

**IMPLEMENTASI ALGORITMA AL-ALAOU
BACKPROPAGATION DALAM PERAMALAN CURAH HUJAN**

SKRIPSI

Untuk memenuhi sebagian persyaratan untuk mencapai gelar Sarjana Komputer



Disusun oleh :

MELATI PUTRI C.

NIM. 0910961005

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS BRAWIJAYA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER**

MALANG

2013

LEMBAR PERSETUJUAN

IMPLEMENTASI ALGORITMA *AL-ALAOU* *BACKPROPAGATION* DALAM PERAMALAN CURAH HUJAN

SKRIPSI

Untuk memenuhi sebagian persyaratan untuk mencapai gelar Sarjana Komputer



Disusun oleh:

MELATI PUTRI C.

NIM.0910961005

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I,

Dosen Pembimbing II,

Candra Dewi, S.Kom., M.Sc.
NIP.197711142003122001

Novanto Yudistira, S.Kom., M.Sc.
NIP. 831110 16110425

LEMBAR PENGESAHAN

IMPLEMENTASI ALGORITMA *AL-ALAOU* *BACKPROPAGATION* DALAM PERAMALAN CURAH HUJAN

SKRIPSI

Untuk memenuhi sebagian persyaratan untuk mencapai gelar Sarjana Komputer

Disusun oleh:

MELATI PUTRI C.

NIM.0910961005

Skrripsi ini telah diuji dan dinyatakan lulus pada tanggal 24 Desember 2013

Penguji I

Penguji II

Drs. Achmad Ridok, M.Kom

NIP. 19680825 199403 1 002

Rekyan Regasari Mardi Putri, ST., MT.

NIP. 19741113 200501 2 001

Penguji III

Ahmad Afif Supianto, S.Si., M.Kom

NIK. 820623 16 1 1 0425

Mengetahui,

Ketua Program Studi Teknik Informatika

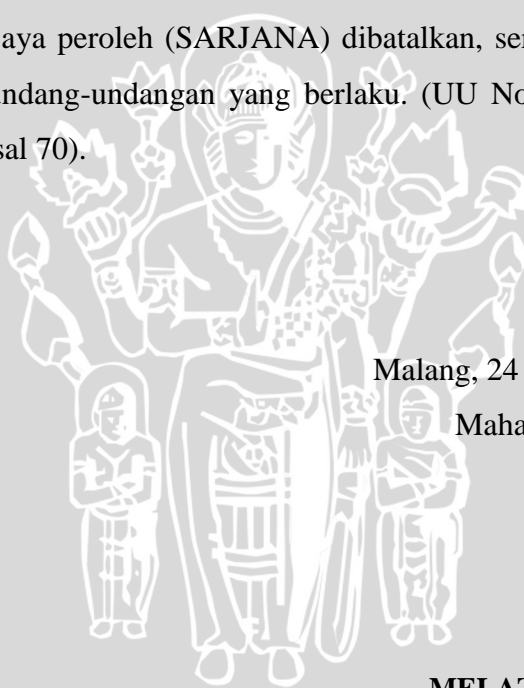
Drs. Marji, M.T.

NIP. 19670801 199203 1 001

PERNYATAAN ORISINALITAS SKRIPSI

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 dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan 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, 24 Desember 2013

Mahasiswa,

MELATI PUTRI C.

NIM. 0910961005

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, karena atas segala rahmat dan limpahan hidayah-Nya, Skripsi yang berjudul **“Implementasi Algoritma Al-Alaoui Backpropagation dalam peramalan curah hujan”** ini dapat disusun dengan baik. Skripsi ini disusun dan diajukan sebagai syarat untuk memperoleh gelar sarjana pada Program Studi Informatika / Ilmu Komputer, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya.

Pada kesempatan ini penulis mengucapkan banyak terima kasih atas segala bantuan dan dedikasi moral maupun material dalam rangka penyusunan skripsi ini. Atas bantuan yang telah diberikan, penulis ingin menyampaikan penghargaan dan ucapan terima kasih yang sedalam-dalamnya kepada:

1. Candra Dewi, S.Kom, M.Sc, selaku dosen pembimbing utama yang telah meluangkan waktu untuk memberikan pengarahan dan masukan bagi penulis.
2. Novanto Yudistira, S.Kom, M.Sc, selaku pembimbing kedua yang telah banyak memberikan bimbingan dalam penulisan skripsi ini.
3. Drs. Marji, MT., selaku Ketua Program Studi Teknik Informatika Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya.
4. Drs. Muh Arif Rahman, M.Kom, selaku Dosen Penasehat Akademik.
5. Ir. Sutrisno, MT, selaku Ketua Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya.
6. Segenap Bapak dan Ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada Penulis selama menempuh pendidikan di Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya.
7. Segenap staf dan karyawan di Program Teknik Informatika Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya yang telah banyak membantu Penulis dalam pelaksanaan penyusunan skripsi ini.
8. Mama Darsih , Bapak Sutoyo ,Mas Soni, Mas Sance dan juga seluruh keluarga tercinta, terima kasih atas semua doa, kasih sayang dan perhatian yang tulus serta dukungan yang telah diberikan.

9. Rekan-rekan Program Studi Informatika/Ilmu Komputer yang telah memberikan dukungannya kepada penulis.
10. Sahabat-sahabat saya yang telah banyak memberikan dukungan dan nasihat.
11. Keluarga kos kertoasri 71 yang telah banyak memberikan motivasi kepada saya.
12. Semua pihak yang telah terlibat baik secara langsung maupun tidak langsung yang tidak dapat penulis sebutkan satu per satu.

Penulis menyadari bahwa skripsi ini tentunya tidak terlepas dari berbagai kekurangan dan kesalahan. Oleh karena itu, segala kritik dan saran yang bersifat membangun sangat penulis harapkan dari berbagai pihak demi penyempurnaan penulisan skripsi ini.

Akhirnya penulis berharap agar skripsi ini dapat memberikan sumbangan dan manfaat bagi semua pihak yang berkepentingan.

Malang, 24 Desember 2013

Penulis

ABSTRAK

Melati Putri C. 2013. Implementasi Algoritma *Al-Alaoui Backpropagation* Dalam Peramalan Curah Hujan. Skripsi Program Studi Teknik Informatika/Ilu Komputer, Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya. Dosen Pembimbing : Candra Dewi, S.Kom, M.Sc dan Novanto Yudistira, S.Kom., M.Sc

Curah hujan merupakan salah satu unsur iklim yang sangat penting terutama dalam pertumbuhan tanaman dan produksi tanaman pangan di bidang pertanian dan peternakan. Curah hujan yang lebat dapat merusak tanaman secara langsung atau mengganggu pembungaan dan penyerbukan. Pada bidang peternakan berpengaruh melalui sumber pakan untuk hewan. Namun untuk mengoptimalkan produktivitas pertanian dan peternakan yang erat kaitannya dengan curah hujan, dan mengantisipasi dampak curah hujan serta tingkat curah hujan yang akan datang juga penting untuk diketahui. Untuk itu dibutuhkan peramalan untuk mengetahui seberapa besar curah hujan yang akan terjadi di Indonesia pada waktu-waktu tertentu. Skripsi ini membahas algoritma *Al-Alaoui Backpropagation* dalam peramalan curah hujan. Sistem yang dibangun menggunakan data curah hujan yang diambil dari BMKG Karang Ploso. Data latih yang digunakan adalah data *time series* curah hujan pada bulan Januari 2011-Maret 2012. Hasil dari pengujian ini mengetahui *learning rate* dan momentum yang terbaik pada MSE agar nilai *learning rate* dan momentum dapat dijadikan acuan nilai dalam pengujian, serta mengetahui tingkat akurasi dari sistem ini. Dari hasil pengujian didapatkan nilai *learning rate* dan momentum terbaik mencapai 0,1 dan 0,1. Akurasi yang didapatkan bisa mencapai sebesar 100% dengan menggunakan data latih pada bulan Januari 2011 – Juni 2011 dan data uji pada bulan Juli, pada bulan ini kondisi cuacanya kemarau, dengan MSE terendah 0.00313611. Dapat disimpulkan bahwa penelitian menggunakan *Al-Alaoui Backpropagation* memiliki kinerja yang baik dalam peramalan.

Kata kunci : *Al-Alaoui Backpropagation*, peramalan, curah hujan

ABSTRACT

Melati Putri C. 2013. *Implementation Of Al-Aloui Backpropagation Algorithm For Predicting Rainfall. Minor Thesis Program of Study Information Technology / Computer Science, Program of Technology Information and Computer Science University of Brawijaya. Advisor : Candra Dewi, S.Kom, M.Sc dan Novanto Yudistira, S.Kom., M.Sc*

Rainfall is one of important element for plant growth and crops production especially in agriculture and also the livestock. Heavy rainfall can be directly destructive for plants or it can influence flowering and pollination process. For the livestock, heavy rainfall influences animal's food supply. To optimalize agriculture and livestock productivity which tightly related with the rainfall and also to anticipate the effect of the rain itself, therefore it is considered important to predict rainfall in Indonesia for the certain time. This thesis discusses about how to implement Al-Aloui Backpropagation algorithm for predicting rainfall. Rainfall data that is used in this thesis are taken from BMKG Karang Ploso. In this thesis, the data used are time series data of rainfall during certain period from Januari 2011 through March 2012. The result of the experimental test that has ben done are the best learning rate and momentum value for MSE so learning rate and momentum value can be used to know the accuracy for this system. From the experiment that has done before, the best value of learning rate and momentum respectively are 0,1 and 0,1. The highest accuracy this system can get is 100 % by using testing data at month of July where it was drought at that month, with the lowest MSE value is 0.00313611. In conclusion, Al-Aloui Backpropagation algorithm has good performance fo prediction.

Keywords: *Al-Aloui Backpropagation, prediction , rain fall*

DAFTAR ISI

LEMBAR PERSETUJUAN.....	ii
LEMBAR PENGESAHAN	iii
PERNYATAAN ORISINALITAS SKRIPSI.....	iv
KATA PENGANTAR	v
ABSTRAK.....	vii
<i>ABSTRACT</i>	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR	xiii
DAFTAR TABEL.....	xiv
DAFTAR <i>SOURCE CODE</i>	xv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	2
1.5 Manfaat Penelitian.....	2
1.6 Sistematika Penulisan.....	3
BAB II KAJIAN PUSTAKA.....	4
2.1 Kajian Pustaka	4
2.2 Pengertian Curah Hujan.....	4
2.2.1 Kriteria Intensitas Curah Hujan	5
2.3 Konsep Jaringan Saraf Tiruan	6
2.3.1 Definisi Jaringan Saraf Tiruan	6
2.3.2 Model Jaringan Saraf Tiruan.....	6
2.3.3 Proses Pembelajaran Jaringan.....	7
2.3.4 Fungsi Aktivasi Jaringan Saraf Tiruan.....	7
2.4 <i>Backpropagation</i>	8
2.4.1 Arsitektur <i>Backpropagation</i>	9

2.4.2 Pelatihan Standar <i>Backpropagation</i>	10
2.5 Algoritma <i>Al-Alaoui</i>	13
2.6 Algoritma <i>Al-Alaoui Backpropagation</i>	13
2.7 Tingkat Akurasi.....	14
2.8 Variasi <i>Backpropagation</i>	14
2.8.1 Normalisasi dan Denormalisasi	14
2.8.2 Momentum.....	15
2.8.3 Jumlah <i>Hidden Layer</i>	15
BAB III METODOLOGI DAN PERANCANGAN	17
3.1 Studi Literatur	17
3.2 Data Penelitian.....	18
3.3 Analisis dan Perancangan Sistem	18
3.3.1. Deskripsi umum sistem	18
3.3.2. Perancangan proses.....	18
3.3.2.1 Proses Normalisasi Data	20
3.3.2.2 Proses Pelatihan Jaringan Syaraf Tiruan <i>Al-Alaoui</i> <i>Backpropagation</i>	22
3.3.2.3 Proses <i>FeedForward</i>	23
3.3.2.4 Proses <i>Backpropagation</i>	25
3.3.2.5 Proses <i>Weight Update</i>	26
3.3.2.6 Proses Pengujian JST <i>Al-Alaoui</i> <i>Backpropagation</i>	26
3.4 Contoh Perhitungan Manual	28
3.4.1. Proses penginputan data	28
3.4.2. Proses Inisialisasi Bobot awal	29
3.4.3 Proses <i>FeedForward</i>	30
3.4.4 Proses <i>Backpropagation</i>	31
3.4.5 Proses <i>Weight Update</i>	34
3.4.6 Proses Denormalisasi.....	36
3.4.7 Perbandingan nilai <i>output</i> jaringan (yk) dengan <i>output</i> data asli.....	36



3.5 Proses Duplikasi Data.....	37
3.6 Rancangan Antarmuka.....	38
3.7 Rancangan Uji Coba.....	40
3.7.1 Uji Pengaruh Learning Rate dan Momentum terhadap MSE.....	40
3.7.2 Uji Pengaruh Jumlah Data Latih Terhadap Akurasi.....	41
BAB IV IMPLEMENTASI	42
4.1 Lingkungan Implementasi	42
4.1.1 Lingkungan Implementasi Perangkat Keras.....	42
4.1.2 Lingkungan Implementasi Perangkat Lunak.....	42
4.2 Implementasi Program.....	42
4.2.1 Proses Baca File	43
4.2.2 Proses mencari nilai minimum dan maksimum.....	44
4.2.3 Proses normalisasi data dan denormalisasi data.....	45
4.2.4 Proses Inisialisasi Bobot.....	46
4.2.5 Proses Pelatihan.....	47
4.2.5.1 Proses <i>FeedForward</i>	47
4.2.5.2 Proses <i>Backpropagation</i>	48
4.2.5.3 Proses <i>Weight Update</i>	49
4.2.5.4 Proses Hitung <i>Error</i>	50
4.2.5.5 Proses Duplikasi.....	50
4.2.6 Proses Pengujian.....	51
4.2.6.1 Proses menghitung Tingkat Akurasi.....	53
4.3 Implementasi Antar Muka.....	55
4.3.1 Form Pelatihan.....	55
4.3.2 Form Pengujian.....	56
BAB V ANALISA HASIL DAN PEMBAHASAN	57
5.1 Hasil Uji Coba	57
5.1.1 Uji Pengaruh <i>Learning Rate</i> dan Momentum terhadap MSE.....	57
5.1.2 Uji Pengaruh Jumlah Data Latih Terhadap Akurasi.....	59
5.2 Analisis dan Interpretasi	60

5.2.1 Analisis Hasil *Learning Rate* dan Momentum terhadap MSE60

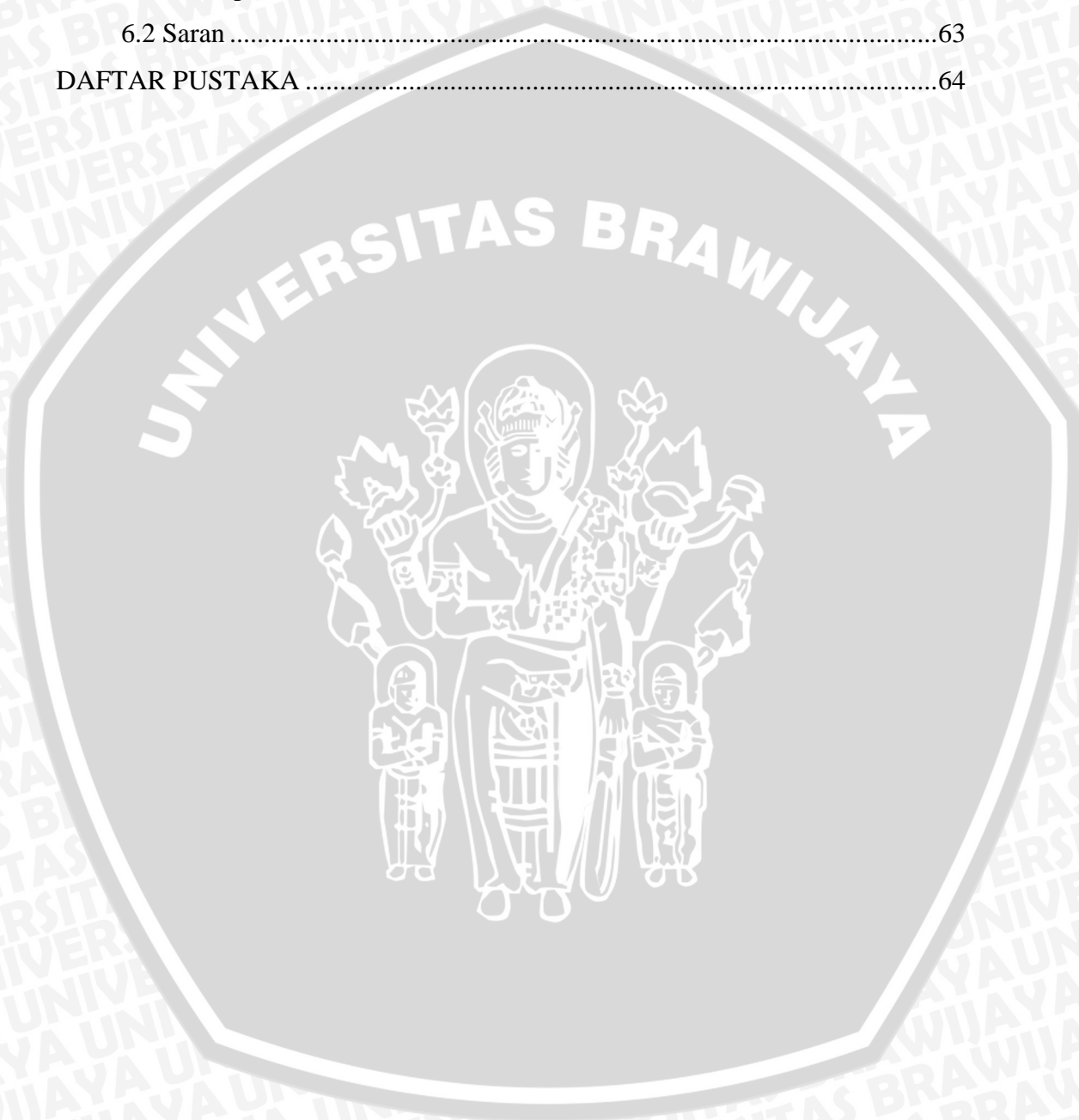
5.2.2 Analisis Hasil Pengaruh Jumlah Data Latih Terhadap Akurasi ...62

