diagram Alir Proses Pelatihan | 27 |
| Gambar 4.4 Diagram Alir Proses Membuat Nilai Random untuk Bobot Masukan..... | 29 |
| Gambar 4.5 Diagram Alir Proses Inisialisasi Bobot Masukan Menggunakan <i>Nguyen-Widrow</i> | 31 |
| Gambar 4.6 Diagram Alir Proses Membuat Nilai <i>Random</i> untuk Bias..... | 32 |
| Gambar 4.7 Diagram Alir Proses Menghitung Keluaran pada <i>Hidden Layer</i> | 33 |
| Gambar 4.8 Diagram Alir Memperbanyak Matriks Bias | 34 |
| Gambar 4.9 Diagram Alir Proses <i>Transpose</i> Matriks | 35 |
| Gambar 4.10 Diagram Alir Proses Menghitung Perkalian Matriks | 36 |
| Gambar 4.11 Diagram Alir Proses Menghitung Penjumlahan Matriks..... | 37 |
| Gambar 4.12 Diagram Alir Proses Menghitung H dengan Fungsi Aktivasi | 38 |
| Gambar 4.13 Diagram Alir Proses Menghitung Bobot Keluaran | 39 |
| Gambar 4.14 Diagram Alir Proses Menghitung Matriks <i>Moore-Penrose Pseudo Inverse</i> | 40 |
| Gambar 4.15 Diagram Alir Proses Pengujian | 41 |
| Gambar 4.16 Diagram Alir Proses Denormalisasi | 42 |
| Gambar 4.17 Diagram Alir Proses Menghitung Nilai Evaluasi | 43 |
| Gambar 4.18 Perancangan <i>User Interface</i> Halaman Utama..... | 55 |
| Gambar 4.19 Perancangan <i>User Interface</i> Tab Data Normalisasi..... | 56 |
| Gambar 4.20 Perancangan <i>User Interface</i> Tab Pembagian Data..... | 57 |
| Gambar 4.21 Perancangan <i>User Interface</i> Tab Proses Pelatihan | 57 |
| Gambar 4.22 Perancangan <i>User Interface</i> Tab Proses Pengujian..... | 58 |
| Gambar 4.23 Perancangan <i>User Interface</i> Tab Hasil Evaluasi | 58 |
| Gambar 5.1 Implementasi <i>Interface</i> Tampilan Halaman Utama | 69 |
| Gambar 5.2 Implementasi <i>Interface</i> Tampilan Menu Tab Data | 70 |

Gambar 5.3 Implementasi *Interface* Tampilan Menu Tab Data Normalisasi 71

Gambar 5.4 Implementasi *Interface* Tampilan Menu Tab Pembagian Data 71

Gambar 5.5 Implementasi *Interface* Tampilan Menu Tab Proses Pelatihan 72

Gambar 5.6 Implementasi *Interface* Tampilan Menu Tab Proses Pengujian 72

Gambar 5.7 Implementasi *Interface* Tampilan Menu Tab Hasil Evaluasi 73

Gambar 6.1 Grafik Hasil Pengujian Jumlah Fitur 74

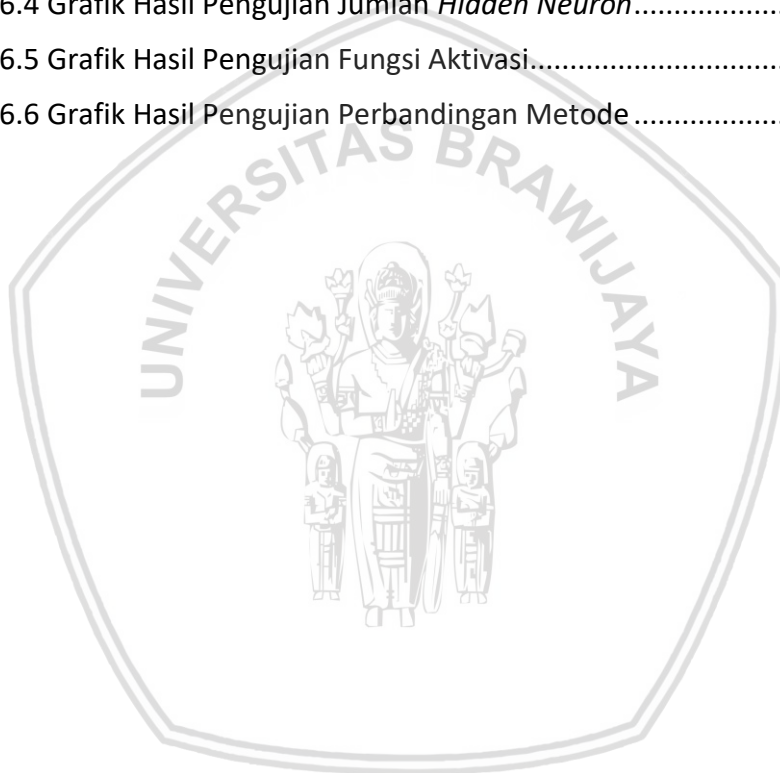
Gambar 6.2 Grafik Hasil Pengujian Jumlah Data Latih..... 75

Gambar 6.3 Grafik Pola Data..... 76

Gambar 6.4 Grafik Hasil Pengujian Jumlah *Hidden Neuron*..... 77

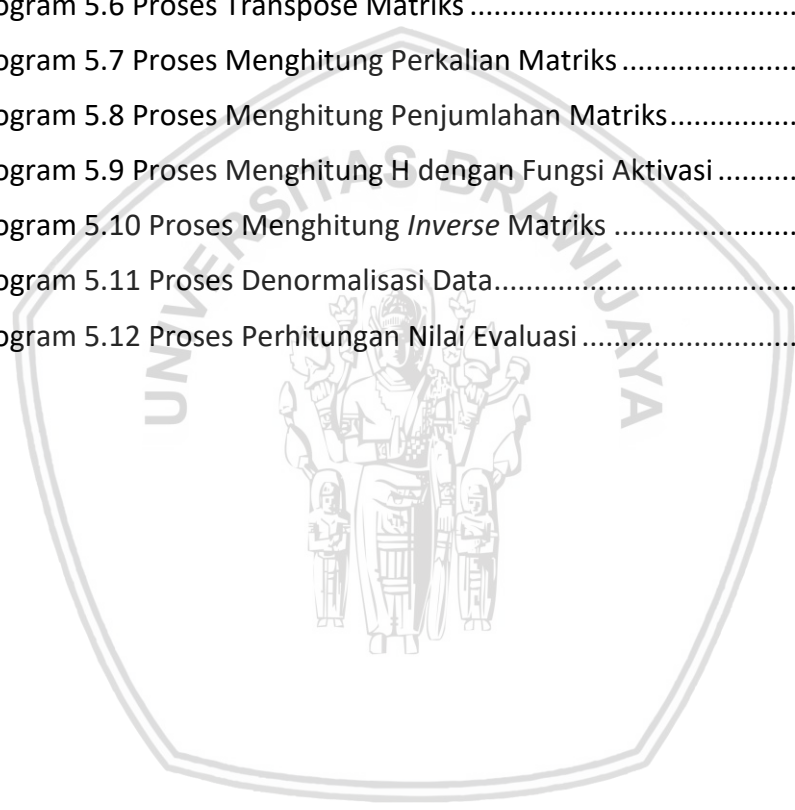
Gambar 6.5 Grafik Hasil Pengujian Fungsi Aktivasi..... 78

Gambar 6.6 Grafik Hasil Pengujian Perbandingan Metode 79



DAFTAR KODE PROGRAM

| | |
|---|----|
| Kode Program 5.1 Proses Normalisasi Data..... | 62 |
| Kode Program 5.2 Proses Membuat Nilai Random untuk Bobot Masukan..... | 63 |
| Kode Program 5.3 Proses Inialisasi Bobot Masukan Menggunakan <i>Nguyen-Widrow</i> | 63 |
| Kode Program 5.4 Proses Membuat Nilai Random untuk Bias..... | 64 |
| Kode Program 5.5 Proses Memperbanyak Matriks Bias..... | 65 |
| Kode Program 5.6 Proses Transpose Matriks | 65 |
| Kode Program 5.7 Proses Menghitung Perkalian Matriks | 65 |
| Kode Program 5.8 Proses Menghitung Penjumlahan Matriks..... | 66 |
| Kode Program 5.9 Proses Menghitung H dengan Fungsi Aktivasi | 67 |
| Kode Program 5.10 Proses Menghitung <i>Inverse</i> Matriks | 68 |
| Kode Program 5.11 Proses Denormalisasi Data..... | 68 |
| Kode Program 5.12 Proses Perhitungan Nilai Evaluasi | 69 |



DAFTAR LAMPIRAN

| | |
|---|----|
| LAMPIRAN A DATA INDEKS HARGA KONSUMEN | 85 |
| LAMPIRAN B BOBOT MASUKAN DAN BIAS | 88 |



BAB 1 PENDAHULUAN

1.1 Latar belakang

Indeks Harga Konsumen (IHK) adalah salah satu indikator ekonomi yang memberikan informasi mengenai harga barang dan jasa yang dikonsumsi oleh rumah tangga (*household*) dalam kurun waktu tertentu (Bank Indonesia, 2016).

Sejak Januari 2014 hingga 2017, Indeks Harga Konsumen secara keseluruhan meliputi 859 komoditas barang dan jasa yang kemudian diklasifikasikan menjadi 7 kelompok, yaitu kelompok bahan makanan; kelompok makanan jadi, minuman, rokok, dan tembakau; kelompok perumahan, air, listrik, gas, dan bahan bakar; kelompok sandang; kelompok kesehatan; kelompok pendidikan, rekreasi, dan olahraga; dan kelompok transport, komunikasi, dan jasa keuangan (Bank Indonesia, 2016).

Perubahan IHK dari waktu ke waktu menggambarkan tingkat kenaikan (inflasi) atau tingkat penurunan (deflasi) dari barang dan jasa (Badan Pusat Statistik, 2017). Komponen inflasi di Indonesia terdiri dari *volatile foods*, *administered price*, *core inflation*, dan *imported inflation*. Ketujuh kelompok dari IHK termasuk ke dalam *volatile foods* atau komponen harga yang bergerak dan *administered price* atau komponen harga yang diatur pemerintah (Editor, 2017), yang berarti jika ada kenaikan harga dari ketujuh kelompok tersebut maka komponen *volatile foods* dan *administered price* akan bergerak naik dan mendorong laju inflasi domestik.

Pada tahun 2017 Inflasi yang terjadi di Indonesia menurut kelompok pengeluaran secara umum adalah sebesar 3,61%. Kelompok perumahan, air, listrik, gas, dan bahan bakar menjadi penyumbang inflasi terbesar sebanyak 5,14% (Badan Pusat Statistik, 2017). Berdasarkan data yang didapatkan dari *website* Bank Indonesia pada tahun 2012 kelompok tersebut memiliki proporsi biaya hidup paling tinggi dalam perhitungan IHK yang dilakukan oleh Badan Pusat Statistik (BPS) dibanding dengan 6 kelompok lainnya yaitu sebesar 25,37%. Maka dari itu berdasarkan data yang telah dijabarkan, prediksi Indeks Harga Konsumen kelompok perumahan, air, listrik, gas, dan bahan bakar sangat penting, karena kelompok tersebut memiliki pengaruh terhadap laju inflasi domestik yang jika diketahui nilainya lebih awal maka dapat diterapkan sebuah kebijakan untuk melakukan antisipasi supaya harga barang yang berada dalam kelompok tersebut tidak mengalami kenaikan dan tidak mendorong laju inflasi domestik.

Prediksi adalah usaha memprediksikan sesuatu yang akan terjadi di masa yang akan datang dengan memanfaatkan informasi-informasi yang bersifat relevan pada waktu sebelumnya (historis) melalui metode ilmiah. Prediksi bertujuan untuk mendapatkan informasi mengenai kejadian yang akan terjadi di masa yang akan datang dengan probabilitas terbesar (Wanto & Windarto, 2017). Beberapa penelitian telah dilakukan untuk melakukan prediksi Indeks Harga Konsumen, penelitian yang dilakukan oleh Anjar Wanto pada tahun 2017 memprediksi Indeks Harga Konsumen kelompok kesehatan menggunakan metode *Backpropagation*. Penelitian ini menggunakan data *time series* Indeks Harga Konsumen kelompok kesehatan tahun 2014 s.d. 2016, menghasilkan hasil

akhir arsitektur terbaik yang diperoleh adalah 12-70-1 dengan MSE (*Mean Squared Error*) 0,3659742.

Penelitian selanjutnya oleh Lina Irawati, Tarno, dan Hasbi Yasin pada tahun 2015 (Irawati, et al., 2015) yang melakukan prediksi Indeks Harga Konsumen 4 kota di Jawa Tengah yaitu Purwokerto, Surakarta, Semarang, dan Tegal. Penelitian ini menghasilkan hasil prediksi dengan nilai MAPE (*Mean Absolute Percentage Error*) 3,93% dan nilai RMSE (*Root Mean Squared Error*) 10,02.

Selain beberapa metode yang sudah dijabarkan sebelumnya, terdapat banyak metode yang digunakan untuk melakukan prediksi IHK. Metode yang sering digunakan untuk melakukan prediksi adalah *Support Vector Machine*, *Support Vector Regression*, *Exponential Smoothing*, *Backpropagation*, *Extreme Learning Machine*, ARIMA, dan beberapa metode prediksi lainnya.

Extreme Learning Machine (ELM) merupakan salah satu metode yang dapat digunakan untuk melakukan prediksi. ELM merupakan salah satu metode pembelajaran *feedforward* dengan satu *hidden layer* atau lebih dikenal dengan istilah *single hidden layer feedforward neural network* (SLFNs). ELM menyediakan *platform* pembelajaran terpadu dengan jenis pemetaan fitur yang tersebar luas dan dapat diterapkan pada aplikasi regresi dan klasifikasi *multiclass* secara langsung dan cenderung memiliki skalabilitas yang baik dan mencapai kesamaan dalam menangani kasus regresi (Huang, et al., 2012).

Beberapa penelitian sudah dilakukan untuk membutuhkan performa dari ELM. Penelitian pertama dilakukan oleh R. Ahsani et al. pada tahun 2017 (Ahsani, et al., 2017). Penelitian ini dilakukan untuk membandingkan kinerja dari metode *Backpropagation* dan ELM dalam melakukan pengaturan ketebalan otomatis pada mesin *press roll* untuk adonan mie instan. Penelitian ini menunjukkan bahwa performa metode ELM lebih cepat dari pada metode *Backpropagation* dengan waktu pelatihan 0,0072812 detik untuk ELM dan 0,501274 detik untuk *Backpropagation*. Hasil akhir dari penelitian ini adalah nilai yang diperoleh dari *average output error* metode ELM sebesar 0,002440673 pada data latih dan 0,6337302 pada data uji, sedangkan metode *Backpropagation* sebesar 0,006638958 pada data latih dan 0,568553072.

Penelitian kedua dilakukan oleh Zaher Mundher Yaseen, et al. pada tahun 2016 (Yaseen, et al., 2016). Penelitian ini dilakukan untuk memprediksi arus aliran air pada studi kasus daerah semi kering di Iraq menggunakan metode ELM (*Extreme Learning Machine*), SVR (*Support Vector Regression*) dan GRNN (*General Regression Neural Network*). Penelitian ini menunjukkan bahwa dibandingkan dengan metode SVR, dan GRNN metode ELM dapat mengurangi nilai RMSE dan MAE secara signifikan yakni sebesar 17,44-29,78% dan 21,3%-30,92% dengan hasil akhir RMSE dan MAE terkecil yang diperoleh dari metode ELM adalah 87,906 dan 71,544.

Metode ELM sangat baik dalam waktu eksekusi dan hasil *error* yang bisa mencapai nilai minimum jika dibandingkan dengan metode jaringan syaraf tiruan *feedforward* tradisional (Huang, et al., 2006). Meskipun begitu, terdapat beberapa tantangan untuk melakukan prediksi dengan metode ini yaitu bagaimana menginisialisasi nilai bobot yang pada metode ELM ini ditentukan secara *random*.

Penelitian sudah pernah dilakukan oleh L. D. Tavares, D. A. G. Vieira, dan R. R. Saldanha pada tahun 2014 (Tavares, et al., 2014) yang melakukan perbandingan antara hasil *error* yang dihasilkan metode *Extreme Learning Machine* menggunakan inisialisasi bobot *random* dan 6 metode inisialisasi lain yaitu metode yang diperkenalkan oleh Nguyen dan Widrow (Nguyen & Widrow, 1990), Drago dan Ridella (Drago & Ridella, 1992), Yam dan Chow (Yam & Chow, 2001), Wessels dan Barnard (Wessels & Barnard, 1992), Le Chun (Chun, 1993), serta Haffner, Shikano, dan Waibel (Haffner, et al., 1988). Dari perbandingan tersebut didapatkan hasil bahwa metode yang di perkenalkan oleh Nguyen dan Widrow serta Drago dan Ridella memiliki hasil yang paling baik dari ke-enam metode lainnya dan dapat meningkatkan performa dari metode ELM dengan mengurangi nilai *error*.

Maka dari itu berdasarkan penjabaran dari teori dan penelitian yang sudah dilakukan sebelumnya, penelitian mengenai prediksi terhadap Indeks Harga Konsumen kelompok perumahan, air, listrik, gas, dan bahan bakar akan dilakukan menggunakan metode *Extreme Learning Machine* dengan inisialisasi bobot menggunakan metode yang diperkenalkan oleh Nguyen dan Widrow untuk mengetahui Indeks Harga Konsumen di periode-periode yang akan datang. Penulis berharap penelitian ini dapat membantu pihak-pihak terkait dalam melakukan perencanaan kebijakan ekonomi di Indonesia.

1.2 Rumusan masalah

1. Berapa jumlah fitur yang paling optimal dalam prediksi Indeks Harga Konsumen Kelompok perumahan, air, listrik, gas, dan bahan bakar?
2. Berapa jumlah data latih yang optimal dalam prediksi Indeks Harga Konsumen Kelompok perumahan, air, listrik, gas, dan bahan bakar?
3. Berapa jumlah *hidden neuron* yang optimal dalam prediksi Indeks Harga Konsumen Kelompok perumahan, air, listrik, gas, dan bahan bakar?
4. Apakah fungsi aktivasi yang paling optimal dalam prediksi Indeks Harga Konsumen Kelompok perumahan, air, listrik, gas, dan bahan bakar?
5. Berapa nilai *error* yang dihasilkan metode *Extreme Learning Machine* menggunakan inisialisasi bobot secara *random* dan inisialisasi bobot menggunakan inisialisasi Nguyen-Widrow dalam prediksi Indeks Harga Konsumen kelompok perumahan, air, listrik, gas, dan bahan bakar?

1.3 Tujuan

1. Menguji jumlah fitur yang paling optimal dalam prediksi Indeks Harga Konsumen kelompok perumahan, air, listrik, gas, dan bahan bakar.
2. Menguji jumlah data latih yang optimal dalam prediksi Indeks Harga Konsumen kelompok perumahan, air, listrik, gas, dan bahan bakar.
3. Menguji jumlah *hidden neuron* yang optimal dalam prediksi Indeks Harga Konsumen elompok perumahan, air, listrik, gas, dan bahan bakar.

4. Mengetahui fungsi aktivasi yang optimal dalam prediksi Indeks Harga Konsumen kelompok perumahan, air, listrik, gas, dan bahan bakar.
5. Mengetahui nilai *error* yang dihasilkan metode *Extreme Learning Machine* menggunakan inisialisasi bobot secara *random* dan inisialisasi bobot menggunakan inisialisasi Nguyen-Widrow dalam prediksi Indeks Harga Konsumen kelompok perumahan, air, listrik, gas, dan bahan bakar.

1.4 Manfaat

Manfaat dari penelitian ini adalah membantu Bank Indonesia dalam melakukan prediksi Indeks Harga Konsumen Kelompok perumahan, air, listrik, gas, dan bahan bakar serta membantu pihak-pihak terkait dalam melakukan perencanaan kebijakan ekonomi di Indonesia.

1.5 Batasan masalah

Batasan masalah dari penelitian ini adalah sebagai berikut.

1. Prediksi yang dilakukan adalah prediksi terhadap Indeks Harga Konsumen Kelompok perumahan, air, listrik, gas, dan bahan bakar.
2. Metode yang digunakan adalah metode *Extreme Learning Machine* dan inisialisasi Nguyen-Widrow.
3. Data yang digunakan berupa *data time series* Indeks Harga Konsumen periode Januari 2011 s.d. Desember 2017.

1.6 Sistematika pembahasan

Sistematika penulisan ini dibagi menjadi tujuh bab dengan masing-masing bab diuraikan sebagai berikut.

BAB I : PENDAHULUAN

Pada bab ini memuat latar belakang permasalahan dan alasan dilakukan penelitian tentang prediksi terhadap Indeks Harga Konsumen Kelompok perumahan, air, listrik, gas, dan bahan bakar dengan metode *Extreme Learning Machine* (ELM) dan inisialisasi Nguyen-Widrow, rumusan masalah yang dibahas dalam penelitian, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika pembahasan.

BAB II : LANDASAN KEPUSTAKAAN

Pada bab ini berisi kajian pustaka terhadap penelitian-penelitian yang telah dilakukan sebelumnya yang memiliki kaitan dengan topik penelitian yang diangkat serta teori-teori yang berhubungan dengan penelitian untuk mendukung penelitian dalam implementasi metode *Extreme Learning Machine* (ELM) dan inisialisasi Nguyen-Widrow pada prediksi Indeks Harga Konsumen Kelompok perumahan, air, listrik, gas, dan bahan bakar.

BAB III : METODOLOGI

Pada bab ini menguraikan langkah-langkah sistematis dalam memprediksi Indeks Harga Konsumen kelompok perumahan, air, listrik, gas, dan bahan bakar. dengan menggunakan metode *Extreme Learning Machine* (ELM) dengan inialisasi bobot menggunakan inialisasi Nguyen-Widrow.

BAB IV : PERANCANGAN

Bab ini berisi perancangan dari sistem yang dibuat dalam penelitian, meliputi formulasi permasalahan, penyelesaian masalah menggunakan metode *Extreme Learning Machine* (ELM) dan inialisasi bobot Nguyen-Widrow, serta menganalisis metode, perancangan antarmuka sistem dan perancangan pengujian sistem.

BAB V : IMPLEMENTASI

Pada bab ini berisi proses implementasi sistem sesuai dengan perancangan sistem yang telah ditentukan sebelumnya.

BAB VI : PENGUJIAN DAN ANALISIS

Pada bab ini dilakukan pengujian serta analisis nilai evaluasi terbaik dari metode *Extreme Learning Machine* (ELM) dengan inialisasi Nguyen-Widrow melalui pengujian jumlah fitur, pengujian jumlah data latih, pengujian jumlah *hidden neuron*, pengujian fungsi aktivasi, dan pengujian perbandingan metode.

BAB VII : PENUTUP

Pada bab ini menguraikan kesimpulan yang diperoleh dari penelitian yang telah dilakukan, serta saran-saran untuk pengembangan lebih lanjut.

BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini akan dibahas landasan kepastakaan yang membahas mengenai teori-teori pendukung dalam penyusunan penelitian ini.

1.1 Kajian Pustaka

Kajian pustaka dilakukan untuk mengkaji penelitian-penelitian yang telah dilakukan sebelumnya. Penelitian-penelitian tersebut berupa penelitian yang berhubungan dengan metode atau objek yang digunakan dalam penelitian ini yaitu Indeks Harga Konsumen, *Extreme Learning Machine* (ELM) atau Nguyen-Widrow.

Penelitian yang dilakukan oleh Anjar Wanto dan Agus Perdana Windarto pada tahun 2017 (Wanto & Windarto, 2017) menggunakan metode *Backpropagation* untuk memprediksi Indeks Harga Konsumen kelompok kesehatan. Data yang digunakan dalam penelitian ini adalah data Indeks Harga Konsumen kelompok kesehatan yang berasal dari Badan Pusat Statistik kota Medan periode Januari 2014 s.d. Desember 2016. Hasil dari penelitian ini adalah nilai prediksi dari Indeks Harga Konsumen kelompok kesehatan. Penelitian ini menggunakan 8 model arsitektur yaitu 12-5-1, 12-26-1, 12-29-1, 12-35-1, 12-40-1, 12-60-1, 12-70-1, 12-75-1. Arsitektur terbaik yang didapatkan pada penelitian ini adalah model 12-70-1 dengan tingkat akurasi sebesar 92% dan MSE 0,3659742.

Penelitian Indeks Harga Konsumen pun telah dilakukan oleh Lina Irawati, Tarno, dan Hasbi Yasin pada tahun 2015 (Irawati, et al., 2015), tetapi pada penelitian ini dilakukan prediksi Indeks Harga Konsumen di 4 kota yaitu Purwokerto, Surakarta, Semarang, dan Tegal menggunakan model *Generalized Space Time Autoregressive* (GSTAR). Data yang digunakan dalam penelitian ini berupa data sekunder yaitu data Indeks Harga Konsumen di Purwokerto, Surakarta, Semarang, dan Tegal periode Juni 2008 s.d. Desember 2013 yang diperoleh dari website Badan Pusat Statistik. Hasil dari penelitian ini berupa nilai prediksi periode Januari 2014-Januari 2015 serta model GSTAR terbaik untuk data Indeks Harga Konsumen di keempat kota tersebut adalah model GSTAR (1₁) I(1) dengan rata-rata nilai MAPE 3,93% dan RMSE 10,20.

Selanjutnya adalah prediksi Indeks Harga Konsumen di Palu, Sulawesi Tengah yang dilakukan oleh Y. Wigati, Rais, dan I. T. Utami pada tahun 2016 (Wigati, et al., 2016) menggunakan metode *Autoregressive Model* (ARIMA). Data yang digunakan pada penelitian ini adalah data Indeks Harga Konsumen di Palu periode Januari 2004 s.d. Agustus 2015. Penelitian ini memperoleh hasil bahwa model ARIMA terbaik yang digunakan untuk melakukan prediksi adalah model ARIMA (1,1,1).

Dari ketiga penelitian sebelumnya dapat dilihat bahwa perlu dilakukan prediksi terhadap Indeks Harga Konsumen karena Indeks Harga Konsumen merupakan salah satu indikator yang dapat digunakan untuk mengukur tingkat inflasi atau deflasi di Indonesia. Sedangkan penelitian mengenai prediksi untuk memprediksi suatu objek menggunakan metode *Extreme Learning Machine* sudah pernah dilakukan. Penjelasan mengenai penelitian tersebut akan dijabarkan pada penjelasan selanjutnya.

Penelitian yang dilakukan oleh Xia Chen, et al. pada tahun 2012 (Chen, et al., 2012), penelitian ini membandingkan metode *Backpropagation Neural Network* (BPNN), *Radial Basis Function Neural Network* (RBFNN), dan *Extreme Learning Machine* (ELM) dalam memprediksi harga listrik. Data yang digunakan dalam penelitian ini adalah data harga pasar yang diambil dari website *Australian Energy Market Operator* (AEMO) periode 1 Juni 2006 sampai 31 Mei 2007 yang dibagi berdasarkan 4 musim. Musim dingin pada bulan Juni-Agustus, musim semi pada bulan September-November, musim panas pada bulan Desember-Februari dan musim gugur pada bulan Maret-Mei, yang mewakili variasi permintaan dan harga listrik yang berbeda di setiap musimnya. Hasil dari penelitian ini adalah nilai prediksi dari ketiga metode yang dibagi berdasarkan 4 musim serta menunjukkan bahwa ELM memiliki performa yang lebih unggul dari BPNN dan RBFNN dengan nilai MAPE, MAE, dan RMSE terendah pada musim dingin (MAPE = 2,0278%; MAE = 8,3372%; RMSE = 2,9371), musim panas (MAPE = 10,1656%; MAE = 21,8798; RMSE = 16,5881), dan musim gugur (MAPE = 7,3193%; MAE = 12,7363%, RMSE = 10,3818).

Hal serupa juga terlihat dari penelitian yang dilakukan oleh Zaher Mundher Yaseen, et al. pada tahun 2016 (Yaseen, et al., 2016). Penelitian ini dilakukan untuk memprediksi arus aliran air pada studi kasus daerah semi kering di Iraq menggunakan metode *Extreme Learning Machine* (ELM), *Support Vector Regression* (SVR) dan *Generalized Regression Neural Network* (GRNN). Hasil dari penelitian ini adalah berupa nilai prediksi arus aliran air. Penelitian ini menunjukkan bahwa dibandingkan dengan metode SVR, dan GRNN, metode ELM dapat mengurangi nilai RMSE dan MAE secara signifikan yakni sebesar 17,44-29,78% dan 21,3-30,92% dengan hasil akhir RMSE dan MAE terkecil yang diperoleh dari metode ELM adalah 87,906 dan 71,544.

Selanjutnya adalah penelitian oleh Jefri Junifer Pangaribuan untuk membandingkan kinerja metode *Extreme Learning Machine* (ELM) dan *Backpropagation* (BP) dalam melakukan diagnosis penyakit diabetes melitus pada tahun 2014 dengan menggunakan data set diabetes dari UCI Repository oleh Vincent Sigillito (Pangaribuan, 2014). Hasil dari penelitian ini adalah hasil diagnosis pada pasien diabetes. Penelitian ini menunjukkan bahwa dari segi kecepatan ELM menunjukkan performa yang lebih baik dengan waktu eksekusi pada data latih 0,1687 detik dan data uji 0,0044 detik sedangkan pada BP memiliki waktu eksekusi data latih sebesar 0,5234 detik dan data uji 0,0226 detik. Hal ini juga selaras dengan tingkat akurasi, yang mana ELM memiliki rata-rata MSE lebih kecil yakni pada data latih sebesar 0,367 dan data uji 0,393 sedangkan untuk BP memiliki MSE pada data latih 0,411 dan data uji 0,842.

Penjelasan terkait penelitian sebelumnya yang ada pada kajian pustaka secara rinci dijelaskan pada Tabel 2.1.