BAB VI PENUTUP63

6.1 Kesimpulan63

6.2 Saran63

DAFTAR PUSTAKA64



DAFTAR GAMBAR

Gambar 2.1 Model Struktur JST6

Gambar 2.2 Fungsi sigmoid unipolar8

Gambar 2.3 Arsitektur *Backpropagation*9

Gambar 3.1 Diagram alir Penelitian17

Gambar 3.2 Arsitektur Jaringan Pada Sistem18

Gambar 3.3 Deskripsi sistem secara umum20

Gambar 3.4 Proses Normalisasi Data21

Gambar 3.5 Proses Pelatihan Jaringan Syaraf Tiruan
Al-Alaoui Bckpropagation22

Gambar 3.6 Proses *FeedForward*24

Gambar 3.7 Proses *Backpropagation*25

Gambar 3.8 Proses *Weight Update*26

Gambar 3.9 Proses Pengujian JST *Al-Alaoui Backpropagation*27



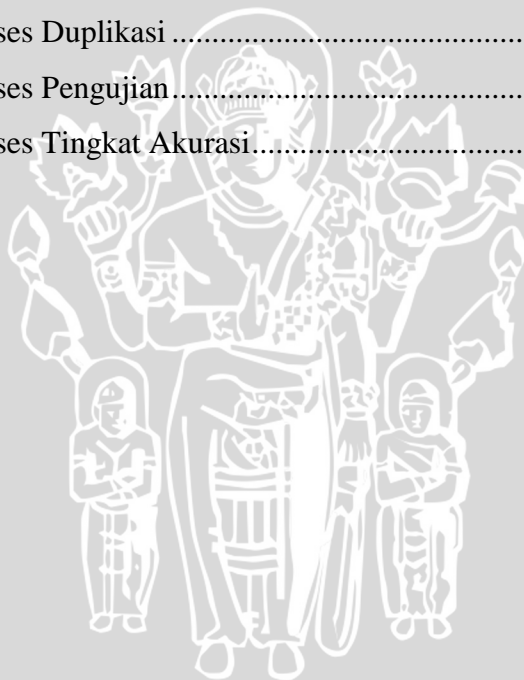
DAFTAR TABEL

Tabel 3.1 Pola Data <i>Input</i> dan Target	19
Tabel 3.2 Pola penginputan data	28
Tabel 3.3 Pola Data asli curah hujan	28
Tabel 3.4 Pola Data Normalisasi	29
Tabel 3.5 Pola Inisialisasi Bobot awal	29
Tabel 3.6 Data Nilai bobot Awal w_{jk}	30
Tabel 3.7 Data Z_{netj}	30
Tabel 3.8 Data Z_j	30
Tabel 3.9 Data y_{netk}	31
Tabel 3.10 Data y_k	31
Tabel 3.11 Data δ_k	32
Tabel 3.12 Data Δw_{kj}	32
Tabel 3.13 Data δ_{netj}	33
Tabel 3.14 Data δ_j	33
Tabel 3.15 Data Δv_{ji}	34
Tabel 3.16 Data Δw_{kj}	34
Tabel 3.17 Data Δv_{ji}	35
Tabel 3.18 Data Δw_{kj}	35
Tabel 3.19 Data Δv_{ji}	35
Tabel 3.20 Data y_k Denormalisasi	36
Tabel 3.21 Kriteria Curah Hujan	37
Tabel 3.22 Hasil perbandingan nilai output dengan kriteria curah hujan	37
Tabel 3.23 Hasil Proses Duplikasi	37
Tabel 3.24 Pengaruh Learning Rate dan Momentum terhadap MSE	40
Tabel 3.25 Uji Pengaruh Jumlah Data Latih terhadap akurasi	41
Tabel 5.1 Tabel Hasil uji pengaruh <i>learning rate</i> dan momentum	55
Tabel 5.2 Tabel Hasil Uji pengaruh jumlah data latih terhadap akurasi	58
Tabel 5.3 Tabel Hasil Uji pengaruh jumlah data latih terhadap akurasi	59



DAFTAR SOURCE CODE

<i>Source Code 4.1</i> Proses Baca File.....	43
<i>Source Code 4.2</i> Proses mencari nilai minimum dan maksimum.....	45
<i>Source Code 4.3</i> Proses Normalisasi Data.....	46
<i>Source Code 4.4</i> Proses Denormalisasi.....	46
<i>Source Code 4.5</i> Proses Inisialisasi Bobot.....	47
<i>Source Code 4.6</i> Proses <i>FeedForward</i>	48
<i>Source Code 4.7</i> Proses <i>Backpropagation</i>	48
<i>Source Code 4.8</i> Proses <i>Weight Update</i>	49
<i>Source Code 4.9</i> Proses hitung <i>Error</i>	50
<i>Source Code 4.10</i> Proses Duplikasi	50
<i>Source Code 4.11</i> Proses Pengujian.....	52
<i>Source Code 4.12</i> Proses Tingkat Akurasi.....	53



BAB I PENDAHULUAN

1.1 Latar Belakang

Curah hujan diukur selama 24 jam, sehingga dapat diukur banyaknya hujan, baik harian, bulanan, maupun tahunan. Curah hujan merupakan salah satu unsur iklim yang sangat penting terutama dalam pertumbuhan tanaman dan produksi tanaman pangan di bidang pertanian dan peternakan karena dari curah hujan ini mensuplay air dalam tanah[KKLI-01].

Curah hujan yang lebat dapat merusak tanaman secara langsung atau mengganggu pembungaan dan penyerbukan. Pada bidang peternakan berpengaruh melalui sumber pakan untuk hewan[TJA-04]. Namun untuk mengoptimalkan produktivitas pertanian dan peternakan yang erat kaitannya dengan curah hujan, dan mengantisipasi dampak curah hujan serta tingkat curah hujan yang akan datang juga penting untuk diketahui. Untuk itu dibutuhkan peramalan untuk mengetahui seberapa besar curah hujan yang akan terjadi di Indonesia pada waktu-waktu tertentu. Salah satu metode *Artificial Intelligence* untuk melakukan peramalan adalah metode *Backpropagation*. Algoritma ini mempunyai kelemahan, yaitu bila *learning rate* kecil, maka pencapaian nilai konvergenya lambat[PRA-99]. Selain itu, juga memiliki tingkat akurasi yang masih rendah [ALA-07]. Namun untuk mengatasinya digunakanlah variasi *backpropagation* yaitu dengan menggunakan metode *Al-Alaoui Backpropagation*

Algoritma *Al-Alaoui Backpropagation* adalah implementasi Algoritma *Al-Alaoui* kedalam sebuah jaringan syaraf tiruan *backpropagation* untuk mempercepat konvergensi dengan jalan mengenalkan ulang dan menduplikasi pasangan data latih yang masih salah dalam proses klasifikasi. Penelitian dengan menggunakan *Al-Alaoui backpropagation* pernah dilakukan dalam permasalahan *subsampling* image compression oleh Mohammad Adnan Al-Alaoui dan Rony Ferzli(2007) dan dihasilkan bahwa dengan iterasi dalam jumlah yang sama, dihasilkan konvergensi yang lebih cepat dan klasifikasi yang lebih akurat dibandingkan *backpropagation* standar [ALA-07].

Dalam menggunakan Algoritma *Al-Alaoui Backpropagation* diharapkan peramalan curah hujan menghasilkan konvergensi yang lebih cepat dan klasifikasi yang lebih akurat. Berdasarkan uraian dari latar belakang di atas, maka skripsi ini diberi judul **“Implementasi Algoritma *Al-Alaoui Backpropagation* dalam peramalan curah hujan”**.

1.2 Rumusan Masalah

Permasalahan yang akan dijadikan objek penelitian pada tugas akhir ini adalah :

1. Bagaimana mengimplementasi sistem jaringan syaraf tiruan *Al-Alaoui backpropagation* untuk melakukan peramalan curah hujan?
2. Berapakah tingkat akurasi sistem jaringan syaraf tiruan *Al-Alaoui backpropagation* untuk peramalan curah hujan?

1.3 Batasan Masalah

Berdasarkan permasalahan diatas akan diberikan batasan masalah sebagai berikut :

- Data yang digunakan dalam sistem algoritma *Al-Alaoui Bacpropagation* menggunakan data curah hujan BMKG Karangploso Kabupaten Malang.

1.4 Tujuan Penelitian

Adapun tujuan yang hendak dicapai dalam penulisan skripsi ini adalah :

1. Mengimplementasikan sistem jaringan syaraf tiruan *Al-Alaoui backpropagation* untuk melakukan peramalan curah hujan.
2. Mengukur tingkat akurasi sistem jaringan syaraf tiruan *Al-Alaoui backpropagation* untuk melakukan peramalan curah hujan.

1.5 Manfaat Penelitian

Dalam penelitian ini, diharapkan sistem yang dibangun dengan menggunakan Algoritma *Al-Alaoui backpropagation* memiliki akurasi peramalan yang baik, sehingga dapat melakukan pendeteksian curah hujan di berbagai bidang.

1.6 Sistematika Penulisan

Skripsi ini disusun berdasarkan sistematika penulisan sebagai berikut :

BAB I PENDAHULUAN

Bab ini berisi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

BAB II KAJIAN PUSTAKA

Bab ini berisi teori-teori yang berkaitan dengan topik penulisan skripsi. Teori tersebut meliputi penjelasan curah hujan serta metode *Al-Alaoui Backpropagation*.

BAB III METODOLOGI DAN PERANCANGAN

Bab ini menjelaskan mengenai metode penelitian sistem untuk peramalan curah hujan dengan metode *Al-Alaoui backpropagation* serta perancangan sistem.

BAB IV IMPLEMENTASI

Bab ini berisi spesifikasi sistem (*hardware*) dan pembahasan dari implementasi metode *Al-Alaoui backpropagation* pada sistem.

BAB V ANALISIS HASIL DAN PEMBAHASAN

Bab ini berisi skenario pengujian perhitungan dalam penggunaan metode yang digunakan, prosedur uji serta menganalisa data hasil pengujian.

BAB VI PENUTUP

Bab ini berisi kesimpulan berdasarkan penelitian yang telah dilakukan, serta saran-saran untuk pengembangan penelitian yang lebih lanjut.

BAB II

KAJIAN PUSTAKA

2.1 Kajian Pustaka

Peneliti mengambil beberapa penelitian terdahulu untuk dijadikan dasar penelitian saat ini. Penelitian pertama peramalan curah hujan menggunakan jaringan saraf tiruan backpropagation pernah diteliti oleh Tito Prasetyo. Dalam penelitian ini diperoleh kombinasi jumlah input neuron dan jumlah hidden neuron yang terbaik yaitu 12 neuron input dan 9 neuron hidden, serta nilai learning rate sebesar 0,2 dengan jumlah epoch sebanyak 10000 dan jumlah data latih sebanyak 208 data (4 tahun). Kombinasi ini menghasilkan MSE pelatihan sebesar 0,004459755, dan ketika dilakukan peramalan sistem mampu mengenali pola dengan baik dan menghasilkan MSE sebesar 0,0267102661781464 [TIT-12].

Penelitian kedua *Subsampling Image Compression Using Al-Alaoui Backpropagation Algorithm* yang dilakukan oleh Mohammad Adnan Al- Alaoui dan Rony Ferzli dengan menggunakan permasalahan subsampling image compression menghasilkan iterasi dalam jumlah yang sama, waktu pelatihan yang lebih cepat dan klasifikasi yang lebih akurat[ALA-07].

Penelitian ketiga Pengenalan Emosi Suara Dengan Jaringan Syaraf Tiruan *Al-Alaoui Backpropagation* yang dilakukan oleh Tony Sahidi untuk pengenalan emosi berdasarkan suara. Dalam penelitian ini, struktur JST yang terbaik adalah struktur JST yang terdiri dari 37 unit neuron input layer, 5 neuron hidden layer, dan 2 neuron output layer, serta learning rate sebesar 0,2. Setelah didapat struktur JST yang optimal, maka akan didapatkan juga tingkat akurasi sebesar 94,386% [SAH-12].

2.2 Pengertian Curah Hujan

Curah hujan merupakan endapan atau deposit air dalam bentuk cair maupun padat yang jatuh ke permukaan bumi. Hal ini berarti curah hujan mencakup hujan, gerimis, salju, dan batu es. Curah hujan merupakan unsur iklim yang sangat penting bagi kehidupan. Unsur – unsur curah hujan yang harus diperhatikan dalam

mempelajari curah hujan ialah : jumlah curah hujan, hari hujan dan intensitas atau kekuatan tetesan hujan. Banyaknya curah hujan dinyatakan dengan satuan millimeter, berdasarkan pengertian klimatologi satu hari hujan adalah periode selama 24 jam terkumpul curah hujan setinggi 0.5 mm atau lebih[WAR-87]. Apabila kurang dari ketentuan tersebut, maka hari hujan dianggap nol. Jumlah curah hujan 1 mm, menunjukkan tinggi air hujan yang menutupi permukaan 1 mm, jadi air tersebut tidak meresap ke dalam tanah atau menguap ke atmosfer[TJA-04].

2.2.1 Kriteria Intensitas Curah Hujan

Hujan adalah bentuk endapan yang sering dijumpai di Indonesia, yang dimaksud endapan adalah curah hujan[TJA-04]. Intensitas curah hujan adalah ketinggian curah hujan yang terjadi pada suatu kurun waktu dimana air tersebut terkonsentrasi. Intensitas curah hujan dinotasikan dengan satuan mm/jam[SUR-06]. Apabila intensitasnya besar berarti hujan lebat dan kondisi ini sangat berbahaya karena berdampak pada efek negatif terhadap tanaman.

Kriteria Intensitas Curah hujan di wilayah Indonesia [BMK-10]:

1. Sangat Ringan (SR)

Jika nilai perbandingan terhadap lama curah hujan(jam) dengan curah hujan maks. dalam 24 jam (mm) adalah <5 mm/hari

2. Ringan (R)

Jika nilai perbandingan terhadap lama curah hujan(jam) dengan curah hujan maks. dalam 24 jam (mm) antara 5 - 20 mm/hari

3. Sedang (S)

Jika nilai perbandingan terhadap lama curah hujan(jam) dengan curah hujan maks. dalam 24 jam (mm) antara 20 – 50 mm/hari

4. Lebat (L)

Jika nilai perbandingan terhadap lama curah hujan(jam) dengan curah hujan maks. dalam 24 jam (mm) antara 50 – 100 mm/hari

5. Sangat Lebat (LB)

Jika nilai perbandingan terhadap lama curah hujan(jam) dengan curah hujan maks. dalam 24 jam (mm) adalah >100 mm/hari.

2.3 Konsep Jaringan Saraf Tiruan

2.3.1 Definisi Jaringan Saraf Tiruan

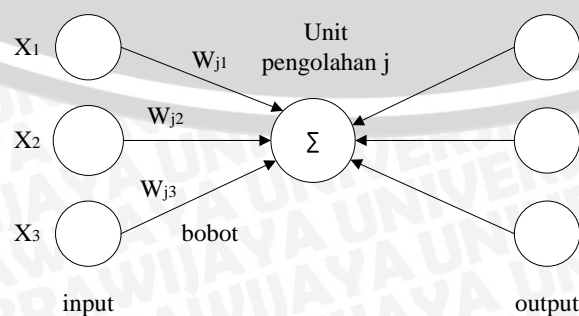
Jaringan saraf tiruan dapat memodifikasi tingkah lakunya sesuai dengan keadaan lingkungannya. Jaringan saraf tiruan dapat mengatur dirinya sendiri untuk menghasilkan suatu respon yang konsisten terhadap serangkaian masukan. Jaringan syaraf prinsip kerjanya sama seperti jaringan syaraf pada manusia. Kemampuan penginderaan manusia mampu mengadaptasi pemrosesan pola data dengan baik. Sehingga mampu mengingat suatu informasi pola secara menyeluruh [SET-93].

Jaringan syaraf tiruan dibentuk sebagai generalisasi model matematika dari jaringan syaraf manusia, dengan asumsi bahwa [GRA-07] :

- Pemrosesan informasi terjadi pada banyak elemen sederhana (neuron).
- Sinyal dikirimkan diantara neuron-neuron melalui penghubung-penghubung.
- Penghubung antara neuron memiliki bobot yang memperkuat atau memperlemah sinyal.
- Untuk menentukan output, setiap neuron menggunakan fungsi aktivasi(biasanya bukan fungsi linier) yang dikenakan pada jumlah input yang diterima.

2.3.2 Model Jaringan Saraf Tiruan

Jaringan saraf terdiri atas beberapa neuron dan masing- masing neuron saling berhubungan. Neuron adalah dasar dari operasi jaringan saraf tiruan yang dilakukan oleh sebuah unit pemrosesan informasi[JUM-09]. Model struktur neuron jaringan saraf tiruan dijelaskan pada gambar 2.1 sebagai berikut :



Gambar 2.1 Model Struktur JST

Jaringan saraf tiruan dapat melakukan generalisasi yang diperoleh dan mengabstraksi karakteristik masukan. Proses pengolahan data pada algoritma jaringan saraf tiruan akan berupa angka yang diubah menjadi data numerik. Jaringan saraf tiruan tidak diprogram menghasilkan keluaran tertentu, semua keluaran atau kesimpulan berasal dari proses pembelajaran[DEI-12]. Proses pembelajaran ini pada jaringan saraf tiruan dimasukkan pola-pola masukan dan keluaran lalu jaringan akan memberikan jawaban yang diterima. Karakteristik Jaringan saraf tiruan ditentukan oleh[SUB-10] :

- Pola hubungan antar neuron yang disebut juga dengan arsitektur jaringan
- Metode penentuan bobot-bobot sambungan sambungan disebut dengan pelatihan atau proses belajar jaringan
- Fungsi aktivasi

2.3.3 Proses Pembelajaran Jaringan

Metode pembelajaran/pelatihan JST adalah sebagai berikut[DEI-12] :

a. Pembelajaran terawasi (*supervised learning*)

Pembelajaran ini kumpulan berupa data masukan yang digunakan, data keluarannya telah diketahui. Keluaran-keluaran yang aktual mencari perbedaannya dengan data keluaran yang diinginkan digunakan untuk mengoreksi bobot JST agar menghasilkan jawaban sedekat mungkin dengan jawaban yang benar.

b. Pembelajaran tak terawasi (*unsupervised learning*)

Pada pembelajaran ini, JST mengorganisasi dirinya sendiri untuk membentuk vektor-vektor masukan yang serupa, tanpa menggunakan data atau contoh-contoh pelatihan. Struktur menggunakan dasar data atau korelasi antara pola-pola data yang dieksplorasi. Paradigma pembelajaran ini mengorganisasi pola-pola ke dalam kategori-kategori berdasarkan korelasi yang ada.

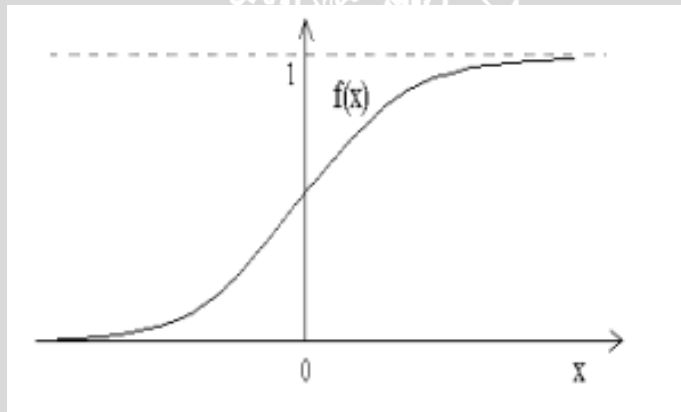
2.3.4 Fungsi Aktivasi Jaringan Saraf Tiruan

Mengaktifkan jaringan saraf tiruan berarti mengaktifkan setiap neuron yang dipakai pada jaringan tersebut. Terdapat banyaknya fungsi yang dipakai

sebagai aktivasi, tetapi fungsi yang digunakan adalah fungsi sigmoid, karena dianggap lebih mendekati kinerja sinyal pada otak.

Dalam melakukan pelatihan terhadap jaringan syaraf tiruan yang menggunakan fungsi aktivasi sigmoid, misalnya dengan algoritma propagasi balik (*backpropagation*). Dalam *backpropagation* fungsi aktivasi harus memenuhi syarat yaitu : kontinu, diferensial dan fungsi tidak turun[SIA-05]. Selanjutnya untuk mendapatkan efisiensi perhitungan, turunannya mudah dihitung. Nilai turunan biasanya diperoleh melalui rentang nilai tertentu[SUB-10]. Terdapat bentuk fungsi yang digunakan jenis fungsi sigmoid yaitu unipolar dan bipolar[FAU-94]. Fungsi sigmoid unipolar atau biner memiliki rentang (0,1) dituliskan pada persamaan (2.1) dan ditunjukkan pada gambar 2.3

$$f(x) = \left(\frac{1}{1 + e^{-x}} \right) \text{ dengan turunan } f'(x) = f(x)(1 - f(x)) \quad (2.1)$$



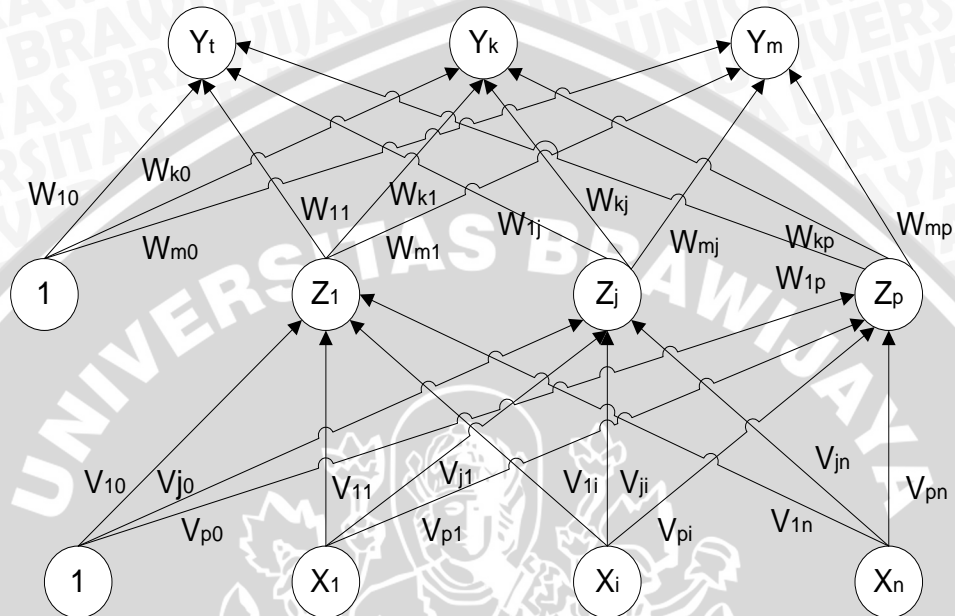
Gambar 2.2 Fungsi sigmoid unipolar

2.4 Backpropagation

Algoritma *Backpropagation* merupakan salah satu variasi dalam jaringan saraf tiruan, untuk proses pembelajarannya menggunakan penyesuaian bobot dari jaringan saraf tiruan dan didasarkan pada error yang terjadi[KUS-09]. *Backpropagation* melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan untuk memberikan respon yang benar terhadap pola masukan dengan pola yang dipakai selama pelatihan[SIA-05].

2.4.1 Arsitektur *Backpropagation*

Arsitektur Jaringan saraf yang digunakan adalah arsitektur *backpropagation* dengan X unit *input*, sebuah layer tersembunyi yang terdiri dari Z dan unit *output* dilambangkan dengan Y . Ditunjukkan pada gambar 2.3



Gambar 2.3 Arsitektur *Backpropagation*

- X_i : Unit ke-i pada lapisan masukan
- V_{ji} : bobot garis dari *input* ke unit layer tersembunyi
- V_{j0} : bobot garis yang menghubungkan bias dari unit layer tersembunyi ke layer tersembunyi
- Z_j : Unit ke-j pada lapisan tersembunyi
- W_{kj} : bobot dari unit layer tersembunyi ke *output*
- W_{k0} : bobot dari bias di layer tersembunyi ke unit keluaran
- Y_k : Unit ke-k pada lapisan keluaran

2.4.2 Pelatihan Standar *Backpropagation*

Pelatihan *Backpropagation* meliputi 3 fase, yaitu :

1. Fase I : *Feedforward*

Fase I adalah menghitung maju dari *input* (x_i) hingga layer keluaran menggunakan fungsi aktivasi yang ditentukan. Keluaran dari setiap unit layer tersembunyi (z_j) dipropagasikan maju lagi ke layer tersembunyi di atasnya, hingga menghasilkan keluaran jaringan (y_k). Keluaran jaringan (y_k) dibandingkan dengan target hingga (t_k). Iterasi akan dihentikan jika terjadi kesalahan antara selisih $t_k - y_k$.

2. Fase II : *Backpropagation*

Fase II adalah selisih antara keluaran jaringan dengan target yang diinginkan akan dipropagasikan mundur, dimulai dari garis yang berhubungan langsung dengan unit-unit di layer keluaran. Berdasarkan kesalahan $t_k - y_k$ dihitung faktor δ_k ($k=1,2,\dots,m$) untuk mendistribusikan kesalahan unit ke semua unit tersembunyi yang terhubung langsung dengan y_k . δ_k dipakai untuk mengubah bobot garis yang berhubungan langsung dengan unit keluaran.

3. Fase III : *Weight Update*

Fase III adalah modifikasi bobot untuk menurunkan kesalahan yang terjadi. Setelah semua faktor δ dihitung, bobot semua garis dimodifikasi bersamaan. Perubahan bobot garis yang menuju layer keluaran didasarkan atas δ yang ada di unit keluaran.

Ketiga Fase tersebut di ulang- ulang terus hingga kondisi penghentian dipenuhi. Kondisi penghentian adalah jumlah iterasi atau kesalahan. Iterasi akan dihentikan jika jumlahnya melebihi jumlah maksimum iterasi yang ditetapkan.

Jong Jek Siang (2005) menjelaskan rincian algoritma pelatihan menggunakan fungsi aktivasi sigmoid unipolar sebagai berikut :

- Langkah 0 : Inisialisasi bobot dengan bilangan acak nilai kecil
- Langkah 1 : Jika kondisi penghentian belum terpenuhi ulangi langkah 2-8
- Langkah 2 : Untuk setiap pasang *training data* lakukan langkah 3-8

Fase I : Feedforward

- Langkah 3 : Tiap unit masukan menerima sinyal input x_i dan diteruskan ke seluruh unit *hidden* layer.
- Langkah 4 : Semua keluaran di unit tersembunyi dihitung z_j ($j=1,2,\dots,p$) akan dijumlahkan sinyal-sinyal *input* yang sudah berbobot.

$$z_{net_j} = v_{j0} + \sum_{i=1}^n x_i \cdot v_{ji} \tag{2.2}$$

dan memakai fungsi aktivasi yang telah ditentukan untuk menghitung sinyal *output* dari hidden unit yang bersangkutan.

$$z_j = f(z_{net_j}) = \frac{1}{1 + \exp(-z_{net_j})}$$

lalu mengirim sinyal *output* ini keseluruhan unit pada unit output.

- Langkah 5 : Setiap unit *output* ($Y_k = 1,2,\dots,m$) akan menjumlahkan sinyal-sinyal input yang sudah berbobot.

$$y_{net_k} = w_{k0} + \sum_{j=1}^p z_j \cdot w_{kj} \tag{2.4}$$

dan memakai fungsi aktivasi yang telah ditentukan untuk menghitung sinyal *output* dari unit output yang bersangkutan :

$$= f(y_{net_k}) = \frac{1}{1 + \exp(-y_{net_k})} \tag{2.5}$$

Fase II :Backpropagation

- Langkah 6

Hitung faktor δ unit keluaran berdasarkan kesalahan pada setiap unit keluaran y_k dimana $k = 1,2,\dots,m$

$$\delta_k = (t_k - y_k) f'(y_{net_k}) = (t_k - y_k) y_k (1 - y_k) \tag{2.6}$$

δ_k merupakan unit kesalahan yang dipakai dalam perubahan bobot layar di bawahnya (langkah 7)

Hitung perubahan bobot w_{kj} dengan percepatan α

$$\Delta w_{kj} = \alpha \delta_k z_j \tag{2.7}$$

$k = 1,2,\dots,m$ dan $j = 0,1,\dots,p$



- Langkah 7

Hitung faktor δ layer tersembunyi berdasarkan kesalahan pada setiap layer tersembunyi z_j dimana $j = 1, 2, \dots, p$

$$\delta_{net_j} = \sum_{k=1}^m \delta_k \cdot w_{kj} \quad (2.8)$$

Faktor δ layer tersembunyi :

$$\delta_j = \delta_{net_j} \cdot f' (z_{net_j}) = \delta_{net_j} \cdot z_j (1 - z_j) \quad (2.9)$$

Hitung perubahan bobot v_{ji}

$$\Delta v_{ji} = \alpha \delta_j x_i \quad (2.10)$$

$j = 1, 2, \dots, p$ dan $i = 0, 1, \dots, n$

Fase III : Weight Update

- Langkah 8

Hitung perubahan bobot garis yang menuju unit keluaran

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj} \quad (2.11)$$

$k = 1, 2, \dots, m$ dan $j = 0, 1, \dots, p$

Hitung perubahan bobot garis yang menuju lapisan tersembunyi

$$v_{ji}(\text{baru}) = v_{ji}(\text{lama}) + \Delta v_{ji} \quad (2.12)$$

$j = 1, 2, \dots, p$ dan $i = 0, 1, \dots, n$

Daftar Notasi :

- z_{net_j} = nilai untuk menghitung *hidden layer*
- z_j = nilai *hidden layer*
- x_i = nilai *input layer* ke i
- v_{j0} = bobot bias antara *input layer* dengan *hidden layer* ($v_{j0} = 1$)
- v_{ji} = bobot antara *input layer* dengan *hidden layer*
- y_{net_k} = nilai untuk menghitung *output layer*
- y_k = nilai *output layer*
- w_{k0} = bobot bias antara *hidden layer* dengan *output layer* ($w_{k0} = 1$)
- w_{kj} = bobot antara *hidden layer* dengan *output layer*

δ_k	= nilai kesalahan pada <i>hidden layer</i>
t_k	= target data ke k
Δw_{kj}	= perubahan bobot antara <i>output layer</i> dengan <i>hidden layer</i>
α	= konstanta <i>learning rate</i> ($0 < \alpha < 1$)
δ_{net_j}	= nilai untuk menghitung kesalahan pada <i>hidden layer</i>
δ_j	= nilai kesalahan pada <i>hidden layer</i>
Δv_{ji}	= perubahan bobot antara <i>hidden layer</i> dengan <i>input layer</i>

2.5 Algoritma Al-Alaoui

Profesor Mohamad Adnan Al-Alaoui menjelaskan bahwa algoritma *Al-Alaoui* berfungsi mengklasifikasi pola. Pada dasarnya menurut Patterson, Womack's dan Wee's telah membuktikan bahwa solusi *Mean Square Error (MSE)* dari permasalahan klasifikasi pola menghasilkan pendekatan mean-square-error minimum terhadap Bayes's discriminant. Untuk itu algoritma ini melakukan proses pengenalan ulang dan penduplikasian pasangan data latih yang masih salah dalam proses klasifikasi [SAH-12].

2.6 Algoritma Al-Alaoui Backpropagation

Berikut ini langkah-langkah Algoritma *Al-Alaoui* untuk jaringan syaraf tiruan sebagai berikut :

1. Inisialisasi parameter dan tentukan nilai misclassification error rate.
2. Lakukan backpropagation sebanyak 100 epoch.
3. Jika jumlah maksimum epochs sudah tercapai, maka berhenti. Jika tidak, maka lanjut ke langkah keempat.
4. Tes input yang asli sebelum melakukan duplikasi, kemudian tentukan persentase nilai error kesalahan klasifikasi.
5. Jika nilai error rate yang diinginkan sudah diperoleh, maka berhenti. Jika tidak lanjutkan ke langkah keenam.
6. Data yang masih mengalami kesalahan klasifikasi akan diduplikasi, kemudian ditambahkan ke dalam data latih. Kembali ke langkah 2.

Langkah 6 merupakan perbedaan algoritma *Al-Alaoui backpropagation* dan *backpropagation*. Pada algoritma *Al-Alaoui backpropagation*, data pada data latih yang masih mengalami kesalahan klasifikasi, diduplikasikan lalu dimasukkan kedalam data latih untuk digunakan pada iterasi selanjutnya, proses duplikasi tersebut menyebabkan jumlah data dalam populasi data latih bertambah di setiap iterasinya[ALA-07].

2.7 Tingkat akurasi

Tingkat akurasi merupakan proses klasifikasi, jika nilai dari tingkat akurasi semakin besar maka menunjukkan semakin akurat proses klasifikasi ini dan hasilnya dinyatakan dalam persentase.

$$\text{Tingkat akurasi} = \frac{\text{data uji benar}}{\text{jumlah total data uji}} \times 100\% \quad (2.13)$$

Akurasi peramalan dapat diukur menggunakan metode lain untuk mengukur tingkat akurasi yaitu *Mean Squared Error* (MSE). *Mean Squared Error* (MSE) merupakan rata-rata jumlah kuadrat kesalahan peramalan[DWS-12]. Semakin kecil nilai MSE maka semakin kecil tingkat kesalahan dalam memprediksi [MOE-04].

$$MSE = \frac{1}{n} \sum_{t=1}^n (Y_t - Y'_t)^2 \quad (2.14)$$

Dimana:

Y_t = nilai observasi

Y'_t = nilai perkiraan

n = jumlah data

t = target

2.8 Variasi Backpropagation

2.8.1 Normalisasi dan Denormalisasi

Normalisasi digunakan untuk menyederhanakan skala data antara 0.1 – 0.9. Berikut ini persamaan 2.15[SIA-05] :

$$X' = \frac{0.8(x - \text{min value})}{(\text{max value} - \text{min value})} + 0.1 \quad (2.15)$$

Denormalisasi data adalah proses mengembalikan data yang telah dinormalisasi menjadi data nyata. Berikut ini persamaan 2.16 [SIA-05] :

$$x'' = \frac{(\max\ value - \min\ value)(x' - 0.1)}{0.8} + \min\ value \quad (2.16)$$

Dimana :

x = data

x' = hasil normalisasi

x'' = hasil denormalisasi

$\min\ value$ = nilai minimum data

$\max\ value$ = nilai maksimum data

2.8.2 Momentum

Penambahan momentum dilakukan untuk menghindari perubahan bobot yang mencolok akibat adanya data yang sangat berbeda dengan yang lain. Berikut ini akan ditunjukkan pada persamaan 2.17 dan 2.18 [SIA-05].

$$\Delta w_{kj} = (\alpha \delta_k z_j) + (\eta \Delta w_{kj}) \quad (2.17)$$

dan

$$\Delta v_{ji} = (\alpha \delta_j x_i) + (\eta \Delta v_{ji}) \quad (2.18)$$

Dimana :

Δw_{kj} = perubahan bobot antara *output layer* dengan *hidden layer*

α = konstanta *learning rate* ($0 < \alpha < 1$)

μ = konstanta momentum ($0.5 \leq \mu \leq 0.9$)

δ_k = nilai kesalahan pada *hidden layer*

δ_j = nilai kesalahan pada *hidden layer*

Δv_{ji} = perubahan bobot antara *hidden layer* dengan *input layer*

2.8.3 Jumlah Hidden Layer

Untuk mengetahui jumlah hidden layer dapat ditunjukkan pada persamaan 2.19. Pada persamaan ini jumlah input yang telah ditetapkan akan di kalikan 2/3 dari jumlah input dan ditambah.jumlah output.

Berikut ini persamaan 2.19 [HEA-08].

$$\text{hidden layer} = \left(\frac{2}{3} \times \text{jumlah input}\right) + \text{jumlah output} \quad (2.19)$$



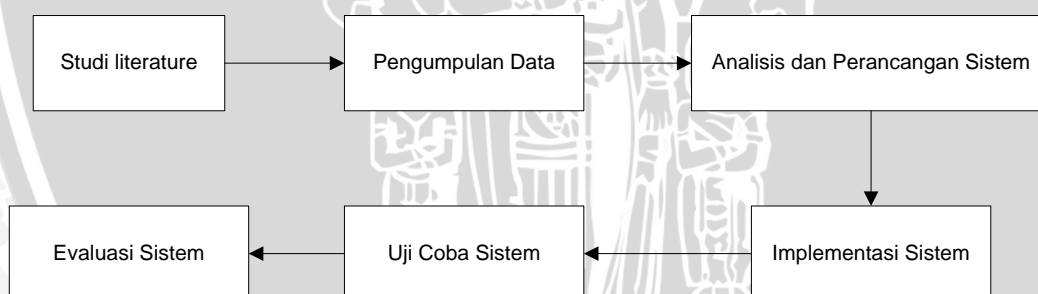
BAB III

METODOLOGI DAN PERANCANGAN

Dalam bab metodologi dan perancangan ini akan dibahas metode perancangan yang digunakan dan langkah-langkah yang dilakukan dalam penelitian menggunakan metode *al-alaoui backpropagation*.

Penelitian dilakukan dengan langkah-langkah sebagai berikut :

1. Mempelajari Studi literature yang terkait dengan masalah curah hujan dan metode *al-alaoui backpropagation*.
2. Melakukan pengumpulan data – data curah hujan dari BMKG Karang Ploso.
3. Menganalisis sistem dari data curah hujan dan melakukan perancangan sistem.
4. Mengimplementasikan sistem untuk membuat perangkat lunak berdasarkan analisis dan perancangan yang telah dilakukan.
5. Melakukan uji coba terhadap perangkat lunak.
6. Mengevaluasi uji coba terhadap perangkat lunak.



Gambar 3.1 Diagram alir penelitian

Sumber : [Perancangan]

3.1 Studi Literatur

Pada tahap ini dilakukan studi literatur yang dapat mendukung proses pembuatan sistem peramalan curah hujan menggunakan metode *al-alaoui backpropagation*. Studi literatur ini diambil dari berbagai sumber, seperti buku, jurnal, serta sumber lain yang dapat memberikan wawasan pada penelitian ini.

3.2 Data Penelitian

Data yang digunakan dalam sistem peramalan curah hujan menggunakan algoritma *Al-Alaoui Backpropagation* dari BMKG Karangploso Kabupaten Malang. Data berupa data curah hujan harian sebanyak 456 data, mulai bulan Januari 2011 sampai maret 2012.