Tabel 2.1 Kajian Pustaka

| No. | Pustaka | Objek | Metode | Output |
|-----|--|--|--|--|
| 1 | Anjar Wanto, Agus Perdana Windarto, 2017 | Indeks Harga Konsumen (IHK) kelompok kesehatan. | <i>Backpropagation</i> | Model arsitektur terbaik untuk melakukan prediksi dan nilai prediksi Indeks Harga Konsumen kelompok kesehatan |
| 2 | Lina Irawati, Tarno, Hasbi, 2015 | Indeks Harga Konsumen (IHK) di Purwokerto, Surakarta, Semarang, dan Tegal. | <i>Generalized Space Time Autoregressive</i> | Model GSTAR terbaik untuk melakukan prediksi dan nilai prediksi Indeks Harga Konsumen di 4 kota |
| 3 | Y. Wigati, Rais, I. T. Utami, 2016 | Indeks Harga Konsumen (IHK) di Palu, Sulawesi Tengah. | <i>Autoregressive Model (ARIMA)</i> | Model ARIMA terbaik untuk melakukan prediksi dan nilai prediksi Indeks Harga Konsumen di Palu |
| 4 | Xia Chen, et al, 2012 | Harga listrik. | <i>Backpropagation Neural Network (BPNN), Radial Basis Function Neural Network (RBFNN), dan Extreme Learning Machine (ELM)</i> | Nilai prediksi harga listrik di 4 musim yaitu musim dingin pada bulan Juni-Agustus, musim semi pada bulan September-November, musim panas pada bulan Desember-Februari, dan musim gugur pada bulan Maret-Mei |
| 5 | Zaher Mundher Yaseem, et al, 2016 | Arus aliran air di daerah semi kering di Iraq. | <i>Extreme Learning Machine (ELM), Support Vector Regression (SVR) dan Generalized Regression Neural Network (GRNN)</i> | Nilai prediksi aliran arus air |
| 6 | Jefri Junifer Pangaribuan, 2014 | Data diabetes. | <i>Extreme Learning Machine (ELM), Backpropagation (BP)</i> | Hasil diagnosis diabetes |

Tabel 2.1 Kajian Pustaka (lanjutan)

| No. | Pustaka | Objek | Metode | Output |
|-----|--|--|--|--|
| 7 | L. D. Tavares, R. R. Saldanha, D. A. G. Vieira, 2014 | <i>Abalone, Auto Price, Bank, Breast Cancer, California Housing, Census, Computer Activity, Delta Ailerons, Delta Elevators, Machine CPU, Servo, Triazines.</i> | <i>Extreme Learning Machine</i> menggunakan inialisasi bobot dengan metode <i>random</i> , <i>Nguyen-Widrow</i> , <i>Drago and Ridella</i> , <i>Yam and Chow</i> , <i>Wessels and Barnard</i> , <i>Le Chun</i> , dan <i>Haffner et al.</i> | Hasil perbandingan performa ELM dengan inialisasi bobot menggunakan metode <i>random</i> , <i>Nguyen-Widrow</i> , <i>Drago and Ridella</i> , <i>Yam and Chow</i> , <i>Wessels and Barnard</i> , <i>Le Chun</i> , dan <i>Haffner et al.</i> |
| 8 | Guang-Bin Huang, Qin-Yu Zhu, Chee-Kheong Siew, 2006 | <i>Abalone, Delta Ailerons, Delta Elevators, Computer Activity, Census (house8L), Auto Price, Triazines, Machine CPU, Servo, Breast Cancer, Bank Domains, California Housing, Stocks Domain.</i> | <i>Extreme Learning Machine</i> , <i>Backpropagation (BP)</i> , <i>Support Vector Regression (SVR)</i> . | Hasil perbandingan performa ELM dengan metode BP dan SVR. |
| 9 | Guang-Bin Huang, Hongming Zhou, Xiaojian Ding, Rui Zhang, 2012 | <ul style="list-style-type: none"> - <i>Binary class datasets</i> <i>Diabetes, Australian Credit, Liver, Banana, Colon, Colon (Gene Sel), Leukimia, Leukimia (Gene Sel), Brightdata, Dimdata, Mushroom, Adult</i> - <i>Multi-class datasets</i> <i>Iris, Glass, Wine, Ecoli, Vowel, Vehicle, Segment, Satimage, DNA, Letter, Shuttle, USPS</i> - <i>Regression datasets</i> <i>Baskball, Cloude, Autoprice, Strike, Pyrim, Bodyfat, Cleveland, Housing, Balloon, Quake, Space-ga, Abalone</i> | <i>Extreme Learning Machine</i> , <i>Support Vector Machine (SVM)</i> , <i>Least Square-Support Vector Machine (LS-SVM)</i> | Hasil perbandingan performa ELM dengan metode SVM dan LS-SVM. Output yang dihasilkan adalah ELM memiliki performa yang lebih baik dari dua metode yang lain, baik dari nilai <i>error</i> dan waktu eksekusi. |

Tabel 2.1 Kajian Pustaka (lanjutan)

| No. | Pustaka | Objek | Metode | Output |
|-----|--|--|---|---|
| 10 | Imam Cholissodin, Sutrisno, Arief Andy Soebroto, Latifah Hanum, Canny Amerilyse Caesar, 2017 | Optimasi kandungan gizi susu kambing Peranakan Etawa (PE) | <i>Extreme Learning Maching</i> (ELM) – <i>Particle Swarm Optimization</i> (PSO) | Pemodelan pakan kambing dalam mengoptimasi kandungan gizi susu kambing. |



1.2 Prediksi

Prediksi adalah usaha memprediksikan sesuatu yang akan terjadi di masa mendatang dengan memanfaatkan informasi-informasi yang bersifat relevan pada sebelumnya (historis) melalui metode ilmiah. Prediksi bertujuan untuk mendapatkan informasi mengenai kejadian yang akan terjadi di masa yang akan datang dengan probabilitas terbesar (Wanto & Windarto, 2017). Prediksi dapat menjadi bahan pertimbangan nilai dan analisis data untuk masa yang akan datang. (Syafuruddin, et al., 2014). Prediksi tidak harus memberikan jawaban secara pasti tentang kejadian yang akan terjadi, melainkan berusaha untuk mencari jawaban sedekat mungkin yang akan terjadi (Herdianto, 2013).

1.3 Indeks Harga Konsumen

Indeks yang menghitung rata-rata perubahan harga dari suatu paket barang dan jasa yang dikonsumsi oleh rumah tangga dalam kurun waktu tertentu. IHK merupakan indikator yang digunakan untuk mengukur tingkat inflasi. Perubahan IHK dari waktu ke waktu menggambarkan tingkat kenaikan (inflasi) atau tingkat penurunan (deflasi) dari barang dan jasa (Badan Pusat Statistik, 2017). Perhitungan IHK dilakukan untuk merekam perubahan harga beli di tingkat konsumen (*purchasing cost*) dari sekelompok tetap barang dan jasa (*fixed basket*) yang pada umumnya dikonsumsi masyarakat (Bank Indonesia, 2016)

Komoditas barang dan jasa yang dipilih dalam perhitungan IHK berdasar pada Survei Biaya Hidup (SBH) 2012 yang digunakan sebagai dasar menentukan paket komoditas, penimbang, cakupan kota, dan tahun dasar dalam pengolahan IHK. Perhitungan IHK dilakukan oleh Badan Pusat Statistik (BPS) setiap bulannya.

Indeks Harga Konsumen diklasifikasikan menjadi 7 kelompok pengeluaran barang dan jasa, yaitu: kelompok bahan makanan; kelompok makanan jadi, minuman, rokok dan tembakau; kelompok perumahan, air, listrik, gas dan bahan bakar; kelompok sandang; kelompok kesehatan; kelompok pendidikan, rekreasi, dan olahraga; dan kelompok transportasi, komunikasi, dan jasa keuangan (Bank Indonesia, 2016).

1.4 Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan (JST) adalah sistem pemroses informasi yang memiliki karakteristik mirip dengan jaringan syaraf biologi. JST pertama kali diperkenalkan oleh McCulloch dan Pitts di tahun 1943. McCulloch dan Pitts menyimpulkan bahwa kombinasi beberapa *neuron* sederhana menjadi sebuah sistem neural akan meningkatkan kemampuan komputasinya. JST dibentuk sebagai generalisasi model matematika dari jaringan syaraf biologi, dengan asumsi bahwa pemrosesan informasi terjadi pada banyak elemen sederhana (*neuron*), sinyal dikirimkan diantara *neuron-neuron* melalui banyak penghubung, Penghubung antar neuron memiliki bobot yang akan memperkuat atau memperlemah sinyal, dan untuk menentukan *output*, setiap neuron menggunakan fungsi aktivasi yang dikenakan

pada jumlah masukan yang diterima. Jaringan Syaraf Tiruan ditentukan oleh 3 hal (Siang, 2005):

- Pola hubungan antar *neuron* (disebut arsitektur jaringan)
- Metode untuk menentukan bobot penghubung (disebut metode pelatihan/*learning*/algoritme)
- Fungsi aktivasi

1.5 Normalisasi dan Denormalisasi Data

Normalisasi data merupakan proses pengolahan data yang bertujuan untuk memetakan data ke dalam *range-range* tertentu. Salah satu metode normalisasi yang dapat digunakan adalah metode *Min-Max*. Metode *Min-Max* adalah teknik sederhana yang mencari nilai minimal dan maksimal pada setiap data (Patro & Sahu, 2015). Cara melakukan normalisasi ditunjukkan pada Persamaan 2.1 (Leung, 2007).

$$x' = \frac{(x - x_{min})}{(x_{max} - x_{min})} \quad (2.1)$$

Keterangan:

- x' = Data normalisasi
- x = Nilai data pada indeks ke- i ; $i = 1, 2, 3, \dots, n$
- n = Jumlah data
- x_{min} = Nilai minimal
- x_{max} = Nilai maksimal

Setelah data sudah melewati proses normalisasi lalu diolah menggunakan metode tertentu, maka yang selanjutnya harus dilakukan adalah melakukan denormalisasi atau mengembalikan nilai menjadi nilai sebenarnya atau nilai dalam bentuk aktual. Cara melakukan denormalisasi ditunjukkan pada Persamaan 2.2.

$$x = x_{min} + (x'(x_{max} - x_{min})) \quad (2.2)$$

1.6 Fungsi Aktivasi

Fungsi aktivasi merupakan suatu fungsi yang akan mentransformasikan suatu masukan menjadi suatu keluaran tertentu. Berikut ini merupakan beberapa jenis fungsi aktivasi (Bahiuddin, et al., 2017).

1. Fungsi Sin

$$H = \sin(H_{init}) \quad (2.3)$$

2. Fungsi Sigmoid Biner

$$H = \frac{1}{1 + \exp^{-H_{init}}} \quad (2.4)$$

3. Fungsi Triangular Basis

$$H = \begin{cases} 1 - abs H_{init}, & \text{jika } 1 \leq H_{init} \leq 0 \\ 0, & \text{jika tidak} \end{cases} \tag{2.5}$$

4. Fungsi Radial Basis

$$H = \exp(-((H_{init})^2)) \tag{2.6}$$

Keterangan:

$$H_{init} = X_{train} W^T + b (\text{ones}(i_{train}, 1), :) \tag{2.7}$$

1.7 Nguyen-Widrow

Nguyen-Widrow adalah sebuah algoritme yang digunakan untuk inialisasi bobot pada jaringan syaraf tiruan (Andrian & Wayahdi, 2014). Metode tersebut adalah metode yang saat ini diterapkan di Matlab *neural network toolbox*. Metode ini melakukan inialisasi ke dalam dua tahap, tahap pertama adalah tahap yang menggunakan bobot acak dan tahap kedua menggunakan Persamaan 2.8 (Tavares, et al., 2014).

$$\begin{aligned} \hat{w} &= U(-1,1) \\ w &= \frac{\beta \hat{w}}{\gamma} \end{aligned} \tag{2.8}$$

Dimana,

$$\beta = 0,7^{\frac{1}{d_{in}}} \tag{2.9}$$

$$\gamma = \sqrt{\sum(\hat{w})^2} \tag{2.10}$$

Keterangan:

$U(-1,1)$ = Nilai random dengan *range* [-1,1]

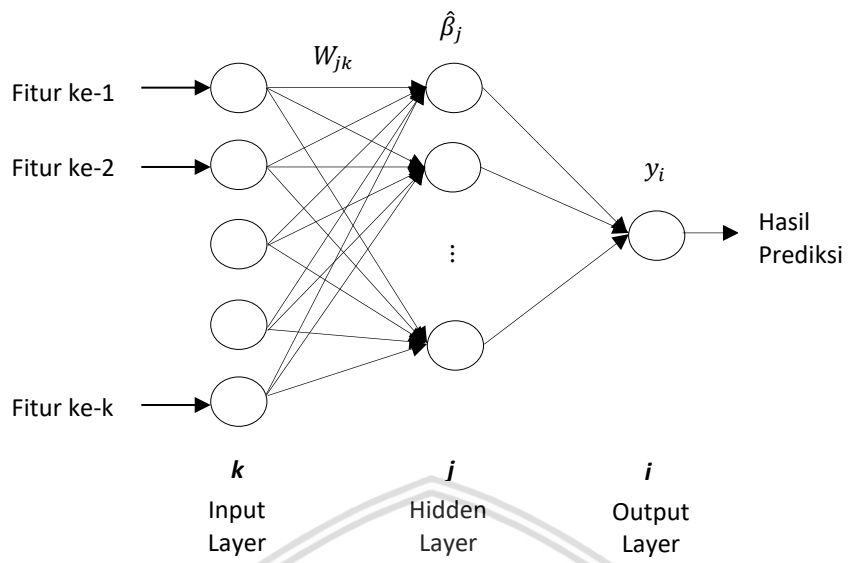
d_{in} = Jumlah fitur

1.8 Extreme Learning Machine

Extreme Learning Machine (ELM) merupakan metode pembelajaran baru jaringan syaraf tiruan yang pertama kali diperkenalkan oleh Huang et al. ELM adalah metode jaringan syaraf tiruan *feedforward* dengan satu *hidden layer* atau biasa disebut dengan *single hidden layer feedforward* (SLFNs). Metode ELM dibuat untuk mengatasi kelemahan-kelemahan metode jaringan syaraf tiruan konvensional terutama dalam hal *learning speed*.

Pada metode konvensional bobot masukan atau bias melalui proses pelatihan terlebih dahulu, tetapi pada algoritma ELM parameter-parameter seperti bobot masukan dan bias dipilih secara *random* dan cara memperoleh bobot keluarannya dengan menggunakan *norm least squares solution* dan *moore-penrose inverse* pada sistem linier secara umum, sehingga ELM memiliki *learning speed* yang lebih baik dibanding metode konvensional seperti *Support Vector Regression* (SVR) dan *Backpropagation* (Huang, et al., 2006).





Gambar 2.1 Struktur Metode ELM
 Sumber: (Cholissodin, et al., 2017)

1.8.1 Proses Pelatihan

Proses pelatihan merupakan proses untuk melakukan pelatihan dengan menggunakan data latih, proses pelatihan dilakukan untuk mendapatkan nilai bobot keluaran yang nantinya akan digunakan pada proses pengujian. Sebelum melakukan proses pelatihan, normalisasi data dilakukan terlebih dahulu dengan *range* [0,1]. Berikut merupakan proses pelatihan (Cholissodin, et al., 2017).

1. Membuat nilai matriks bobot W_{jk} yang nilainya di inialisasi secara *random* sebagai bobot masukan dengan *range* [-1,1] berukuran j (jumlah *hidden neuron*) $\times k$ (jumlah *neuron* masukan) dan matriks bias b yang juga diinisialisasi secara *random* dengan *range* [0,1] berukuran $1 \times$ jumlah *hidden neuron*.
2. Melakukan inialisasi bobot masukan dengan menggunakan inialisasi Nguyen-Widrow menggunakan Persamaan 2.8.
3. Menghitung matriks keluaran pada *hidden layer*.

$$H = \frac{1}{1 + \exp^{-X_{train} W^T + b (ones(i_{train}, 1), :)}} \tag{2.11}$$

Keterangan:

- H = Matriks keluaran pada *hidden layer*
- X_{train} = Matriks masukan pada data latih
- W^T = Matriks *transpose* bobot
- i_{train} = Jumlah data latih
- $b (ones(i_{train}, 1), :)$ = Perhitungan untuk memperbanyak matriks bias sejumlah data latih

b = Matriks bias

4. Menghitung matriks *pseudo-inverse* dengan *Moore-Penrose*

$$H^+ = (H^T H)^{-1} H^T \tag{2.12}$$

Keterangan:

H^+ = Matriks *pseudo-inverse*

H = Matriks keluaran pada *hidden layer*

H^T = Matriks keluaran yang telah di *transpose*

5. Menghitung $\hat{\beta}$ sebagai bobot keluaran dengan persamaan 2.13.

$$\hat{\beta} = H^+ T \tag{2.13}$$

Keterangan:

$\hat{\beta}$ = Matriks bobot keluaran

T = Matriks target

1.8.2 Proses Pengujian

Setelah dilakukan proses pelatihan maka proses selanjutnya adalah melakukan proses pengujian. Tahapan-tahapan yang dilakukan adalah sebagai berikut (Cholissodin, et al., 2017).

1. Menggunakan nilai matriks bobot W_{jk} dengan *range* [-1,1] dan matriks bias b dengan *range* [0,1], dan matriks bobot keluaran yang sama seperti pada proses pelatihan.
2. Menghitung matriks keluaran pada *hidden layer*.

$$H = \frac{1}{1 + \exp^{-X_{train} W^T + b(\text{ones}(i_{train}, 1), :)}} \tag{2.14}$$

Keterangan:

H = Matriks keluaran pada *hidden layer*

X_{test} = Matriks masukan pada data uji

W^T = Matriks *transpose* bobot

i_{test} = Jumlah data uji

$b(\text{ones}(i_{test}, 1), :)$ = Perhitungan untuk memperbanyak matriks bias sejumlah data uji

b = Matriks bias

3. Menghitung \hat{Y} sebagai hasil keluaran *output layer*

$$\hat{Y} = H \hat{\beta} \tag{2.15}$$

Keterangan:

\hat{Y} = Hasil prediksi



H = Matriks keluaran pada *hidden layer*

$\hat{\beta}$ = Matriks bobot keluaran

Setelah proses pengujian dilakukan maka selanjutnya adalah melakukan denormalisasi hasil keluaran *output layer* untuk mengubah nilai menjadi hasil prediksi aktual menggunakan Persamaan 2.2. Kemudian menghitung nilai evaluasi menggunakan RMSE.

1.8.3 Root Mean Squared Error (RMSE)

Root Mean Squared Error (RMSE) merupakan aturan penskalaan kuadrat yang juga mengukur besaran kesalahan rata-rata. RMSE adalah akar kuadrat dari rata-rata perbedaan kuadrat antara prediksi dan observasi aktual (JJ, 2016). Root Mean Square Error (RMSE) telah digunakan sebagai metrik statistik standar untuk mengukur kinerja model dalam penelitian meteorologi, kualitas udara, dan penelitian iklim (Tianfeng & Draxler, 2014). Persamaan untuk menghitung RMSE adalah seperti yang ditunjukkan oleh Persamaan 2.16 (Yaseen, et al., 2016).

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (Y_i - \hat{Y}_i)^2} \quad (2.16)$$

Keterangan:

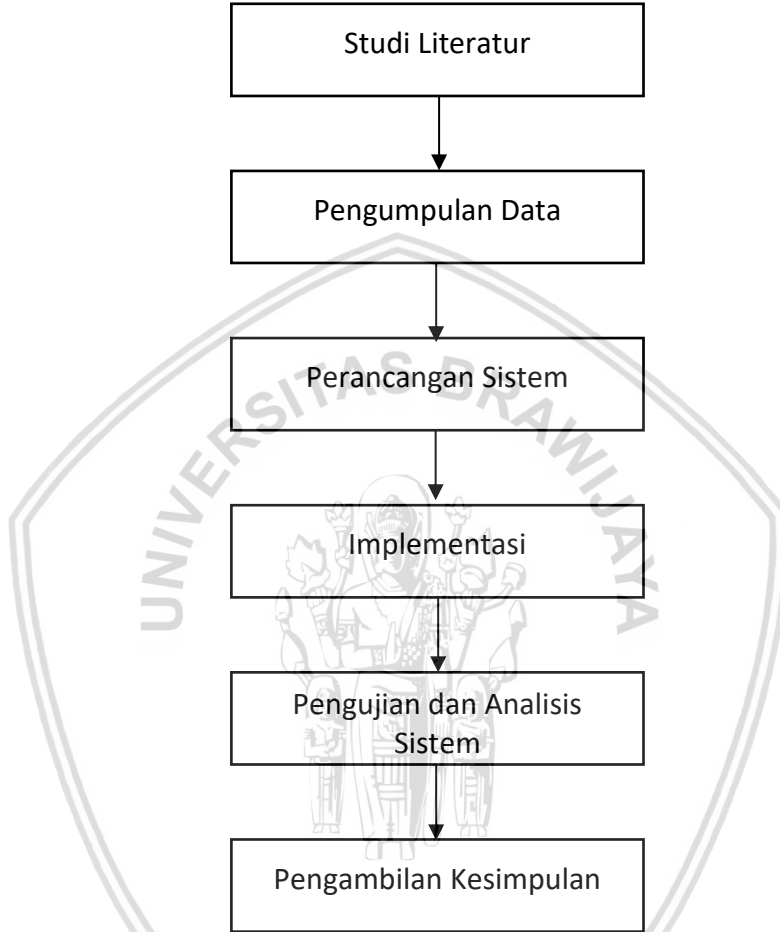
Y_i = Data aktual ke- i

\hat{Y}_i = Nilai prediksi ke- i

N = Jumlah data

BAB 3 METODOLOGI

Pada bab ini akan dibahas mengenai metode dan tahapan yang dilakukan dalam prediksi Indeks Harga Konsumen kelompok perumahan, air, listrik, gas dan bahan bakar dengan metode *Extreme Learning Machine* dan inisialisasi bobot Nguyen-Widrow. Tahapan penelitian ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram Alir Metode Penelitian

1.1 Studi Literatur

Studi literatur dilakukan untuk mempelajari teori-teori yang berkaitan dengan penelitian untuk digunakan sebagai pedoman penelitian. Studi literatur yang dibutuhkan dalam penelitian ini meliputi: Indeks Harga Konsumen (IHK), Prediksi, *Extreme Learning Machine* dan *Root Mean Squared Error* (RMSE). Studi literatur diperoleh melalui buku, *paper*, jurnal penelitian-penelitian sebelumnya dan informasi dari situs-situs resmi yang tersedia secara *online*.



1.2 Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah data *time series* Indeks Harga Konsumen kelompok perumahan, air, listrik, gas, dan bahan bakar yang didapatkan melalui website resmi Bank Indonesia (<http://bi.go.id>). Data tersebut merupakan data kuantitatif berupa angka indeks per bulan dari Indeks Harga Konsumen kelompok perumahan, air, listrik, gas, dan bahan bakar periode Januari 2011 s.d. Desember 2017. Contoh data Indeks Harga Konsumen kelompok perumahan, air, listrik, gas, dan bahan bakar ditunjukkan pada Tabel 3.1.

Tabel 3.1 Data Indeks Harga Konsumen Tahun 2014

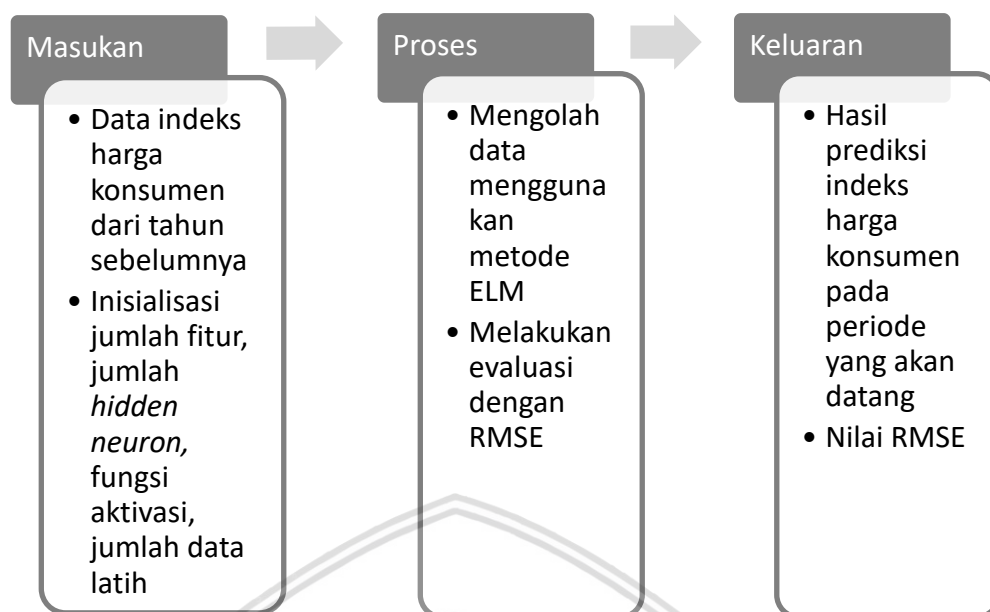
| Tahun | Bulan | Indeks |
|-------|-----------|--------|
| 2014 | Januari | 108,72 |
| | Februari | 108,9 |
| | Maret | 109,07 |
| | April | 109,34 |
| | Mei | 109,59 |
| | Juni | 110,01 |
| | Juli | 110,5 |
| | Agustus | 111,31 |
| | September | 112,17 |
| | Oktober | 113,34 |
| | November | 113,9 |
| | Desember | 115,55 |

1.3 Perancangan Sistem

Perancangan merupakan tahap yang menggambarkan rancangan penelitian secara detail. Perancangan sistem dilakukan untuk mempermudah proses implementasi, pengujian, dan analisis.

1.3.1 Mekanisme Penyelesaian Masalah

Mekanisme penyelesaian masalah diawali dengan melakukan identifikasi masalah secara rinci, lalu menyediakan kebutuhan penyelesaian berupa data masukan dan metode yang akan digunakan untuk menyelesaikan masalah tersebut. Penyelesaian masalah meliputi langkah-langkah metode ELM yang digambarkan dalam bentuk diagram alir pada proses pelatihan dan pengujian data menggunakan metode ELM serta perhitungan nilai evaluasi menggunakan RMSE. Mekanisme penyelesaian masalah pada penelitian ini digambarkan melalui diagram blok pada Gambar 3.2.



Gambar 3.2 Diagram Blok Mekanisme Penyelesaian Masalah

Penjelasan dari Gambar 2.3 adalah sebagai berikut.

- Masukan sistem adalah data latih dan data uji yang diperoleh dari keseluruhan data yang disimpan dalam file .xls, jumlah fitur, jumlah *hidden neuron*, jumlah data latih, serta jenis fungsi aktivasi.
- Proses penyelesaian masalah dengan metode ELM untuk data latih dan data uji serta pengujian untuk mengetahui nilai RMSE.
- Keluaran yang dihasilkan adalah hasil prediksi Indeks Harga Konsumen beserta nilai RMSE.

1.3.2 Perancangan Antarmuka Pengguna

Perancangan antarmuka merupakan sebuah penghubung yang dapat menjelaskan bagaimana pengguna akan berinteraksi dengan sistem. Perancangan antarmuka dilakukan untuk memudahkan pengguna dalam menggunakan sistem.

1.3.3 Perancangan Pengujian

Perancangan pengujian dilakukan untuk menguji sistem untuk menghasilkan nilai RMSE seminimal mungkin. Perancangan pengujian sistem yang dilakukan adalah pengujian jumlah fitur, jumlah data latih, jumlah *hidden neuron*, dan fungsi aktivasi.

1.4 Implementasi

Implementasi dilakukan untuk mentransformasikan hasil analisis dan perancangan menjadi bentuk yang nyata. Sistem prediksi diimplementasikan ke dalam perangkat lunak dengan implementasi metode ELM yang selanjutnya dioperasikan untuk dilakukan pengujian dan evaluasi hasil. Sistem

diimplementasikan ke dalam perangkat keras dengan spesifikasi 4 GB RAM, *Processor* Intel Core i5 2.40 GHz melalui perangkat lunak sistem operasi Microsoft Windows 10 64bit, editor pemrograman Netbeans IDE, editor dokumentasi Microsoft Office Word 2016 dan Microsoft Excel 2016. Sistem diimplementasikan dengan bahasa pemrograman Java dengan menerapkan algoritma ELM dalam kode program serta pembuatan antarmuka yang disesuaikan dengan analisis kebutuhan dan perancangan yang telah dibuat.

1.5 Pengujian dan Analisis

Pengujian dilakukan untuk menilai tingkat keberhasilan sistem yang telah dibangun kemudian melakukan evaluasi untuk mencapai tujuan yang telah ditentukan. Pengujian yang dilakukan dalam penelitian ini adalah sebagai berikut.

1. Pengujian jumlah fitur dan pengaruhnya terhadap nilai RMSE.

Tabel 3.2 Tabel Pengujian Jumlah Fitur

| Jumlah Fitur | RMSE |
|--------------|------|
| 1 Bulan | |
| 2 Bulan | |
| 3 Bulan | |
| 4 Bulan | |
| 5 Bulan | |
| 6 Bulan | |
| 7 Bulan | |
| 8 Bulan | |
| 9 Bulan | |
| 10 Bulan | |

2. Pengujian jumlah data latih untuk mengetahui berapa jumlah data latih paling optimal yang digunakan dalam proses pelatihan.

Tabel 3.3 Tabel Pengujian Jumlah Data Latih

| Jumlah Data latih | RMSE |
|-------------------|------|
| 10 | |
| 20 | |
| 30 | |
| 40 | |
| 50 | |
| 60 | |

- Pengujian jumlah *hidden neuron* pada *hidden layer* dan pengaruhnya terhadap nilai RMSE.

Tabel 3.4 Tabel Pengujian Jumlah *Hidden Neuron*

| Jumlah Hidden Neuron | RMSE |
|----------------------|------|
| 1 | |
| 2 | |
| 3 | |
| 4 | |
| 5 | |
| ... | |
| 20 | |

- Pengujian fungsi aktivasi dan pengaruhnya terhadap nilai RMSE.

Tabel 3.5 Tabel Pengujian Fungsi Aktivasi

| Fungsi Aktivasi | RMSE |
|-------------------------|------|
| Fungsi Sin | |
| Fungsi Sigmoid | |
| Fungsi Triangular Basis | |
| Fungsi Radial Basis | |

- Pengujian perbandingan metode dan pengaruhnya terhadap nilai RMSE.

Tabel 3.6 Tabel Pengujian Perbandingan Metode

| Fungsi Aktivasi | RMSE |
|-----------------|------|
| ELM-NW | |
| ELM | |

1.6 Kesimpulan dan Saran

Pengambilan kesimpulan dilakukan untuk menjelaskan apa yang telah dicapai dalam penelitian. Kesimpulan merupakan jawaban dari rumusan masalah yang telah didefinisikan sebelumnya. Evaluasi sistem berupa kekurangan dan kesalahan yang terjadi pada sistem juga dijelaskan. Penulisan saran juga dituliskan untuk pengembangan sistem selanjutnya.

BAB 4 PERANCANGAN

1.1 Formulasi Permasalahan

Permasalahan yang akan diselesaikan adalah bagaimana melakukan prediksi terhadap Indeks Harga Konsumen kelompok perumahan, air, listrik, gas, dan bahan bakar. Dalam menyelesaikan permasalahan tersebut dibutuhkan masukan berupa data *time series* Indeks Harga Konsumen kelompok perumahan, air, listrik, gas, dan bahan bakar dimulai dari periode Januari 2011 s.d. Desember 2017 dan beberapa parameter lain yaitu jumlah fitur, jumlah data latih, jumlah *hidden neuron*, jenis fungsi aktivasi, nilai bobot masukan yang didapatkan secara *random* dengan *range* $[-1,1]$ dan nilai bias yang juga didapatkan secara *random* dengan *range* $[0,1]$.

Data dan parameter tersebut kemudian akan diolah menggunakan Inisialisasi Nguyen-Widrow dan *Extreme Learning Machine* yakni dengan melakukan proses pelatihan dan dilanjutkan dengan proses pengujian. Tetapi, sebelum melakukan proses tersebut perlu dilakukan proses normalisasi data pada *range* $[0,1]$. Setelah itu, proses pelatihan dilakukan menggunakan data latih yang akan menghasilkan matriks bobot keluaran yang dibutuhkan pada proses pengujian. Proses pengujian dilakukan menggunakan data uji sampai didapatkan nilai prediksi dan nilai evaluasi dengan RMSE. Tahap terakhir yaitu melakukan denormalisasi hasil prediksi untuk mengembalikan nilai normalisasi menjadi nilai prediksi dalam bentuk aktual.

Adapun sampel data Indeks Harga Konsumen yang digunakan pada penelitian ini adalah data dengan parameter teknikal yang berjumlah 12 data dengan 5 fitur. Sampel data ini ditunjukkan pada Tabel 4.1.

Tabel 4.1 Data Indeks Harga Konsumen Kelompok Perumahan, Air, Gas, dan Bahan Bakar

| Data ke-i | Bln-Th | Fitur 1 | Fitur 2 | Fitur 3 | Fitur 4 | Fitur 5 | Indeks Harga Konsumen |
|-----------|----------------|---------|---------|---------|---------|---------|-----------------------|
| 1 | Januari-2015 | 111,31 | 112,17 | 113,34 | 113,9 | 115,55 | 116,48 |
| 2 | Februari-2015 | 112,17 | 113,34 | 113,9 | 115,55 | 116,48 | 116,96 |
| 3 | Maret-2015 | 113,34 | 113,9 | 115,55 | 116,48 | 116,96 | 117,3 |
| 4 | April-2015 | 113,9 | 115,55 | 116,48 | 116,96 | 117,3 | 117,56 |
| 5 | Mei-2015 | 115,55 | 116,48 | 116,96 | 117,3 | 117,56 | 117,8 |
| 6 | Juni-2015 | 116,48 | 116,96 | 117,3 | 117,56 | 117,8 | 118,07 |
| 7 | Juli-2015 | 116,96 | 117,3 | 117,56 | 117,8 | 118,07 | 118,22 |
| 8 | Agustus-2015 | 117,3 | 117,56 | 117,8 | 118,07 | 118,22 | 118,41 |
| 9 | September-2015 | 117,56 | 117,8 | 118,07 | 118,22 | 118,41 | 118,65 |
| 10 | Oktober-2015 | 117,8 | 118,07 | 118,22 | 118,41 | 118,65 | 118,76 |
| 11 | November-2015 | 118,07 | 118,22 | 118,41 | 118,65 | 118,76 | 118,94 |

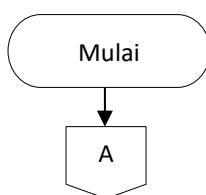
Tabel 4.1 Data Indeks Harga Konsumen Kelompok Perumahan, Air, Gas, dan Bahan Bakar (lanjutan)

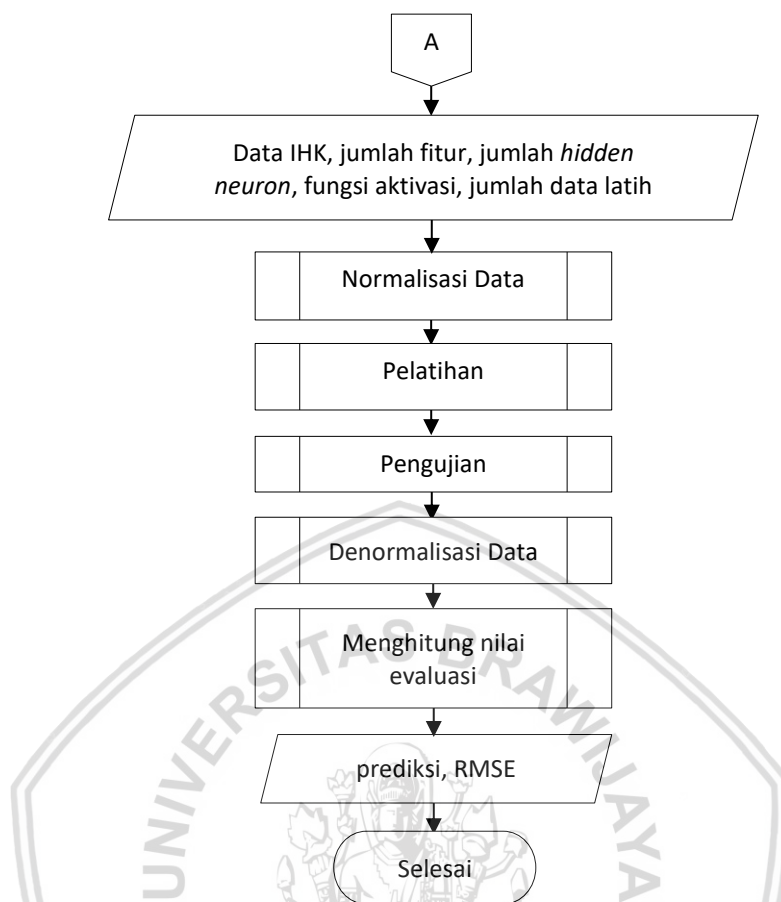
| Data ke-i | Bln-Th | Fitur 1 | Fitur 2 | Fitur 3 | Fitur 4 | Fitur 5 | Indeks Harga Konsumen |
|-----------|---------------|---------|---------|---------|---------|---------|-----------------------|
| 12 | Desember-2015 | 118,22 | 118,41 | 118,65 | 118,76 | 118,94 | 119,41 |

Nilai pada fitur 5 didapatkan dari nilai index 1 bulan sebelumnya, sedangkan nilai pada fitur 4 didapatkan dari nilai index 2 bulan sebelumnya, lalu nilai pada fitur 3 didapatkan dari nilai index 3 bulan sebelumnya, nilai pada fitur 4 didapatkan dari nilai index 2 bulan sebelumnya, dan nilai pada fitur 1 didapatkan dari nilai index 5 bulan sebelumnya. Keterangan setiap parameter data pada Tabel 4.1 adalah sebagai berikut.

1. Bln-Th : Menerangkan bulan dan tahun dari Indeks Harga Konsumen (date)
2. Fitur 1 : Menerangkan fitur pertama yang merupakan nilai Indeks Harga Konsumen 5 bulan sebelum bulan dan tahun pada parameter pertama (double)
3. Fitur 2 : Menerangkan fitur kedua yang merupakan nilai Indeks Harga Konsumen 4 bulan sebelum bulan dan tahun pada parameter pertama (double)
4. Fitur 3 : Menerangkan fitur ketiga yang merupakan nilai Indeks Harga Konsumen 3 bulan sebelum bulan dan tahun pada parameter pertama (double)
5. Fitur 4 : Menerangkan fitur keempat yang merupakan nilai Indeks Harga Konsumen 2 bulan sebelum bulan dan tahun pada parameter pertama (double)
6. Fitur 5 : Menerangkan fitur kelima yang merupakan nilai Indeks Harga Konsumen 1 bulan sebelum bulan dan tahun pada parameter pertama (double)
7. Nilai Indeks : Menerangkan nilai Indeks Harga Konsumen dari bulan dan tahun pada parameter pertama (double)

1.2 Tahapan *Extreme Learning Machine*





Gambar 4.1 Diagram Alir *Extreme Learning Machine*

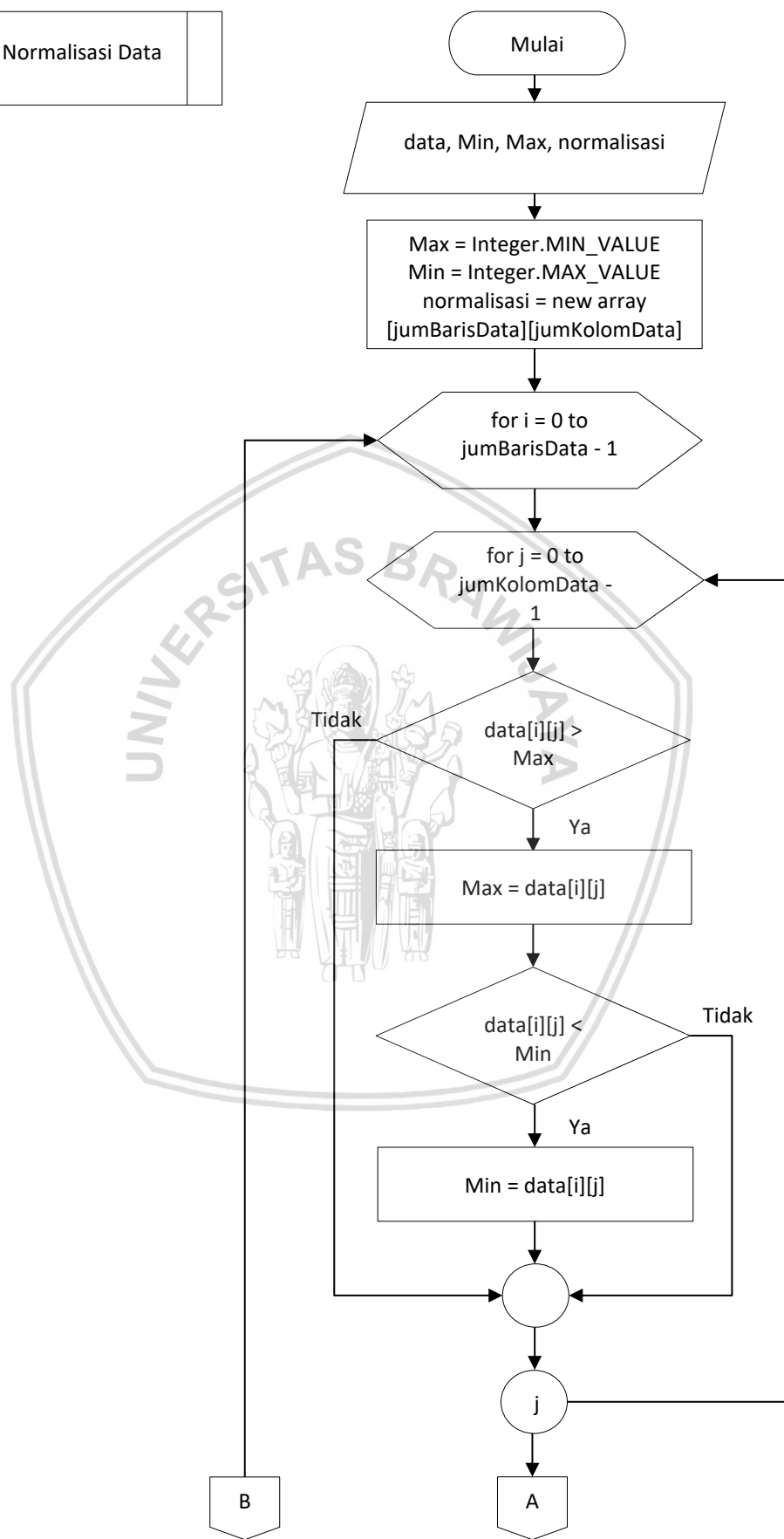
Berdasarkan Gambar 4.1 di atas, langkah-langkah proses pada *Extreme Learning Machine* adalah sebagai berikut.

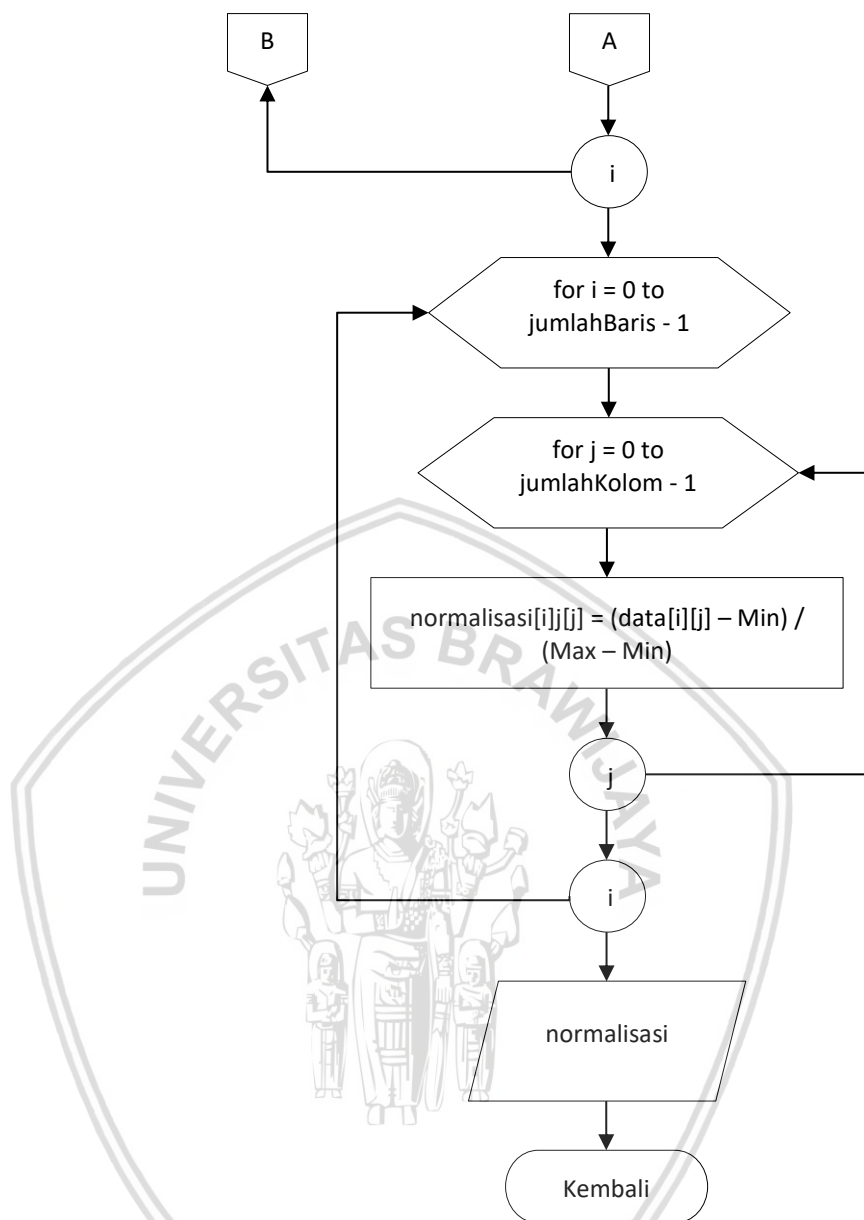
1. Masukan pada proses ELM adalah data IHK, jumlah fitur, jumlah *hidden neuron*, fungsi aktivasi, dan jumlah data latih yang digunakan.
2. Melakukan proses normalisasi pada data Indeks Harga Konsumen menggunakan Persamaan 2.1.
3. Melakukan proses pelatihan dengan menggunakan data latih yang jumlahnya sudah ditentukan untuk mendapatkan bobot keluaran ($\hat{\beta}$).
4. Melakukan proses pengujian dengan menggunakan data uji untuk mendapatkan nilai prediksi.
5. Melakukan denormalisasi data dari nilai prediksi dan menghitung nilai evaluasi menggunakan RMSE.
6. Keluaran pada proses ini adalah hasil prediksi dan nilai RMSE.

1.3 Normalisasi Data

Normalisasi data dilakukan untuk memetakan data ke dalam *range-range* tertentu. Pada penelitian ini data akan dipetakan ke dalam *range* [0,1]. Diagram alir dari proses normalisasi data ditunjukkan pada Gambar 4.2.

| |
|------------------|
| Normalisasi Data |
|------------------|





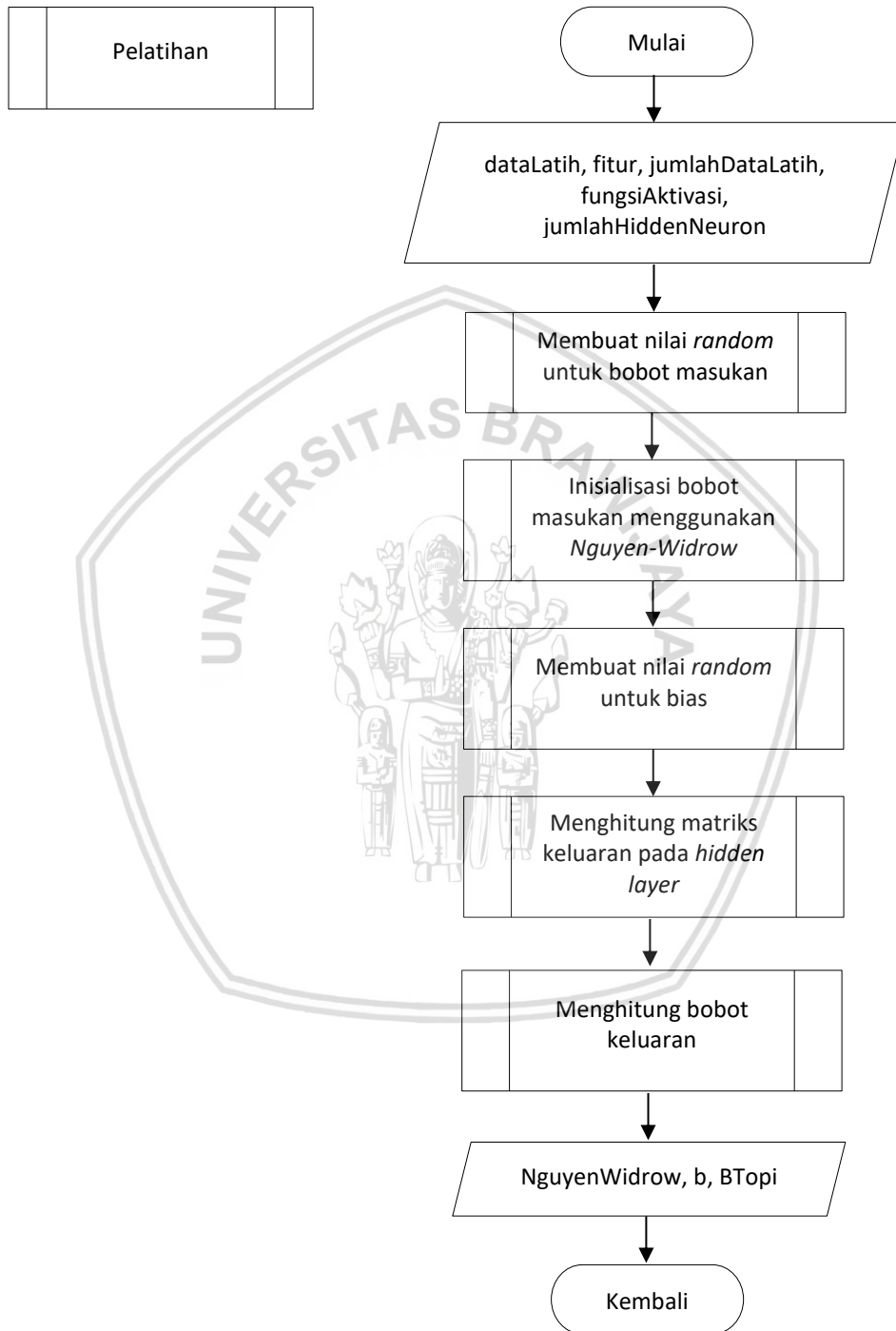
Gambar 4.2 Diagram Alir Proses Normalisasi

Berdasarkan Gambar 4.2 di atas, langkah-langkah pada proses normalisasi adalah sebagai berikut.

1. Masukan pada proses normalisasi adalah data Indeks Harga Konsumen (*data*), serta deklarasi variabel *Min*, *Max*, dan *normalisasi* untuk menyimpan data hasil normalisasi.
2. Melakukan perulangan untuk mencari nilai maksimal dan nilai minimal dari data.
3. Melakukan perhitungan normalisasi dengan Persamaan 2.1.
4. Nilai keluaran berupa data yang sudah dinormalisasi dengan *range* [0,1].

1.4 Pelatihan

Pelatihan dilakukan untuk mendapatkan matriks bobot keluaran untuk selanjutnya diolah di proses pengujian. Diagram alir dari proses pelatihan ditunjukkan pada Gambar 4.3.



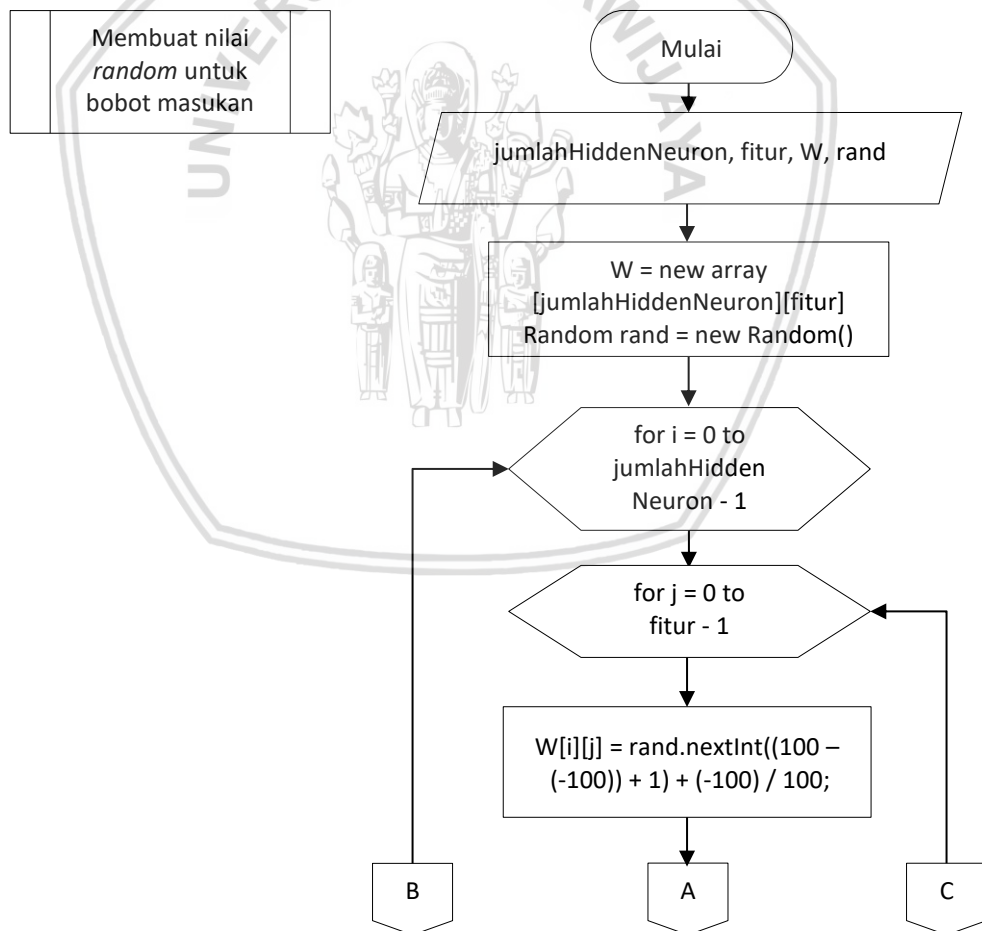
Gambar 4.3 Diagram Alir Proses Pelatihan

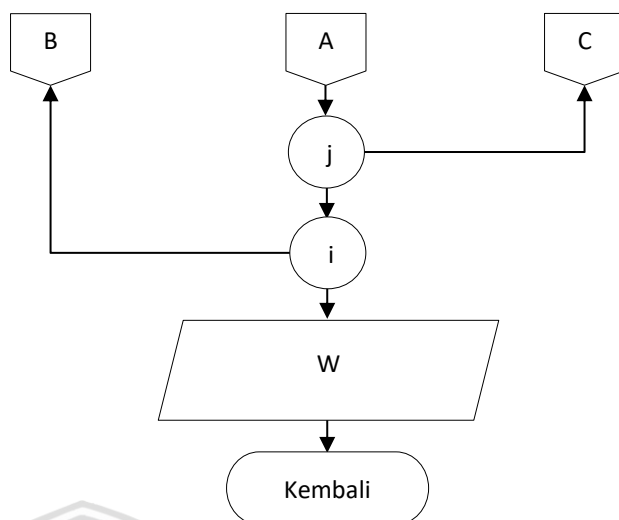
Berdasarkan Gambar 4.3 di atas, langkah-langkah pada proses pelatihan adalah sebagai berikut.



1. Masukan berupa *dataLatih*, data latih yang telah di normalisasi, jumlah fitur, jumlah data latih, fungsi aktivasi, dan jumlah *hidden neuron*.
2. Membuat matriks bobot masukan ($W_{m \times n}$) secara random dengan *range* [-1,1], *m* adalah jumlah *hidden neuron* dan *n* adalah jumlah fitur.
3. Inisialisasi bobot masukan ($W_{m \times n}$) yang sebelumnya telah ditentukan secara *random* menggunakan inisialisasi *Nguyen-Widrow*.
4. Selanjutnya membuat matriks bias ($b_{1 \times m}$) secara random dengan *range* [0,1], *m* adalah jumlah *hidden neuron*.
5. Menghitung nilai matriks keluaran pada *hidden layer* dengan menggunakan Persamaan 2.11.
6. Menghitung bobot keluaran (BT_{opi}) dengan Persamaan 2.13.
7. Keluaran dari proses pelatihan adalah matriks bobot masukan (*NguyenWidrow*), matriks bias (*b*), bobot keluaran (BT_{opi}).

1.4.1 Membuat Nilai *Random* untuk Bobot Masukan



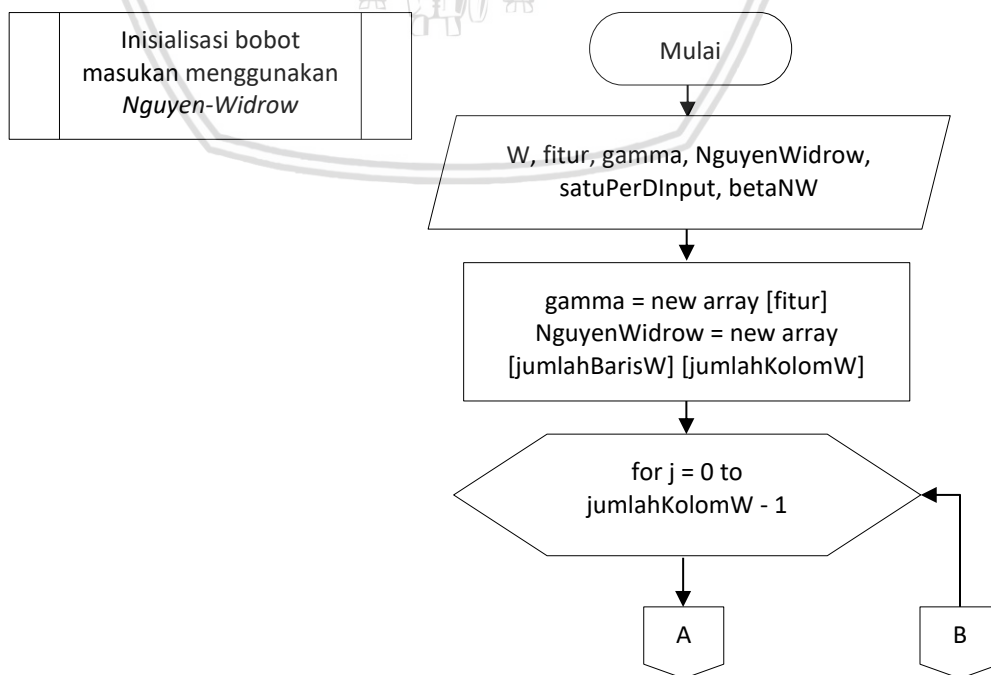


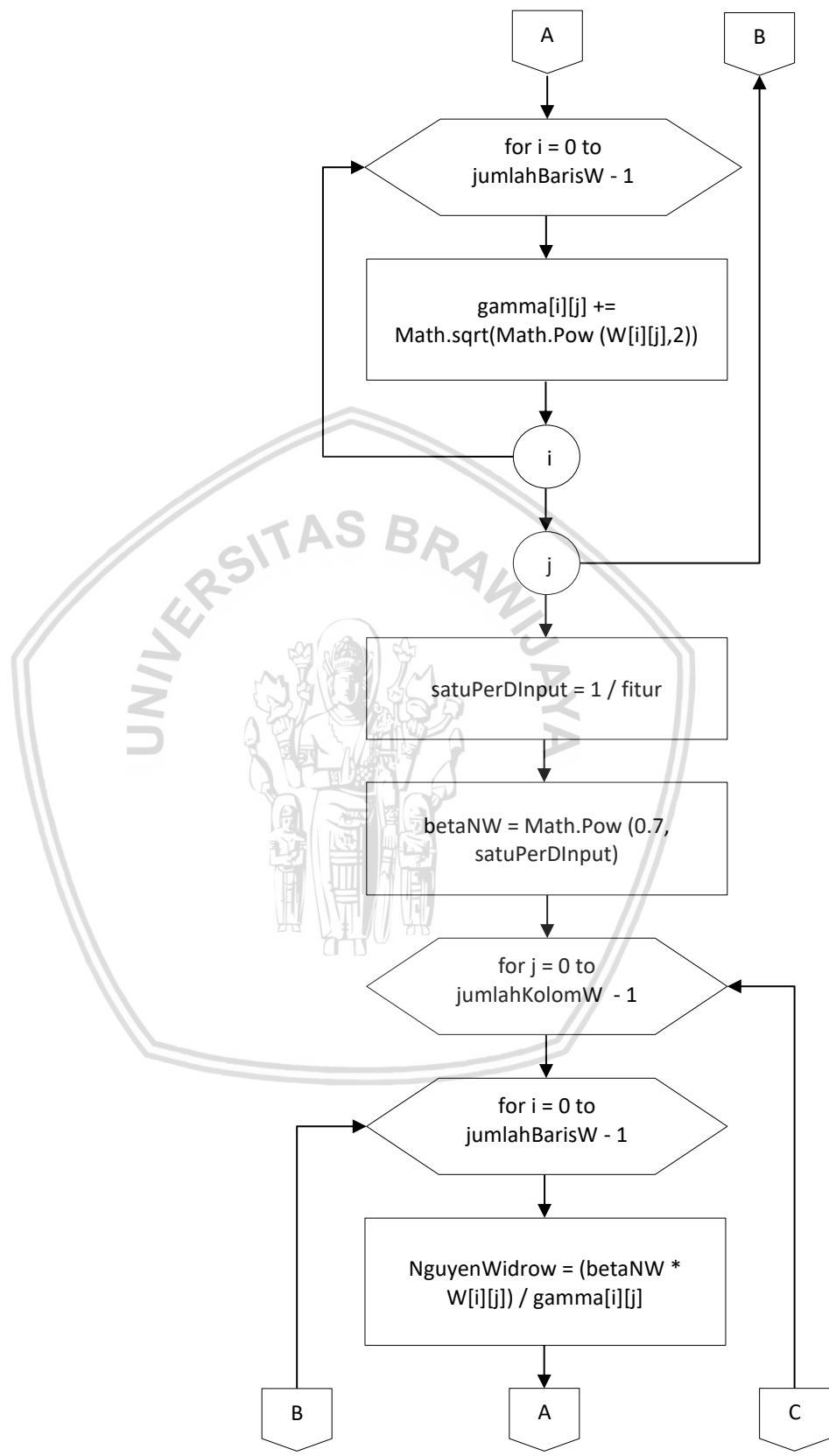
Gambar 4.4 Diagram Alir Proses Membuat Nilai Random untuk Bobot Masukan

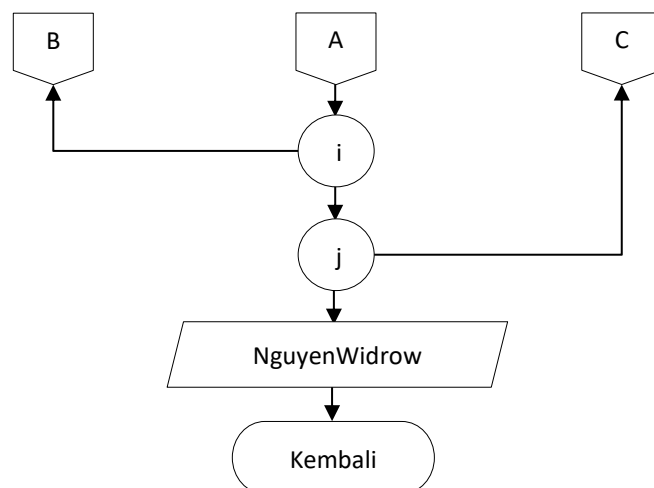
Berdasarkan Gambar 4.4 di atas, langkah-langkah pada proses membuat nilai random untuk bobot masukan adalah sebagai berikut.

1. Masukan pada proses ini adalah jumlah *hidden neuron* dan jumlah fitur serta deklarasi *array* bobot masukan (W) dan *rand*.
2. Melakukan inisialisasi *array* ($W_{m \times n}$), m sama dengan jumlah *hidden neuron* dan n sama dengan jumlah fitur
3. Melakukan perulangan untuk mendapatkan nilai *random* di seluruh indeks matriks bobot masukan (W) dengan range $[-1,1]$.
4. Keluaran pada proses ini adalah matriks bobot masukan (W).

1.4.2 Inisialisasi Bobot Masukan Menggunakan *Nguyen-Widrow*





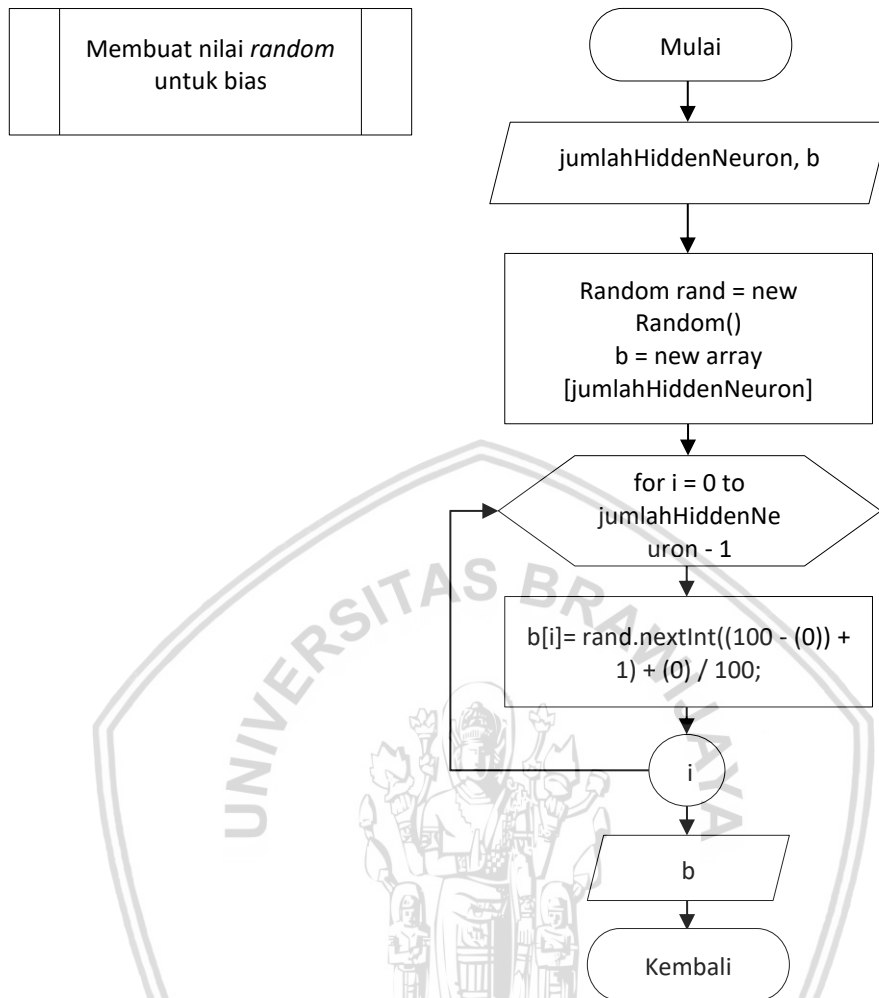


Gambar 4.5 Diagram Alir Proses Inisialisasi Bobot Masukan Menggunakan Nguyen-Widrow

Berdasarkan Gambar 4.5 di atas, langkah-langkah pada proses inisialisasi bobot masukan menggunakan *Nguyen-Widrow* adalah sebagai berikut.