3.3 Analisis dan Perancangan Sistem

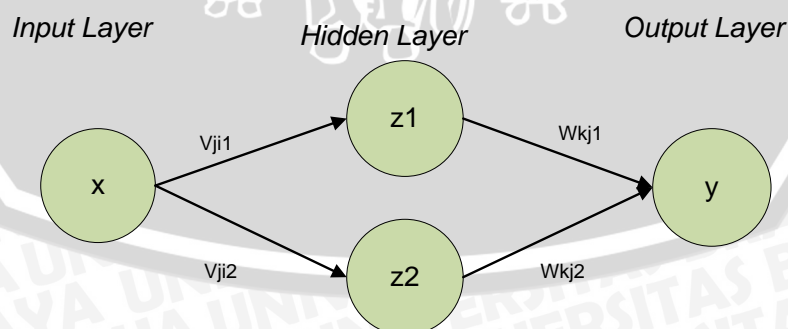
3.3.1 Deskripsi umum sistem

Sistem yang akan dibuat berupa program aplikasi yang digunakan untuk peramalan curah hujan menggunakan metode *al-alaoui backpropagation*. Metode ini dapat mengenali pola *output* untuk menentukan jumlah curah hujan dari data curah hujan harian, sehingga akan dapat diramalkan curah hujan pada hari berikutnya.

Proses awal dari sistem ini harus memperoleh data curah hujan terlebih dahulu. Setelah data diperoleh, akan dilakukan proses pelatihan Jaringan Syaraf Tiruan (JST) yang terdiri dari JST *backpropagation* dengan algoritma *al-alaoui*. Proses pelatihan ini akan menghasilkan nilai yang sudah tersimpan, digunakan untuk melakukan peramalan.

3.3.2 Perancangan proses

Arsitektur Jaringan Syaraf Tiruan yang akan digunakan ditunjukkan pada gambar 3.2



Gambar 3.2 Arsitektur Jaringan Pada Sistem

Sumber : [Perancangan]

Arsitektur yang digunakan dalam sistem ini terdiri dari 3 lapisan yaitu *input* (X) adalah curah hujan, lapisan *hidden* (Z1) adalah hidden layer ke 1, (Z2) adalah hidden layer ke 2 serta lapisan *output* (Y) adalah target curah hujan. Pada setiap *hidden layer* jumlahnya akan sesuai dengan persamaan 2.19, sedangkan pada *input* dan *output* jumlah *neuron* hanya 1. Proses yang ada dalam sistem peramalan curah hujan ini di bagi beberapa tahap yaitu pengambilan data curah hujan yang selanjutnya dinormalisasi, maka data siap untuk dilatih dalam jaringan saraf tiruan. Ditentukan data latih dengan jumlah curah hujan harian sebagai *input* dan hari selanjutnya sebagai target.

Data dari BMKG dapat disusun suatu pola data latih sebagai masukan dalam pelatihan jaringan syaraf tiruan. Sebagai contoh, apabila menggunakan 5 *input* data dari jumlah curah hujan harian, maka hari ke-1 sebagai x_1 dan pola target/keluaran yaitu jumlah curah hujan hari ke-2 sebagai t . Hari ke-2 sebagai x_2 dan pola target pada hari ke-3, proses ini akan berhenti jika mencapai hari ke-5 sebagai x_5 dan pola target/keluaran adalah hari ke-6. Tabel pola data ditunjukkan pada table 3.1.

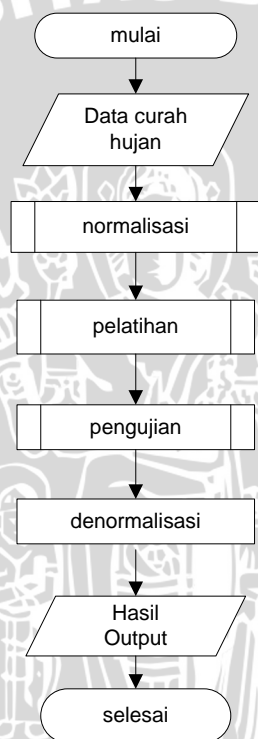
Tabel 3.1 Pola Data *Input* dan Target

No	Data	Target
X_1		
X_2		
X_3		
X_4		
X_5		

Setiap data *input* satu persatu akan di *feed forward* ke dalam JST. *Output* (Y_k) akan dibandingkan dengan target, apabila selesih *output* dengan target tidak sesuai dengan galat yang ditetapkan, maka akan dilakukan *backpropagation* untuk memperbaiki bobot jaringan. Setelah itu data akan diduplikasikan dan ditambah untuk proses pelatihan di *epoch* selanjutnya.

Pelatihan ini akan berhenti hingga jika tidak ada data yang mengalami misklarifikasi dan *epoch* maksimum telah telah tercapai. Pengujian akan dilakukan untuk menghitung tingkat akurasi yang terbaik. Data yang di uji dibuat dengan

cara yang sama dengan data latih. Proses pengujian struktur JST sama seperti proses pelatihan ditentukan *input*, *hidden* dan *output*, kemudian untuk perhitungan tingkat akurasi JST setiap data di *feedforward*kan ke dalam JST. Nilai *output* akan dibandingkan dengan nilai target. Setelah itu dengan membandingkan jumlah setiap data uji dengan jumlah seluruh data yang di uji akan didapatkan tingkat akurasi. Hasil pengujian akan menghasilkan *output* berupa curah hujan yang diramalkan, tetapi hasil yang di uji masih berupa data normalisasi sehingga perlu denormalisasi agar didapatkan nilai *output* yang nyata. Alur proses deskripsi sistem secara umum ditunjukkan pada gambar 3.3.



Gambar 3.3 Deskripsi sistem secara umum

Sumber : [Perancangan]

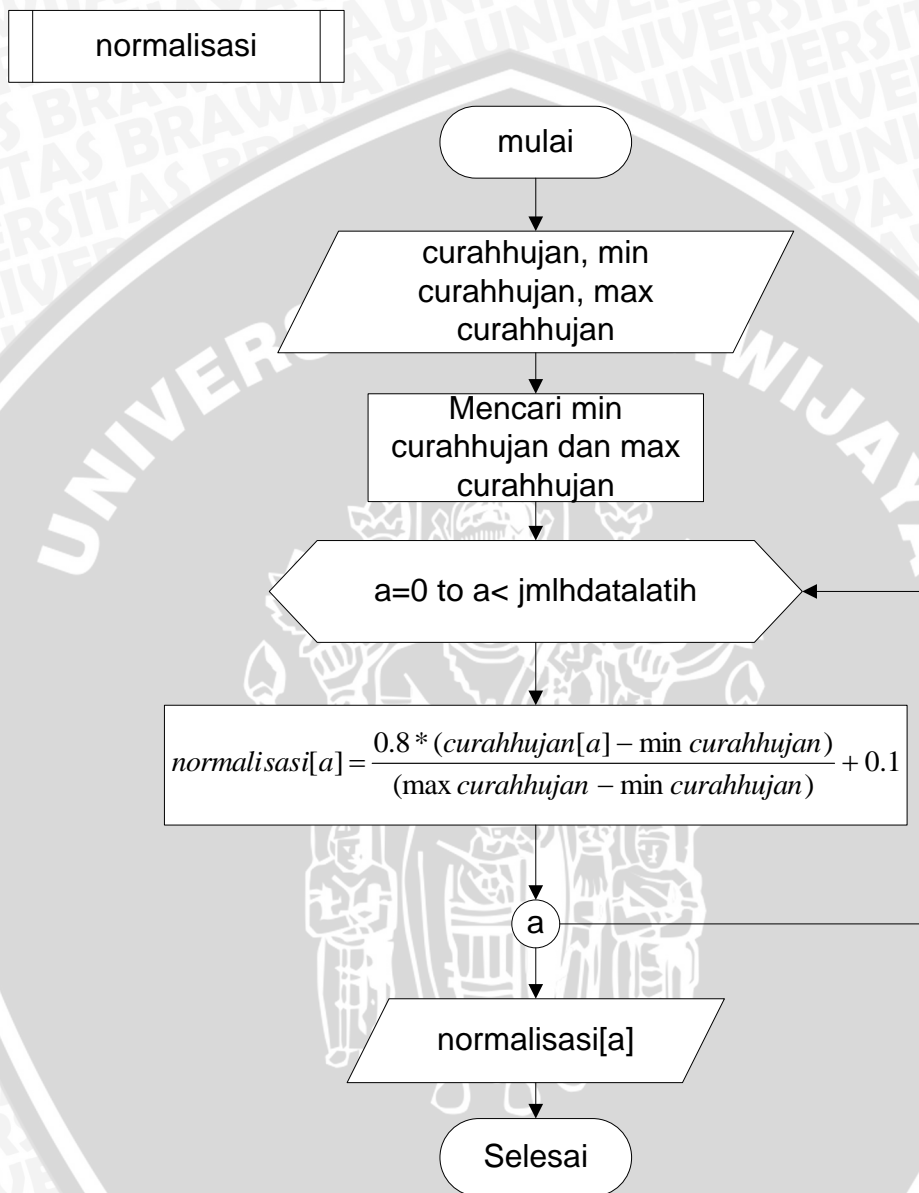
3.3.2.1 Proses Normalisasi Data

Proses normalisasi dilakukan sebelum proses pelatihan, semua data curah hujan akan dinormalisasi agar berada pada range 0,1-0,9.

Proses dari perhitungannya akan dijelaskan sebagai berikut :

1. Melakukan *input* data curah hujan.
2. Mencari nilai data yang maksimal dan minimum.

3. Melakukan perulangan
4. Menghitung normalisasi curah hujan sebanyak data curah hujan yang di inputkan.

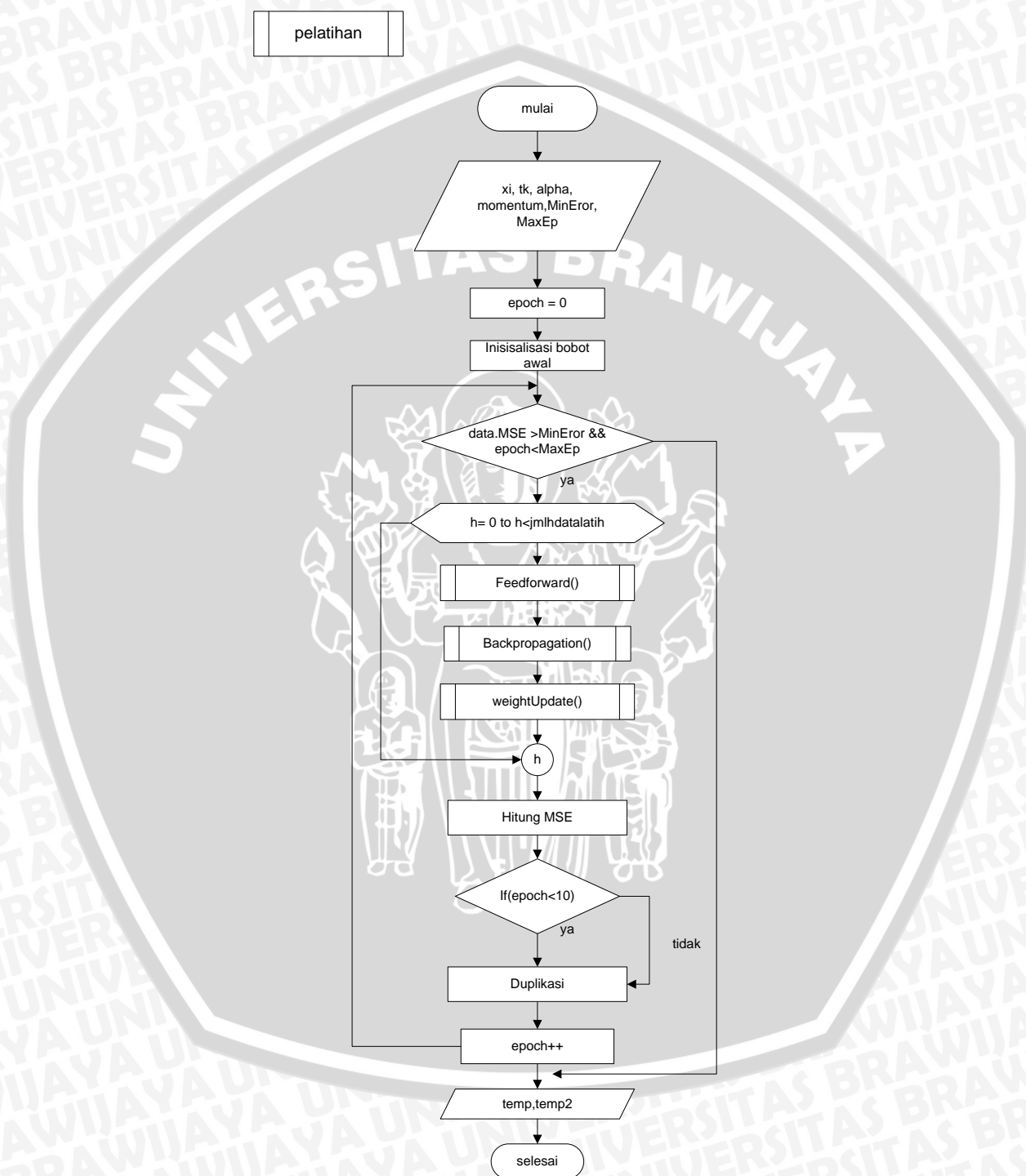


Gambar 3.4 Proses Normalisasi Data

Sumber : [Perancangan]

3.3.2.2 Proses Pelatihan Jaringan Syaraf Tiruan *Al-Alaoui Backpropagation*

Flowchart dari proses pelatihan *Al-Alaoui backpropagation* dapat dilihat pada gambar 3.5.



Gambar 3.5 Proses Pelatihan JST *Al-Alaoui backpropagation*

Sumber : [Perancangan]

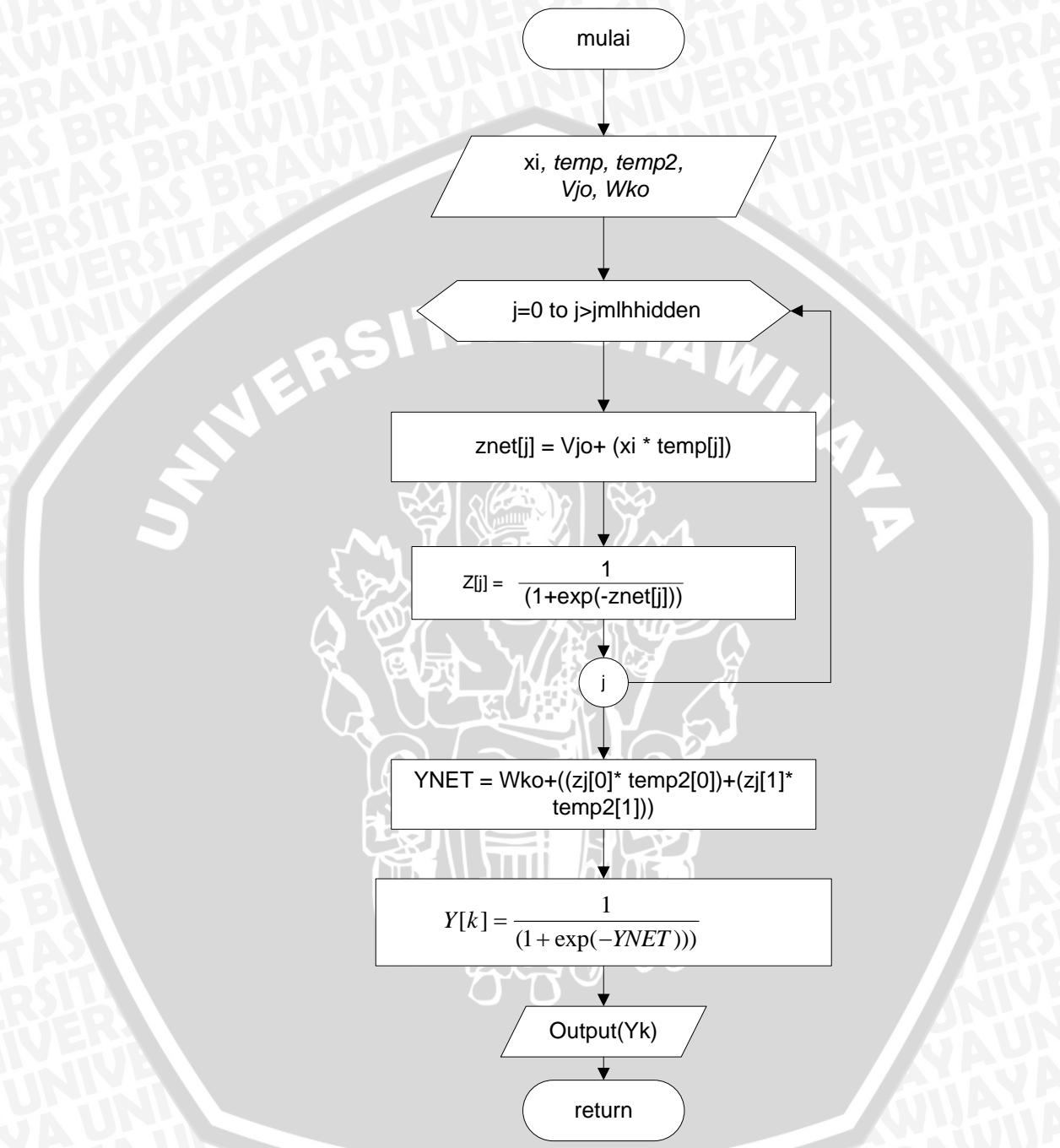
Setelah semua data dinormalisasi, maka akan ditentukan data latih yang kemudian akan menjadi masukan dalam proses pelatihan JST. Proses pelatihan Jaringan Syaraf Tiruan memiliki beberapa langkah, yaitu :

1. Masukkan data input yang akan dilatih, learning rate, momentum, maksimal *epoch*, minimal *error*.
2. Menentukan inisialisasi bobot awal v_{ji} dan w_{kj} .
3. Nilai *epoch* untuk iterasi pertama mulai dari 0
4. Membandingkan nilai error dari MSE dengan minimal eror yang telah ditetapkan. Jika nilai error lebih besar dari minimal eror dan $epoch < Maxepoch$, maka dilakukan langkah selanjutnya lanjut langkah ke 4 hingga langkah 12. Jika tidak proses pelatihan selesai.
5. Dilakukan perhitungan *feedforward* kedalam JST yang menghasilkan y_k (nilai output).
6. Melakukan perhitungan *backpropagation* untuk menghitung nilai perbaikan setiap bobot.
7. Tahap selanjutnya menghitung *weight update* untuk mendapatkan bobot baru.
8. Kemudian nilai output (y_k) dari perhitungan *feedforward* dihitung nilai error (MSE) menggunakan persamaan 2.14.
9. Duplikasi data digunakan untuk memperoleh iterasi selanjutnya.
10. Jika seluruh data telah diproses maka dilakukan langkah selanjutnya. Jika belum kembali ke langkah 4.
11. Nilai iterasi ditambahkan.
12. Kembali ke langkah 3 dengan membandingkan nilai *epoch*.