1. Masukan pada proses ini adalah matriks bobot masukan yang telah didapatkan secara *random* (W), dan jumlah fitur (*fitur*), serta deklarasi variabel γ , *NguyenWidrow*, *satuPerDInput*, β_{NW} .
2. Inisialisasi dan deklarasi *array* satu dimensi dengan nama γ yang memiliki jumlah baris sama dengan jumlah fitur.
3. Inisialisasi *array* ($NguyenWidrow_{i \times j}$), i sama dengan jumlah baris *array* W , dan j sama dengan jumlah kolom *array* W .
4. Menghitung γ yakni penjumlahan dari nilai kuadrat setiap bobot random (W) dalam satu kolom yang sama, kemudian hasil dari penjumlahan tersebut di akar kuadrat.
5. Menghitung nilai *satuPerDInput* yakni 1 dibagi dengan jumlah fitur. Hasil perhitungan ini akan digunakan untuk menghitung β_{NW} .
6. Menghitung nilai β_{NW} yakni 0.7 di akar pangkatkan dengan hasil perhitungan variabel *satuPerDInput*.
7. Menghitung nilai inisialisasi bobot masukan menggunakan rumus dari inisialisasi *Nguyen-Widrow*.
8. Keluaran pada proses ini berupa nilai inisialisasi bobot masukan menggunakan inisialisasi *Nguyen-Widrow* (*NguyenWidrow*).

1.4.3 Membuat Nilai *Random* untuk Bias



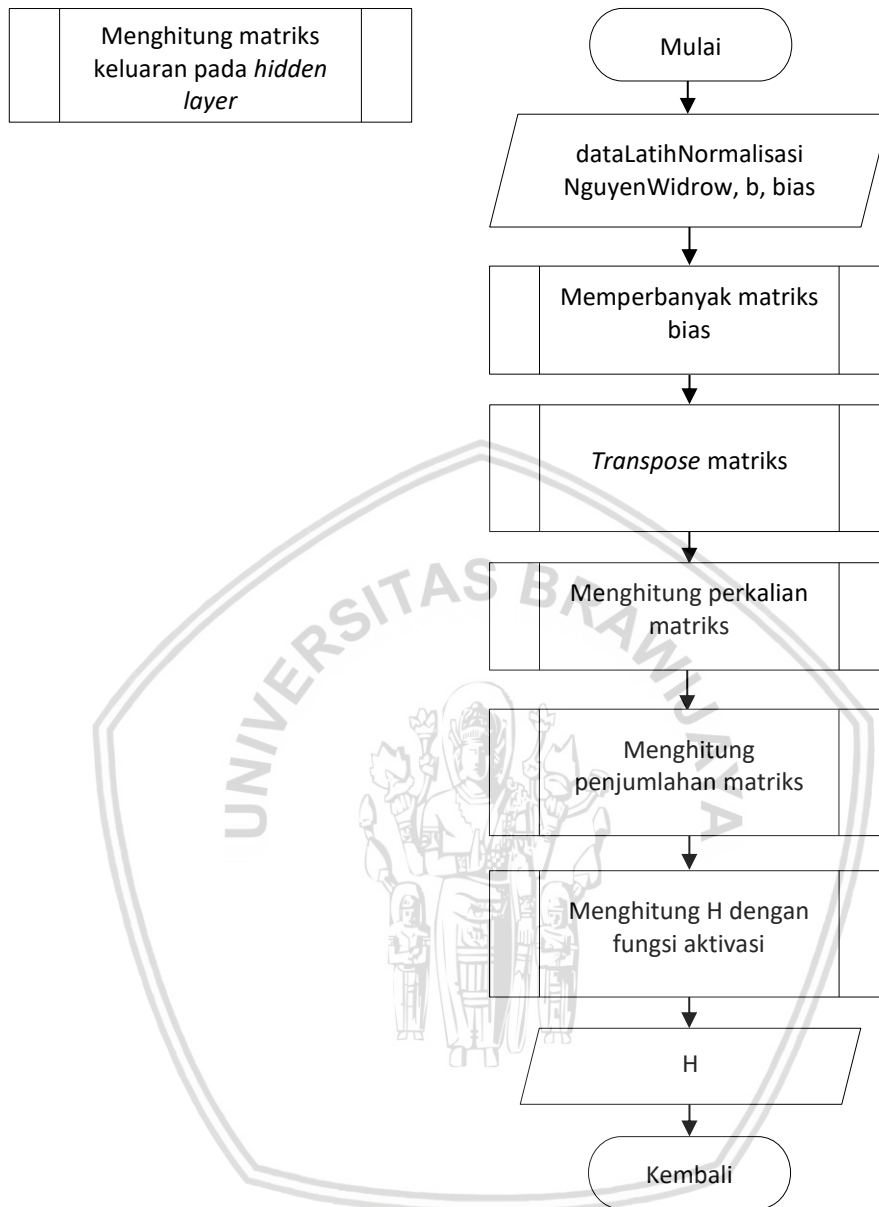
Gambar 4.6 Diagram Alir Proses Membuat Nilai *Random* untuk Bias

Berdasarkan Gambar 4.6 di atas, langkah-langkah pada proses membuat nilai *random* untuk bias adalah sebagai berikut.

1. Masukan pada proses ini adalah jumlah dari *hidden neuron* dan deklarasi *array* bias satu dimensi (*b*).
2. Melakukan inisialisasi *array* bias satu dimensi dengan nama *b* yang memiliki panjang kolom sama dengan jumlah *hidden neuron* (*jumlahHiddenNeuron*).
3. Selanjutnya adalah melakukan perulangan sebanyak jumlah *jumlahHiddenNeuron*.
4. Melakukan proses *random* dengan *range* [0,1], kemudian disimpan di dalam *array b*.
5. Keluaran dari proses ini adalah *array b*.



1.4.4 Menghitung Matriks Keluaran pada *Hidden Layer*



Gambar 4.7 Diagram Alir Proses Menghitung Keluaran pada *Hidden Layer*

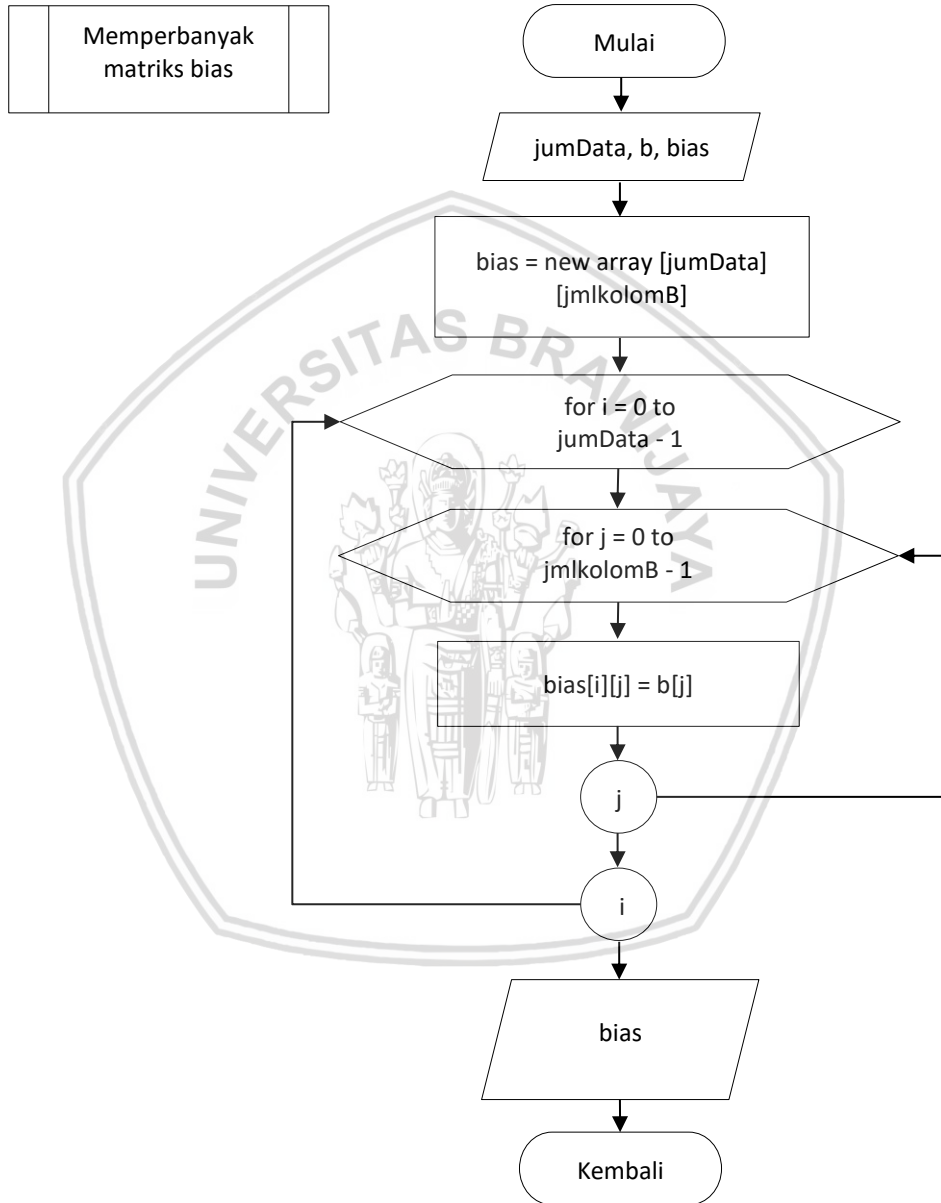
Berdasarkan Gambar 4.7 di atas, langkah-langkah pada proses menghitung keluaran pada *hidden layer* adalah sebagai berikut.

1. Masukan pada proses ini adalah data latih normalisasi (*dataLatihNormalisasi*), matriks bobot masukan (*NguyenWidrow*), dan matriks bias (*b*), seta deklarasi matriks bias dua dimensi (*bias*).
2. Selanjutnya memperbanyak matriks bias (*b*) sejumlah data latih.
3. Melakukan *transpose* matriks bobot masukan (*NguyenWidrow*).
4. Menghitung perkalian matriks antara data latih yang telah dinormalisasi (*dataLatihNormalisasi*) dengan matriks bobot masukan (*NguyenWidrow*) yang telah ditranspose.



5. Menghitung H_{init} yaitu dengan melakukan penjumlahan antara matriks hasil perkalian $dataLatihNormalisasi$ dan bobot masukan ($NguyenWidrow$) dengan matriks bias ($bias$).
6. Menghitung H yaitu menghitung H_{init} menggunakan fungsi aktivasi.
7. Keluaran pada proses ini adalah matriks keluaran pada *hidden layer* (H).

1.4.4.1 Memperbanyak Matriks Bias



Gambar 4.8 Diagram Alir Memperbanyak Matriks Bias

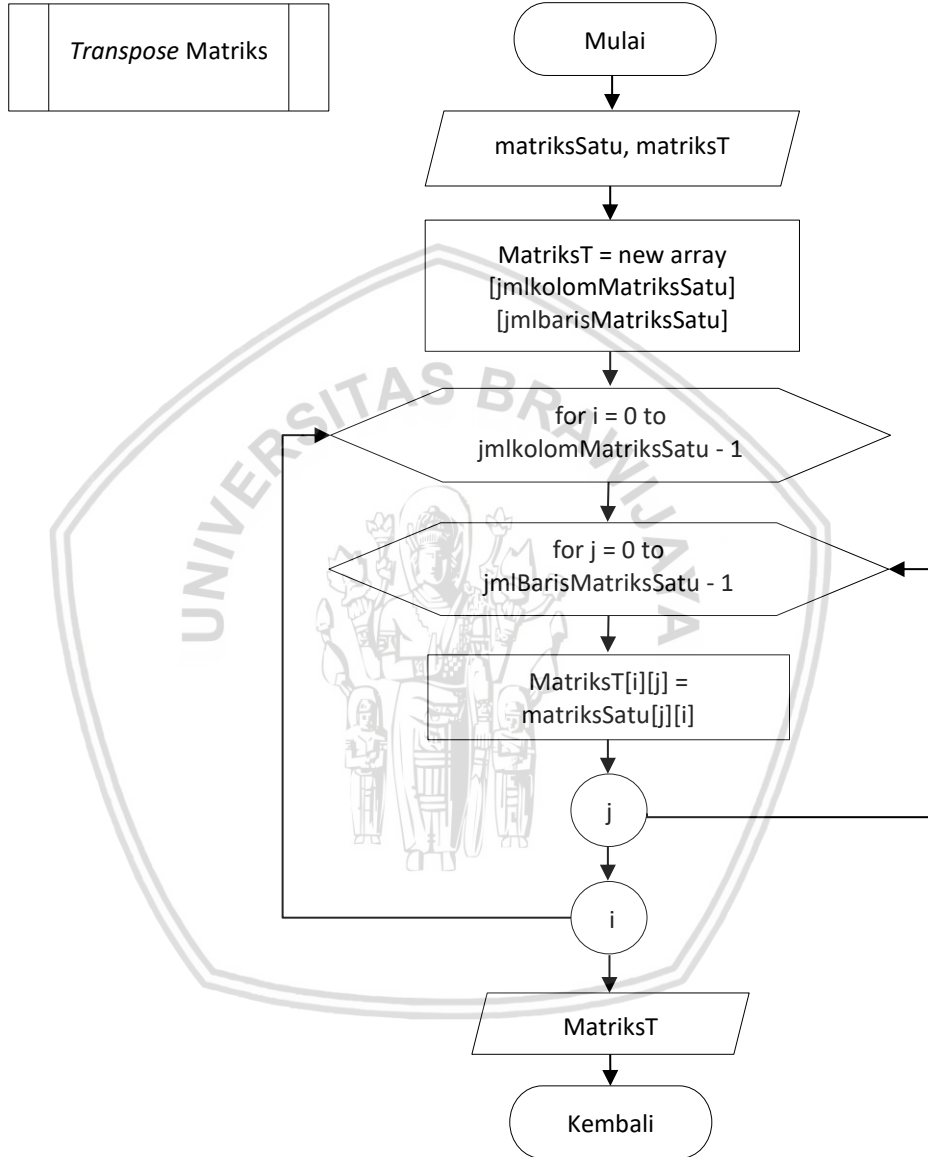
Berdasarkan Gambar 4.8, langkah-langkah pada proses memperbanyak matriks bias adalah sebagai berikut.

1. Masukan pada proses ini adalah jumlah data ($jumData$), matriks bias satu dimensi (b), dan deklarasi matriks bias dua dimensi ($bias$).



2. Membuat matriks bias baru ($bias_{i \times j}$), i sama dengan $jumData$ dan j sama dengan jumlah kolom matriks bias (b).
3. Melakukan duplikasi matriks bias (b) sejumlah $jumData$.
4. Keluaran pada proses ini adalah matriks bias baru ($bias$).

1.4.4.2 Transpose Matriks



Gambar 4.9 Diagram Alir Proses *Transpose* Matriks

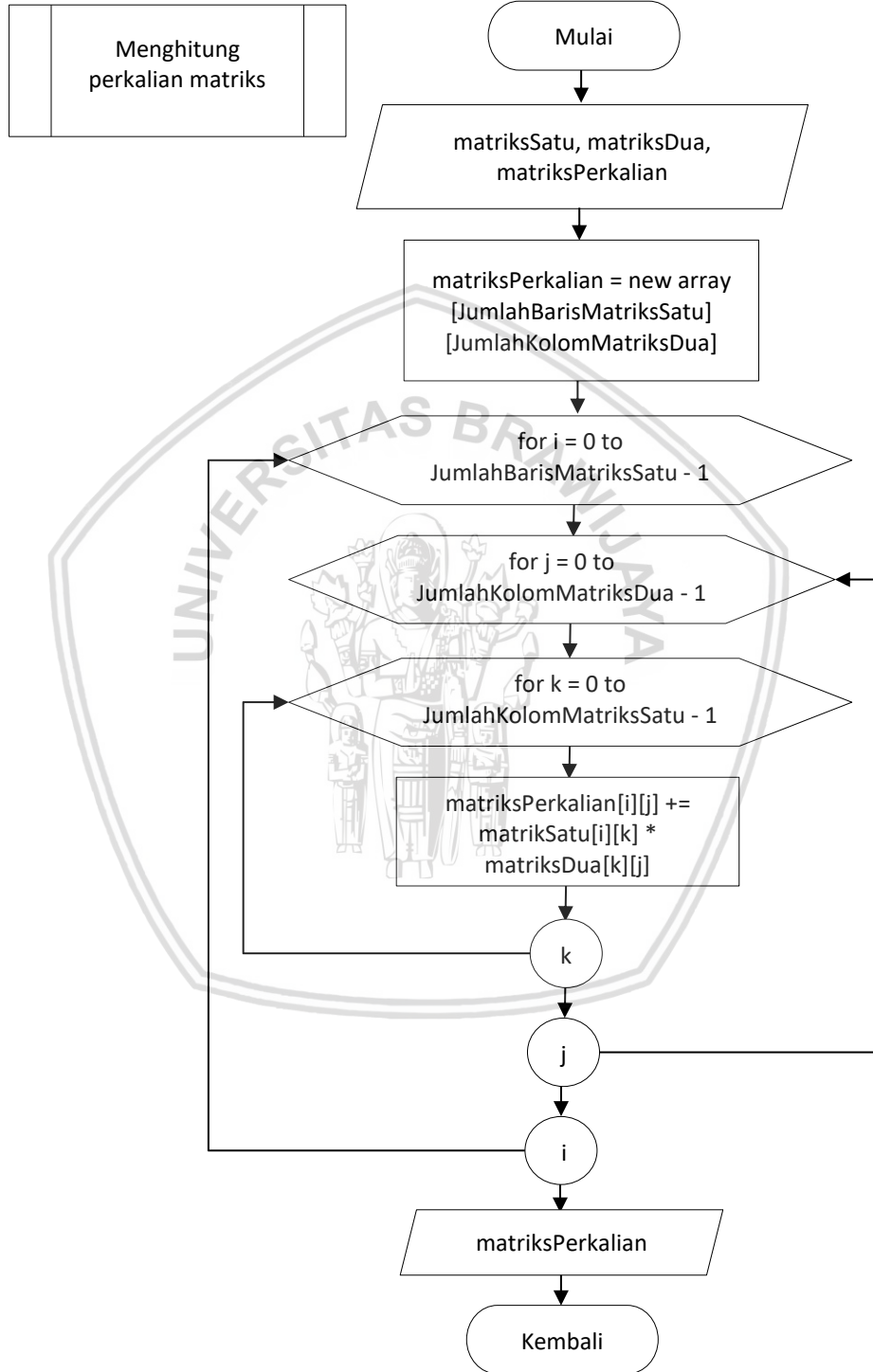
Berdasarkan Gambar 4.9 langkah-langkah pada proses *transpose* matriks adalah sebagai berikut.

1. Masukan pada proses ini adalah matriks *matriksSatu* dan deklarasi *array matriksT*.
2. Membuat matriks ($matriksT_{i \times j}$), i sama dengan jumlah kolom *matriksSatu* dan j sama dengan jumlah baris *matriksSatu*.



3. Melakukan proses *transpose array matriksSatu* dan disimpan di dalam *array matriksT*.
4. Keluaran dari proses ini adalah matriks *transpose* dengan nama *matriksT*.

1.4.4.3 Menghitung Perkalian Matriks



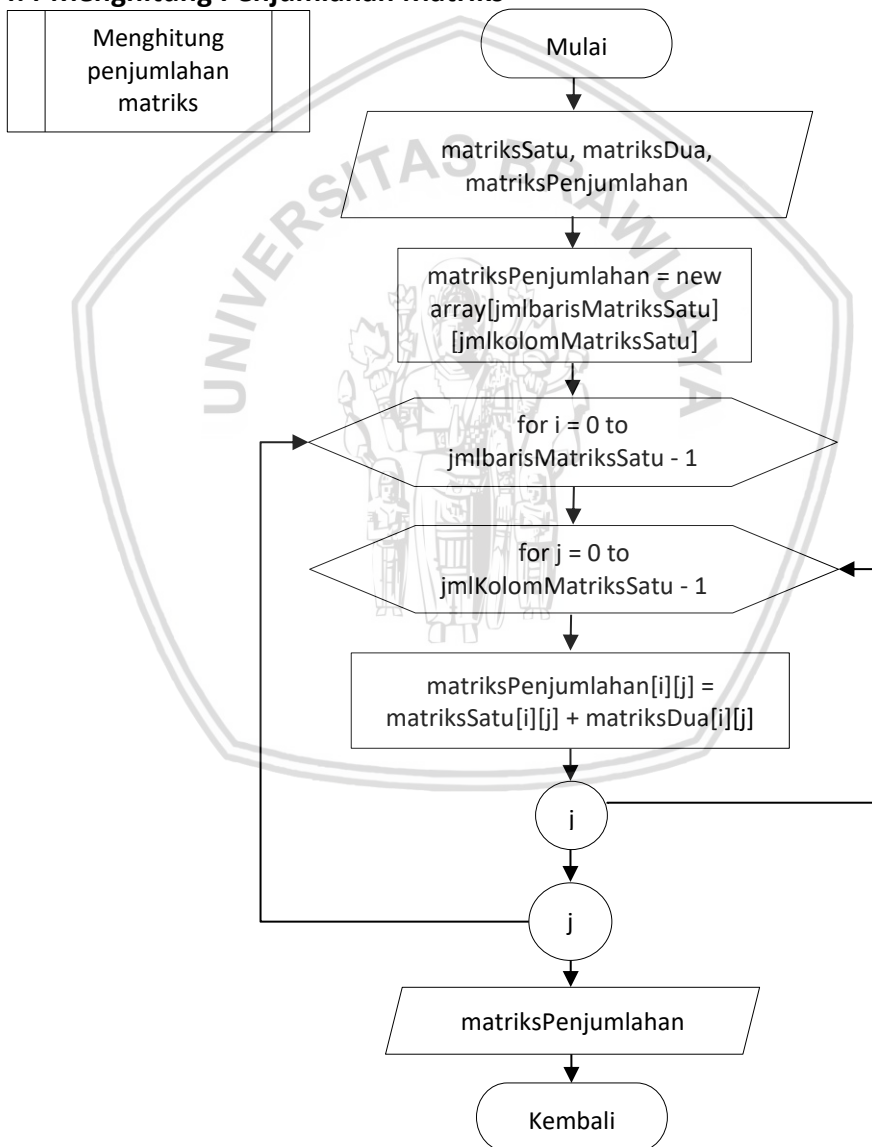
Gambar 4.10 Diagram Alir Proses Menghitung Perkalian Matriks



Berdasarkan Gambar 4.10, langkah-langkah pada proses menghitung perkalian matriks adalah sebagai berikut.

1. Masukan pada proses ini adalah *matriksSatu* dan *matriksDua*, dan deklarasi *array matriksPerkalian*.
2. Membuat matriks baru (*matriksPerkalian_{i×j}*), *i* sama dengan jumlah baris matriks *matriksSatu* dan *j* sama dengan jumlah kolom matriks *matriksDua*.
3. Melakukan proses perkalian antara *matriksSatu* dengan *matriksDua*.
4. Keluaran pada proses ini adalah *matriksPerkalian*, yaitu hasil perkalian antara *matriksSatu* dengan *matriksDua*.

1.4.4.4 Menghitung Penjumlahan Matriks



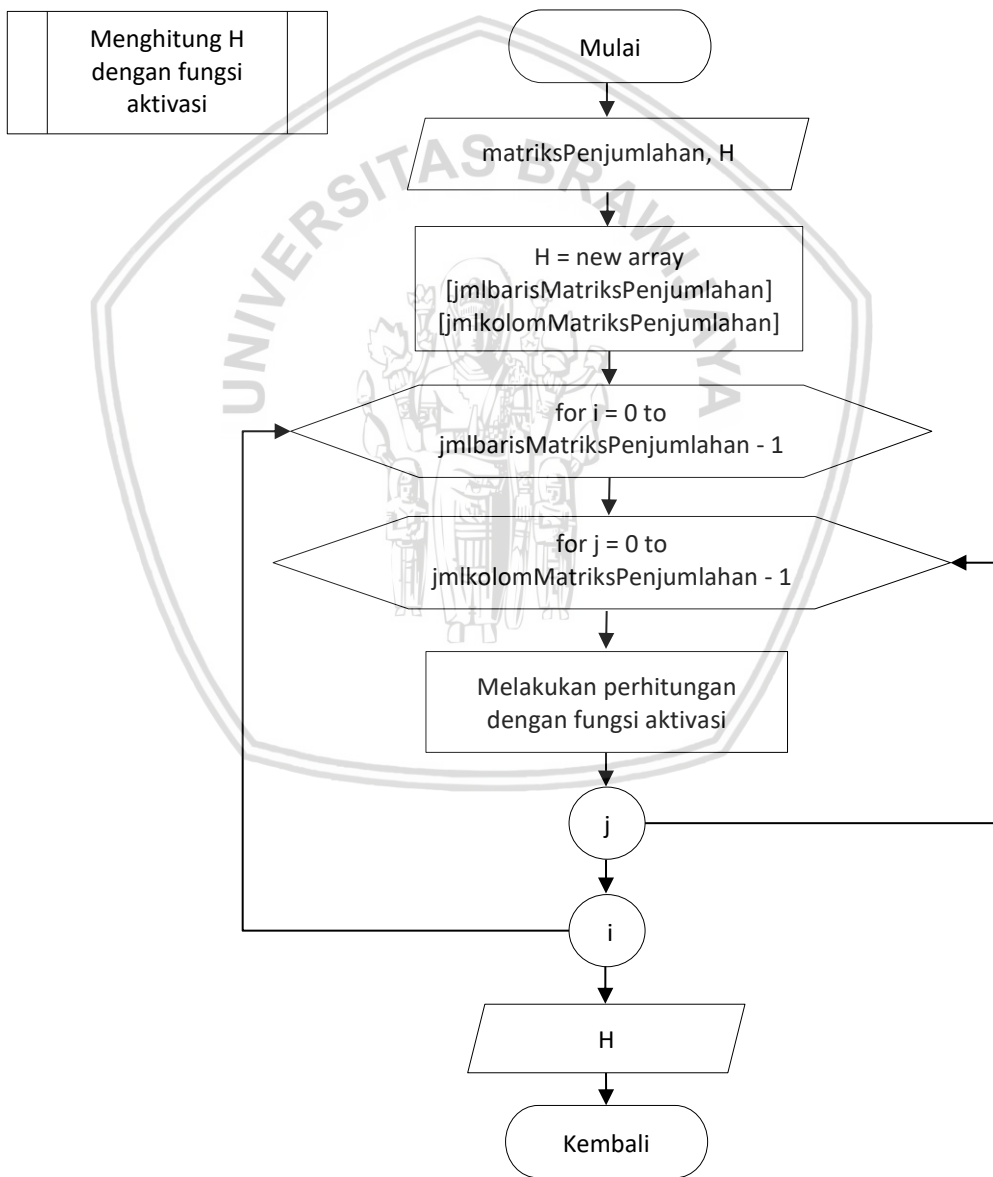
Gambar 4.11 Diagram Alir Proses Menghitung Penjumlahan Matriks

Berdasarkan Gambar 4.11, langkah-langkah pada proses menghitung penjumlahan matriks adalah sebagai berikut.



1. Masukan pada proses ini adalah *matriksSatu* dan *matriksDua*, serta deklarasi *array matriksPenjumlahan*.
2. Membuat matriks baru (*matriksPenjumlahan_{i×j}*), *i* sama dengan jumlah baris matriks *matriksSatu* dan *j* sama dengan jumlah kolom matriks *matriksSatu*.
3. Melakukan proses penjumlahan antara matriks *matriksSatu* dan matriks *matriksDua*.
4. Keluaran dari proses ini adalah matriks *matriksPenjumlahan*, yaitu hasil perjumlahan antara *matriksSatu* dengan matriks *matriksDua*.

1.4.4.5 Menghitung H dengan Fungsi Aktivasi



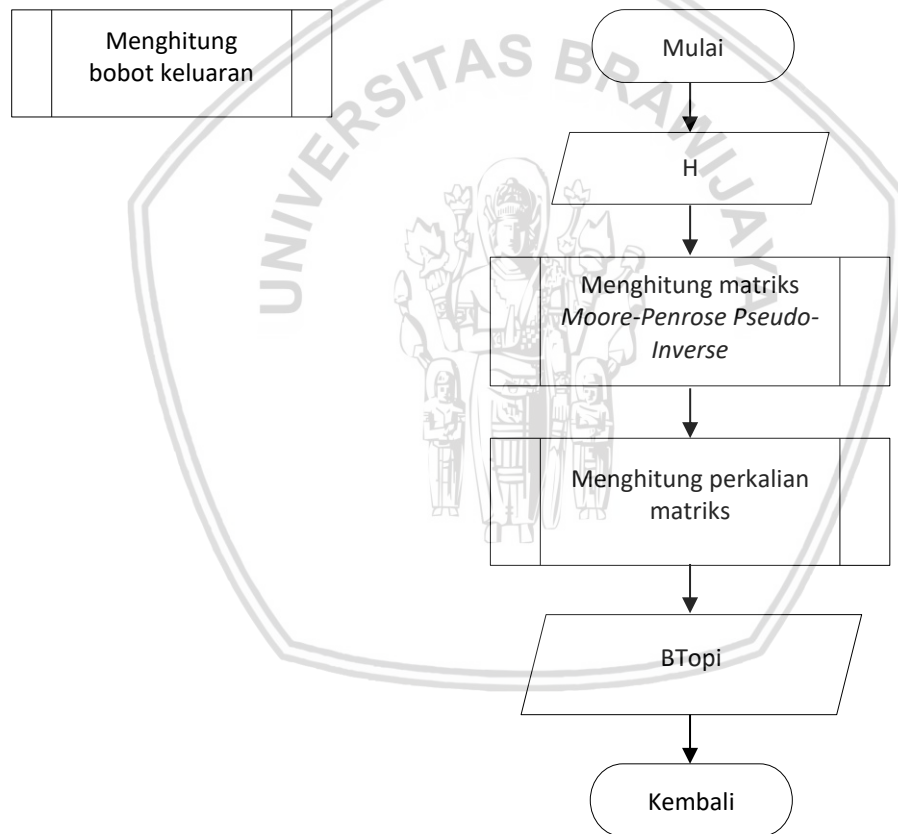
Gambar 4.12 Diagram Alir Proses Menghitung H dengan Fungsi Aktivasi



Berdasarkan Gambar 4.12, langkah-langkah pada proses menghitung H dengan fungsi aktivasi adalah sebagai berikut.

1. Masukan pada proses ini adalah *matriksPenjumlahan* yaitu matriks *Hinit*, dan deklarasi matriks *H*.
2. Membuat matriks baru $(H_{i \times j})$, *i* sama dengan jumlah baris *matriksPenjumlahan* dan *j* sama dengan jumlah kolom *matriksPenjumlahan*.
3. Menghitung fungsi aktivasi dengan fungsi Sin menggunakan Persamaan 2.3, Sigmoid Biner menggunakan Persamaan 2.4, Tringular Basis menggunakan Persamaan 2.5, Radial Basis menggunakan Persamaan 2.6.
4. Keluaran pada proses ini adalah *H* yaitu matriks keluaran *hidden layer*.

1.4.5 Menghitung Bobot Keluaran



Gambar 4.13 Diagram Alir Proses Menghitung Bobot Keluaran

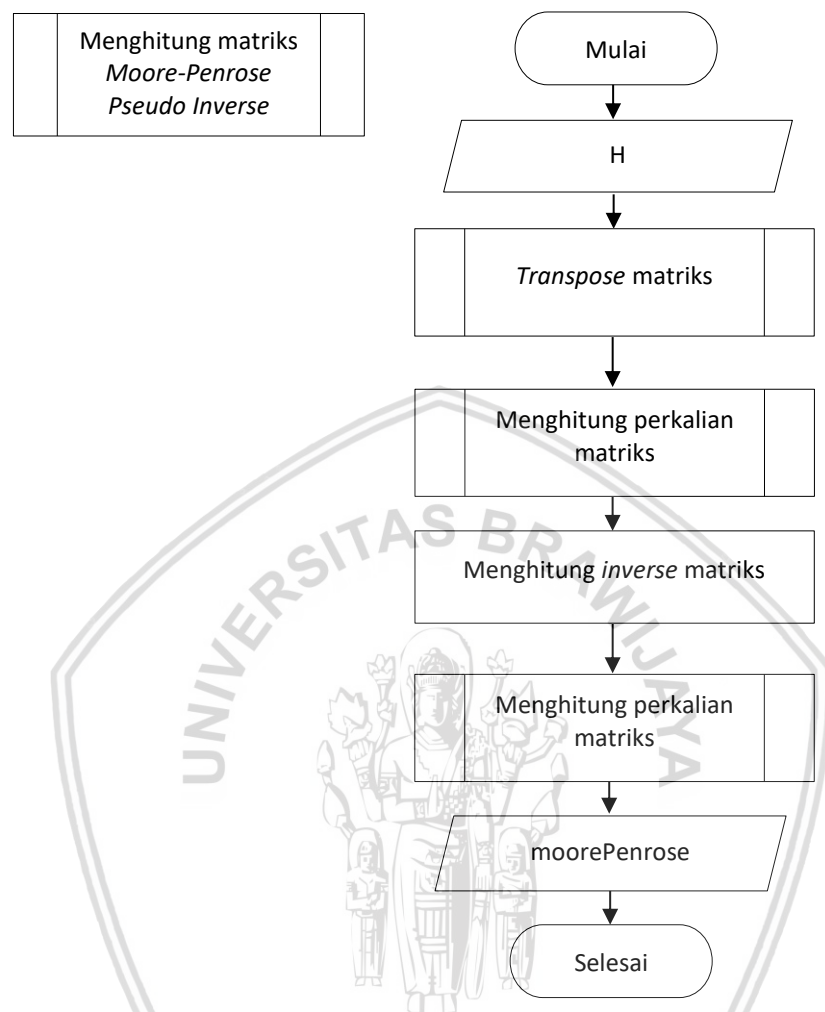
Berdasarkan Gambar 4.13, langkah-langkah pada proses menghitung bobot keluaran adalah sebagai berikut.

1. Masukan pada proses ini adalah matriks *H*.
2. Menghitung matriks *Moore-Penrose Pseudo Inverse*.
3. Menghitung perkalian matriks untuk menghitung nilai bobot keluaran (*BTopi*).



4. Keluaran pada proses ini adalah BT_{opi} atau bobot keluaran ($\hat{\beta}$).

1.4.5.1 Menghitung Matriks Moore-Penrose Pseudo Inverse

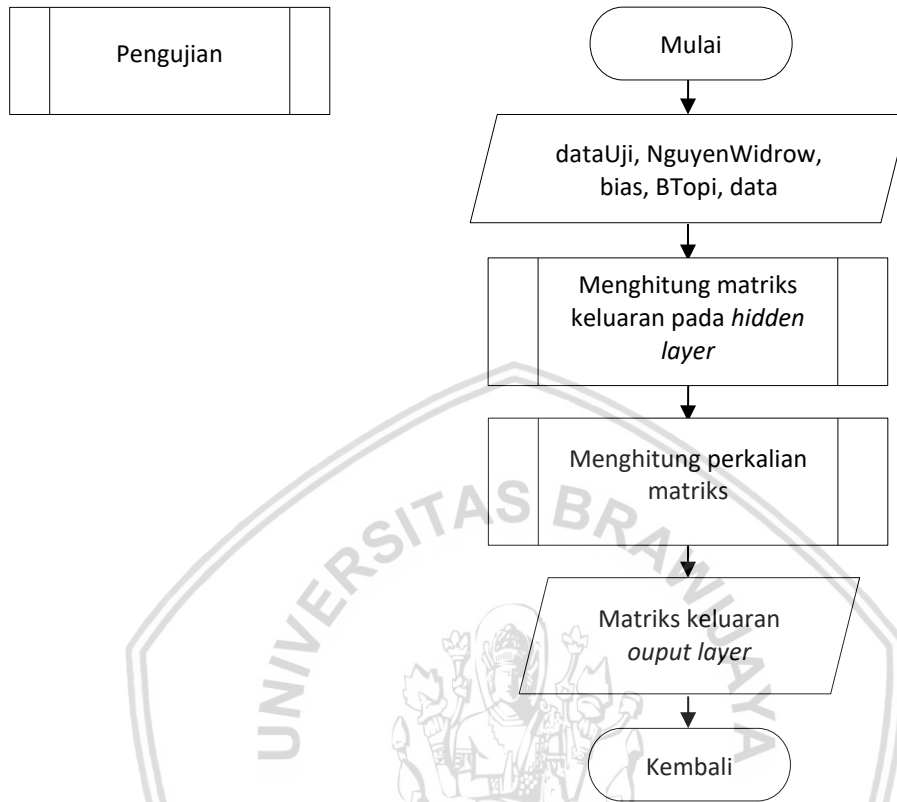


Gambar 4.14 Diagram Alir Proses Menghitung Matriks Moore-Penrose Pseudo Inverse

Berdasarkan Gambar 4.14, langkah-langkah pada proses menghitung matriks moore-penrose *pseudo inverse* adalah sebagai berikut.

1. Masukan pada proses ini adalah matriks H .
2. Melakukan *transpose* matriks H .
3. Melakukan perkalian matriks antara matriks H *transpose* dengan matriks H .
4. Menghitung *inverse* matriks dari hasil perkalian matriks pada langkah nomor 3.
5. Menghitung perkalian matriks antara hasil dari *inverse* matriks pada langkah nomor 4 dan hasil dari *transpose* matriks pada langkah nomor 2.
6. Keluaran dari proses ini adalah matriks *moorePenrose* yaitu hasil dari proses perhitungan *Moore-Penrose Pseudo Inverse*.

1.5 Pengujian



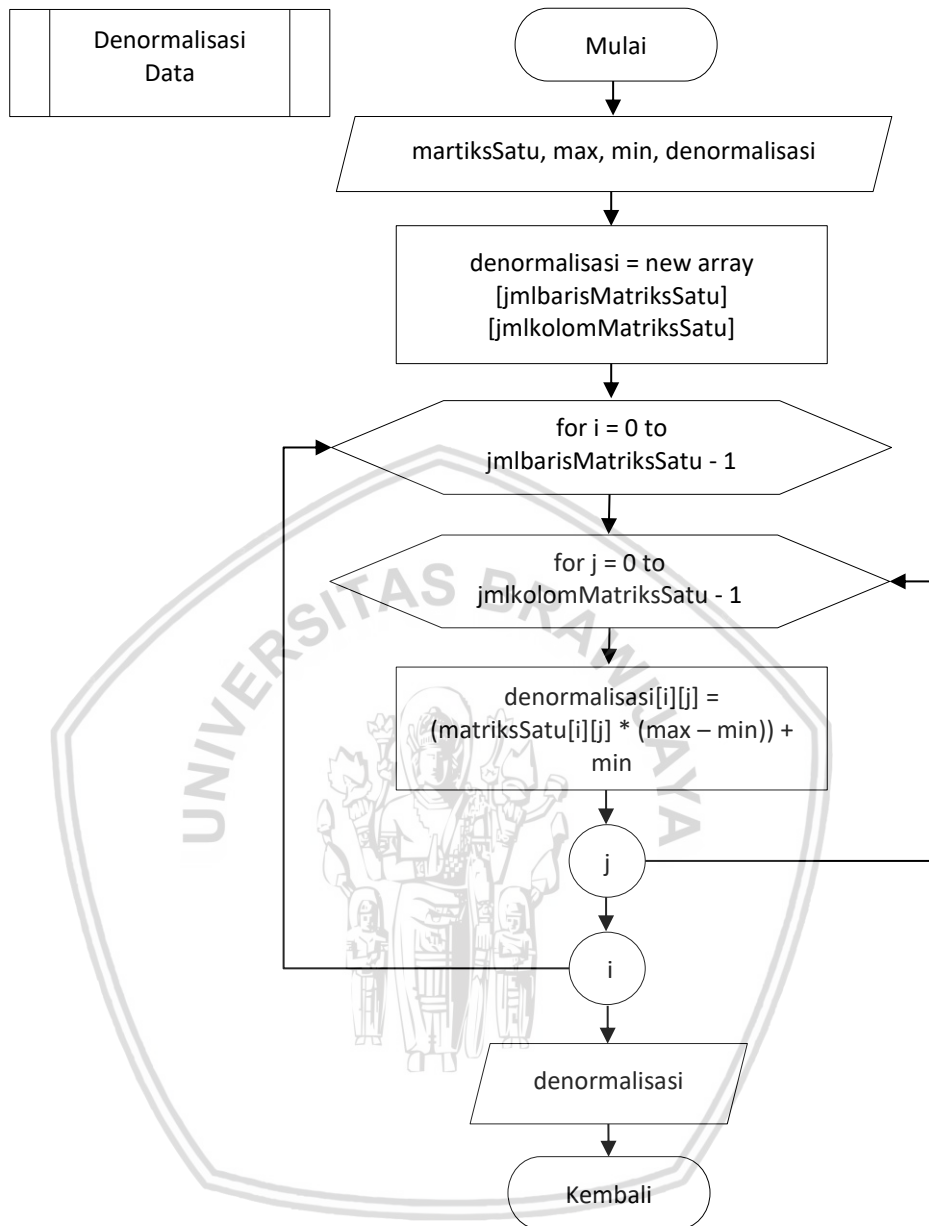
Gambar 4.15 Diagram Alir Proses Pengujian

Berdasarkan Gambar 4.15, langkah-langkah pada proses pengujian adalah sebagai berikut.

1. Masukan pada proses ini adalah data uji, matriks bobot *Nguyen-Widrow* (*NguyenWidrow*), matriks bias (*bias*), bobot keluaran (*BTopi*) yang sudah didapatkan ketika proses pelatihan, dan matriks *data* yang merupakan target dalam bentuk aktual.
2. Menghitung matriks keluaran pada *hidden layer* (*H*) yang prosesnya sama seperti pada proses pelatihan.
3. Menghitung perkalian matriks antara matriks *H* dan bobot keluaran (*BTopi*) untuk mencari keluaran *output layer*.



1.5.1 Denormalisasi Data

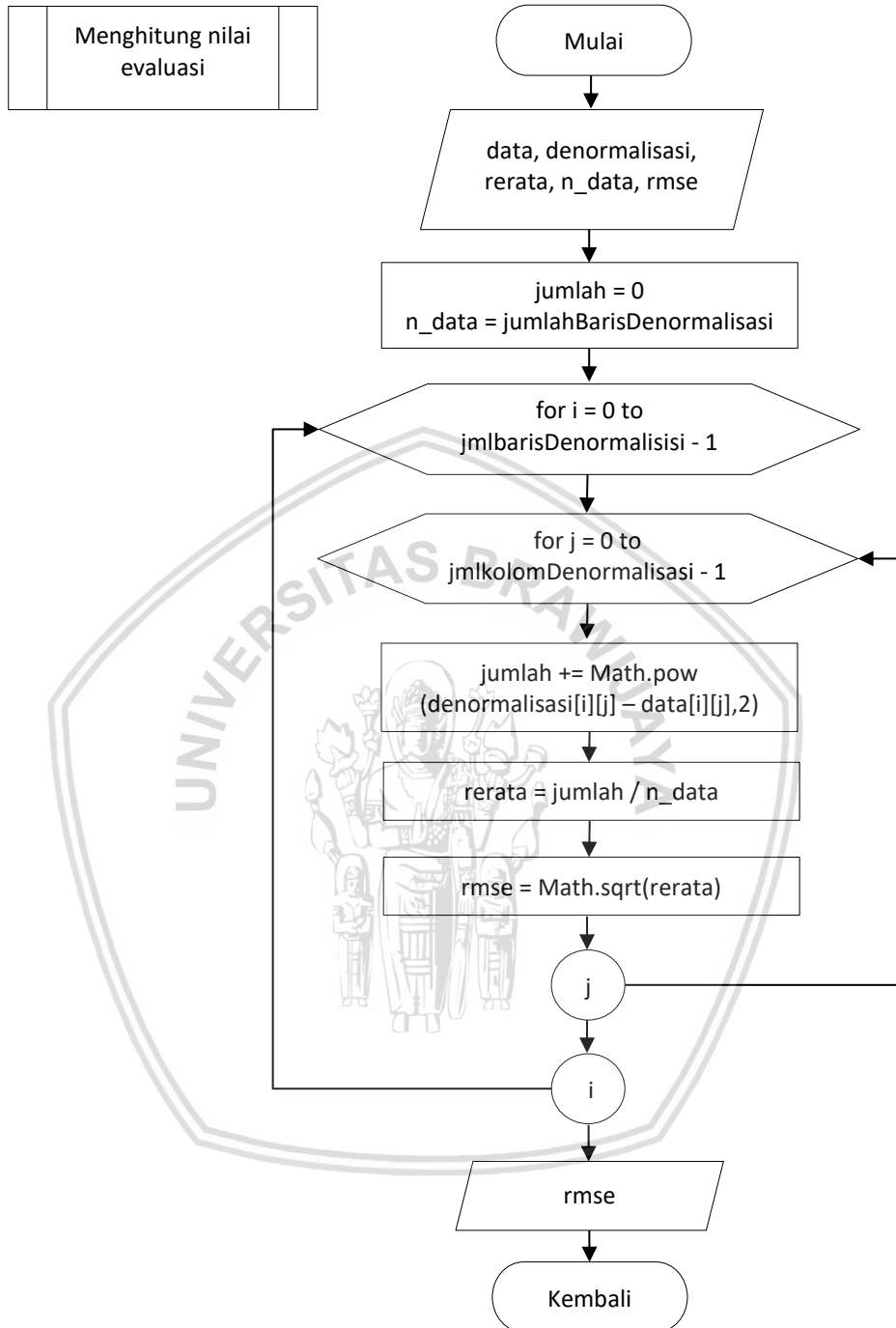


Gambar 4.16 Diagram Alir Proses Denormalisasi

Berdasarkan Gambar 4.16, langkah-langkah pada proses denormalisasi adalah sebagai berikut.

1. Masukan pada proses ini adalah matriks *matriksSatu*, variabel *max*, *min*, dan deklarasi *denormalisasi*.
2. Melakukan perulangan untuk menghitung denormalisasi dengan Persamaan 2.2.
3. Keluaran dari proses ini matriks *denormalisasi* yaitu hasil prediksi dalam bentuk nilai aktual.

1.5.2 Menghitung Nilai Evaluasi



Gambar 4.17 Diagram Alir Proses Menghitung Nilai Evaluasi

Berdasarkan Gambar 4.17, langkah-langkah pada proses menghitung nilai evaluasi adalah sebagai berikut.

1. Masukan pada proses ini adalah matriks *data* yang merupakan target testing aktual, jumlah data (*n_data*) dan *denormalisasi*, serta deklarasi variabel *rerata* dan *rmse*.

2. Melakukan inisialisasi untuk variabel *jumlah* dan *n_data* sama dengan jumlah baris *denormalisasi*.
3. Melakukan proses perhitungan nilai RMSE seperti pada Persamaan 2.16.
4. Keluaran dari proses ini adalah nilai RMSE yaitu nilai hasil evaluasi.

1.6 Contoh Perhitungan Manualisasi

Perhitungan manualisasi merupakan contoh perhitungan dari tahapan implementasi metode ELM dan inisialisasi Nguyen-Widrow terhadap prediksi Indeks Harga Konsumen kelompok, perumahan, air, listrik, gas, dan bahan bakar. Pada contoh perhitungan manualisasi ini digunakan metode ELM dengan inisialisasi bobot awal menggunakan inisialisasi Nguyen-Widrow. Data yang digunakan pada perhitungan manualisasi ini adalah data Indeks Harga Konsumen kelompok perumahan, air, listrik, gas, dan bahan bakar periode Agustus 2014 s.d. Desember 2015.

Tabel 4.2 Data IHK Kelompok Perumahan, Air, Gas, dan Bahan Bakar

| Tahun | Bulan | Indeks |
|-------|-----------|--------|
| 2014 | Agustus | 111,31 |
| | September | 112,17 |
| | Oktober | 113,34 |
| | November | 113,9 |
| | Desember | 115,55 |
| 2015 | Januari | 116,48 |
| | Februari | 116,96 |
| | Maret | 117,3 |
| | April | 117,56 |
| | Mei | 117,8 |
| | Juni | 118,07 |
| | Juli | 118,22 |
| | Agustus | 118,41 |
| | September | 118,65 |
| | Oktober | 118,76 |
| | November | 118,94 |
| | Desember | 119,41 |

Langkah-langkah perhitungan manualisasi untuk menghitung prediksi IHK kelompok perumahan, air, listrik, gas, dan bahan bakar adalah sebagai berikut.

1. Menentukan jumlah fitur yang digunakan

Fitur yang digunakan dalam perhitungan manualisasi berjumlah 5 fitur; X_1 , X_2 , X_3 , X_4 , dan X_5 . Kelima fitur tersebut adalah data 5 bulan sebelum bulan yang ingin diprediksi nilai dari indeks harga konsumen. Penggunaan 5 fitur ini digunakan berdasarkan penelitian yang dilakukan oleh Yaseen et al., (2016).

2. Menentukan jumlah *hidden neuron*, data latih, dan fungsi aktivasi

Pada manualisasi ini *hidden neuron* yang digunakan sebanyak 2, jumlah data latih dan data uji masing-masing adalah 6. Data latih terdiri dari data bulan Januari 2015 sampai dengan Juni 2015, sedangkan data uji terdiri dari data bulan Juli 2015 sampai dengan Desember 2015. Sedangkan fungsi aktivasi yang digunakan adalah fungsi aktivasi sigmoid biner.

3. Proses normalisasi

Sebelum melakukan proses normalisasi dilakukan pencarian nilai minimum dan nilai maksimum dari keseluruhan data seperti ditunjukkan pada Tabel 4.3. Selanjutnya adalah melakukan proses normalisasi menggunakan Persamaan 2.1 dan memisahkan data untuk digunakan sebagai data latih dan data uji seperti ditunjukkan pada Tabel 4.4 dan Tabel 4.5.

Tabel 4.3 Nilai Min dan Max

| | Nilai |
|------------|--------|
| Min | 111,31 |
| Max | 119,41 |

Perhitungan normalisasi data menggunakan Persamaan 2.1. Pada Tabel 4.4 dan Tabel 4.5 menunjukkan hasil perhitungan normalisasi untuk setiap data latih dan data uji.

$$y_{1,1} = \frac{111,31 - 111,31}{119,41 - 111,31}$$

$$y_{1,1} = \frac{0}{8,1}$$

$$y_{1,1} = 0$$

Tabel 4.4 Data Latih Normalisasi

| Bln Thn | X1 | X2 | X3 | X4 | X5 | Y |
|---------------|-------------|----------|----------|---------|----------|----------|
| Januari 2015 | 0 | 0,106173 | 0,250617 | 0,31975 | 0,523457 | 0,638272 |
| Februari 2015 | 0,10617284 | 0,250617 | 0,319753 | 0,52346 | 0,638272 | 0,697531 |
| Maret 2015 | 0,250617284 | 0,319753 | 0,523457 | 0,63827 | 0,697531 | 0,739506 |
| April 2015 | 0,319753086 | 0,523457 | 0,638272 | 0,69753 | 0,739506 | 0,771605 |
| Mei 2015 | 0,52345679 | 0,638272 | 0,697531 | 0,73951 | 0,771605 | 0,801235 |
| Juni 2015 | 0,638271605 | 0,697531 | 0,739506 | 0,7716 | 0,801235 | 0,834568 |

Tabel 4.5 Data Uji Normalisasi

| Bln Thn | X1 | X2 | X3 | X4 | X5 | Y |
|----------------|-------------|----------|----------|---------|----------|----------|
| Juli 2015 | 0,697530864 | 0,739506 | 0,771605 | 0,80123 | 0,834568 | 0,853086 |
| Agustus 2015 | 0,739506173 | 0,771605 | 0,801235 | 0,83457 | 0,853086 | 0,876543 |
| September 2015 | 0,771604938 | 0,801235 | 0,834568 | 0,85309 | 0,876543 | 0,906173 |
| Oktober 2015 | 0,801234568 | 0,834568 | 0,853086 | 0,87654 | 0,906173 | 0,919753 |
| November 2015 | 0,834567901 | 0,853086 | 0,876543 | 0,90617 | 0,919753 | 0,941975 |
| Desember 2015 | 0,85308642 | 0,876543 | 0,906173 | 0,91975 | 0,941975 | 1 |

4. Proses pelatihan

Pada proses pelatihan yang pertama kali dilakukan adalah membuat nilai *random* untuk matriks W_{jk} sebagai bobot masukan, nilai j adalah sama dengan jumlah *hidden neuron* sedangkan nilai k adalah sama dengan jumlah fitur. Dalam perhitungan manualisasi ini jumlah *hidden neuron* sebanyak 2 dan jumlah fitur yang digunakan adalah 5. Jadi matriks bobot masukan memiliki ordo 2×5 .

4.1 Membuat nilai *random* untuk matriks bobot masukan

Hasil nilai *random* untuk matriks bobot masukan ditunjukkan pada Tabel 4.6.

Tabel 4.6 Matriks Bobot Masukan

| $j \backslash k$ | 1 | 2 | 3 | 4 | 5 |
|------------------|-------|-------|-------|-------|------|
| 1 | 0,12 | 0,07 | -0,56 | -0,3 | 0,6 |
| 2 | -0,64 | -0,02 | 0,68 | -0,21 | 0,79 |

Setelah itu membuat nilai *random* untuk matriks bias $b_{j \times k}$, nilai j adalah 1 dan nilai k adalah sama dengan jumlah *hidden neuron*. Dalam perhitungan manualisasi ini matriks bias memiliki ordo 1×2 .

4.2 Membuat nilai *random* untuk matriks bias

Hasil nilai *random* untuk matriks bobot masukan ditunjukkan pada Tabel 4.7.

Tabel 4.7 Matriks Bias

| $j \backslash k$ | 1 | 2 |
|------------------|------|------|
| 1 | 0,48 | 0,56 |

4.3 Melakukan inialisasi bobot masukan dengan *Nguyen-Widrow*

Perhitungan manualisasi inialisasi bobot masukan dengan inialisasi *Nguyen-Widrow* diawali dengan mencari nilai beta (β) menggunakan Persamaan 2.9 dan



nilai gamma (γ) dari setiap kolom pada bobot masukan *random* menggunakan Persamaan 2.10. Nilai gamma tersebut ditunjukkan pada Tabel 4.8.

$$\begin{aligned}\beta &= 0,7^{\frac{1}{5}} \\ \beta &= 0,931149915 \\ \gamma_1 &= \sqrt{(0,12^2) + (-0,64^2)} \\ &= \sqrt{0,424} \\ &= 0,65115\end{aligned}$$

Tabel 4.8 Nilai Gamma (γ)

| Fitur ke- | | | | |
|-------------|-------------|-------------|-------------|-------------|
| 1 | 2 | 3 | 4 | 5 |
| 0,651152824 | 0,072801099 | 0,880908622 | 0,366196668 | 0,992018145 |

Selanjutnya melakukan perhitungan inisialisasi untuk bobot masukan menggunakan Persamaan 2.8. Nilai bobot masukan menggunakan inisialisasi *Nguyen-Widrow* ditunjukkan pada Tabel 4.9.

$$\begin{aligned}w_{1,1} &= \frac{0,931149915 \times 0,12}{0,651152823843988} \\ &= \frac{0,11173799}{0,651152823843988} \\ &= 0,171600254\end{aligned}$$

Tabel 4.9 Matriks Bobot Masukan Menggunakan *Nguyen-Widrow*

| $j \backslash k$ | 1 | 2 | 3 | 4 | 5 |
|------------------|--------------|--------------|--------------|--------------|-------------|
| 1 | 0,171600254 | 0,895322942 | -0,591938755 | -0,762827733 | 0,563185212 |
| 2 | -0,915201353 | -0,255806555 | 0,718782773 | -0,533979413 | 0,741527195 |

4.4 Memperbanyak matriks bias sebanyak jumlah data latih

Sebelumnya matriks bias dengan ordo 1×2 sudah terbentuk, langkah selanjutnya adalah memperbanyak matriks bias sebanyak $j \times k$, j adalah jumlah data latih dan k adalah jumlah *hidden neuron*. Karena data latih yang digunakan sebanyak 6 data maka matriks bias diperbanyak menjadi 6 kali. Jadi setelah diperbanyak matriks bias memiliki ordo 6×2 . Matriks bias pada proses pelatihan ditunjukkan pada Tabel 4.10.

Tabel 4.10 Matriks Bias Proses Pelatihan

| $j \backslash k$ | 1 | 2 |
|------------------|------|------|
| 1 | 0,48 | 0,56 |
| 2 | 0,48 | 0,56 |
| 3 | 0,48 | 0,56 |
| 4 | 0,48 | 0,56 |
| 5 | 0,48 | 0,56 |
| 6 | 0,48 | 0,56 |

4.2 Menghitung H_{init}

Proses perhitungan H_{init} dilakukan menggunakan Persamaan 2.7. Berdasarkan persamaan tersebut maka yang pertama harus dilakukan adalah melakukan perkalian antara data latih (X_{train}) dan matriks bobot masukan *transpose* (W^T). Sebelum melakukan perkalian, bobot masukan harus *ditranspose* terlebih dahulu menjadi seperti Tabel 4.11.

Tabel 4.11 Matriks *Transpose* Bobot Masukan

| | 1 | 2 |
|---|--------------|--------------|
| 1 | 0,171600254 | -0,915201353 |
| 2 | 0,895322942 | -0,255806555 |
| 3 | -0,591938755 | 0,718782773 |
| 4 | -0,762827733 | -0,533979413 |
| 5 | 0,563185212 | 0,741527195 |

Hasil perkalian dari data latih dengan matriks bobot masukan *transpose* ditunjukkan pada Tabel 4.12. Contoh dari perhitungan perkalian tersebut adalah sebagai berikut.

$$\begin{aligned}
 (X_{train} \times W^T)_{1,1} &= (0 \times 0,171600254) + (0,106173 \times 0,895322942) + \\
 &\quad (0,250617 \times -0,591938755) + \\
 &\quad (0,31975 \times -0,762827733) + \\
 &\quad (0,523457 \times 0,563185212) \\
 &= -0,002404503
 \end{aligned}$$

Tabel 4.12 Hasil Perkalian Bobot Masukan *Transpose* dengan Data Latih

| | 1 | 2 |
|---|--------------|-------------|
| 1 | -0,002404503 | 0,370395558 |
| 2 | 0,013486219 | 0,262334543 |



Tabel 4.12 Hasil Perkalian Bobot Masukan *Transpose* dengan Data Latih (lanjutan)

| | 1 | 2 |
|---|--------------|--------------|
| 3 | -0,074618311 | 0,241505719 |
| 4 | 0,030097938 | 0,208133316 |
| 5 | 0,118829651 | 0,03631572 |
| 6 | 0,158942406 | -0,048919636 |

Hasil perkalian antara X_{train} dan W^T kemudian dijumlahkan dengan matriks bias. Hasil perhitungan H_{init} ditunjukkan pada Tabel 4.13. Berikut adalah contoh perhitungan H_{init} .

$$\begin{aligned}
 H_{init_{1,1}} &= -0,002404503 + 0,48 \\
 &= 0,477595497
 \end{aligned}$$

Tabel 4.13 Matriks Hinit

| | 1 | 2 |
|---|-------------|-------------|
| 1 | 0,477595497 | 0,930395558 |
| 2 | 0,493486219 | 0,822334543 |
| 3 | 0,405381689 | 0,801505719 |
| 4 | 0,510097938 | 0,768133316 |
| 5 | 0,598829651 | 0,59631572 |
| 6 | 0,638942406 | 0,511080364 |

4.1 Menghitung H dengan fungsi aktivasi

Perhitungan fungsi aktivasi dari H_{init} dilakukan menggunakan Persamaan 2.11 untuk mendapatkan matriks keluaran pada *hidden layer*. Hasil perhitungan fungsi aktivasi ditunjukkan pada Tabel 4.14. Berikut adalah contoh perhitungan H dengan fungsi aktivasi.

$$\begin{aligned}
 H_{1,1} &= \frac{1}{1 + \exp(-0,477595497)} \\
 &= 0,61718
 \end{aligned}$$

Tabel 4.14 Matriks Keluaran *Hidden Layer* (H)

| | 1 | 2 |
|---|---------|---------|
| 1 | 0,61718 | 0,71716 |
| 2 | 0,62093 | 0,69473 |
| 3 | 0,59998 | 0,6903 |



Tabel 4.14 Matriks Keluaran *Hidden Layer* (H) (lanjutan)

| | | |
|----------|----------|----------|
| | 1 | 2 |
| 4 | 0,62483 | 0,68312 |
| 5 | 0,64539 | 0,64481 |
| 6 | 0,65451 | 0,62506 |

4.7 Menghitung matriks *pseudo-inverse* dengan *Moore-Penrose*

Perhitungan matriks *pseudo-inverse* dengan *Moore-Penrose* dapat dilakukan menggunakan Persamaan 2.12. Berdasarkan persamaan tersebut maka matriks keluaran *hidden layer* harus ditranspose terlebih dahulu seperti yang ditunjukkan pada Tabel 4.15.

Tabel 4.15 Matriks Keluaran *Hidden Layer* Transpose

| | | | | | | |
|----------|----------|----------|----------|----------|----------|----------|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 0,61718 | 0,62093 | 0,59998 | 0,62483 | 0,64539 | 0,65451 |
| 2 | 0,71716 | 0,69473 | 0,6903 | 0,68312 | 0,64481 | 0,62506 |

Setelah ditranspose, maka selanjutnya adalah melakukan perkalian antara H^T dan H seperti pada contoh perhitungan berikut ini.

$$\begin{aligned}
 H^T H_{1,1} &= (0,61718 \times 0,61718) + (0,62093 \times 0,62093) + \dots + \\
 &\quad (0,65451 \times 0,65451) \\
 &= 2,36176
 \end{aligned}$$

Tabel 4.16 Matriks Hasil Perkalian H^T dan H

| | | |
|----------|----------|----------|
| | 1 | 2 |
| 1 | 2,36176 | 2,54025 |
| 2 | 2,54025 | 2,74661 |

Selanjutnya adalah melakukan *inverse* dari hasil perkalian H^T dan H . Hasil dari *inverse* tersebut ditunjukkan pada Tabel 4.17 Berikut adalah contoh perhitungannya.

$$\begin{aligned}
 (H^T H)^{-1} &= \frac{1}{(2,36176 \times 2,74661) - (2,54025 \times 2,54025)} \times \\
 &\quad \begin{bmatrix} 2,74661 & -2,54025 \\ -2,54025 & 2,36176 \end{bmatrix} \\
 &= \begin{bmatrix} 80,89740912 & -74,81958385 \\ -74,81958385 & 69,56247158 \end{bmatrix}
 \end{aligned}$$



Tabel 4.17 Matriks Hasil Inverse Perkalian H^T dan H

| | | |
|----------|--------------|--------------|
| | 1 | 2 |
| 1 | 80,89740912 | -74,81958385 |
| 2 | -74,81958385 | 69,56247158 |

Langkah selanjutnya adalah menghitung H^+ dengan melakukan perkalian antara matriks *inverse* perkalian H^T dan H dan matriks H *transpose*. Hasil dari perhitungan tersebut ditunjukkan pada Tabel 4.18. Berikut adalah contoh perhitungannya.

$$H^+_{1,1} = (80,89740912 \times 0,61718) + (-74,81958385 \times 0,71716) = -3,729$$

Tabel 4.18 Hasil Perhitungan H^+

| | | | | | | |
|----------|----------|----------|----------|----------|----------|----------|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | -3,729 | -1,7481 | -3,11087 | -0,5634 | 3,96562 | 6,18181 |
| 2 | 3,70997 | 1,86973 | 3,12848 | 0,76983 | -3,4329 | -5,48979 |

4.8 Menghitung $\hat{\beta}$ sebagai bobot keluaran

Menghitung $\hat{\beta}$ dilakukan menggunakan Persamaan 2.13. Hasil perhitungan $\hat{\beta}$ ditunjukkan pada Tabel 4.19. Berikut adalah contoh perhitungannya.

$$\hat{\beta} = (-3,729 \times 0,638272) + (-1,7481 \times 0,697531) + \dots + (6,18181 \times 0,834568) = 2,00177$$

Tabel 4.19 Matriks Bobot Keluaran

| | |
|----------|----------|
| | 1 |
| 1 | 2,00177 |
| 2 | -0,75249 |

5. Proses Pengujian

Proses pengujian dilakukan dengan menggunakan nilai dari matriks bobot masukan Nguyen-Widrow (Tabel 4.9), matriks bias (Tabel 4.7), dan bobot keluaran (Tabel 4.19) yang sama dengan proses pelatihan.