3.3.2.3 Proses Feedforward

Pada proses ini, data *input* akan menuju sampai *hidden layer* dan akan di aktivasi menggunakan fungsi sigmoid yang ditunjukkan pada persamaan 2.1. Hasil aktivasi kemudian menuju lapisan *output*. Langkah-langkah perhitungan feed forward dapat dilihat pada sub bab 2.5.2

FeedForward



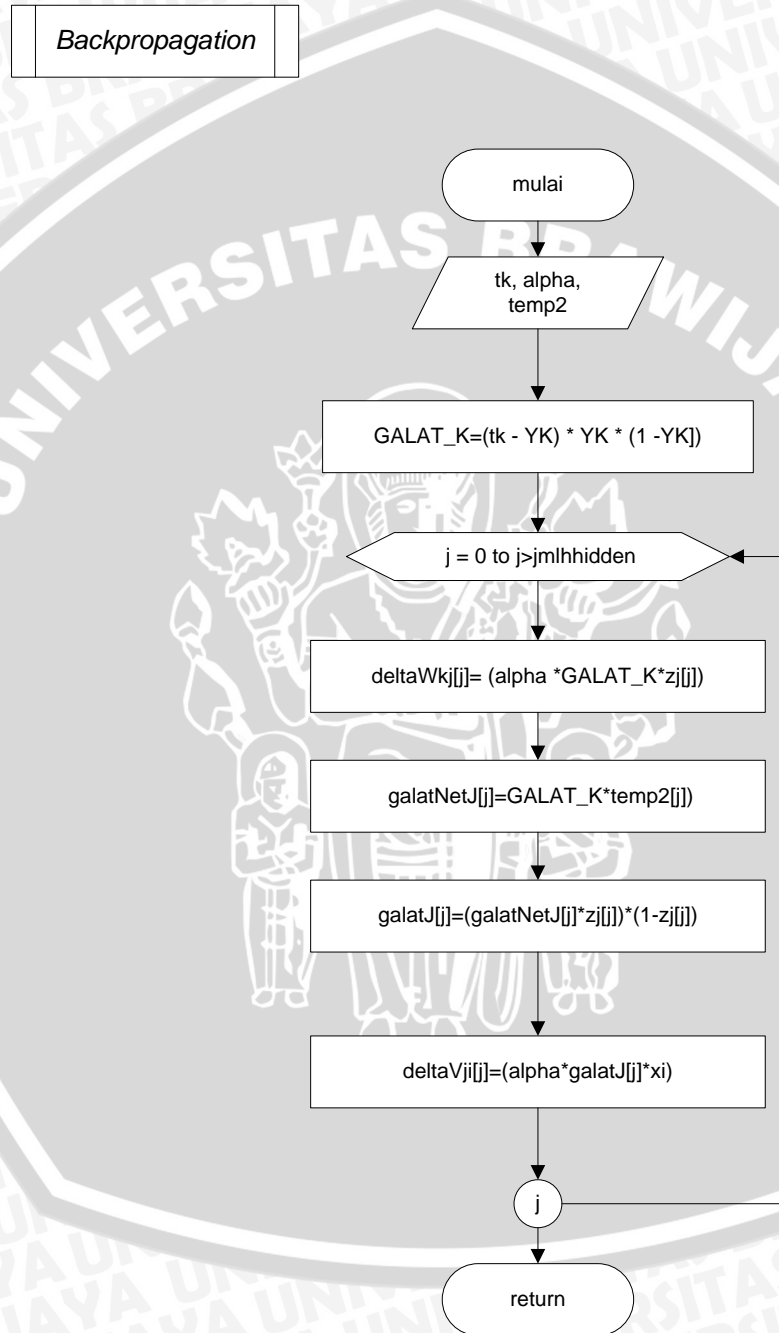
Gambar 3.6 Proses *feedforward*

Sumber : [Perancangan]



3.3.2.4 Proses Backpropagation

Proses *backpropagation* ini untuk memperbaiki bobot-bobot sebelumnya agar lebih optimal. Flowchart untuk *backpropagation* dapat dilihat pada gambar 3.7 dan langkah-langkah perhitungannya dijelaskan pada sub bab 2.5.2.

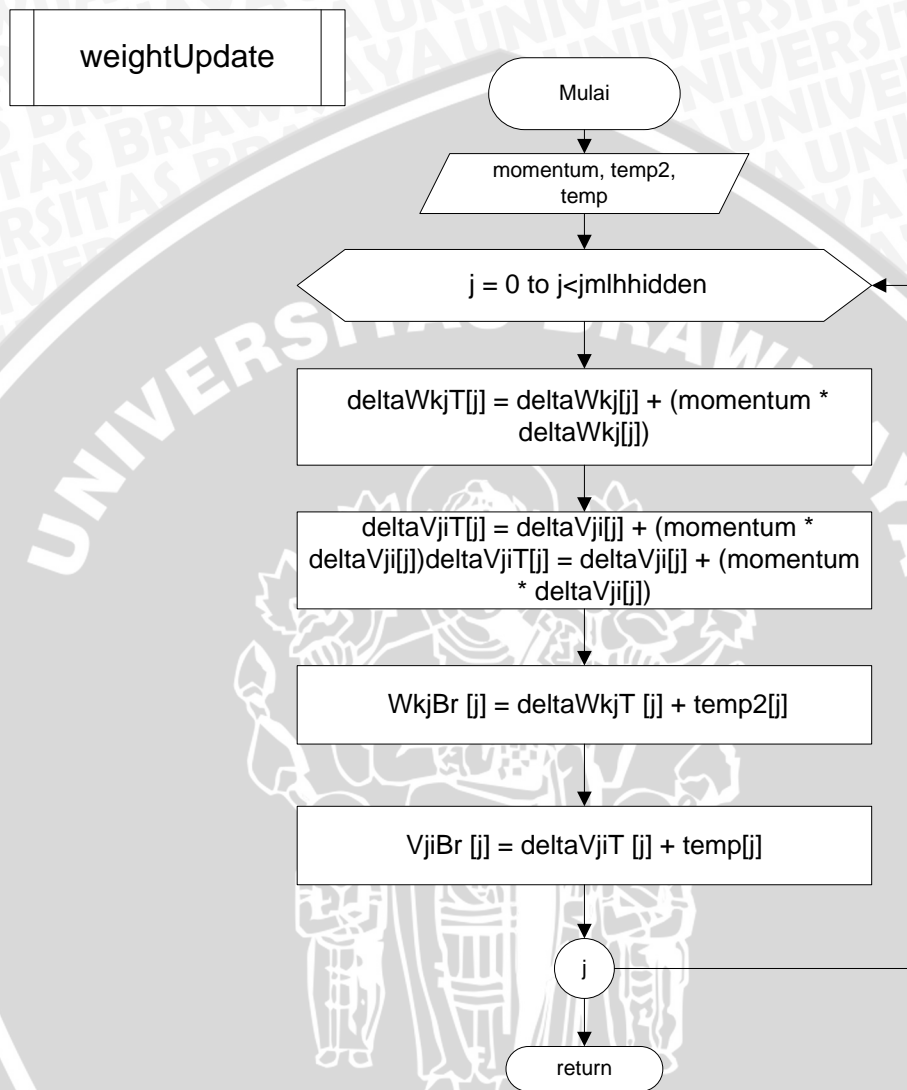


Gambar 3.7 Proses *Backpropagation*

Sumber : [Perancangan]

3.3.2.5 Proses *Weight Update*

Flowchart untuk *weight update* dapat dilihat pada gambar 3.8 dan langkah-langkah perhitungannya dijelaskan pada sub bab 2.5.2.



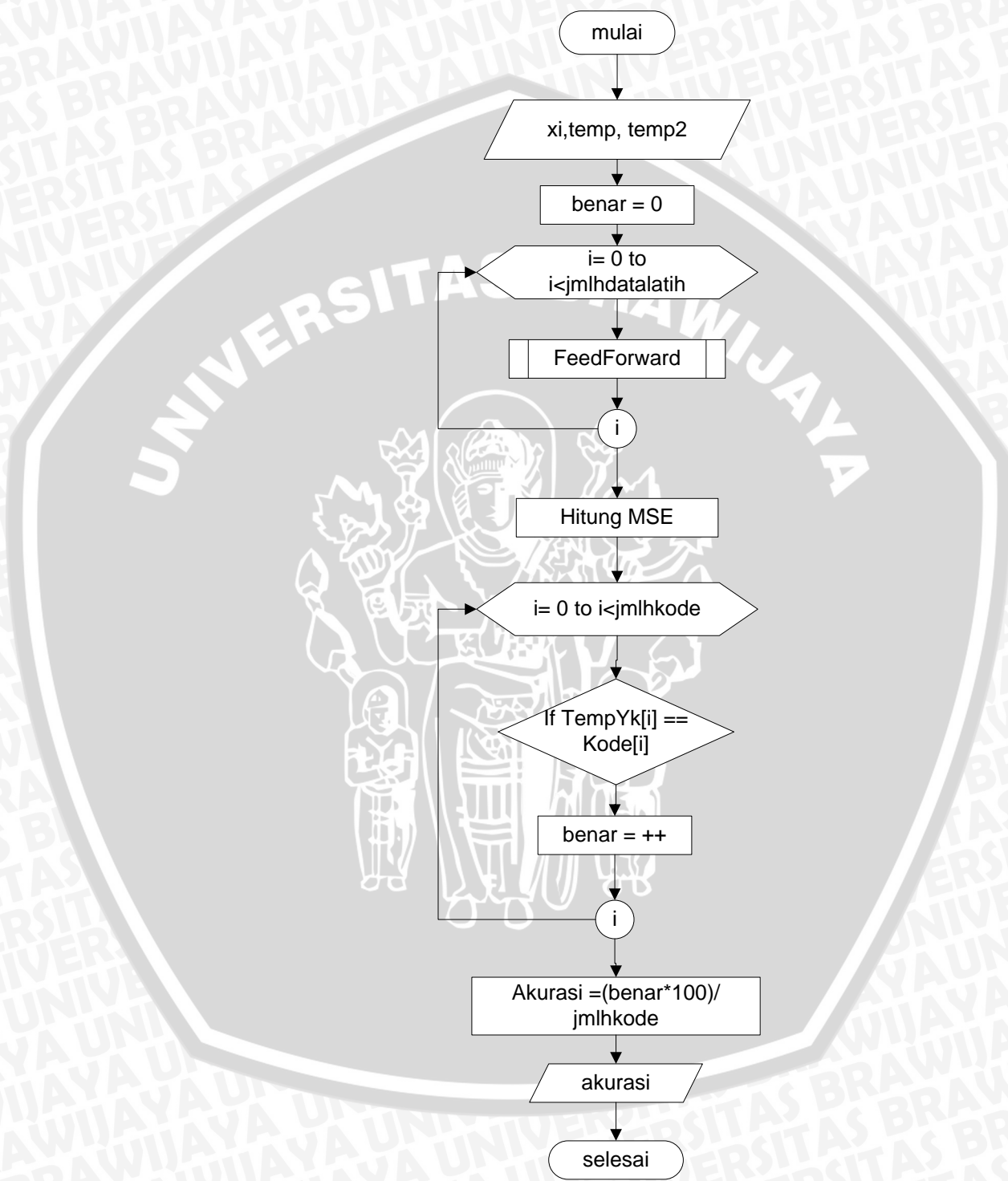
Gambar 3.8 Proses *weight update*

Sumber : [Perancangan]

3.3.2.6 Proses Pengujian JST *Al-alaoui Backpropagation*

Proses pengujian dari hasil penelitian ini merupakan pengecekan tingkat kebenaran sistem dalam peramalan curah hujan pada hari berikutnya. Data hasil pelatihan kemudian dihitung tingkat kebenarannya dengan data *real*. Langkah-langkah pengujian ditunjukkan pada gambar 3.9.

pengujian



Gambar 3.9 Proses Pengujian JST Al-Aloui Backpropagation

Sumber : [Perancangan]



Untuk menghitung tingkat akurasi diperlukan data uji yang memiliki strukturnya sama dengan data latih. Proses pengujian JST *al-alaoui backpropagation* dijelaskan pada langkah berikut ini :

1. Memasukkan x_i berupa *input* dan t_k berupa target yang akan di uji.
2. Variabel benar untuk inisialisasi awal adalah 0 dan variabel ini akan digunakan untuk menghitung tingkat akurasi.
3. Menghitung *feedforward* ke dalam JST untuk menghasilkan nilai *output*. Proses Flowchart ditunjukkan pada gambar 3.8.
4. Jika hasil *output* memiliki nilai yang sama dengan target maka dijalankan tahap selanjutnya nilai benar akan bertambah dan jika tidak proses akan berulang.
5. Tingkat akurasi JST dihitung menggunakan persamaan 2.13

3.4 Contoh Perhitungan Manual

3.4.1 Proses penginputan data

Diketahui data curah hujan hari ke-1 hingga hari ke-5 bulan januari 2011. Berikut ini contoh perhitungan hari ke-1 dengan *epoch* ke 0 sebagai berikut:

Tabel 3.2 Data asli curah hujan

Hari ke-	Jumlah curah hujan (mm)
1	12.3
2	0.8
3	1.2
4	4.6
5	0.0

Dari data asli tersebut akan dicari proses pelatihan hari ke-1 sebagai x_1 dan hari ke-2 sebagai target, proses ini akan berlanjut sampai hari ke-5 sebagai x_5 dan hari ke-6 sebagai target. Berikut ini pola data pada table 3.3.

Tabel 3.3 Pola Data asli curah hujan

No	Data	Target
x_1	12.3	0.8
x_2	0.8	1.2
x_3	1.2	4.6
x_4	4.6	0.0

x_5	0.0	3.8
-------	-----	-----

Data kemudian dinormalisasi menggunakan persamaan. Berikut ini table hasil normalisasi data terlihat pada table 3.4

$$x' = \frac{0.8(x - hmin)}{(hmax - hmin)} + 0.1$$

$$x' = \frac{0.8(12.3 - 0.0)}{(12.3 - 0.0)} + 0.1$$

$$= 0.9$$

Tabel 3.4 Pola Data Normalisasi

No	Data	Target
x1	0.9	0.15203252
x2	0.15203252	0.17804878
x3	0.17804878	0.399186992
x4	0.399186992	0.1
x5	0.1	0.347154472

Struktur Jaringan Syaraf Tiruan akan ditentukan untuk pelatihan sebagai berikut :

- Jumlah *input* = 5
- Jumlah *hidden layer* = 2
- Jumlah *output* = 1
- α (learning rate) = 0.1
- η (momentum) = 0.5
- δ (galat) = 0.01

3.4.2 Proses Inisialisasi Bobot awal

Pertama inisialisasi bobot awal v_{ij} dan w_{jk} dengan mengacak nilai data.

Dapat dilihat pada tabel 3.5 dan table 3.6 sebagai berikut :

Tabel 3.5 Data Nilai bobot Awal v_{ij}

I	J	
	1	2
1	0.065509423	0.612158292

Tabel 3.6 Data Nilai bobot Awal w_{jk}

K	J	
	1	2
1	0.403209801	0.139595752

3.4.3 Proses FeedForward

Proses feedforward bertujuan untuk mendapatkan nilai sebuah neuron pada lapisan berikutnya. Semua keluaran unit tersembunyi dihitung Z_j pada persamaan 2.2 dan fungsi aktivasi persamaan 2.3. Berikut ini contoh perhitungan Data ke-x1 yang ditunjukkan pada pada 3.7 dan tabel 3.8.

$$\begin{aligned}
 z_{\text{net}_j} &= V_{j0} + \sum_{i=1}^n x_i \cdot v_{ji} \\
 &= 1 + (0.9 \times 0.065509423) \\
 &= 1.058958481
 \end{aligned}$$

Tabel 3.7 Data Znetj

znetj	J	
	1	2
1	1.058958481	1.550942463
2	1.009797182	1.093025527
3	1.01144268	1.10893468
4	1.025603473	1.244217734
5	1.006364729	1.061166027

$$\begin{aligned}
 z_j &= f(z_{\text{net}_j}) = \frac{1}{1 + \exp(-z_{\text{net}_j})} \\
 &= \frac{1}{1 + \exp(1.058958481)} \\
 &= 0.742491459
 \end{aligned}$$

Tabel 3.8 Data Zj

Zj	J	
	1	2
1	0.742491459	0.825049811
2	0.732980456	0.74895102
3	0.733302389	0.751930449

4	0.73606265	0.776297319
5	0.732308118	0.742913311

Untuk keluaran unit output dihitung dengan dengan persamaan 2.4 dan fungsi aktivasi dihitung menggunakan persamaan 2.5, sehingga menghasilkan nilai y_{netk} dan y_k .

$$\begin{aligned}
 y_{netk} &= W_{ko} + \sum_{j=1}^p z_j \cdot w_{kj} \\
 &= (1 + (0.742491459 \times 0.403209801) \\
 &\quad + (0.825049811 \times 0.139595752)) \\
 &= 1.414553282
 \end{aligned}$$

Tabel 3.9 Data y_{netk}

YnetK	
1	1.414553282
2	1.382205561
3	1.366257502
4	1.359278618
5	1.335325231

$$\begin{aligned}
 y_k &= f(y_{netk}) = \frac{1}{1 + \exp(-y_{netk})} \\
 &= \frac{1}{1 + \exp(-1.414553282)} \\
 &= 0.804483123
 \end{aligned}$$

Tabel 3.10 Data y_k

Yk	
1	0.804483123
2	0.799344989
3	0.796774824
4	0.795642429
5	0.791720126

3.4.4 Proses Backpropagation

Berdasarkan persamaan 2.6 dapat dihitung nilai aktivasi output dari nilai δ_k masing-masing neuron output data ke-1. δ_k merupakan unit kesalahan yang dipakai dalam perubahan bobot layar δ_{netj} .

$$\begin{aligned} \delta_k &= (t_k - y_k) y_k (1 - y_k) \\ &= (0.152032520325203 - 0.804483123)0.804483123 (1 - 0.804483123) \\ &= -0.102623974 \end{aligned}$$

Tabel 3.11 Data δ_k

δ_k	
1	-0.102623974
2	-0.0996513
3	-0.064379292
4	-0.113108366
5	-0.073308595

Menghitung perubahan bobot w_{kj} dengan percepatan α dengan persamaan 2.7, hasilnya akan ditunjukkan pada tabel 3.12.

$$\begin{aligned} \Delta w_{kj} &= \alpha \delta_k z_j \\ &= 0.5 \times (-0.102623974) \times 0.742491459371648 \\ &= -0.00761974238867841 \end{aligned}$$

Tabel 3.12 Data Δw_{kj}

ΔW_{kj}	J	
	1	2
1	-0.007619742	-0.008466989
2	-0.007304246	-0.007463394
3	-0.004720949	-0.004840875
4	-0.008325484	-0.008780572
5	-0.005368448	-0.005446193

Menghitung faktor δ layer tersembunyi data ke-x1 berdasarkan kesalahan pada setiap layer tersembunyi z_j menggunakan persamaan 2.8. Hasilnya ditunjukkan pada tabel 3.13.

$$\begin{aligned} \delta_{net_j} &= \sum_{k=1}^m \delta_k \cdot w_{kj} \\ &= -0.102623974 \times 0.403209800907084 \end{aligned}$$

$$= 0.733302389$$

Tabel 3.13 Data δ_{netj}

δ_{netj}	J	
	1	2
1	0.733302389	0.751930449
2	-0.039041405	-0.012645278
3	-0.024517168	-0.007448695
4	-0.042273392	-0.012265343
5	-0.026483036	-0.006983964

Dihitung juga faktor kesalahan δ_j di layer tersembunyi dengan persamaan

2.9 dan hasilnya diperoleh pada tabel 3.14

$$\begin{aligned}\delta_j &= \delta_{netj} \cdot z_j(1 - z_j) \\ &= 0.733302389 \times 0.742491459(1 - 0.742491459) \\ &= -0.00791157603802451\end{aligned}$$

Tabel 3.14 Data δ_j

δ_j	J	
	1	2
1	-0.007911576	-0.002067834
2	-0.007641188	-0.002377608
3	-0.004794822	-0.001389413
4	-0.008212639	-0.002129997
5	-0.005191547	-0.001333889

Setelah didapatkan nilai kemelesetan tiap-tiap layer, maka dihitung perubahan bobot dari v_{ji} data ke- 1 dari persamaan 2.10 dan hasilnya diperoleh pada tabel 3.15.

$$\begin{aligned}\Delta v_{ji} &= \alpha \delta_j x_i \\ &= 0.5 \times (-0.0764) \times 0.9 \\ &= -0.000712042\end{aligned}$$

Tabel 3.15 Data Δv_{ji}

Δv_{ji}	J	
	1	2
1	-0.000712042	-0.000186105
2	-0.000116171	-3.61474E-05
3	-8.53712E-05	-2.47383E-05
4	-0.000327838	-8.50267E-05
5	-5.19155E-05	-1.33389E-05

3.4.5 Proses Weight Update

Proses weight update digunakan untuk menghitung perubahan bobot baris yang menuju unit keluaran dan lapisan tersembunyi pada persamaan 2.11 dan 2.12. Sebelum menghitung persamaan di atas dihitung nilai Δw_{kj} dan Δv_{ji} .