5.1 Memperbanyak matriks bias sebanyak jumlah data uji

Matriks bias proses pelatihan ditunjukkan pada Tabel 4.20.



Tabel 4.20 Matriks Bias Proses Pengujian

| | 1 | 2 |
|---|------|------|
| 1 | 0,48 | 0,56 |
| 2 | 0,48 | 0,56 |
| 3 | 0,48 | 0,56 |
| 4 | 0,48 | 0,56 |
| 5 | 0,48 | 0,56 |
| 6 | 0,48 | 0,56 |

5.2 Menghitung H_{init}

Proses perhitungan H_{init} dilakukan menggunakan Persamaan 2.7. Berdasarkan persamaan tersebut maka yang pertama harus dilakukan adalah melakukan perkalian antara data latih (X_{test}) dan matriks bobot masukan *transpose* (W^T). Sebelum melakukan perkalian, bobot masukan harus ditranspose terlebih dahulu menjadi seperti Tabel 4.21.

Tabel 4.21 Matriks *Transpose* Bobot Masukan Pada Proses Pengujian

| | 1 | 2 |
|---|--------------|--------------|
| 1 | 0,171600254 | -0,915201353 |
| 2 | 0,895322942 | -0,255806555 |
| 3 | -0,591938755 | 0,718782773 |
| 4 | -0,762827733 | -0,533979413 |
| 5 | 0,563185212 | 0,741527195 |

Hasil perkalian dari data uji dengan matriks bobot masukan *transpose* ditunjukkan pada Tabel 4.22. Contoh dari perhitungan perkalian tersebut adalah sebagai berikut.

$$\begin{aligned}
 (X_{test} \times W^T)_{1,1} &= (0,739506173 \times 0,171600254) + \\
 &\quad (0,771604938 \times 0,895322942) + \dots + \\
 &\quad (0,85308642 \times 0,563185212) \\
 &= 0,183862801
 \end{aligned}$$

Tabel 4.22 Hasil Perkalian Bobot Masukan *Transpose* dengan Data Uji

| | 1 | 2 |
|---|-------------|--------------|
| 1 | 0,183862801 | -0,081923349 |
| 2 | 0,187267374 | -0,111320344 |
| 3 | 0,198656403 | -0,116811865 |



Tabel 4.22 Hasil Perkalian Bobot Masukan *Transpose* dengan Data Uji (lanjutan)

| | 1 | 2 |
|---|-------------|--------------|
| 4 | 0,221416604 | -0,129699303 |
| 5 | 0,214877575 | -0,153834326 |
| 6 | 0,223673671 | -0,146258823 |

Hasil perkalian antara X_{test} dan W^T kemudian dijumlahkan dengan matriks bias yang digunakan pada proses pelatihan. Hasil perhitungan H_{init} ditunjukkan pada Tabel 4.23. Berikut adalah contoh perhitungan H_{init} .

$$\begin{aligned}
 H_{init_{1,1}} &= 0,183862801 + 0,48 \\
 &= 0,663862801
 \end{aligned}$$

Tabel 4.23 Matriks Hinit

| | 1 | 2 |
|---|-------------|-------------|
| 1 | 0,663862801 | 0,478076651 |
| 2 | 0,667267374 | 0,448679656 |
| 3 | 0,678656403 | 0,443188135 |
| 4 | 0,701416604 | 0,430300697 |
| 5 | 0,694877575 | 0,406165674 |
| 6 | 0,703673671 | 0,413741177 |

5.3 Menghitung H dengan fungsi aktivasi

Perhitungan fungsi aktivasi dari H_{init} dilakukan menggunakan Persamaan 2.11 untuk mendapatkan matriks keluaran pada *hidden layer*. Hasil perhitungan fungsi aktivasi ditunjukkan pada Tabel 4.24. Berikut adalah contoh perhitungan H dengan fungsi aktivasi.

$$\begin{aligned}
 H_{1,1} &= \frac{1}{1 + \exp(-0,663862801)} \\
 &= 0,660127578
 \end{aligned}$$

Tabel 4.24 Matriks Keluaran *Hidden Layer* (H)

| | 1 | 2 |
|---|-------------|-------------|
| 1 | 0,660127578 | 0,617293601 |
| 2 | 0,660891009 | 0,610325264 |
| 3 | 0,663438755 | 0,609018437 |
| 4 | 0,668501777 | 0,60594547 |



Tabel 4.24 Matriks Keluaran *Hidden Layer (H)* (lanjutan)

| | 1 | 2 |
|---|-------------|-------------|
| 5 | 0,667051088 | 0,600168124 |
| 6 | 0,669001769 | 0,601984603 |

4.8 Menghitung keluaran *output layer* atau hasil prediksi

Menghitung keluaran *output layer* menggunakan Persamaan 2.15. Hasil keluaran *output layer* ditunjukkan pada Tabel 4.25. Berikut adalah contoh perhitungannya.

$$\hat{Y}_{1,1} = (0,660127578 \times 2,00177) + (0,617293601 \times -0,75249)$$

$$= 0,856917385$$

Tabel 4.25 Matriks Keluaran *Output Layer*

| | 1 |
|---|-------------|
| 1 | 0,856917385 |
| 2 | 0,863689205 |
| 3 | 0,869772587 |
| 4 | 0,882219981 |
| 5 | 0,88366343 |
| 6 | 0,886201365 |

5.5 Denormalisasi hasil prediksi

Melakukan denormalisasi hasil prediksi menggunakan Persamaan 2.2. Hasil denormalisasi prediksi ditunjukkan pada Tabel 4.26. Berikut adalah contoh perhitungannya.

$$y_{1,1} = 111,31 + (0,856917385 \times (119,41 - 111,31))$$

$$= 118,2510308$$

Tabel 4.26 Matriks Denormalisasi Hasil Prediksi

| | Hasil Prediksi | Nilai Aktual |
|---|----------------|--------------|
| 1 | 118,2510308 | 118,22 |
| 2 | 118,3058826 | 118,41 |
| 3 | 118,355158 | 118,65 |
| 4 | 118,4559818 | 118,76 |
| 5 | 118,4676738 | 118,94 |
| 6 | 118,4882311 | 119,41 |



5.6 Perhitungan nilai evaluasi

Perhitungan nilai evaluasi menggunakan RMSE dan dilakukan dengan menggunakan Persamaan 2.16.

$$RMSE = \sqrt{\frac{1}{6} \times ((118,22 - 118,25)^2 + \dots + (119,41 - 118,488)^2)}$$

$$RMSE = 0,45897$$

Berdasarkan perhitungan manualisasi, didapatkan nilai RMSE 0,45897 dengan jumlah fitur = 5, jumlah data latih = 6, jumlah data uji = 6, jumlah *hidden neuron* = 2, dan fungsi aktivasi yang digunakan adalah fungsi sigmoid biner.

1.7 Perancangan User Interface

Perancangan *user interface* adalah proses untuk merancang antarmuka pengguna dengan sistem. Percangan *user interface* pada penelitian ini terdiri dari 2 bagian. Bagian pertama berisi *form* untuk memasukkan parameter ELM yaitu jumlah fitur, jumlah *hidden neuron*, jumlah data latih, fungsi aktivasi, serta 2 tombol yaitu tombol Proses dan tombol Default. Tombol proses berfungsi untuk memproses perhitungan ELM secara keseluruhan, sedangkan tombol default untuk mengambil parameter ELM secara *default* yang sudah di-*set* sebelumnya. Bagian kedua terdiri dari 6 tab menu yaitu tab menu Data, Data Normalisasi, Pembagian Data, Proses Pelatihan, Proses Pengujian, dan Hasil Evaluasi.

1 PREDIKSI INDEKS HARGA KONSUMEN KELOMPOK PERUMAHAN, AIR, GAS, DAN BAHAN BAKAR

| | |
|----------------------|-----------|
| Jumlah Fitur | 2 |
| Jumlah Hidden Neuron | 3 |
| Jumlah Data Latih | 4 |
| Fungsi Aktivasi | 5 |
| | 6 Proses |
| | 7 Default |

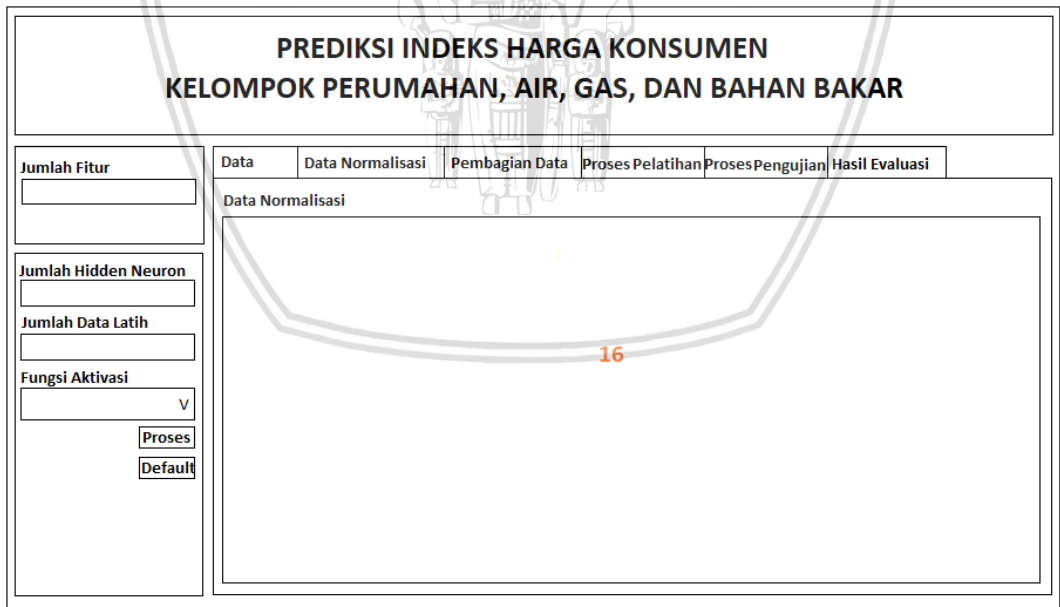
| | | | | | |
|---------|------------------|---|------------------|------------------|----------------|
| Data | Data Normalisasi | Pembagian Data | Proses Pelatihan | Proses Pengujian | Hasil Evaluasi |
| Dataset | | Pembentukan Data Berdasarkan Jumlah Fitur | | | |
| 14 | | 15 | | | |

Gambar 4.18 Perancangan User Interface Halaman Utama

Keterangan:

1. Judul penelitian
2. *Form* untuk memasukkan jumlah fitur

3. *Form* untuk memasukkan jumlah *hidden neuron*
4. *Form* untuk memasukkan jumlah data latih
5. *Form* untuk memasukkan jenis fungsi aktivasi
6. Tombol untuk melakukan proses perhitungan ELM
7. Tombol untuk mengambil nilai parameter yang telah di-set sebelumnya
8. *Tab* menu untuk menampilkan data aktual yang digunakan dalam penelitian
9. *Tab* menu untuk menampilkan data yang telah dinormalisasi
10. *Tab* menu untuk menampilkan hasil pembagian data
11. *Tab* menu untuk menampilkan seluruh hasil perhitungan pada proses pelatihan
12. *Tab* menu untuk menampilkan seluruh hasil perhitungan pada proses pengujian
13. *Tab* menu untuk menampilkan hasil evaluasi penelitian
14. Panel untuk menampilkan seluruh *dataset* aktual yang digunakan
15. Panel untuk menampilkan hasil dari pembentukan data aktual berdasarkan jumlah fitur yang sudah dimasukkan pada *form* nomor 3



Gambar 4.19 Perancangan *User Interface* Tab Data Normalisasi

Keterangan:

16. Panel untuk menampilkan seluruh data yang sudah dinormalisasi

PREDIKSI INDEKS HARGA KONSUMEN KELOMPOK PERUMAHAN, AIR, GAS, DAN BAHAN BAKAR

| | Data | Data Normalisasi | Pembagian Data | Proses Pelatihan | Proses Pengujian | Hasil Evaluasi |
|--|--|------------------|---|------------------|------------------|----------------|
| <p>Jumlah Fitur</p> <input style="width: 100%;" type="text"/> | <p>Data Training</p> <div style="border: 1px solid black; height: 150px; display: flex; align-items: center; justify-content: center; font-size: 24px; color: orange;">17</div> | | <p>Data Testing</p> <div style="border: 1px solid black; height: 150px; display: flex; align-items: center; justify-content: center; font-size: 24px; color: orange;">18</div> | | | |
| <p>Jumlah Hidden Neuron</p> <input style="width: 100%;" type="text"/> | | | | | | |
| <p>Jumlah Data Latih</p> <input style="width: 100%;" type="text"/> | | | | | | |
| <p>Fungsi Aktivasi</p> <input style="width: 100%;" type="text" value="v"/> | | | | | | |
| <p><input type="button" value="Proses"/></p> <p><input type="button" value="Default"/></p> | | | | | | |

Gambar 4.20 Perancangan User Interface Tab Pembagian Data

Keterangan:

- 17. Panel untuk menampilkan data latih yang digunakan dalam penelitian, jumlah data latih yang digunakan berdasarkan nilai yang dimasukkan pada form 4.
- 18. Panel untuk menampilkan data uji yang digunakan dalam penelitian.

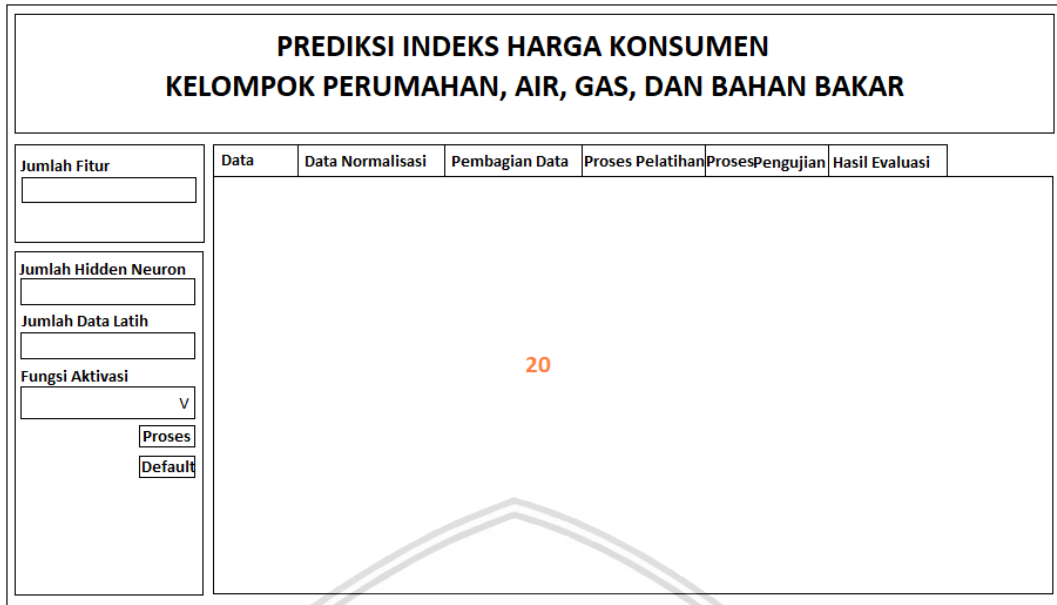
PREDIKSI INDEKS HARGA KONSUMEN KELOMPOK PERUMAHAN, AIR, GAS, DAN BAHAN BAKAR

| | Data | Data Normalisasi | Pembagian Data | Proses Pelatihan | Proses Pengujian | Hasil Evaluasi |
|--|------|------------------|----------------|------------------|------------------|----------------|
| <p>Jumlah Fitur</p> <input style="width: 100%;" type="text"/> | 19 | | | | | |
| <p>Jumlah Hidden Neuron</p> <input style="width: 100%;" type="text"/> | | | | | | |
| <p>Jumlah Data Latih</p> <input style="width: 100%;" type="text"/> | | | | | | |
| <p>Fungsi Aktivasi</p> <input style="width: 100%;" type="text" value="v"/> | | | | | | |
| <p><input type="button" value="Proses"/></p> <p><input type="button" value="Default"/></p> | | | | | | |

Gambar 4.21 Perancangan User Interface Tab Proses Pelatihan

Keterangan:

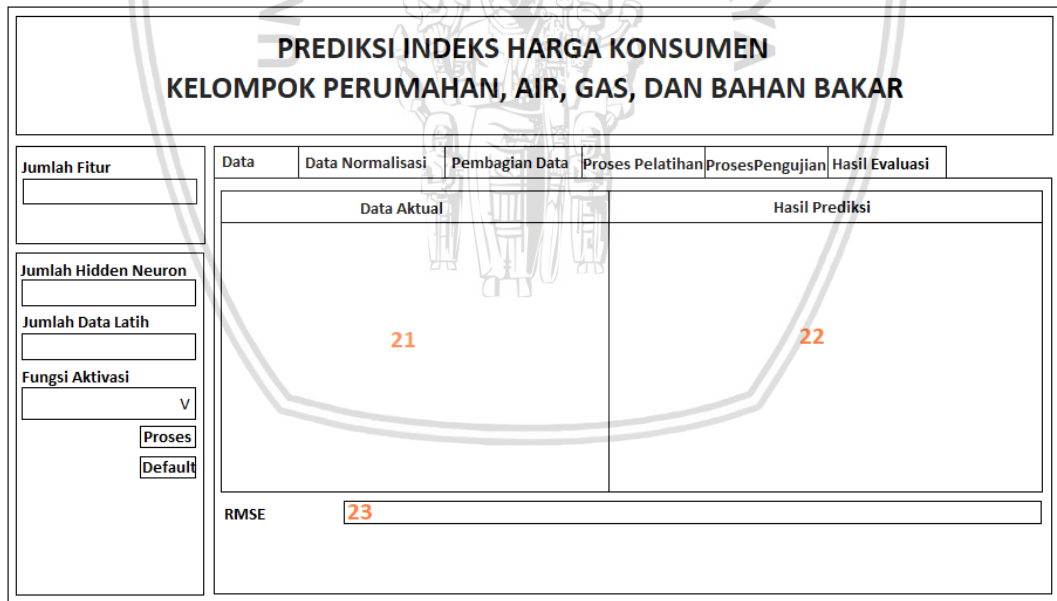
- 19. Panel untuk menampilkan seluruh hasil perhitungan dari proses pelatihan.



Gambar 4.22 Perancangan *User Interface* Tab Proses Pengujian

Keterangan:

- 20. Panel untuk menampilkan seluruh hasil perhitungan dari proses pengujian.



Gambar 4.23 Perancangan *User Interface* Tab Hasil Evaluasi

Keterangan:

- 21. Bagian kolom pada tabel untuk menampilkan nilai data aktual.
- 22. Bagian kolom pada tabel untuk menampilkan hasil prediksi berdasarkan perhitungan menggunakan inisialisasi Nguyen-Widrow dan ELM.
- 23. *Form* untuk menampilkan nilai RMSE.



1.8 Perancangan Pengujian

Perancangan pengujian digunakan untuk menguji hasil prediksi yang dilakukan oleh sistem. Pengujian yang dilakukan terdiri dari 4 pengujian sebagai berikut.

1. Pengujian jumlah fitur.
2. Pengujian jumlah data latih.
3. Pengujian jumlah *hidden neuron*.
4. Pengujian fungsi aktivasi.
5. Pengujian perbandingan metode.

1.8.1 Rancangan Pengujian Jumlah Fitur

Perancangan pengujian jumlah fitur dilakukan untuk menemukan jumlah fitur paling optimal dalam melakukan prediksi dengan nilai RMSE seminimal mungkin. Pengujian akan dilakukan sebanyak 10 kali dengan fitur yang digunakan adalah 1 sampai dengan 10. Rancangan pengujian fungsi aktivasi ditunjukkan pada Tabel 4.27.

Tabel 4.27 Rancangan Pengujian Jumlah Fitur

| Jumlah Fitur | RMSE Percobaan ke- | | | | | | | | | | Rata-rata RMSE |
|--------------|--------------------|---|---|---|---|---|---|---|---|----|----------------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| 1 bulan | | | | | | | | | | | |
| 2 bulan | | | | | | | | | | | |
| 3 bulan | | | | | | | | | | | |
| 4 bulan | | | | | | | | | | | |
| 5 bulan | | | | | | | | | | | |
| 6 bulan | | | | | | | | | | | |
| 7 bulan | | | | | | | | | | | |
| 8 bulan | | | | | | | | | | | |
| 10 bulan | | | | | | | | | | | |

1.8.2 Rancangan Pengujian Jumlah Data Latih

Perancangan pengujian jumlah data latih dilakukan untuk mengetahui jumlah data latih yang paling optimal untuk melakukan prediksi sehingga menghasilkan nilai RMSE seminimal mungkin. Pengujian jumlah data latih dilakukan dengan menggunakan jumlah fitur paling optimal yang didapatkan ketika pengujian jumlah fitur dengan RMSE terkecil. Pengujian ini akan dilakukan sebanyak 10 kali percobaan dengan 6 variasi jumlah data latih yaitu 10, 20, 30, 40, 50, dan 60. Rancangan pengujian jumlah data latih ditunjukkan pada Tabel 4.28.

Tabel 4.28 Rancangan Pengujian Jumlah Data Latih

| Jumlah Data Latih | RMSE Percobaan ke- | | | | | | | | | | Rata-rata RMSE |
|-------------------|--------------------|---|---|---|---|---|---|---|---|----|----------------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| 10 | | | | | | | | | | | |
| 20 | | | | | | | | | | | |

Tabel 4.28 Rancangan Pengujian Jumlah Data Latih (lanjutan)

| Jumlah Data Latih | RMSE Percobaan ke- | | | | | | | | | | Rata-rata RMSE |
|-------------------|--------------------|---|---|---|---|---|---|---|---|----|----------------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| 30 | | | | | | | | | | | |
| 40 | | | | | | | | | | | |
| 50 | | | | | | | | | | | |
| 60 | | | | | | | | | | | |

1.8.3 Rancangan Pengujian Jumlah *Hidden Neuron*

Perancangan pengujian jumlah *hidden neuron* dilakukan untuk mengetahui jumlah *hidden neuron* yang paling optimal untuk melakukan prediksi sehingga menghasilkan nilai RMSE seminimal mungkin. Pengujian *hidden neuron* dilakukan dengan menggunakan jumlah fitur yang menghasilkan nilai RMSE terkecil pada pengujian jumlah fitur dan jumlah data latih yang digunakan adalah jumlah data latih paling optimal yang didapatkan ketika melakukan pengujian jumlah data latih. Pengujian ini akan dilakukan sebanyak 10 kali percobaan dengan jumlah *hidden neuron* 1 sampai dengan 20. Rancangan pengujian jumlah *hidden neuron* ditunjukkan pada Tabel 4.29.

Tabel 4.29 Rancangan Pengujian Jumlah *Hidden Neuron*

| Jumlah <i>Hidden Neuron</i> | Waktu Eksekusi Percobaan ke- | | | | | | | | | | Rata-rata RMSE |
|-----------------------------|------------------------------|---|---|---|---|---|---|---|---|----|----------------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| 1 | | | | | | | | | | | |
| 2 | | | | | | | | | | | |
| 3 | | | | | | | | | | | |
| 4 | | | | | | | | | | | |
| 5 | | | | | | | | | | | |
| 6 | | | | | | | | | | | |
| 7 | | | | | | | | | | | |
| 8 | | | | | | | | | | | |
| | | | | | | | | | | | |
| 20 | | | | | | | | | | | |

1.8.4 Rancangan Pengujian Fungsi Aktivasi

Perancangan pengujian fungsi aktivasi dilakukan untuk mengetahui fungsi aktivasi yang paling cocok untuk melakukan prediksi dengan nilai RMSE seminimal mungkin. Pengujian fungsi aktivasi dilakukan dengan menggunakan jumlah fitur yang menghasilkan nilai RMSE terkecil ketika melakukan pengujian jumlah fitur, jumlah data latih yang digunakan adalah jumlah data latih paling optimal yang didapatkan ketika pengujian jumlah data latih sedangkan untuk jumlah *hidden neuron* yang digunakan untuk pengujian adalah *hidden neuron* yang menghasilkan RMSE terkecil berdasarkan pengujian jumlah *hidden neuron*. Pengujian akan dilakukan sebanyak 10 kali dengan 4 variasi fungsi aktivasi. Rancangan pengujian fungsi aktivasi ditunjukkan pada Tabel 4.30.



Tabel 4.30 Rancangan Pengujian Fungsi Aktivasi

| Fungsi Aktivasi | RMSE Percobaan ke- | | | | | | | | | | Rata-rata RMSE |
|-------------------------|--------------------|---|---|---|---|---|---|---|---|----|----------------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| Fungsi Sin | | | | | | | | | | | |
| Fungsi Sigmoid | | | | | | | | | | | |
| Fungsi Triangular Basis | | | | | | | | | | | |
| Fungsi Radial Basis | | | | | | | | | | | |

1.8.5 Rancangan Pengujian Perbandingan Metode

Perancangan pengujian perbandingan metode dilakukan untuk mengetahui nilai *error* antara metode ELM menggunakan inisialisasi bobot *random* dan inisialisasi Nguyen-Widrow. Pengujian perbandingan metode dilakukan dengan menggunakan jumlah fitur yang menghasilkan nilai RMSE terkecil ketika melakukan pengujian jumlah fitur, jumlah data latih yang digunakan adalah jumlah data latih paling optimal yang didapatkan ketika pengujian jumlah data latih sedangkan untuk jumlah *hidden neuron* yang digunakan untuk pengujian adalah *hidden neuron* yang menghasilkan RMSE terkecil berdasarkan pengujian jumlah *hidden neuron*. Masing-masing pengujian akan dilakukan sebanyak 10 kali percobaan. Rancangan pengujian fungsi aktivasi ditunjukkan pada Tabel 4.31.

Tabel 4.31 Rancangan Pengujian Perbandingan Metode

| Metode | RMSE Percobaan ke- | | | | | | | | | | Rata-rata RMSE |
|--------|--------------------|---|---|---|---|---|---|---|---|----|----------------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| ELM-NW | | | | | | | | | | | |
| ELM | | | | | | | | | | | |



BAB 5 IMPLEMENTASI

Bab ini menjelaskan tentang implementasi sistem yang berdasarkan analisis kebutuhan serta proses perancangan sistem sebelumnya. Pada bab ini juga menjelaskan tentang implementasi dari metode *Extreme Learning Machine* dan inialisasi Nguyen-Widrow serta antarmuka yang digunakan dalam menyelesaikan proses prediksi Indeks Harga Konsumen kelompok perumahan, air, listrik, gas, dan bahan bakar.

1.1 Implementasi Kode Program

Sistem prediksi Indeks Harga Konsumen kelompok perumahan, air, listrik, dan bahan bakar diimplementasikan berupa aplikasi berbasis desktop yang dibangun dengan *tools* pemrograman Netbeans 8.1 dan bahasa pemrograman Java.

1.1.1 Implementasi Proses Normalisasi Data

```

1 public double[][] normalisasi(double[][] data) {
2     normalisasi = new double[data.length][data[0].length];
3     double max = Integer.MIN_VALUE, min =
4     Integer.MAX_VALUE;
5     for (int i = 0; i < data.length; i++) {
6         for (int j = 0; j < data[0].length; j++) {
7             if (data[i][j] > max) {
8                 max = data[i][j];
9                 this.max = max;
10            }
11            if (data[i][j] < min) {
12                min = data[i][j];
13                this.min = min;
14            }
15        }
16    }
17    for (int i = 0; i < data.length; i++) {
18        for (int j = 0; j < data[0].length; j++) {
19            normalisasi[i][j] = (data[i][j] - min) / (max
20            - min);
21        }
22    }
23    return normalisasi;
24 }

```

Kode Program 5.1 Proses Normalisasi Data

Penjelasan dari Kode Program 5.1 adalah sebagai berikut.

1. Baris 2 merupakan inialisasi dan deklarasi *array normalisasi* dengan panjang baris dan panjang kolom sama dengan *array* yang ingin dinormalisasi
2. Baris 3 merupakan deklarasi variabel *max* dan *min*
3. Baris 4-15 merupakan perulangan untuk mencari nilai maksimal dan minimal dari array yang ingin dinormalisasi

4. Baris 16-20 merupakan perulangan untuk menghitung proses normalisasi menggunakan metode *Min-Max Normalization*

1.1.2 Implementasi Proses Membuat Nilai Random untuk Bobot Masukan

```

1 public double[][] randomBobotMasukan(int fitur, int
2 jumlahHiddenNeuron) {
3     Random rand = new Random();
4     W = new double[jumlahHiddenNeuron][fitur];
5     for (int i = 0; i < W.length; i++) {
6         for (int j = 0; j < W[i].length; j++) {
7             W[i][j] = rand.nextInt((100 - (-100)) + 1) +
8                 (-100);
9             W[i][j] /= 100;
10        }
11    }
12    return W;
13 }

```

Kode Program 5.2 Proses Membuat Nilai Random untuk Bobot Masukan

Penjelasan dari Kode Program 5.2 adalah sebagai berikut.

1. Baris 3 merupakan inisialisasi variabel *rand* untuk menyimpan hasil dari proses *random* angka. Pada proses *random* ini menggunakan *library Random*.
2. Baris 4 merupakan inisialisasi dan deklarasi *array* dengan nama *W* dengan jumlah baris sama dengan nilai variabel *jumlahHiddenNeuron* dan jumlah kolom sama dengan nilai variabel *fitur*.
3. Baris 5-12 merupakan proses perulangan untuk melakukan proses *random* dengan batas bawah adalah -1 dan batas atas adalah 1.

1.1.3 Implementasi Proses Inisialisasi Bobot Masukan Menggunakan *Nguyen-Widrow*

```

1 public double[][] bobotNguyenWidrow(double[][] W, int fitur)
2 {
3     double satuPerDInput;
4     satuPerDInput = (double) 1/fitur;
5     double betaNW = Math.pow(0.7, satuPerDInput);
6     double[] gamma = new double[fitur];
7     NguyenWidrow = new double[W.length][W[0].length];
8
9     int a = 0;
10    for (int j = 0; j < W[0].length; j++) {
11        double total = 0;
12        for (int i = 0; i < W.length; i++) {
13            total += Math.pow(W[i][j], 2);
14            gamma[j] = Math.sqrt(total);
15        }
16    }
17 }

```

Kode Program 5.3 Proses Inisialisasi Bobot Masukan Menggunakan *Nguyen-Widrow*

```

15     for (int i = 0; i < W.length; i++) {
16         for (int j = 0; j < W[0].length; j++) {
17             NguyenWidrow[i][j] = (betaNW * W[i][j]) /
                gamma[j];
18         }
19     }
20     return NguyenWidrow;
21 }

```

Kode Program 5.4 Proses Inialisasi Bobot Masukan Menggunakan Nguyen-Widrow (lanjutan)

Penjelasan dari Kode Program 5.3 adalah sebagai berikut.

1. Baris 2-4 merupakan deklarasi dan inialisasi variabel *satuPerDInput* dan *betaNW*.
2. Baris 5-6 merupakan deklarasi dan inialisasi *array* satu dimensi dengan nama *gamma* yang memiliki panjang *array* sama dengan nilai variabel *fitur* dan *array* dua dimensi dengan nama *NguyenWidrow* yang memiliki jumlah baris sama dengan jumlah baris *array W* serta jumlah baris kolom sama dengan jumlah kolom *array W*.
3. Baris 7 merupakan deklarasi variabel *a*.
4. Baris 8-14 merupakan proses perulangan dan perhitungan dari *array gamma*.
5. Baris 15-21 merupakan proses perulangan dan perhitungan dari bobot *random* menggunakan inialisasi Nguyen-Widrow yang akan disimpan di dalam *array NguyenWidrow*.

1.1.4 Implementasi Proses Membuat Nilai Random untuk Bias

```

1 public double[] bias(int jumlahHiddenNeuron) {
2     Random rand = new Random();
3     b = new double[jumlahHiddenNeuron];
4
5     for (int i = 0; i < b.length; i++) {
6         b[i] = rand.nextInt((100 - (0)) + 1) + (0);
7         b[i] /= 100;
8     }
9     return b;
}

```

Kode Program 5.4 Proses Membuat Nilai Random untuk Bias

Penjelasan dari Kode Program 5.4 adalah sebagai berikut.