Hasilnya dapat dilihat dalam tabel 3.16, tabel 3.17, tabel 3.18 dan tabel 3.19

$$\begin{aligned}
 \Delta w_{kj} &= (\alpha \delta_k z_j) + (\eta \Delta w_{kj}) \\
 &= (0.5 \times (-0.102623973548825) \times 0.742491459371648) \\
 &\quad + (0.1 \times (-0.00761974238867841)) \\
 &= -0.011429614
 \end{aligned}$$

Tabel 3.16 Data Δw_{kj}

$\Delta W_{kj}(t)$	J	
	1	2
1	-0.011429614	-0.012700483
2	-0.010956368	-0.011195091
3	-0.007081423	-0.007261312
4	-0.012488227	-0.013170858
5	-0.008052672	-0.00816929

$$\begin{aligned}
 \Delta v_{ji} &= (\alpha \delta_j x_i) + (\eta \Delta v_{ji}) \\
 &= (0.5 \times (-0.00791157603802451) \times 0.9) \\
 &\quad + (0.1 \times (-0.000712041843422206)) \\
 &= -0.00106806276513331
 \end{aligned}$$

Tabel 3.17 Tabel Data Δv_{ji}

$\Delta v_{ji}(t)$	J	
	1	2
1	-0.001068063	-0.000279158
2	-0.000174256	-5.42211E-05
3	-0.000128057	-3.71075E-05
4	-0.000491757	-0.00012754
5	-7.78732E-05	-2.00083E-05

$$\begin{aligned}
 w_{kj}(\text{baru}) &= w_{kj}(\text{lama}) + \Delta w_{kj} \\
 &= 0.403209800907084 + (-0.0114296135830176) \\
 &= 0.391780187
 \end{aligned}$$

Tabel 3.18 Data Δw_{kj}

wkj (baru)	J	
	1	2
1	0.391780187	0.126895268
2	0.380823819	0.115700177
3	0.373742396	0.108438864
4	0.361254169	0.095268006
5	0.353201497	0.087098716

$$\begin{aligned}
 v_{ji}(\text{baru}) &= v_{ji}(\text{lama}) + \Delta v_{ji} \\
 &= 0.0655094233760277 + (-0.00106806276513331) \\
 &= 0.0644413606108944
 \end{aligned}$$

Tabel 3.19 Tabel Data Δv_{ji}

vji (baru)	J	
	1	2
1	0.064441361	0.611879135
2	0.064267104	0.611824914
3	0.064139047	0.611787806
4	0.063647291	0.611660266
5	0.063569417	0.611640258

$$MSE = \frac{1}{n} \sum_{t=1}^n (Y_t - \hat{Y}'_t)^2$$

$$\begin{aligned}
&= (0.152032520325203 - 0.804483122539015)^2 \\
&\quad + (0.178048780487804 - 0.799344989374883)^2 + \dots \\
&\quad + (0.347154471544715 - 0.79172012614916)^2 \\
&= 0.530266772415335
\end{aligned}$$

Hasil dari nilai *error* MSE dibandingkan dengan minimal *error* yang telah ditetapkan. Jika *error* $0.530266772415335 > 0.01$, karena nilai *error* lebih besar dari minimal *error* maka proses dilakukan sampai nilai MSE kurang dari minimal *error*.

3.4.6 Perhitungan Denormalisasi

Pada perhitungan diatas nilai *output* (*yk*) *epoch* terakhir pada data latih masih dalam bentuk normalisasi, untuk mengembalikan data yang telah dinormalisasi menjadi data sesungguhnya maka dilakukanlah perhitungan denormalisasi dapat ditunjukkan pada persamaan 2.16 dan pada tabel 3.20.

$$\begin{aligned}
x &= \frac{(\max\ value - \min\ value)(x' - 0.1)}{0.8} + \min\ value \\
&= \frac{(12.3 - 0.0)(0.8044831225390147 - 0.1)}{0.8} + 0.0 \\
&= 10.831428009037351
\end{aligned}$$

Tabel 3.20 Data y_k denormalisasi

Data ke -	Output Jaringan
x1	10.831428009037351
x2	10.752429211638825
x3	10.71291291494711
x4	10.695502348558966
x5	10.63519693954333

3.4.7 Perbandingan nilai *output* jaringan (*yk*) dengan *output* data asli

Kemudian langkah selanjutnya melakukan duplikasi, sebelum melakukan proses duplikasi, dilakukanlah perbandingan nilai *output* jaringan(*yk*) dengan *output* data asli berdasarkan kriteria curah hujan. Berikut ini kriteria curah hujan yang ditunjukkan pada tabel 3.21 dan hasil perbandingannya pada tabel 3.22

Tabel 3.21 kriteria curah hujan

No	Kriteria curah hujan
1	Sangat Ringan = < 5 mm
2	Ringan = $5 - < 20$ mm
3	Sedang = $20 - < 50$ mm
4	Lebat = $50 - < 100$ mm
5	Sangat Lebat = > 100 mm

Tabel 3.22 Hasil perbandingan nilai output dengan kriteria curah hujan.

Data ke -	Output Jaringan	Kriteria output jaringan	Output Data Asli	Kriteria output data asli
x1	10.831428009037351	R	0.8	SR
x2	10.752429211638825	R	1.2	SR
x3	10.71291291494711	R	4.6	SR
x4	10.695502348558966	R	0	SR
x5	10.63519693954333	R	3.8	SR

Dari hasil perbandingan diatas dapat disimpulkan bahwa nilai kedua *output* berdasarkan kriteria curah hujan mengalami perubahan sehingga perlu melakukan proses duplikasi dengan metode *al-alaoui backpropagation* sampai tidak terjadi *misclassification* atau sampai jumlah minimal *error* terpenuhi.

3.5 Proses Duplikasi Data

Pada tabel 3.22 terdapat 5 data yang salah jadi ke 5 data tersebut di duplikasi yang akan ditempatkan ke dalam index terakhir, sehingga data akan berjumlah sebanyak 10 data. Seperti ditunjukkan pada tabel 3.23 proses duplikasinya.

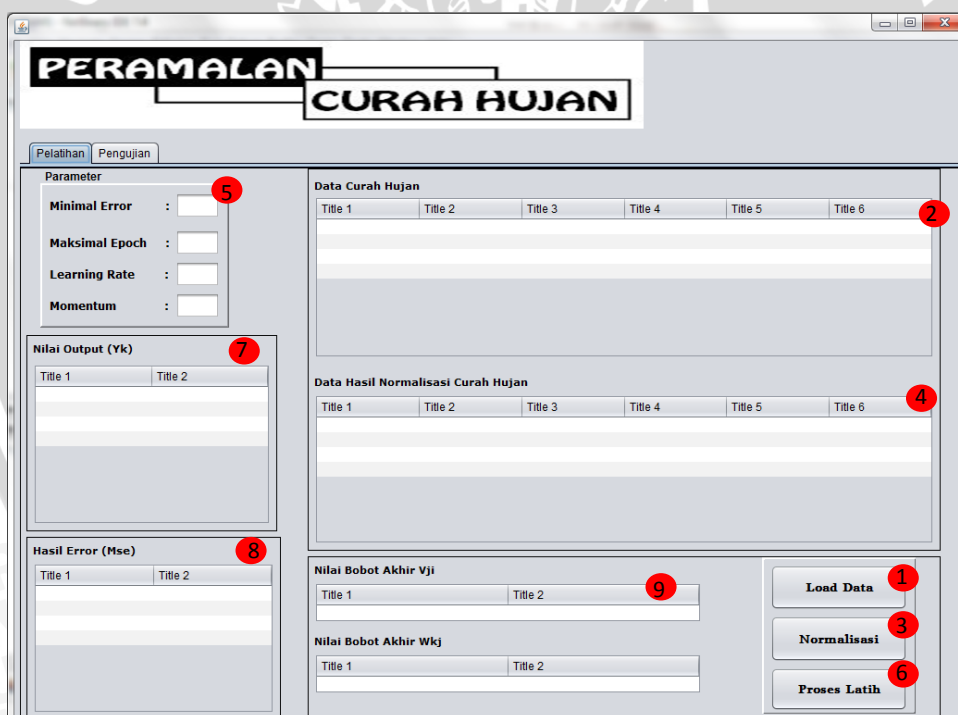
Tabel 3.23 Hasil Proses Duplikasi.

Data ke -	Output Jaringan	Kriteria output jaringan	Output Data Asli	Kriteria output data asli
x1	10.831428009037351	R	0.8	SR
x2	10.752429211638825	R	1.2	SR
x3	10.71291291494711	R	4.6	SR
x4	10.695502348558966	R	0	SR

x5	10.63519693954333	R	3.8	SR
x6	10.417880039992728	R	0.8	SR
x7	10.3643575242724	R	1.2	SR
x8	10.309966201504698	R	4.6	SR
x9	10.254707073361823	R	0	SR
x10	10.198582324558547	R	3.8	SR
x11	10.14159537592231	R	0.8	SR
x12	10.088704249633413	R	1.2	SR
x13	10.038389647730575	R	4.6	SR
x14	10.00393426357172	R	0	SR
x15	9.94950290883863	R	3.8	SR

3.6 Rancangan Antarmuka

Rancangan antarmuka yang digunakan pada sistem akan ditunjukkan pada terdiri dari 2 bagian yaitu proses pelatihan dan proses pengujian. Rancangan antarmuka untuk proses pelatihan akan ditunjukkan pada gambar 3.9 berikut ini:

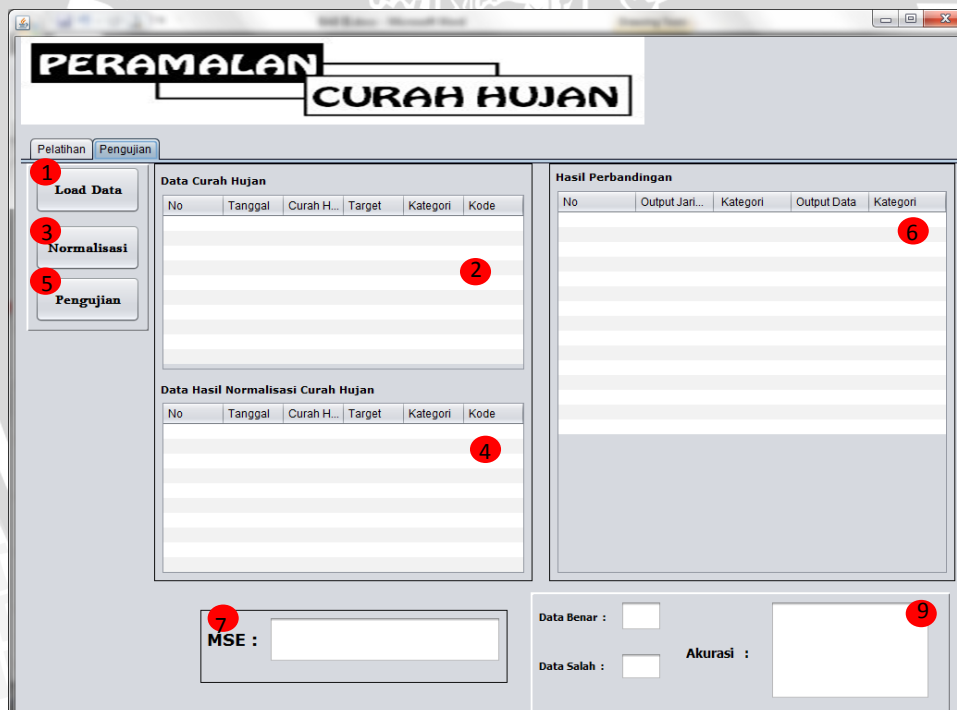


Gambar 3.9 Proses Pelatihan

1. *JButton* Load Data digunakan untuk mencari data latih yang akan diinputkan ke dalam sistem
2. *Jtable* digunakan untuk menampilkan data curah hujan.

3. *JButton* Normalisasi digunakan untuk memproses data curah hujan agar berada pada range 0.1 – 0.9.
4. *Jtable* digunakan untuk menampilkan data curah hujan yang telah dinormalisasi.
5. *JTextField* digunakan untuk menginputkan nilai minimal error, maksimal epoch, learning rate dan momentum.
6. *JButton* Proses Latih digunakan untuk memulai proses pelatihan.
7. *Jtable* digunakan untuk menampilkan nilai output (yk).
8. *Jtable* digunakan untuk menampilkan nilai error (MSE).
9. *Jtable* digunakan untuk menampilkan bobot akhir nilai Vji dan Wkj.

Rancangan antarmuka untuk proses pengujian ditunjukkan pada gambar 3.10 berikut ini:



Gambar 3.10 Proses Pengujian

1. *JButton* Load Data digunakan untuk mencari data uji yang akan diinputkan ke dalam sistem.
2. *Jtable* digunakan untuk menampilkan data curah hujan.
3. *JButton* Normalisasi digunakan untuk memproses data curah hujan agar berada pada range 0.1 – 0.9.

4. *Jtable* digunakan untuk menampilkan data curah hujan yang telah dinormalisasi.
5. *JButton* Pengujian digunakan untuk memulai proses pengujian.
6. *Jtable* digunakan untuk menampilkan data perbandingan hasil akhir output jaringan dengan nilai target data uji.
7. *JTextField* digunakan untuk menampilkan nilaii erro (MSE) data uji.
8. *JTextField* digunakan untuk menampilkan jumlah benar dan jumlah salah dari hasil perbandingan output jaringan dengan nilai data target, kemudian akan menampilkan nilai akurasi pada data.

3.7 Rancangan Uji Coba

Setelah sistem telah selesai dibuat, maka dilanjutkan dengan proses pengujian sistem. Pengujian ini bertujuan untuk mengetahui tingkat akurasi dari hasil peramalan curah hujan menggunakan metode *Al-Aloui Backpropagation*. Terdapat dua macam proses pengujian yaitu proses untuk mengetahui pengaruh learning rate dan momentum terhadap MSE dan pengaruh data uji terhadap MSE. Hasil pengujian nilai output akan dibandingkan dengan data target untuk dihitung nilai akurasinya dengan menggunakan persamaan 2.13.

3.7.1 Uji Pengaruh Learning Rate dan Momentum terhadap MSE

Uji pengaruh learning rate dan momentum merupakan proses pengujian yang dilakukan terhadap beberapa data latih yang berbeda dengan beberapa nilai learning rate antara 0.1 - 0.9 dan momentum antara 0.1 - 0.9 terhadap MSE. Tujuannya untuk membandingkan learning rate dan momentum yang akan digunakan dengan mengambil MSE yang terkecil. Tabel 3.24 menggambarkan rancangan tabel pengaruh learning rate dan momentum terhadap MSE.

Tabel 3.24 Pengaruh Learning Rate dan Momentum terhadap MSE

Learning Rate	Momentum								
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1									
0.2									
0.3									
0.4									

0.5								
0.6								
0.7								
0.8								
0.9								

Keterangan:

1. Learning Rate dan Momentum : Jumlah nilai *learning rate* dan momentum yang digunakan
2. MSE : Nilai Error pada sistem

3.7.2 Uji Pengaruh Jumlah Data latih terhadap akurasi

Uji pengaruh data latih merupakan tahap membandingkan jumlah data latih terhadap akurasi

Tabel 3.25 Uji Pengaruh Jumlah Data Latih terhadap akurasi

Jumlah Data Latih	Jumlah Data Uji	Akurasi	MSE
1 bulan			
2 bulan			
3 bulan			
4 bulan			
5 bulan			
6 bulan			

Keterangan:

1. Jumlah Data latih : Jumlah Data latih yang digunakan
2. Akurasi : Jumlah akurasi pada data latih
3. MSE : Nilai Error pada sistem

BAB IV IMPLEMENTASI

4.1 Lingkungan Implementasi

Implementasi ini berupa sistem aplikasi yang menerapkan metode *Algoritma Al-Aloui backpropagation* untuk meramalkan curah hujan yang terjadi pada hari berikutnya. Variabel yang digunakan pada aplikasi ini adalah curah hujan harian. Beberapa aspek yang perlu diperhatikan dalam metode penelitian dan perancangan sistem yaitu dari segi perangkat keras (*hardware*) maupun perangkat lunak (*software*).

4.1.1 Lingkungan Implementasi Perangkat Keras

Dalam mengembangkan sistem dan penerapan metode penelitian ini digunakan beberapa komponen perangkat keras sebagai berikut :

1. Processor : Intel ® Core™ i5 CPU @2.50GHz
2. Memory : 4.00 GB
3. Hard disk : 500 GB

4.1.2 Lingkungan Implementasi Perangkat Lunak

Pengembangan sistem dan penelitian ini dibutuhkan beberapa perangkat lunak yang digunakan sebagai berikut ini :

1. Sistem Operasi yang digunakan Windows 7 Profesional
2. Aplikasi pembangunan GUI dan code menggunakan NetBeans IDE 7.3
3. Bahasa pemrograman yang dipakai yaitu bahasa pemrograman java.
4. Komponen java yang digunakan yaitu JDK 1.7.0.

4.2 Implementasi Program

Berdasarkan metode penelitian dan perancangan proses yang terdapat dalam sub bab 3.3, maka pada sub bab ini akan dijelaskan implementasi proses-proses tersebut.