1. Baris 2 merupakan inialisasi variabel *rand* untuk menyimpan hasil dari proses *random* angka. Pada proses *random* ini menggunakan *library Random*.
2. Baris 3 merupakan inialisasi dan deklarasi *array* satu dimensi dengan nama *b* dengan jumlah baris sama dengan nilai variabel *jumlahHiddenNeuron*.
3. Baris 4-6 merupakan proses perulangan untuk melakukan proses *random* dengan batas bawah adalah 0 dan batas atas adalah 1.

1.1.5 Implementasi Proses Memperbanyak Matriks Bias

```

1 public double[][] memperbanyakBias(int jumData, double[] b) {
2     Random rand = new Random();
3     bias = new double[jumData][bias.length];
4
5     for (int i = 0; i < jumData; i++) {
6         for (int j = 0; j < bias.length; j++) {
7             bias[i][j] = b[j];
8         }
9     }
10    return bias;
11 }

```

Kode Program 5.5 Proses Memperbanyak Matriks Bias

Penjelasan dari Kode Program 5.6 adalah sebagai berikut.

1. Baris 2-3 merupakan proses untuk mengambil nilai *bias* yang diperoleh dari proses *random*.
2. Baris 4-9 merupakan proses perulangan dan perhitungan untuk memperbanyak bias sejumlah variabel *jumData*.

1.1.6 Implementasi Proses Transpose Matriks

```

1 public double[][] transposeMatriks(double[][] matriksSatu) {
2     matriksT = new double[matriksSatu[0].length]
3     [matriksSatu.length];
4
5     for (int i = 0; i < matriksT.length; i++) {
6         for (int j = 0; j < matriksT[0].length; j++) {
7             matriksT[i][j] = matriksSatu[j][i];
8         }
9     }
10    return matriksTranspose;
11 }

```

Kode Program 5.6 Proses Transpose Matriks

Penjelasan dari Kode Program 5.5 adalah sebagai berikut.

1. Baris 2 merupakan inisialisasi dan deklarasi *array matriksT* dengan jumlah baris sama dengan jumlah kolom *array matriks* dan jumlah kolom sama dengan jumlah baris *array matriks*.
2. Baris 3-6 merupakan proses perulangan dan perhitungan untuk melakukan transpose matriks yang nilainya diperoleh dari *array matriks* dan disimpan di dalam *array matriksT*.

1.1.7 Implementasi Proses Menghitung Perkalian Matriks

```

1 public double[][] perkalianMatriks(double[][] matriksSatu,
2 double[][] matriksDua) {
3     matriksPerkalian = new double[matriksSatu.length]

```

Kode Program 5.7 Proses Menghitung Perkalian Matriks

```

3      [matriksDua[0].length];
4          for (int i = 0; i < matriksSatu.length; i++) {
5              for (int j = 0; j < matriksDua[0].length; j++) {
6                  for (int k = 0; k < matriksSatu[0].length;
7                      k++) {
8                      matriksPerkalian[i][j] +=
9                          matriksSatu[i][k] * matriksDua[k][j];
10                 }
11             }
12         }

```

Kode Program 5.7 Proses Menghitung Perkalian Matriks (lanjutan)

Penjelasan dari Kode Program 5.7 adalah sebagai berikut.

1. Baris 2 merupakan inisialisasi dan deklarasi *array matriksPerkalian* dengan jumlah baris sama dengan jumlah baris *array matriksSatu* dan jumlah kolom sama dengan jumlah kolom *array matriksDua*.
2. Baris 4-10 merupakan perulangan untuk melakukan proses perkalian antara *matriksSatu* dan *matriksDua* yang nantinya akan disimpan di dalam array *matriksPerkalian*.

1.1.8 Implementasi Proses Menghitung Penjumlahan Matriks

```

1  public double[][] penjumlahanMatriks(double[][] matriksSatu,
2  double[][] matriksDua) {
3      matriksPenjumlahan = new double[matriksSatu.length]
4      [matriksSatu[0].length];
5
6      for (int i = 0; i < matriksSatu.length; i++) {
7          for (int j = 0; j < matriksSatu[0].length; j++) {
8              matriksPenjumlahan[i][j] = matriksSatu[i][j]
9              + matriksDua[i][j];
10         }
11     }
12
13     return matriksPenjumlahan;
14 }

```

Kode Program 5.8 Proses Menghitung Penjumlahan Matriks

Penjelasan dari Kode Program 5.8 adalah sebagai berikut.

1. Baris 3 merupakan inisialisasi dan deklarasi *array matriksPenjumlahan* dengan jumlah baris sama dengan jumlah baris *array matriksSatu* dan jumlah kolom sama dengan jumlah kolom *array matriksDua*.
2. Baris 4-8 merupakan perulangan untuk melakukan proses penjumlahan antara *matriksSatu* dan *matriksDua* yang nantinya akan disimpan di dalam array *matriksPenjumlahan*.

1.1.9 Implementasi Proses Menghitung H dengan Fungsi Aktivasi

```

1 public double[][] hitungH(int fungsiAktivasi, double[][]
2 Hinit) {
3     H = new double[Hinit.length][Hinit[0].length];
4
5     if (fungsiAktivasi == 0) { // sigmoid biner
6         for (int i = 0; i < H.length; i++) {
7             for (int j = 0; j < H[0].length; j++) {
8                 H[i][j] = 1 / (1 + Math.pow(Math.E, (-1 *
9                     Hinit[i][j])));
10            }
11        }
12    } else if (fungsiAktivasi == 1) { // sin
13        for (int i = 0; i < H.length; i++) {
14            for (int j = 0; j < H[0].length; j++) {
15                H[i][j] = Math.sin(Hinit[i][j]);
16            }
17        }
18    } else if (fungsiAktivasi == 2) { // tribas
19        for (int i = 0; i < H.length; i++) {
20            for (int j = 0; j < H[0].length; j++) {
21                if ((1 - Math.abs(Hinit[i][j])) <= 1 &&
22                    (1 - Math.abs(Hinit[i][j])) >= -1) {
23                    H[i][j] = 1 - Math.abs(Hinit[i][j]);
24                } else {
25                    H[i][j] = 0;
26                }
27            }
28        }
29    } else if (fungsiAktivasi == 3) { // radbas
30        for (int i = 0; i < H.length; i++) {
31            for (int j = 0; j < H[i].length; j++) {
32                double pows = Math.pow(-Hinit[i][j],
33                    2);
34                H[i][j] = Math.exp(pows);
35            }
36        }
37    }
38    return H;
39 }

```

Kode Program 5.9 Proses Menghitung H dengan Fungsi Aktivasi

Penjelasan dari Kode Program 5.9 adalah sebagai berikut.

1. Baris 3 merupakan inisialisasi dan deklarasi *array H* dengan panjang baris dan panjang kolom sama dengan *array Hinit*.
2. Baris 4 merupakan proses seleksi jika nilai dari variabel *fungsiAktivasi* = 0.
3. Baris 5-9 merupakan proses perulangan dan perhitungan fungsi aktivasi dari *array Hinit*.
4. Baris 10 merupakan proses seleksi jika nilai dari variabel *fungsiAktivasi* = 1.
5. Baris 11-15 merupakan proses perulangan dan perhitungan fungsi aktivasi dari *array Hinit*.
6. Baris 16 merupakan proses seleksi jika nilai dari variabel *fungsiAktivasi* = 2.

7. Baris 17-25 merupakan proses perulangan dan perhitungan fungsi aktivasi dari array *Hinit*.
8. Baris 16 merupakan proses seleksi jika nilai dari variabel *fungsiAktivasi* = 3.
9. Baris 27-33 merupakan proses perulangan dan perhitungan fungsi aktivasi dari array *Hinit*.

1.1.10 Implementasi Proses Menghitung *Inverse* Matriks

```

1 import org.apache.commons.math3.linear.Array2DRowRealMatrix;
2 import org.apache.commons.math3.linear.MatrixUtils;
3 import org.apache.commons.math3.linear.RealMatrix;

4 public double[][] inverseMatriks(double[][] matriks) {
5     RealMatrix a = new Array2DRowRealMatrix(matriks);
6     matriksInverse = MatrixUtils.inverse(a).getData();
7     return matriksInverse;
8 }

```

Kode Program 5.10 Proses Menghitung *Inverse* Matriks

Penjelasan dari Kode Program 5.10 adalah sebagai berikut.

1. Baris 1-3 merupakan proses *import library* matriks.
2. Baris 5 merupakan proses deklarasi dan inialisasi variabel *a* untuk mengambil data matriks yang ingin di-*inverse* yaitu array *matriks*.
3. Baris 6 merupakan proses *inverse* menggunakan *library*.

1.1.11 Implementasi Proses Denormalisasi Data

```

1 public double[][] denormalisasi(double[][] matriksSatu) {
2     denormalisasi = new double[matriksSatu.length]
3     [matriksSatu[0].length];

4     for (int i = 0; i < denormalisasi.length; i++) {
5         for (int j = 0; j < denormalisasi[0].length; j++)
6         {
7             denormalisasi[i][j] = (matriksSatu[i][j] *
8             (this.max - this.min)) + this.min;
9         }
10    }
11    return denormalisasi;
12 }

```

Kode Program 5.11 Proses Denormalisasi Data

Penjelasan dari Kode Program 5.11 adalah sebagai berikut.

1. Baris 2 merupakan inialisasi dan deklarasi *array denormalisasi* dengan jumlah baris sama dengan jumlah baris *array matriksSatu* dan jumlah kolom sama dengan jumlah kolom *array matriksSatu*.
2. Baris 4-9 merupakan perulangan untuk melakukan denormalisasi dari *array matriksSatu* yang nantinya akan disimpan di dalam array *denormalisasi*.

1.1.12 Implementasi Proses Perhitungan Nilai Evaluasi

```

1 public double rmse (double[][] data, double[][]
  denormalisasi) {
2     double jumlah = 0;
3     double n_data = denormalisasi.length;

4     for (int i = 0; i < denormalisasi.length; i++) {
5         for (int j = 0; j < denormalisasi[0].length; j++)
6         {
7             jumlah += Math.pow(denormalisasi[i][j] -
8                 data[i][j], 2);
9         }
10    }
11    double rerata = jumlah / n_data;
12    rmse = Math.sqrt(rerata);
13    return rmse;
14 }

```

Kode Program 5.12 Proses Perhitungan Nilai Evaluasi

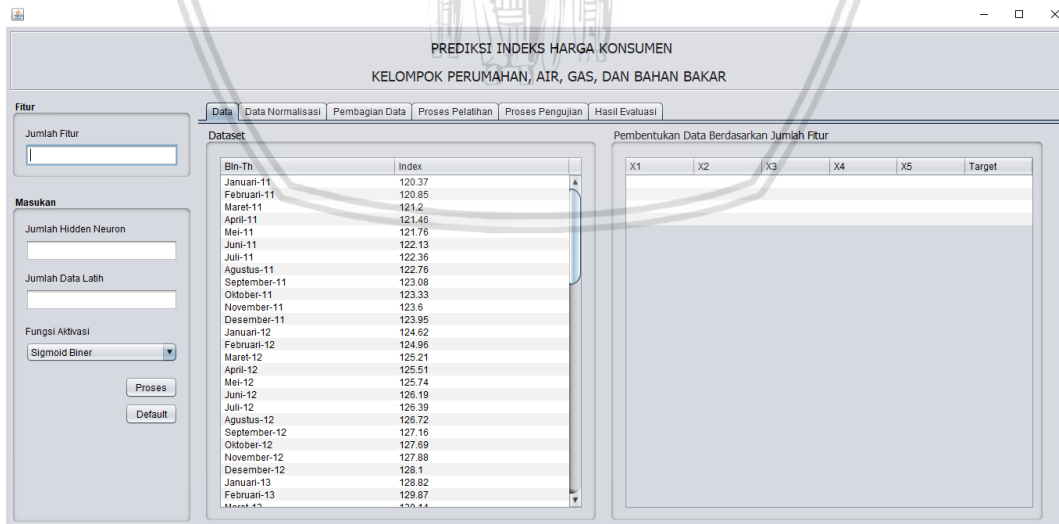
Penjelasan dari Kode Program 5.12 adalah sebagai berikut.

1. Baris 2-3 merupakan proses deklarasi dan inisialisasi variabel *jumlah* dan *n_data*.
2. Baris 4-10 merupakan perulangan untuk melakukan proses perhitungan hasil evaluasi menggunakan RMSE.

1.2 Implementasi User Interface

1.2.1 Implementasi Interface Tampilan Halaman Utama

Implementasi *interface* halaman utama sistem ditunjukkan pada Gambar 5.1.



Gambar 5.1 Implementasi Interface Tampilan Halaman Utama

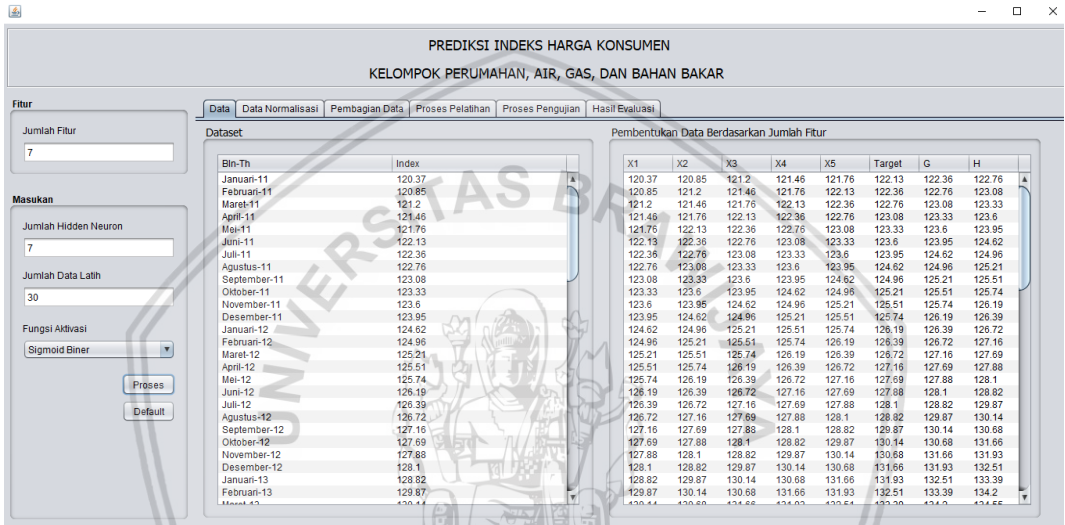
Pada implementasi interface tampilan awal sistem terdapat panel yang berisi beberapa kolom yang berfungsi untuk pengguna memasukkan jumlah fitur, jumlah *hidden neuron*, jumlah data latihan, serta fungsi aktivasi yang akan digunakan untuk



melakukan proses perhitungan. Pada panel Masukan terdapat 2 tombol yaitu tombol Proses dan Default. Tombol Proses adalah tombol yang berfungsi untuk melakukan proses perhitungan metode ELM sedangkan tombol Default berfungsi untuk mengeset nilai-nilai masukan yang sudah ditentukan sebelumnya. Lalu terdapat panel yang berisi beberapa tab yaitu tab Data, tab Data Normalisasi, tab Pembagian Data, tab Proses Pelatihan, tab Proses Pengujian, dan tab Hasil Evaluasi.

1.2.2 Implementasi Interface Tampilan Menu Tab Data

Implementasi *interface* Tampilan Menu Tab Data ditunjukkan pada Gambar 5.2.



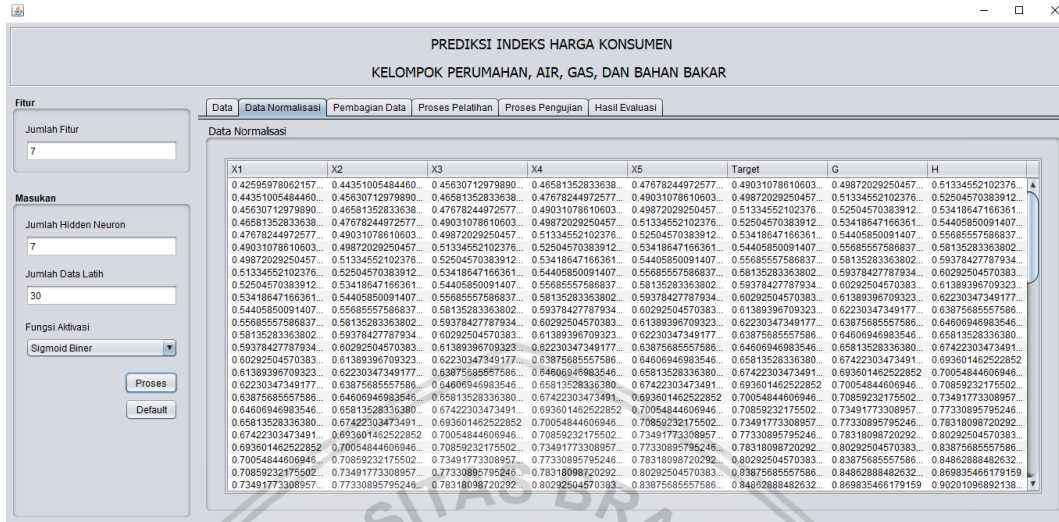
Gambar 5.2 Implementasi Interface Tampilan Menu Tab Data

Pada menu tab Data, terdapat 2 tabel, yaitu tabel Dataset yang diperoleh dengan cara mengambil data IHK pada *file excel* dan tabel Pembentukan Data berdasarkan jumlah fitur yang dibentuk dari nilai masukan pada kolom Jumlah Fitur. Pada Gambar 5.2 nilai masukan pada kolom Jumlah Fitur adalah 7, maka hasil dari pembentukan fitur adalah data yang memiliki 5 fitur yaitu X1, X2, X3, X4, X5, X6, dan X7. Implementasi Interface pada menu tab Data Normalisasi, Pembagian Data, Proses Pelatihan, Proses Pengujian, dan Hasil Evaluasi akan dijelaskan pada penjelasan selanjutnya.



1.2.3 Implementasi Interface Tampilan Menu Tab Data Normalisasi

Implementasi *interface* Tampilan Menu Tab Data Normalisasi ditunjukkan pada Gambar 5.3.

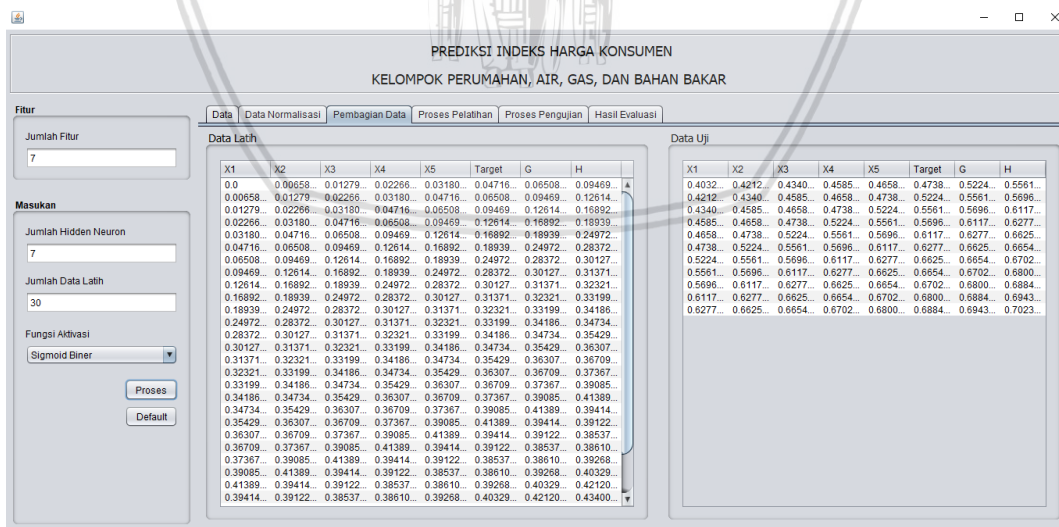


Gambar 5.3 Implementasi *Interface* Tampilan Menu Tab Data Normalisasi

Pada menu tab Data Normalisasi terdapat tabel yang berisi nilai dalam bentuk normalisasi yang diperoleh dari tabel Pembentukan Data Berdasarkan Jumlah Fitur pada tab Data.

1.2.4 Implementasi Interface Tampilan Menu Tab Pembagian Data

Implementasi *interface* Tampilan Menu Tab Pembagian Data ditunjukkan pada Gambar 5.4.



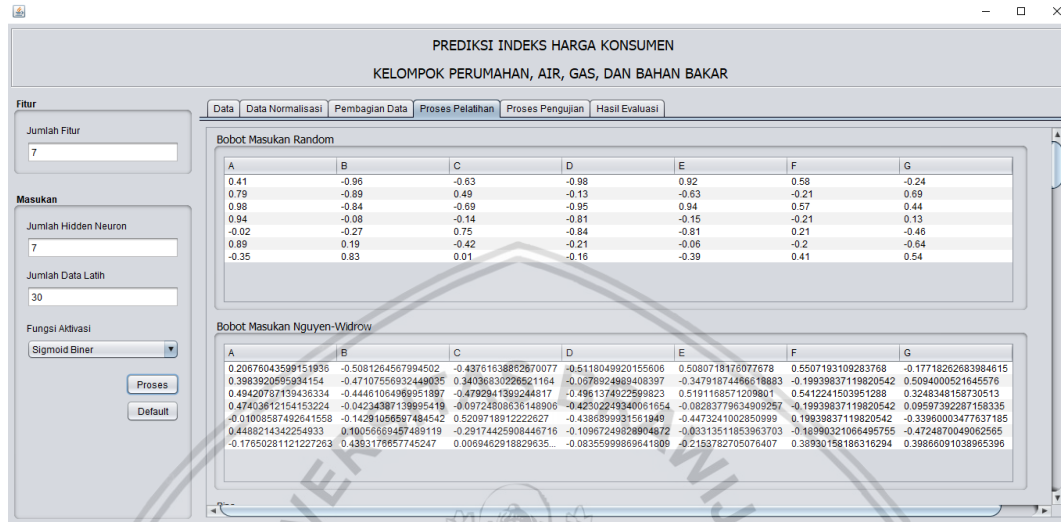
Gambar 5.4 Implementasi *Interface* Tampilan Menu Tab Pembagian Data

Pada menu tab Pembagian Data terdapat 2 tabel, yaitu tabel Data Lath yang berisi data latih yang digunakan untuk melakukan proses pelatihan yang akan ditampilkan pada menu tab Proses Pelatihan. Jumlah data latih berdasarkan nilai

masukannya pada kolom Jumlah Data Latih yang dimasukkan oleh pengguna. Dan tabel Data Uji yang digunakan untuk proses pengujian.

1.2.5 Implementasi *Interface* Tampilan Menu Tab Proses Pelatihan

Implementasi *interface* Tampilan Menu Tab Proses Pelatihan ditunjukkan pada Gambar 5.5.

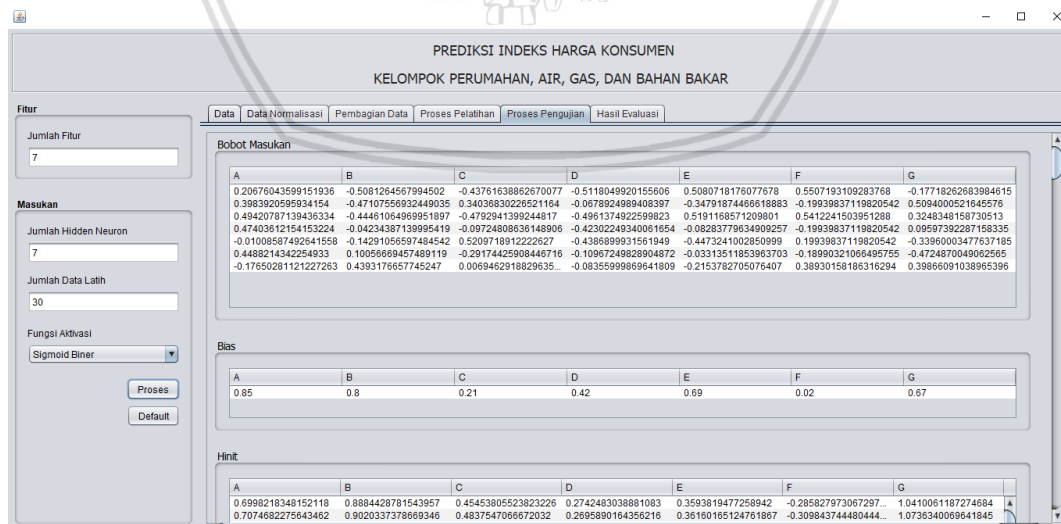


Gambar 5.5 Implementasi *Interface* Tampilan Menu Tab Proses Pelatihan

Pada tab menu Proses Pelatihan terdapat tabel-tabel yang merupakan hasil perhitungan dari proses pelatihan.

1.2.6 Implementasi *Interface* Tampilan Menu Tab Proses Pengujian

Implementasi *interface* Tampilan Menu Tab Proses Pengujian ditunjukkan pada Gambar 5.6.

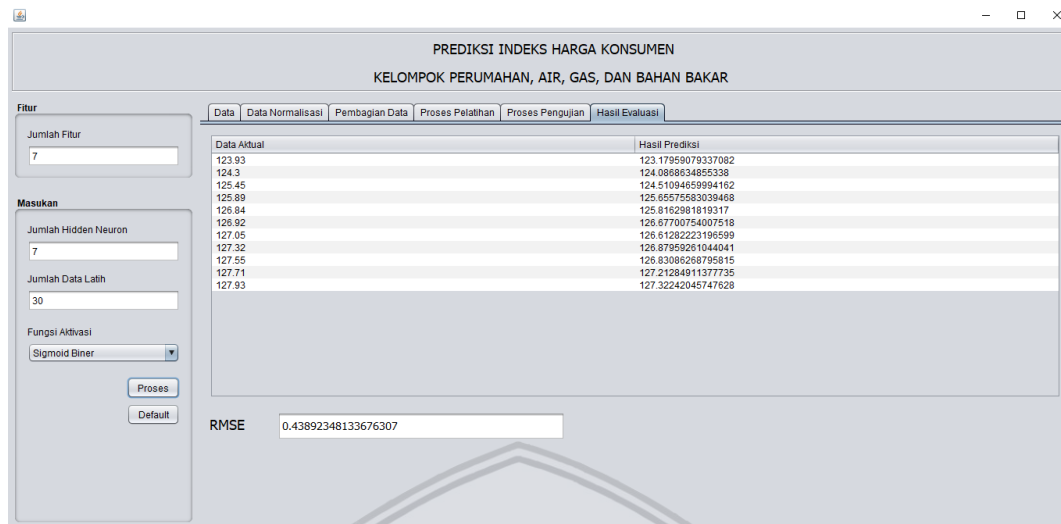


Gambar 5.6 Implementasi *Interface* Tampilan Menu Tab Proses Pengujian

Pada tab menu Proses Pengujian terdapat tabel-tabel yang merupakan hasil perhitungan dari proses pengujian.



1.2.7 Implementasi *Interface* Tampilan Menu Tab Hasil Evaluasi



Gambar 5.7 Implementasi *Interface* Tampilan Menu Tab Hasil Evaluasi

Pada tab menu Hasil Evaluasi terdapat tabel yang berisi data aktual dan hasil prediksi. Kolom data aktual adalah data uji dalam bentuk aktual sedangkan kolom hasil prediksi adalah nilai prediksi yang diperoleh dari proses pelatihan dan pengujian yang ditampilkan pada tab Proses Pelatihan dan tab Proses Pengujian. Selanjutnya terdapat kolom RMSE yang menampilkan hasil evaluasi antara nilai prediksi dan nilai data aktual.

BAB 6 PENGUJIAN DAN ANALISIS

Bab ini akan menjelaskan tentang pengujian yang dilakukan dan hasil yang didapatkan. Pengujian pada penelitian menggunakan data Indeks Harga Konsumen kelompok perumahan, air, gas, dan bahan bakar periode Januari 2011 s.d. Desember 2017. Pengujian tersebut meliputi pengujian jumlah fitur, pengujian jumlah data latih, pengujian jumlah *hidden neuron*, pengujian fungsi aktivasi dan pengujian perbandingan metode.

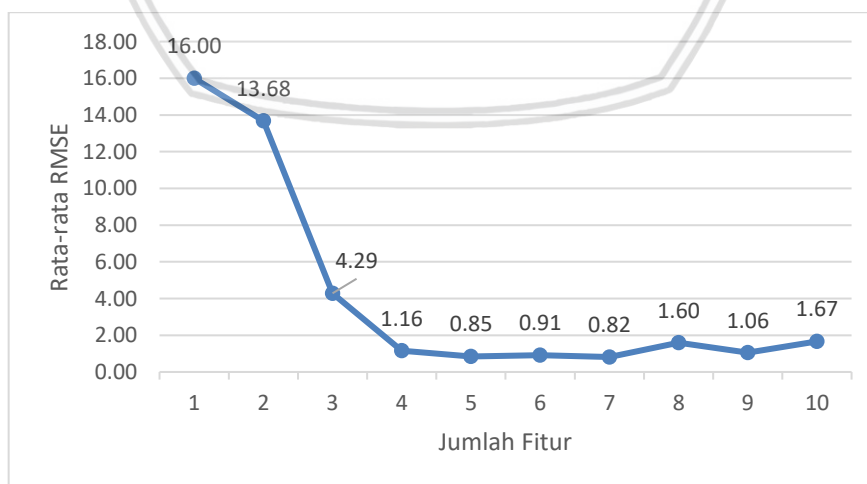
1.1 Pengujian Jumlah Fitur

Pengujian jumlah fitur yang dilakukan yaitu 1, 2, 3, 4, 5, 6, 7, 8, 9, dan 10. Jumlah fitur ini merepresentasikan jumlah bulan sebelum bulan target yang akan diprediksi. Pengujian jumlah fitur dilakukan sebanyak 10 kali percobaan dengan jumlah data latih sebanyak 20, jumlah *hidden neuron* adalah 6, dan fungsi aktivasi adalah sigmoid biner. Hasil pengujian jumlah fitur ditunjukkan pada Tabel 6.1.

Tabel 6.1 Tabel Pengujian Jumlah Fitur

| Jumlah Fitur | RMSE Percobaan ke- | | | | | | | | | | RMSE |
|--------------|--------------------|------|------|------|-------|-------|-------|-------|-------|------|-------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| 1 bulan | 35,0 | 17,4 | 10,9 | 23,1 | 8,15 | 13,62 | 12,83 | 14,45 | 15,35 | 9,23 | 16,00 |
| 2 bulan | 9,3 | 14,3 | 4,0 | 14,5 | 24,45 | 20,88 | 8,61 | 20,07 | 12,13 | 8,58 | 13,68 |
| 3 bulan | 5,91 | 5,08 | 4,51 | 4,77 | 0,80 | 2,20 | 8,13 | 4,33 | 5,36 | 1,78 | 4,29 |
| 4 bulan | 0,96 | 0,58 | 0,58 | 2,94 | 2,50 | 0,52 | 0,55 | 1,13 | 1,35 | 0,51 | 1,16 |
| 5 bulan | 0,66 | 1,69 | 0,71 | 0,62 | 0,59 | 0,52 | 1,10 | 0,83 | 1,05 | 0,77 | 0,85 |
| 6 bulan | 1,42 | 0,85 | 0,58 | 1,16 | 0,63 | 1,21 | 0,93 | 0,91 | 0,68 | 0,74 | 0,91 |
| 7 bulan | 0,87 | 0,60 | 0,41 | 0,41 | 0,53 | 0,69 | 2,37 | 0,50 | 1,03 | 0,72 | 0,82 |
| 8 bulan | 1,30 | 1,11 | 1,18 | 0,93 | 1,42 | 1,07 | 2,36 | 1,08 | 0,52 | 5,00 | 1,60 |
| 9 bulan | 0,57 | 1,83 | 1,12 | 1,01 | 1,04 | 0,32 | 0,72 | 1,59 | 1,24 | 1,15 | 1,06 |
| 10 bulan | 1,20 | 0,96 | 1,48 | 2,50 | 1,67 | 2,02 | 0,95 | 1,54 | 3,02 | 1,33 | 1,67 |

Grafik hasil pengujian jumlah fitur dapat dilihat pada Gambar 6.1.



Gambar 6.1 Grafik Hasil Pengujian Jumlah Fitur

Berdasarkan hasil pengujian jumlah fitur yang ditunjukkan pada Tabel 6.1, diperoleh rata-rata nilai RMSE minimum sebesar 0,82 dengan jumlah fitur = 7.