4.2.1. Proses Baca File

Proses Baca File ini merupakan proses pertama untuk memanggil data yang akan digunakan untuk proses pelatihan dengan data latih dan proses pengujian dengan data uji. Pembacaan data latih dan data uji menggunakan file jxl yang akan di inputkan ke dalam java. File yang terbaca berisi atribut curah hujan, target dan kategori setiap data. Tahapan proses Baca File ditunjukkan pada *source code 4.1*

Proses Baca File

```
import java.io.*;
import jxl.Workbook;
import jxl.Sheet;

public class BacaFile
{
    String fileName;
    double curahhujan[]= null;
    double target[]= null;
    String tanggal []=null;
    String kategori []=null;
    int kode [] =null;

    public BacaFile(String fileName)
    {
        this.fileName=fileName;
    }

    public Atribut[] baca()
    {
        File file = new File(fileName);
        String cell = " ";
        Atribut[]dataSet=null;

        try
        {
            Workbook workbook = jxl.Workbook.
            getWorkbook(file);
            Sheet[] sheets = workbook.getSheets();
            int row = sheets[0].getRows();
            int column = sheets[0].getColumns();

            dataSet=new Atribut[row-1];
            tanggal = new String[row-1];
            curahhujan = new double[row-1];
            target = new double[row-1];
            kategori = new String[row-1];
            kode = new int [row-1];
```

```
for(int i=0;i<dataSet.length;i++)
{
    dataSet[i]=new Atribut(0,"", 0,0, "",0);
}

for(int i=1 ; i< row ; i++)
{
String a= sheets[0].getCell(1, i)
    .getContents();
double b = Double.parseDouble
    (sheets[0].getCell (2, i)
    .getContents());
double d =Double.parseDouble(sheets[0]
    .getCel(3, i).getContents());
String c= sheets[0].getCell(4, i)
    .getContents();
int e= Integer.parseInt(sheets[0].getCell
    (5, i).getContents());

    dataSet[i-1]=new Atribut(i, a, b, d, c, e);
    tanggal[i-1]=dataSet[i-1].tanggal;
    curahhujan[i-1]= dataSet[i-1].curahHujan;
    target[i-1]= dataSet[i-1].Target;
    kategori[i-1]=dataSet[i-1].kategori;
    kode[i-1]= dataSet[i-1].kode;
}
}
catch (Exception ex)
{
    System.out.println("Gagal membaca XLS!"
        + "\nAda Error: "+ex.getMessage());
}
return dataSet;
}
}
```

Source Code 4.1 Proses Baca File

4.2.2. Proses mencari nilai minimum dan maksimum

Proses ini digunakan untuk mencari nilai minimum dan maksimum dari data curah hujan yang nantinya akan digunakan untuk proses normalisasi. Tahapan proses mencari nilai minimum dan maksimum ditunjukkan pada *source code 4.2.2*

Proses mencari nilai minimum dan maksimum

```
public class normalisasi
{
    double [] curahHujan,target,b,c =null;
    double hmax,hmin,max,min;

    public void Hmaxmin (double curahHujan[])
    {
        this.curahHujan= curahHujan;

        max = 0;
        min = 100;

        for (int a = 1; a < curahHujan.length; a++)
        {
            if (curahHujan[a] > max)
            {
                max = curahHujan[a] ;
            }
            else if (curahHujan[a] < min)
            {
                min = curahHujan[a] ;
            }
            hmax =max;
            hmin =min;
        }
    }
}
```

Source Code 4.2 Proses Mencari Nilai minimum dan maksimum

4.2.3. Proses Normalisasi Data dan Denormalisasi Data

Proses ini digunakan untuk melakukan normalisasi terhadap data curah hujan agar range data yang didapatkan berkisar antara 0,1 hingga 0,9. Demikian pula sebaliknya, denormalisasi berfungsi merubah data normalisasi antara 0,1 hingga 0,9 menjadi data normal. Atribut yang digunakan untuk proses normalisasi dan denormalisasi adalah curah hujan dan target hari selanjutnya. Proses Denormalisasi dilakukan setelah mendapatkan nilai *output* yk saat proses pelatihan. Tahapan untuk proses ini ditunjukkan pada *source code* 4.3 dan *source code* 4.4.

Proses Normalisasi Data

```
public void Hnormalis (double curahHujan [], double
target [])
{
    this.curahHujan= curahHujan;
    this.target = target;

    b = new double [curahHujan.length];
    c = new double [target.length];

    hmax =max;
    hmin =min;

    for (int a = 0; a <curahHujan.length; a++)
    {
        b[a]=(0.8*(curahHujan[a]-hmin)/(hmax-hmin))
            +0.1;
    }
    for (int i = 0; i <target.length; i++)
    {
        c[i]=(0.8*(target[i]-hmin)/(hmax-hmin))+0.1;
    }
}
```

*Source Code 4.3 Normalisasi Data***Proses Denormalisasi Data**

```
for (int h=0; h<Kode.size(); h++)
{
    denormalisasi.set(h, ((hmax-hmin)*(YK.get(h)-0.1)
/0.8)+hmin);
}
```

*Source Code 4.4 Denormalisasi Data***4.2.4. Proses Inisialisasi Bobot**

Inisialisasi bobot dilakukan untuk menentukan nilai awal bobot dari *input layer* ke *hidden layer* (V_{ji}) dan dari *hidden layer* ke *output layer* (W_{kj}). Kemudian nilai bobot V_{ji} dan W_{kj} di acak dengan nilai antara 0.1 – 0.9. Tahapan proses mencari inisialisasi bobot dapat ditunjukkan pada *source code 4.5*.

Proses inialisasi bobot

```
public void random ()
{
    nilai_Vji = new double [2];
    nilai_Wkj = new double [2];

    for (int j=0;j<2; j++)//kolom
    {
        nilai_Vji[j]= Math.random();
    }

    for (int j=0;j<2; j++)//kolom
    {
        nilai_Wkj[j]= Math.random();
    }
    temp=nilai_Vji;
    temp2=nilai_Wkj;
}
```

*Source Code 4.5 inialisasi bobot***4.2.5. Proses Pelatihan**

Langkah pertama adalah user memilih unit yang akan dilatih dan memasukan data-data yang akan dipakai untuk proses pelatihan. Data-data tersebut meliputi data learning rate, maksimum epoch, minimum eror dan momentum. Dalam proses pelatihan terdapat beberapa proses yang dijalankan diantaranya adalah *feedforward*, *backpropagation*, perhitungan *error(MSE)*, *weight update* dan duplikasi.

4.2.5.1 Proses FeedForward

Jika sudah diketahui bobot dari V_{ji} dan W_{kj} , kemudian melakukan inialisasi bobot bias. Setelah itu nilai bias dan nilai V_{ji} dan W_{kj} akan dikalikan dengan data curah hujan. Proses *feedforward* merupakan fungsi untuk mengaktifkan neuron-neuron pada *hidden layer* dan *output layer*. Tahapan proses ini ditunjukkan pada *source code 4.6*.

Proses FeedForward

```

public void FeedForward (ArrayList<Double>xi ,
ArrayList<Double>YNET,ArrayList<Double>YK,int h)
{
    znet_j = new double[2];
    zj = new double [2];

    for (int j=0; j<jmlhHidden ; j++)// hidden
    {
        znet_j[j]= Vjo + (xi.get(h)*temp[j]);

        zj[j]= 1/(1+Math.exp(-znet_j[j]));
    }

    YNET.set(h,Wko +((zj[0]* temp2[0])+(zj[1]*
temp2[1])));

    YK.set(h, 1/(1+Math.exp(-YNET.get(h))));
}

```

*Source Code 4.6 Proses Feedforward***4.2.5.2 Proses Backpropagation**

Proses *Backpropagation* terdiri dari perhitungan faktor kesalahan di unit *yk* ke semua unit layer *hidden* dan perhitungan nilai perubahan bobot lapisan tersembunyi ke *layer output*, Proses *Backpropagation* dapat dilihat pada *source code 4.7*.

Proses Backpropagation

```

public void Backpropagation (ArrayList<Double>tk,double
alpha,ArrayList<Double>xi, ArrayList<Double>YK,
ArrayList<Double>GALAT_K,int h)
{
    this.alpha=alpha;

    deltaWkj= new double [2];
    galatNetJ = new double [2];
    galatJ =new double [2];
    deltaVji= new double [2];

    GALAT_K.set(h, (tk.get(h)-YK.get(h))*YK.get(h)*
(1-YK.get(h)) );

    for (int j=0; j<jmlhHidden ; j++)
    {
        deltaWkj[j]= (alpha *GALAT_K.get(h)
*zj[j]);
    }
}

```

```

galatNetJ[j]=(GALAT_K.get(h)*temp2[j]);
galatJ[j]=(galatNetJ[j]*zj[j])*(1-zj[j]);
deltaVji[j]=(alpha*galatJ[j]*xi.get(h));
}
}

```

Source Code 4.7 Proses Backpropagation

4.2.5.3 Proses Weight Update

Proses weight update digunakan untuk menghitung bobot baru dari unit-unit yang berada pada layer *input* ke unit-unit yang berada pada layer *hidden*, dan juga bobot baru dari unit-unit yang berada pada layer *hidden* ke unit-unit yang berada pada layer *output*. Proses weight update dapat dilihat pada *source code* 4.8.

Proses Weight Update

```

public void weightUpdate(ArrayList<Double>xi, double
momentum,
int h)
{
    this.momentum = momentum;

    deltaWkjT = new double [2];
    deltaVjiT = new double [2];
    WkjBr =new double [2];
    VjiBr= new double [2];

    for (int j=0; j<jmlhHidden ; j++)// hidden
    {
        deltaWkjT[j] = deltaWkj[j] + (momentum *
            deltaWkj[j]);

        deltaVjiT[j] = deltaVji[j] + (momentum *
            deltaVji[j]);

        WkjBr [j] = deltaWkjT [j] + temp2[j];

        VjiBr [j] = deltaVjiT [j] + temp[j];
    }
    temp=VjiBr;
    temp2=WkjBr;
}

```

Source Code 4.8 Proses Weight Update

4.2.5.4 Proses Hitung Error

Nilai output (y_k) yang diperoleh dari proses *feedforward*, digunakan untuk menghitung informasi error pada *output* layer. Nilai ini diambil dari nilai target dikurangi *output* (y_k), kemudian dibagi jumlah data yang digunakan. Tahapan prosesnya ditunjukkan pada *source code* 4.9.

Proses Hitung Error

```
public void MSE ( ArrayList<Double>tk,
  ArrayList<Double>YK)
  {
    for (int h=0; h<tk.size(); h++)
    {
      Mse = (Mse+(Math.pow(tk.get(h)-YK
        .get(h),2)));
    }
    Mse = Mse/tk.size();
  }
```

Source Code 4.9 Proses Hitung Error

4.2.5.5 Proses Duplikasi

Proses duplikasi ini berfungsi untuk menghasilkan data latihan baru yang akan dijadikan pelatihan pada iterasi selanjutnya. Proses duplikasi ini merupakan pembeda antara JST Backpropagation dengan JST Al-alaoui Backpropagation. Proses duplikasi akan ditunjukkan pada *source code* 4.10

Proses Duplikasi

```
public void Duplikasi( ArrayList<Double> denormalisasi,
  double hmax, double hmin, ArrayList<Integer> Kode,
  ArrayList<Double>YK,ArrayList<Double>xi,
  ArrayList<Double>tk,ArrayList<Double>YNET,
  ArrayList<Double> GALAT_K,ArrayList<Integer> TempYk,
  ArrayList<Integer> tempH)

  for (int h=0; h<Kode.size(); h++)
  {
    if (denormalisasi.get(h)<5)
    {
      TempYk.set(h, 1);
    }
    else if (denormalisasi.get(h)>=5 &&
      denormalisasi.get(h)<20)
    {
      TempYk.set(h, 2);
    }
    else if (denormalisasi.get(h)>=20 &&
      denormalisasi.get(h)<50)
```

```
{
    TempYk.set(h, 3);
}

else if (denormalisasi.get(h)>=50 &&
        denormalisasi.get(h)<100)
{
    TempYk.set(h, 4);
}

else if (denormalisasi.get(h)>100)
{
    TempYk.set(h, 5);
}
}
for (int h=0; h<Kode.size(); h++)
{
    if (TempYk.get(h) != Kode.get(h))
    {
        tempH.add(h);
        salah++;
    }
}
for (int h=0; h<tempH.size(); h++)
{
    xi.add(xi.get(tempH.get(h)));
    tk.add(tk.get(tempH.get(h)));
    TempYk.add(TempYk.get(tempH.get(h)));
    Kode.add(Kode.get(tempH.get(h)));
    GALAT_K.add(GALAT_K.get(tempH.get(h)));
    YNET.add(YNET.get(tempH.get(h)));
    YK.add(YK.get(tempH.get(h)));
    denormalisasi.add(denormalisasi
        .get(tempH.get(h)));
}
}
```

Source Code 4.10 Proses Duplikasi

4.2.6. Proses Pengujian

Proses pengujian digunakan untuk menguji data curah hujan. Proses pengujian ini dilakukan menggunakan proses *feedforward*. Tahap proses pengujian dapat dilihat pada *source code 4.11*

Proses Pengujian

```
public class pengujian
{
    double znet_j[]=null;
    double zj[]=null;
    double Ynet_k[]=null;
    double Yk[]=null;
    double xi[]=null;
    double curahhujan[]=null;
    double TempYk[]=null;
    int [] Kode=null;
    double [] denormalisasi=null;
    double hmax,hmin;
    double temp[]=null;
    double temp2[]=null;
    int TingkatAkurasi=0;
    double jmlhHidden = 2;
    double jmlhInput = 1;
    double jmlhOutput = 1;
    int Vjo = 1;
    int Wko = 1;
    int benar = 0;
    int salah = 0;
    String [] SR;

    public void FeedForward (double [] curahhujan,
double temp[],double temp2 [])
    {
        this.curahhujan=curahhujan;
        this.temp=temp;
        this.temp2=temp2;

        znet_j = new double[2];
        zj = new double [2];
        Ynet_k = new double [curahhujan.length];
        Yk = new double [curahhujan.length];
        xi = curahhujan;

        System.out.println();

        for (int i=0; i<curahhujan.length; i++)
        {
            for (int j=0; j<jmlhHidden ; j++)// hidden
            {
                znet_j[j]= Vjo + (xi[i]*temp[j]
                zj[j]=1/(1+Math.exp(-znet_j[j]));
            }
            Ynet_k[i]=Wko +((zj[0]* temp2[0])+
            (zj[1]* temp2[1]));
            Yk[i]= 1/(1+Math.exp(-Ynet k[i]));
        }
    }
}
```

```

    }
}

```

Source Code 4.11 Proses Pengujian

4.2.6.1 Proses menghitung Tingkat Akurasi

Tingkat akurasi merupakan proses klasifikasi, jika nilai dari tingkat akurasi semakin besar maka menunjukkan semakin akurat proses klasifikasi ini

Proses menghitung Tingkat Akurasi

```

public void tingkatAkurasi (int [] Kode,double hmax,
double hmin)
{
    this.Kode=Kode;
    this.hmax=hmax;
    this.hmin=hmin;

    denormalisasi = new double [Kode.length];
    TempYk = new double [Kode.length];
    SR = new String [Kode.length];

    for (int h=0; h<Kode.length; h++)
    {
        denormalisasi[h] = ((hmax-hmin)*(Yk[h]-0.1)/0.8)
        +hmin;
    }

    for (int h=0; h<Kode.length; h++)
    {
        if (denormalisasi[h]<5)
        {
            TempYk[h]= 1;
            if (TempYk[h]==1)
            {
                SR[h]="SR";
            }
        }
        else if (denormalisasi[h]>=5 &&
        denormalisasi[h]<20)
        {
            TempYk[h]=2;
            if (TempYk[h]== 2)
            {
                SR[h]="S";
            }
        }
        else if (denormalisasi[h]>=20 &&
        denormalisasi[h]<50)
        {
            TempYk[h]=3;

```

```
        if(TempYk[h]== 3)
        {
            SR[h]="R";
        }
    }
    else if (denormalisasi[h]>=50 &&
        denormalisasi[h]<100)
    {
        TempYk[h]=4;
        if (TempYk[h]==4)
        {
            SR[h]="L";
        }
    }

    else if (denormalisasi[h]>100)
    {
        TempYk [h]=5;
        if(TempYk[h]==5)
        {
            SR[h]="SL";
        }
    }
}

for (int i=0; i<Kode.length; i++)
{
    if (TempYk[i] == Kode[i])
    {
        benar++;
    }

    TingkatAkurasi = (benar*100/Kode.length);
}

for (int h=0; h<Kode.length; h++)
{
    if (TempYk[h] != Kode[h])
    {
        salah++;
    }
}
}
```

Source Code 4.12 Proses menghitung Tingkat Akurasi

4.3 Implementasi Antar muka

Berdasarkan rancangan antarmuka pada sub bab 3.5, dihasilkan antarmuka untuk peramalan curah hujan. Pada tampilan implementasi program terdiri 2 bagian utama yaitu :

1. *Form* Proses pelatihan

Form ini bertujuan untuk melakukan pelatihan penentuan curah hujan dengan memasukkan parameternya.

2. *Form* Proses pengujian

Form ini bertujuan untuk mengetahui akurasi dan perbandingan sistem dari data curah hujan yang dimasukkan.

4.3.1 Form Pelatihan

Pada Form ini terdapat form load data untuk menginputkan data latih ke dalam tabel pelatihan. Form normalisasi merupakan tempat untuk nilai awal data latih yang sudah dinormalisasi. Langkah pertama yang perlu dilakukan adalah menentukan arsitektur jaringan dan mengatur nilai parameter yang akan digunakan, seperti minimum error, maksimal epoch, *learning rate* dan momentum. Langkah kedua setelah menentukan parameter jaringan maka proses pelatihan dapat diproses. Pada proses pelatihan akan ditunjukkan pada gambar hasil pelatihan seperti gambar 4.1

PERAMALAN CURAH HUJAN					
Pelatihan		Pengujian			
Parameter					
Minimal Error	:	<input type="text"/>			
Maksimal Epoch	:	<input type="text"/>			
Learning Rate	:	<input type="text"/>			
Momentum	:	<input type="text"/>			
Nilai Output (Yk)					
Title 1	Title 2				
<input type="text"/>	<input type="text"/>				
Hasil Error (Mse)					
Title 1	Title 2				
<input type="text"/>	<input type="text"/>				
Data Curah Hujan					
Title 1	Title 2	Title 3	Title 4	Title 5	Title 6
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Data Hasil Normalisasi Curah Hujan					
Title 1	Title 2	Title 3	Title 4	Title 5	Title 6
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Nilai Bobot Akhir Vj					
Title 1	Title 2				
<input type="text"/>	<input type="text"/>				
Nilai Bobot Akhir Wkj					
Title 1	Title 2				
<input type="text"/>	<input type="text"/>				
<input type="button" value="Load Data"/>					
<input type="button" value="Normalisasi"/>					
<input type="button" value="Proses Latih"/>					

Gambar 4.1 Form Pelatihan

Dari antarmuka tersebut terlihat hasil pelatihan menghasilkan nilai *output* yang telah di duplikasi. Proses ini selesai jika nilai *output* sesuai dengan target, bahwa proses pelatihan berhasil. Setelah proses pelatihan selesai, maka bobot akhir pada V_{ji} dan W_{kj} akan tersimpan dan dijadikan nilai bobot awal pada proses pengujian.

4.3.2 Form Pengujian

Langkah pertama menginputkan data uji pada form pengujian, setelah itu data uji tersebut akan dinormalisasi. Pada proses pengujian akan ditunjukkan pada gambar hasil pelatihan seperti gambar 4.2

Data Curah Hujan					
No	Tanggal	Curah H...	Target	Kategori	Kode

Data Hasil Normalisasi Curah Hujan					
No	Tanggal	Curah H...	Target	Kategori	Kode

Hasil Perbandingan				
No	Output Jari...	Kategori	Output Data	Kategori

MSE :

Data Benar :

Data Salah :

Akurasi :

Gambar 4.2 Form Pengujian

Nilai *output* yang telah di uji akan di denormalisasi agar nilainya menjadi data normal. Data denormalisasi beserta kategorinya akan akan dibandingkan dengan data target awal, hasil perbandingan ini untuk mengetahui jumlah data uji yang salah dan data uji yang benar, serta nilai akurasi dari metode ini.

BAB V

ANALISIS HASIL DAN PEMBAHASAN

Pada bab ini dilakukan pengujian dan analisis hasil pengujian sistem peramalan curah hujan dengan menggunakan metode *Al-aloui Backpropagation*.

5.1 Hasil Uji Coba

Pada pengujian sistem peramalan curah hujan digunakan 242 data, masing-masing jumlah data yang telah ditentukan, akan dibagi secara seimbang dengan data latih yang berbeda-beda sebesar 31 data, 59 data, 90 data, 120 data, 151 data serta 181 data dan data uji tetap sebesar 31 data dengan menggunakan bulan selanjutnya.

Untuk memperoleh struktur jaringan syaraf tiruan terbaik yang digunakan untuk peramalan, maka dilakukan pengujian terhadap sistem. Pengujian dilakukan dengan cara melatih jaringan syaraf tiruan dengan parameter-parameter yang berbeda yaitu uji pengaruh learning rate dan momentum terhadap MSE dan jumlah data terhadap tingkat akurasi.

5.1.1 Uji Pengaruh Learning Rate dan Momentum terhadap MSE

Pada pengujian ini yang dilakukan adalah membandingkan nilai MSE yang diperoleh pada learning rate yaitu antara 0.1 sampai 0.9 dan momentum antara 0.1 – 0.9 . Proses pelatihan menggunakan data latih sebanyak 181 data (bulan Januari-Juni 2011) data . Berikut ini pengujian dengan jumlah data latih yang berbeda dapat dilihat pada tabel 5.1.

Tabel 5.1 Tabel Hasil uji pengaruh *learning rate* dan momentum

<i>Learning Rate</i>	Momentum								
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	0.015387498	0.01548079	0.015478082	0.015520611	0.015516598	0.01553387	0.015533814	0.01553526	0.01556955
0.2	0.015637958	0.01545643	0.015432092	0.015579506	0.01568202	0.01582621	0.015839471	0.01584995	0.01591695
0.3	0.0158401	0.01584995	0.015452374	0.01549749	0.015528057	0.01560646	0.015793114	0.015676	0.01566761
0.4	0.015540418	0.01560646	0.015568791	0.015689496	0.015694633	0.01598011	0.016209324	0.01645046	0.01609979
0.5	0.015713327	0.01569463	0.016114971	0.016344966	0.016481047	0.01619685	0.016277933	0.01657684	0.01644661
0.6	0.016129326	0.01645046	0.016131625	0.016262416	0.016576841	0.01647282	0.016524958	0.01670994	0.01642456
0.7	0.016123963	0.01626242	0.016620136	0.016526597	0.016618726	0.01636785	0.01661614	0.01635761	0.01643872
0.8	0.016398717	0.01647282	0.016594414	0.016367846	0.016613786	0.01633088	0.016464635	0.01655905	0.01650323
0.9	0.016552943	0.01670994	0.016556579	0.016357608	0.016459669	0.01655905	0.016513036	0.01670935	0.01677604

Pada tabel 5.1 dapat dilihat bahwa nilai MSE yang paling kecil yang dihasilkan oleh pengujian ini adalah 0.015387498 yaitu pengujian dengan *learning rate* sebesar 0.1 dan momentum 0.1. Sedangkan MSE terbesar yang dihasilkan yaitu 0.01677604 pengujian dengan *learning rate* sebesar 0.9 dan momentum 0.9. Sehingga dapat disimpulkan bahwa *learning rate* dan momentum yang terbaik untuk pengujian saat *learning rate* dan momentum mencapai 0.1 dan 0.1.

5.1.2 Uji pengaruh jumlah data latih terhadap akurasi

Pada pengujian digunakan 4 data latih dengan *learningrate* dan *momentum* terbaik yaitu 0.1 dan 0.1, dengan data latih sebanyak 1 bulan pada bulan januari, 2 bulan pada bulan januari-februari, 3 bulan pada bulan januari-maret, 4 bulan pada bulan januari-april, 5 bulan pada bulan januari-mei, 6 bulan pada bulan januari – juni dan data uji yang digunakan tetap dengan menggunakan 1 bulan selanjutnya. Misalkan data latih yang digunakan pada bulan januari maka data uji yang digunakan bulan februari. Berikut ini uji pengaruh jumlah data latih terhadap akurasi yang akan ditunjukkan pada tabel 5.2

Tabel 5.2 Tabel Hasil Uji pengaruh jumlah data latih terhadap akurasi

Jumlah Data Latih	Jumlah DataUji	Akurasi	MSE
1 bulan	1 bulan	17%	0.02476
2 bulan	1 bulan	35%	0.03119
3 bulan	1 bulan	56%	0.02753
4 bulan	1 bulan	19%	0.02626
5 bulan	1 bulan	100%	0.0034
6 bulan	1 bulan	100%	0.03398

Pada pengujian ke 2 menggunakan data latih sebanyak 1 bulan pada bulan januari, 2 bulan pada bulan Januari-Februari, 3 bulan pada bulan Januari-Maret, 4 bulan pada bulan Januari-April, 5 bulan pada bulan Januari-Mei, 6 bulan pada bulan Januari – Juni dan tetapi data uji yang digunakan tetap dengan menggunakan 2 bulan selanjutnya. Misalkan data latih yang digunakan pada bulan Januari-Maret maka data uji yang digunakan bulan April-Mei. Berikut Berikut ini

uji pengaruh jumlah data latih terhadap akurasi yang akan ditunjukkan pada tabel 5.3

Tabel 5.3 Tabel Hasil Uji pengaruh jumlah data latih terhadap akurasi

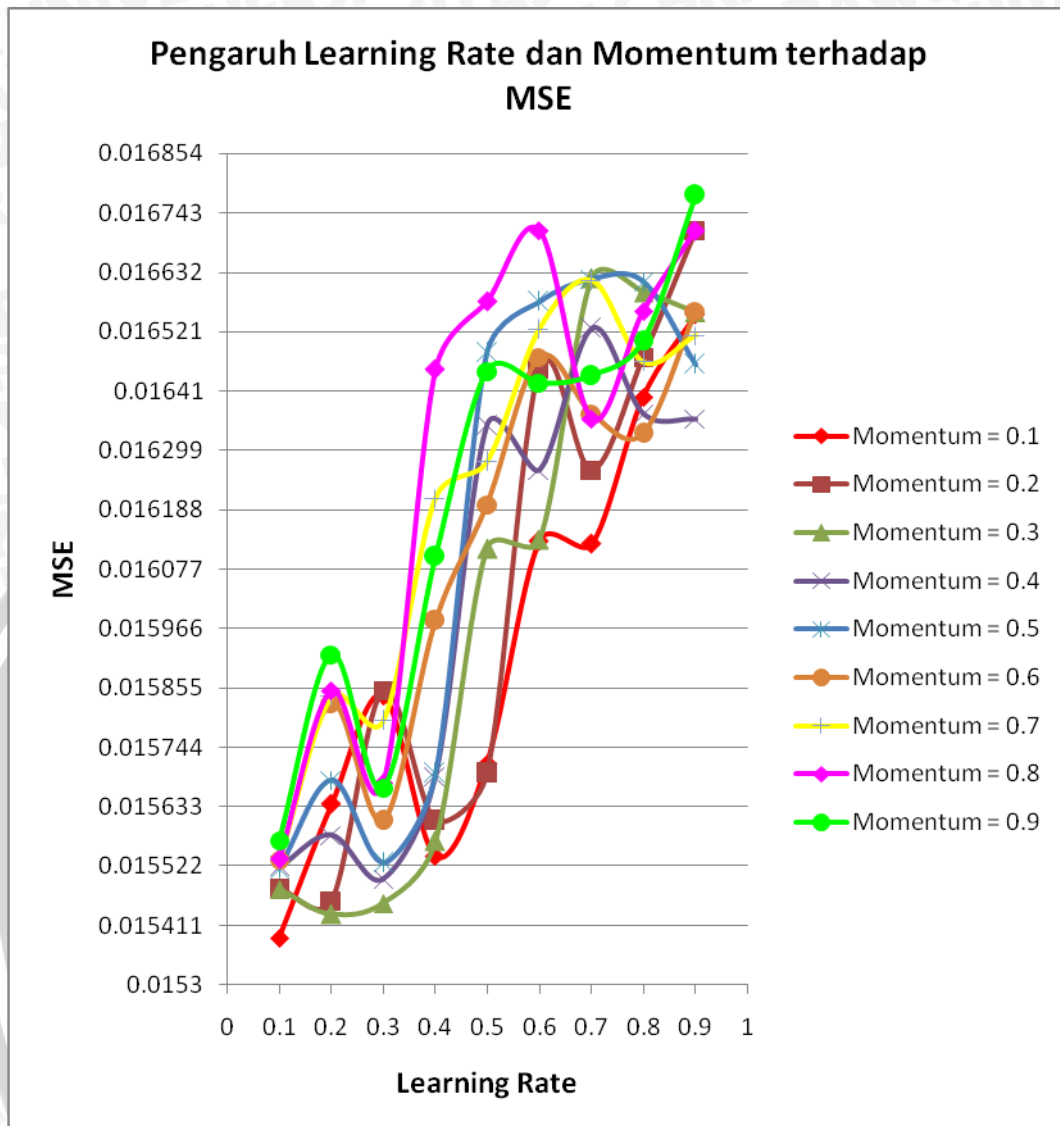
Jumlah Data Latih	Jumlah Data Uji	Akurasi	MSE
1 bulan	2 bulan	27%	0.02641
2 bulan	2 bulan	32%	0.02163
3 bulan	2 bulan	24%	0.02083
4 bulan	2 bulan	9%	0.01514
5 bulan	2 bulan	100%	0.00314
6 bulan	2 bulan	100%	0.01701

Dapat dilihat dari tabel 5.2 dan tabel 5.3 bahwa jumlah akurasi terbesar pada jumlah data 5 bulan dan data 6 bulan akurasi mencapai 100%. Dari kedua tabel dapat dilihat MSE minimum saat jumlah data latih 5 bulan dengan data uji 2 bulan sebesar 0.00313611.

5.2 Analisis dan Interpretasi

5.2.1 Analisis Hasil Pengaruh *Learning Rate* dan Momentum terhadap MSE

Pada proses pengujian yang telah dilakukan dengan menggunakan 6 macam jumlah data latih yang berbeda, maka dapat diketahui nilai kombinasi *learning rate* dan momentum. Proses pelatihan dilakukan menggunakan jumlah duplikasi sebanyak 10, dikarenakan keterbatasan perangkat yang tidak mampu menampung memori yang banyak jika lebih dari 10 duplikasi, maksimal epoch yang digunakan sebanyak 100 karena pada epoch selanjutnya memiliki nilai MSE yang tidak berbeda jauh dengan epoch sebelumnya, proses ini menggunakan data latih 181 data(bulan Januari-Juni) data, nilai MSE terbentuk naik-turun ketika nilai *learning rate* mengalami kenaikan untuk setiap nilai momentum. Grafik pengaruh *learning rate* dan momentum terhadap MSE dapat dilihat pada gambar grafik 5.1

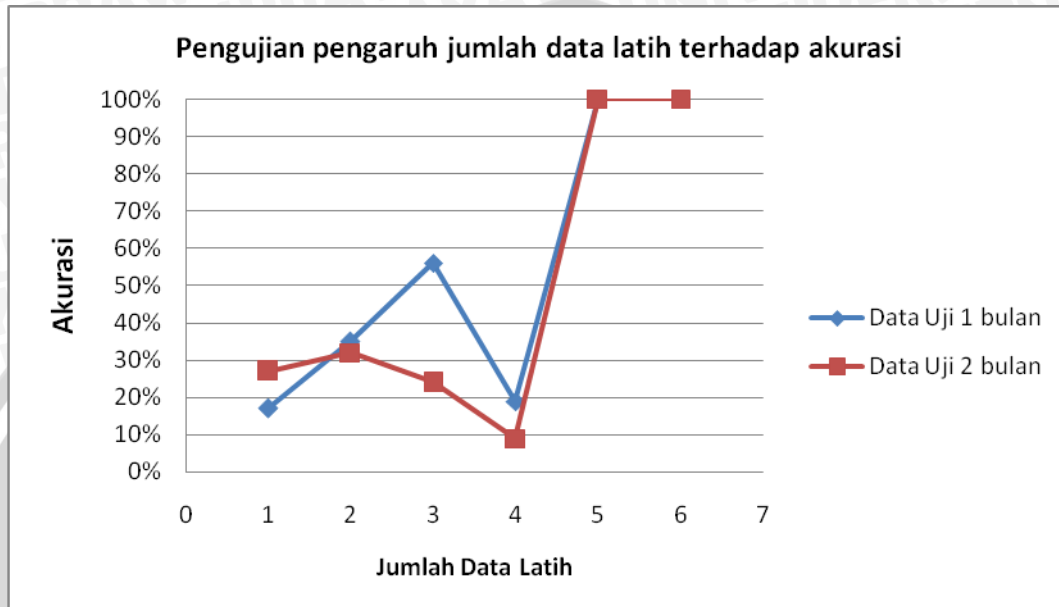


Gambar 5.1. Grafik pengaruh *learning rate* dan momentum terhadap mse

Berdasarkan grafik 5.1. dapat diketahui bahwa semakin tinggi nilai *learning rate* dan momentum jumlah mse dapat berubah. Saat *learning rate* berada pada range 0,2 – 0,3 mengalami penurunan, kemudian mengalami kenaikan saat range berada 0,4 - 0,9. Hal ini terjadi karena nilai *learning rate* dan momentum berpengaruh terhadap mse, karena pada dasarnya *learning rate* berfungsi untuk mempercepat perubahan nilai bobot sistem pada proses pelatihan, sedangkan momentum berguna untuk mencegah local optimum (bobot yang diinginkan belum tercapai, namun tidak bisa berubah lagi).

5.2.2 Analisis Hasil Pengaruh Jumlah Data Latih terhadap akurasi

Berdasarkan pengujian yang telah dilakukan, terlihat bahwa jumlah data uji berpengaruh terhadap akurasi yang dihasilkan. Grafik hasil jumlah data latih terhadap akurasi terlihat pada gambar 5.2



Gambar 5.2. Grafik pengaruh jumlah data latih terhadap akurasi

Dari grafik dapat dilihat bahwa tingkat akurasi mengalami penurunan ketika data latih menggunakan data 4 bulan yaitu bulan Januari - April dan data uji yang digunakan adalah bulan Mei, tetapi kemudian naik saat data latih 5 bulan, hal ini disebabkan bahwa jumlah data latih yang digunakan dapat mempengaruhi tingkat akurasi sistem, semakin banyak data latih semakin banyak data yang akan dilatih dalam jaringan syaraf tiruan sehingga semakin banyak pola-pola yang akan dikenal. Grafik tersebut juga menunjukkan bahwa tingkat akurasi tertinggi dicapai ketika jumlah data latih sebanyak 5 bulan dan 6 bulan, hal ini dikarenakan nilai curah hujan pada bulan Juli dan Agustus tingkat curah hujan rendahnya banyak, ketika bobot akhir dari proses tersebut memiliki nilai dengan kategori SR (sangat rendah), dan data uji juga memiliki nilai kategori SR (sangat rendah) akan mengakibatkan tinggi tingkat akurasinya.

BAB VI

PENUTUP

6.1 Kesimpulan

Hasil dari implementasi metode backpropagation dengan Algoritma *Alaoui Backpropagation* pada kasus peramalan curah hujan dapat disimpulkan bahwa:

1. Struktur jaringan yang diperoleh menggunakan 1 *neuron* pada *input layer*, 2 *neuron* pada *hidden layer* dan 1 *neuron* pada *output layer*.
2. *Learning rate* dan momentum yang terbaik untuk pengujian saat *learning rate* dan momentum mencapai 0.1 dan 0.1.
3. Peramalan sistem mampu mengenali pola dengan baik dan menghasilkan MSE sebesar 0.00313611 dan tingkat akurasi mencapai 100% dengan menggunakan data latih pada bulan Januari 2011 – Juni 2011 dan data uji pada bulan Juli, pada bulan ini kondisi cuacanya kemarau.

6.2 Saran

Beberapa saran yang dapat diberikan penulis untuk pengembangan lebih lanjut:

1. Diharapkan penelitian lebih lanjut bisa menggunakan Arsitektur jaringan syaraf terbaik dengan menggunakan data curah hujan.
2. Diharapkan penelitian lebih lanjut bisa melakukan peramalan curah hujan tidak terbatas pada satu daerah tertentu.
3. Untuk memaksimalkan hasil peramalan diharapkan melakukan pengujian terhadap jumlah duplikasi.

DAFTAR PUSTAKA

- [ALA-07] Al-Alaoui, M.A., Ferzli, R. 2007. *Subsampling Image Compression Using Al-Alaoui Backpropagation Algorithm*. 14Th IEEE International Conference on Electronics, Circuits and Systems, Marrakech, Morocco
- [BMK-10] BMKG. 2010. *Kondisi Cuaca Ekstrem dan iklim tahun 2010-2011*. Jakarta Pusat
- [DEI-12] Dessy WM dan Irawan A. 2012. *Perbandingan metode jaringan syaraf tiruan backpropagation dan learning vector quantization pada pengenalan wajah*. Jurnal komputer dan Informatika (komputa) edisi.1 volume 1
- [DWS-12] Dwi, JA dan Syukriyadin. 2012. *Peramalan beban puncak transformator daya gardu induk lampeuneurut menggunakan metode time series stokastik vol .1. No.1.2012:59-56*. Universitas Syiah Kuala
- [FAU-94] Fausett, Laurene. 1994. *Fundamentals of Neural Networks*. New Jersey: Prentice Hall Internasional, Inc.
- [GRA-07] Graupe, Daniel. 2007, *