Masing-masing jumlah fitur memiliki nilai rata-rata RMSE yang berbeda, hal ini menunjukkan bahwa jumlah fitur memiliki pengaruh terhadap nilai RMSE dalam melakukan prediksi dengan data historis. Pada Gambar 6.1 terlihat bahwa semakin sedikit jumlah fitur yang digunakan maka nilai RMSE akan semakin besar, sedangkan semakin bertambah jumlah fitur maka nilai RMSE akan semakin kecil. Hal ini terjadi karena jika jumlah fitur dalam melakukan prediksi terlalu sedikit menyebabkan pola dari data sebelumnya tidak bisa dikenali terlalu jelas sehingga nilai RMSE yang dihasilkan akan semakin tinggi. Tetapi jumlah fitur yang terlalu banyak juga tidak menjamin nilai RMSE akan semakin kecil, selain dapat mengurangi jumlah data latih, fitur yang terlalu banyak dapat menyebabkan pola yang dikenali dari data sebelumnya akan terlalu luas sehingga dapat menyebabkan hasil prediksi tidak akurat.

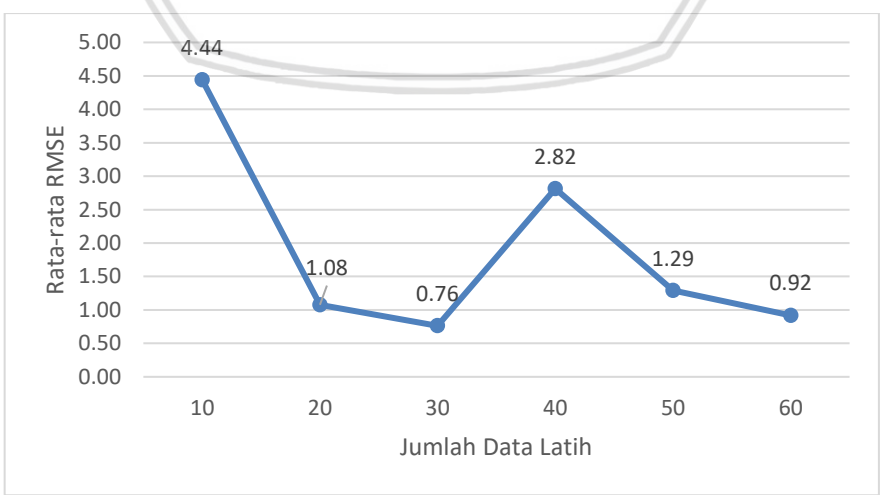
1.2 Pengujian Jumlah Data Latih

Pengujian jumlah data latih yang dilakukan adalah 10, 20, 30, 40, 50, dan 60. Pengujian jumlah data latih dilakukan sebanyak 10 kali percobaan menggunakan jumlah fitur dengan nilai RMSE paling minimum yang diperoleh dari pengujian jumlah fitur yaitu 7, jumlah *hidden neuron* adalah 6, dan fungsi aktivasi adalah sigmoid biner. Hasil pengujian jumlah data latih ditunjukkan pada Tabel 6.2.

Tabel 6.2 Tabel Pengujian Jumlah Data Latih

| Jumlah Data Latih | RMSE Percobaan ke- | | | | | | | | | | RMSE |
|-------------------|--------------------|------|------|------|------|------|------|------|------|------|------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| 10 | 4,79 | 4,65 | 2,77 | 4,98 | 4,88 | 4,63 | 4,08 | 4,83 | 4,64 | 4,15 | 4,44 |
| 20 | 1,42 | 0,92 | 1,31 | 0,78 | 1,08 | 2,24 | 1,24 | 0,36 | 0,63 | 0,81 | 1,08 |
| 30 | 0,89 | 0,77 | 0,66 | 0,67 | 0,61 | 0,68 | 0,82 | 0,95 | 0,83 | 0,76 | 0,76 |
| 40 | 3,04 | 3,14 | 3,37 | 3,29 | 3,20 | 3,26 | 3,47 | 1,44 | 1,80 | 2,15 | 2,82 |
| 50 | 1,46 | 1,50 | 1,05 | 1,60 | 1,20 | 0,99 | 1,15 | 0,72 | 2,23 | 1,00 | 1,29 |
| 60 | 1,23 | 1,14 | 0,44 | 0,92 | 1,47 | 1,20 | 0,44 | 0,97 | 0,78 | 0,56 | 0,92 |

Grafik hasil pengujian jumlah data latih dapat dilihat pada Gambar 6.2.

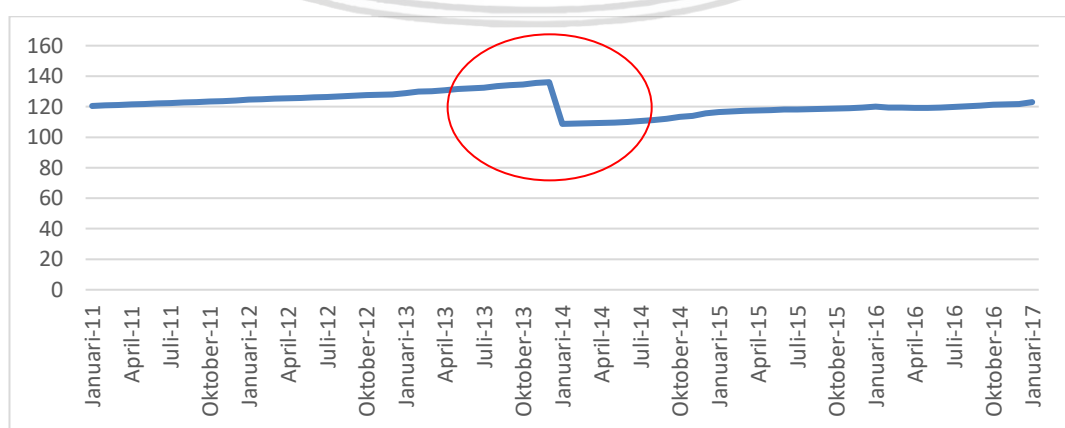


Gambar 6.2 Grafik Hasil Pengujian Jumlah Data Latih

Berdasarkan hasil pengujian jumlah data latih yang ditunjukkan pada Tabel 6.2, diperoleh rata-rata nilai RMSE minimum sebesar 0,76 dengan jumlah data latih = 30. Masing-masing jumlah data latih memiliki nilai rata-rata RMSE yang berbeda, hal ini menunjukkan bahwa jumlah data latih memiliki pengaruh terhadap tingkat *error*. Jumlah data latih yang digunakan untuk melakukan pelatihan jika terlalu sedikit dapat menyebabkan tingkat *error* yang diperoleh semakin besar karena variasi data yang dikenali semakin sedikit dan data yang digunakan untuk proses pembelajaran pun berkurang sehingga keluaran pada *ouput layer* akan semakin menjauhi target tetapi semakin banyak jumlah data latih pun belum tentu akan menghasilkan nilai prediksi yang semakin akurat.

Pada Gambar 6.2 dapat dilihat bahwa ketika jumlah data latih berjumlah 10, 20, dan 30 secara berurutan nilai RMSE mengalami penurunan tetapi mengalami kenaikan kembali ketika jumlah data latih berjumlah 40, lalu kembali mengalami penurunan ketika jumlah data latih berjumlah 50 dan 60. Hal ini terjadi karena ketika data latih berjumlah 10, 20, dan 30, data yang digunakan adalah data periode Januari 2014 s.d. Januari 2017. Pola data yang terbentuk ketika periode tersebut adalah cenderung naik secara konstan. Namun, ketika jumlah data latih berjumlah 40, data yang digunakan untuk proses pelatihan adalah data periode Maret 2013 s.d. Januari 2017. Berdasarkan metadata Bank Indonesia terjadi perubahan cakupan data di tahun 2013 dan tahun 2014. Pada tahun 2013 cakupan data pada IHK mencakup 66 kota di seluruh Indonesia yang meliputi 774 komoditas barang dan jasa, sedangkan pada tahun 2014 mencakup 82 kota di seluruh Indonesia yang meliputi 859 komoditas barang dan jasa. Hal tersebut mengakibatkan pola data yang dibentuk oleh data latih berubah karena terdapat perbedaan pola data di bulan Desember 2013 dan Januari 2014 yang diakibatkan oleh perbedaan cakupan data IHK seperti yang ditunjukkan pada Gambar 6.3.

Karena perubahan pada pola data tersebut jaringan akan mengenali pola data baru yang terbentuk, hal tersebut menyebabkan nilai RMSE semakin meningkat, lalu kembali mengalami penurunan ketika jumlah data latih berjumlah 50 dan 60 karena pola data yang terbentuk ketika jumlah data latih berjumlah tersebut sudah dikenali oleh jaringan.



Gambar 6.3 Grafik Pola Data

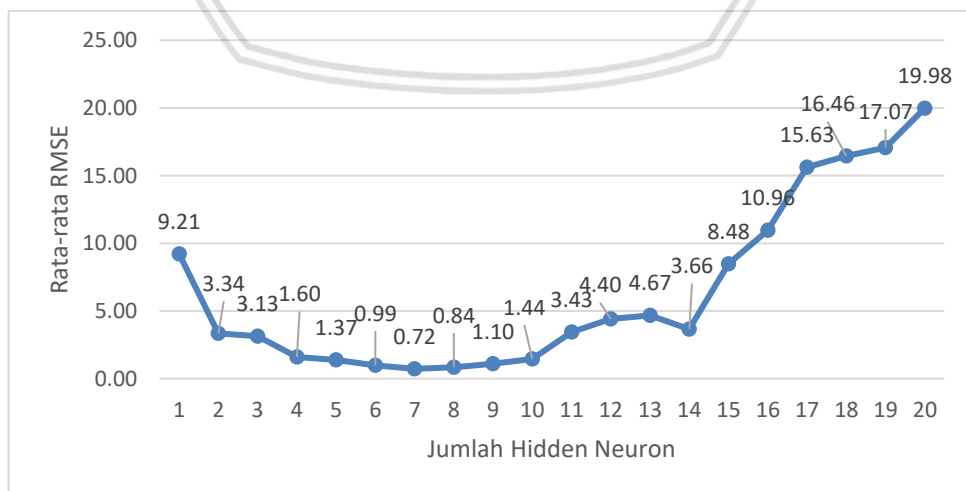
1.3 Pengujian Jumlah *Hidden Neuron*

Pengujian jumlah *hidden neuron* yang dilakukan adalah 1 hingga 20. Pengujian jumlah *hidden neuron* dilakukan sebanyak 10 kali percobaan menggunakan jumlah fitur dengan nilai RMSE paling minimum yang diperoleh dari pengujian jumlah fitur yaitu 7, jumlah data latih dengan nilai RMSE paling minimum yang diperoleh dari pengujian jumlah data latih yaitu 30, dan fungsi aktivasi adalah sigmoid biner. Hasil pengujian jumlah *hidden neuron* ditunjukkan pada Tabel 6.3.

Tabel 6.3 Tabel Pengujian Jumlah *Hidden Neuron*

| Jumlah <i>Hidden Neuron</i> | RMSE Percobaan ke- | | | | | | | | | | RMSE |
|-----------------------------|--------------------|-------|------|------|------|-------|-------|------|------|-------|-------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| 1 | 7,65 | 9,75 | 6,64 | 13,1 | 7,45 | 9,31 | 7,66 | 7,59 | 9,49 | 13,48 | 9,21 |
| 2 | 0,64 | 3,42 | 4,20 | 1,61 | 4,58 | 1,29 | 3,50 | 4,46 | 5,65 | 4,10 | 3,34 |
| 3 | 2,55 | 1,81 | 4,26 | 0,87 | 3,24 | 5,08 | 4,36 | 1,98 | 3,26 | 3,91 | 3,13 |
| 4 | 0,92 | 3,32 | 1,65 | 0,61 | 2,04 | 0,68 | 1,19 | 1,23 | 2,94 | 1,40 | 1,60 |
| 5 | 0,68 | 2,45 | 0,57 | 1,56 | 0,49 | 3,61 | 1,39 | 0,85 | 1,51 | 0,62 | 1,37 |
| 6 | 2,25 | 0,85 | 1,32 | 1,16 | 0,70 | 0,66 | 0,92 | 0,52 | 0,51 | 1,02 | 0,99 |
| 7 | 0,54 | 0,44 | 0,83 | 0,80 | 0,42 | 1,15 | 0,42 | 1,16 | 0,41 | 1,04 | 0,72 |
| 8 | 1,64 | 0,43 | 1,44 | 0,80 | 0,95 | 0,64 | 0,61 | 1,05 | 0,44 | 0,38 | 0,84 |
| 9 | 1,31 | 0,51 | 1,49 | 1,75 | 0,66 | 1,27 | 0,76 | 0,37 | 1,52 | 1,36 | 1,10 |
| 10 | 2,89 | 1,98 | 0,53 | 1,43 | 1,90 | 1,63 | 1,58 | 0,67 | 0,71 | 1,10 | 1,44 |
| 11 | 1,97 | 1,88 | 1,95 | 7,16 | 0,63 | 4,40 | 12,16 | 1,19 | 0,94 | 2,06 | 3,43 |
| 12 | 1,63 | 14,60 | 2,62 | 1,66 | 6,58 | 8,25 | 2,22 | 2,93 | 1,09 | 2,46 | 4,40 |
| 13 | 10,3 | 5,59 | 0,63 | 1,92 | 3,02 | 10,16 | 2,99 | 2,60 | 7,60 | 1,95 | 4,67 |
| 14 | 1,47 | 2,86 | 4,06 | 3,89 | 5,04 | 1,67 | 4,55 | 3,96 | 1,63 | 7,48 | 3,66 |
| 15 | 7,67 | 16,3 | 4,71 | 1,68 | 2,39 | 4,55 | 11,2 | 3,39 | 17,1 | 15,9 | 8,48 |
| 16 | 5,98 | 17,4 | 17,5 | 1,39 | 11,3 | 4,19 | 21,6 | 12,2 | 11,7 | 6,19 | 10,96 |
| 17 | 14,1 | 17,5 | 24,8 | 28,7 | 21,6 | 2,63 | 9,92 | 16,2 | 15,4 | 5,45 | 15,63 |
| 18 | 24,5 | 6,5 | 12,2 | 28,8 | 16,2 | 7,27 | 16,3 | 23,0 | 12,4 | 17,5 | 16,46 |
| 19 | 7,31 | 26,7 | 19,8 | 16,2 | 20,4 | 20,5 | 7,99 | 19,4 | 15,3 | 17,0 | 17,07 |
| 20 | 32,4 | 22,3 | 15,8 | 19,0 | 20,6 | 10,9 | 29,4 | 11,5 | 18,6 | 19,2 | 19,98 |

Grafik hasil pengujian jumlah *hidden neuron* dapat dilihat pada Gambar 6.4.



Gambar 6.4 Grafik Hasil Pengujian Jumlah *Hidden Neuron*



Berdasarkan hasil pengujian jumlah *hidden neuron* yang ditunjukkan pada Tabel 6.3, diperoleh rata-rata nilai RMSE minimum sebesar 0,72 dengan jumlah *hidden neuron* = 7. Berdasarkan grafik pengujian jumlah *hidden neuron* yang ditunjukkan pada Gambar 6.4, dapat dilihat bahwa nilai RMSE cenderung mengalami penurunan ketika *hidden neuron* berjumlah 1 sampai 7, tetapi mengalami kenaikan ketika *hidden neuron* berjumlah 8, 9, 10 dan seterusnya. Hal ini menunjukkan bahwa perbedaan kecil pada jumlah *hidden neuron* memiliki pengaruh terhadap nilai RMSE dan semakin besar jumlah *hidden neuron* maka proses perhitungan semakin kompleks sehingga berpengaruh terhadap nilai RMSE.

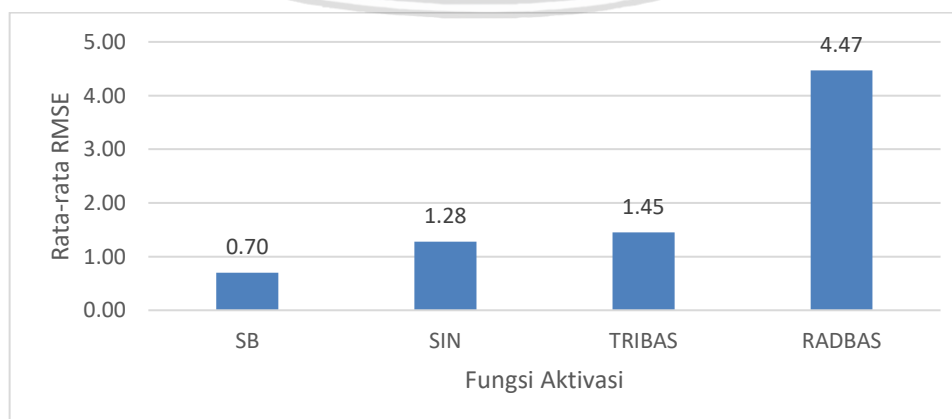
1.4 Pengujian Fungsi Aktivasi

Pengujian fungsi aktivasi yang dilakukan meliputi 4 fungsi aktivasi, yaitu fungsi aktivasi Sigmoid Biner (SB), Sin (SIN), Tringular Basis (TRIBAS), dan Radial Basis (RADBAS). Fungsi aktivasi ini diperoleh berdasarkan penelitian yang dilakukan oleh Bahiuddin et al, pada tahun 2017 yang melakukan perbandingan antara 5 fungsi aktivasi. Diperoleh hasil 4 fungsi aktivasi memiliki nilai RMSE yang tidak terlalu jauh yaitu fungsi aktivasi Sigmoid Biner, Sin, Tringular Basis, dan Radial Basis. Pengujian fungsi aktivasi dilakukan sebanyak 10 kali percobaan menggunakan jumlah fitur dengan nilai RMSE paling minimum yang diperoleh dari pengujian jumlah fitur yaitu 7, jumlah data latih dengan nilai RMSE paling minimum yang diperoleh dari pengujian jumlah data latih yaitu 30, jumlah *hidden neuron* dengan nilai RMSE paling minimum yang diperoleh dari pengujian jumlah *hidden neuron* yaitu 7. Hasil pengujian jumlah fitur ditunjukkan pada Tabel 6.4.

Tabel 6.4 Tabel Pengujian Fungsi Aktivasi

| Fungsi Aktivasi | RMSE Percobaan ke- | | | | | | | | | | RMSE |
|-----------------|--------------------|------|------|------|------|------|------|------|------|------|------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| SB | 0,59 | 1,11 | 0,51 | 0,52 | 0,66 | 0,63 | 0,47 | 0,57 | 0,44 | 1,50 | 0,70 |
| SIN | 1,15 | 0,55 | 1,86 | 1,01 | 1,39 | 1,61 | 2,33 | 0,97 | 1,13 | 0,79 | 1,28 |
| TRIBAS | 4,54 | 0,58 | 2,18 | 0,57 | 0,37 | 3,29 | 0,58 | 0,54 | 0,61 | 1,25 | 1,45 |
| RADBAS | 13,1 | 4,53 | 8,92 | 5,14 | 4,89 | 2,26 | 1,29 | 2,71 | 1,11 | 0,72 | 4,47 |

Grafik hasil pengujian fungsi aktivasi dapat dilihat pada Gambar 6.5.



Gambar 6.5 Grafik Hasil Pengujian Fungsi Aktivasi

Berdasarkan hasil pengujian fungsi aktivasi yang ditunjukkan pada Tabel 6.4, dapat dilihat bahwa jenis fungsi aktivasi yang digunakan untuk melakukan prediksi memiliki pengaruh terhadap nilai RMSE. Fungsi aktivasi yang cocok adalah adalah Sigmoid Biner (SB) Dengan diperoleh rata-rata nilai RMSE minimum sebesar 0,70. Hal ini disebabkan karena rentang nilai ouput dari fungsi aktivasi sigmoid biner sama dengan rentang nilai neuron masukan yaitu [0,1].

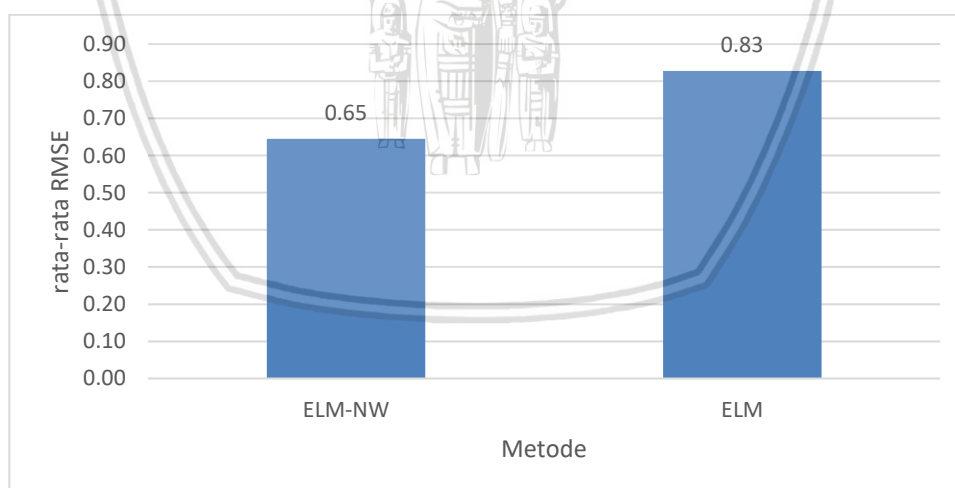
1.5 Pengujian Perbandingan Metode

Pengujian perbandingan metode dilakukan antara metode *Extreme Learning Machine* menggunakan inisialisasi bobot secara *random* dan inisialisasi bobot menggunakan Nguyen-Widrow. Pengujian perbandingan metode dilakukan sebanyak 10 kali percobaan menggunakan jumlah fitur dengan nilai RMSE paling minimum yang diperoleh dari pengujian jumlah fitur yaitu 7, jumlah data latih dengan nilai RMSE paling minimum yang diperoleh dari pengujian jumlah data latih yaitu 30, jumlah *hidden neuron* dengan nilai RMSE paling minimum yang diperoleh dari pengujian jumlah *hidden neuron* yaitu 7. Hasil pengujian perbandingan metode ditunjukkan pada Tabel 6.5.

Tabel 6.5 Tabel Pengujian Perbandingan Metode

| Metode | RMSE Percobaan ke- | | | | | | | | | | RMSE |
|--------|--------------------|------|------|------|------|------|------|------|------|------|------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| ELM-NW | 0,72 | 0,73 | 0,78 | 0,57 | 0,56 | 0,85 | 0,43 | 0,87 | 0,39 | 0,56 | 0,65 |
| ELM | 0,79 | 1,17 | 0,67 | 0,44 | 0,52 | 0,86 | 0,59 | 0,44 | 2,10 | 0,70 | 0,83 |

Grafik hasil pengujian perbandingan metode dapat dilihat pada Gambar 6.6.



Gambar 6.6 Grafik Hasil Pengujian Perbandingan Metode

Berdasarkan hasil pengujian perbandingan metode yang ditunjukkan pada Tabel 6.5, dapat dilihat bahwa melakukan inisialisasi bobot secara *random* dan menggunakan Nguyen-Widrow memiliki perbedaan pada nilai RMSE. Grafik yang ditunjukkan pada Gambar 6.6, grafik menunjukkan bahwa metode inisialisasi bobot menggunakan Nguyen-Widrow memiliki nilai RMSE yang lebih baik dibandingkan dengan inisialisasi bobot secara *random* yaitu sebesar 0,65.



BAB 7 PENUTUP

1.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan mengenai prediksi Indeks Harga Konsumen kelompok perumahan, air, gas, dan, bahan bakar didapatkan kesimpulan sebagai bahwa metode *Extreme Learning Machine* dengan inialisasi bobot *Nguyen-Widrow* dapat diimplementasikan dalam melakukan prediksi terhadap Indeks Harga Konsumen kelompok perumahan, air, listrik, gas, dan bahan bakar. Berdasarkan 5 pengujian yang telah dilakukan yaitu pengujian jumlah fitur, pengujian jumlah data latih, pengujian jumlah *hidden neuron*, pengujian fungsi aktivasi, dan pengujian perbandingan metode didapatkan hasil sebagai berikut.

1. Pengujian jumlah fitur yang telah dilakukan yaitu 1 sampai dengan 10 fitur yang masing-masing merepresentasikan jumlah bulan sebelum bulan target yang akan diprediksi. Setelah dilakukan masing-masing 10 kali uji coba, jumlah fitur terbaik yang didapatkan adalah 7 fitur dengan nilai rata-rata RMSE adalah 0,82.
2. Pengujian jumlah data latih yang telah dilakukan yaitu 10, 20, 30, 40, 50, dan 60 data latih. Setelah dilakukan masing-masing 10 kali uji coba, jumlah data latih terbaik yang didapatkan adalah 30 data latih dengan nilai rata-rata RMSE adalah 0,76.
3. Pengujian jumlah *hidden neuron* yang telah dilakukan yaitu berjumlah 1 hingga 20 *hidden neuron*. Setelah dilakukan masing-masing 10 kali uji coba, jumlah *hidden neuron* terbaik yang didapatkan adalah 7 *hidden neuron* dengan nilai rata-rata RMSE adalah 0,72.
4. Pengujian fungsi aktivasi yang telah dilakukan yaitu terdiri dari 4 fungsi jenis aktivasi yaitu fungsi Sigmoid Biner, fungsi Sin, fungsi Tringular Basis, dan fungsi Radial Basis. Setelah dilakukan masing-masing 10 kali uji coba, jumlah fungsi aktivasi yang paling cocok untuk melakukan prediksi adalah fungsi Sigmoid Biner dengan nilai rata-rata RMSE adalah 0,70.
5. Pengujian perbandingan metode yang dilakukan yaitu antara metode *Extreme Learning Machine* menggunakan inialisasi bobot *random* dan inialisasi *Nguyen-Widrow*. Setelah dilakukan masing-masing 10 kali uji coba diperoleh hasil bahwa metode *Extreme Learning Machine* menggunakan inialisasi *Nguyen-Widrow* memiliki nilai RMSE yang lebih kecil yaitu sebesar 0,65 sedangkan metode *Extreme Learning Machine* menggunakan inialisasi *random* memiliki nilai RMSE yang lebih besar yaitu sebesar 0,83.

1.2 Saran

Untuk pengembangan pada penelitian selanjutnya terdapat beberapa saran yang diharapkan dapat memperbaiki hasil penelitian yaitu sebagai berikut.

1. Dapat dilakukan perbandingan antara inialisasi bobot *Nguyen-Widrow* dengan inialisasi bobot yang diperkenalkan oleh (Drago & Ridella, 1992).

2. Dapat dilakukan penambahan fitur relevan lain untuk melihat pengaruh dari penambahan fitur tersebut terhadap prediksi Indeks Harga Konsumen kelompok perumahan, air, listrik, dan bahan bakar.



DAFTAR PUSTAKA

- Ahsani, R. et al., 2017. Comparison of Neural Neutwork (NN) and Extreme Learning Machine (ELM) on Thickness Auto Adjustment of Instant Noodle Dough on Roll Press Machine. *IEEE*, Volume 111, p. 107.
- Andrian, Y. & Wayahdi, M. R., 2014. Analisis Algoritma Inisialisasi Nguyen-Widrow Pada Proses Prediksi Curah Hujan Kota Medan Menggunakan Metode Backpropagation Neural Network. *Seminar Nasional Informatika 2014*, pp. 57-63.
- Badan Pusat Statistik, 2017. *Metadata Indikator Inflasi*. [Online] Available at: <https://www.bps.go.id/subject/3/inflasi.html> [Accessed 17 February 2018].
- Bahiuddin, I. et al., 2017. Study of Extreme Learning Machine Activation Functions for Magnetorheological Fluid Modelling. *IEEE*.
- Bank Indonesia, D. S., 2016. *Metadata Indeks Harga Konsumen (IHK)*, Jakarta: Bank Indonesia.
- Chen, X. et al., 2012. Electricity Price Forecasting With Extreme Learning Machine and Bootstrapping. *IEEE Transactions on Power Systems*, Volume 27, pp. 2055-2062.
- Cholissodin, I. et al., 2017. Optimasi Kandungan Gizi Susu Kambing Peranakan Etawa (PE) Menggunakan ELM-PSO di UPT Pembibitan Ternak dan Hijauan Makanan Ternak Singosari-Malang. *Teknologi Informasi dan Ilmu Komputer (JTIIK)*, pp. 31-36.
- Chun, Y. L., 1993. Efficient Learning and Second-order Methods.
- Drago, G. P. & Ridella, S., 1992. Statistically Controlled Activation Weight Initialization (SCAWI). *Transaction on Neural Networks*, Volume 3, pp. 627-631.
- Editor, D., 2017. *Macroeconomic Dashboard*. [Online] Available at: <http://macroeconomicdashboard.feb.ugm.ac.id/inflasi-dan-indeks-harga-konsumen/> [Accessed 16 Februari 2018].
- Haffner, P., Shikano, K. & Waibel, A., 1988. Fast Back-Propagation Learning Methods for Neural Networks in Speech. *Proceedings of the Fall Meeting of the Acoustical Society of Japan*, pp. 619-624.
- Herdianto, 2013. Prediksi Kerusakan Motor Induksi Menggunakan Metode Jaringan Saraf Tiruan Backpropagation. *Repository Universitas Sumatera Utara*.
- Huang, G. B., Zhu, Q. Y. & Siew, C. K., 2006. Extreme Learning Machine: Theory and Applications. *ScienceDirect*, Volume 70, pp. 489-501.

- Huang, G.-B., Zhou, H., Ding, X. & Zhang, R., 2012. Extreme Learning Machine for Regression and Multiclass Classification. *IEEE*, Volume 42, pp. 513-529.
- Irawati, L., Tarno & Yasin, H., 2015. Peramalan Indeks Harga Konsumen 4 Kota di Jawa Tengah Menggunakan Model Generalized Space Time Autoregressive (GSTAR). *Jurnal Gaussian*, Volume 4, pp. 553-562.
- JJ, 2016. *Medium*. [Online] Available at: <https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d> [Accessed 16 July 2018].
- Leung, K. M., 2007. *Tandon School of Engineering Polytechnic Institute*. [Online] Available at: <http://cis.poly.edu/~mleung/FRE7851/f07/preparingData.pdf> [Accessed 20 March 2018].
- Nguyen, D. & Widrow, B., 1990. Improving The Learning Speed of 2-Layer Neural Networks by Choosing Initial Values of The Adaptive Weights. *International Joint Conference on Neural Networks, 1990*, Volume 3, pp. 21-26.
- Pangaribuan, J. J., 2014. Diagnosis of Diabetes Mellitus using Extreme Learning Machine. *IEEE*.
- Patro, S. G. K. & Sahu, K. K., 2015. Normalization: A Preprocessing Stage. *International Advanced Research Journal in Science, Engineering, and Technology*, 2(3).
- Siang, J. J., 2005. *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab*. Yogyakarta: ANDI.
- Syafruddin, M., Hakim, L. & Despa, D., 2014. Metode Regresi Linier untuk Prediksi Kebutuhan Energi Listrik Jangka Panjang (Studi Kasus Provinsi Lampung). *Jurnal Informatika dan Teknik Elektro Terapan*, Volume 2.
- Tavares, L. D., Saldanha, R. R. & Vieira, D. A. G., 2014. Extreme Learning Machine with Initialized Hidden Weight. *IEEE*, pp. 43-47.
- Tianfeng, C. & Draxler, R. R., 2014. Root Mean Square Error (RMSE) or Mean Absolute Error (MAE)? – Arguments Against Avoiding RMSE in The Literature. *Geoscientific Model Development*, pp. 1247-1250.
- Wanto, A. & Windarto, A. P., 2017. Analisis Prediksi Indeks Harga Konsumen Berdasarkan Kelompok Kesehatan Dengan Menggunakan Metode Backpropagation. *Jurnal & Penelitian Teknik Informatika*, Volume 2, pp. 37-44.
- Wessels, L. F. & Barnard, E., 1992. Avoiding False Local Minima by Proper Initialization of Connections. *Transaction Neural Networks*, Volume 3, pp. 899-905.
- Wigati, Y., Rais & Utami, I. T., 2016. Pemodelan Time Series dengan Proses ARIMA untuk Prediksi Indeks Harga Konsumen (IHK) di Palu, Sulawesi Tengah. *Jurnal Ilmiah Matematika dan Terapan*, Volume 12, pp. 149-159.

- Yam, J. Y. F. & Chow, T. W. S., 2001. Feedforward Networks Training Speed Enhancement by Optimal Initialization of The Synaptic Coefficients. *IEEE Transactions on Neural Networks*, Volume 12, pp. 430-434.
- Yaseen, Z. M. et al., 2016. Stream-flow Forecasting Using Extreme Learning Machines: A Case Study in A Semi-arid Region in Iraq. *Journal of Hydrology*, pp. 603-614.

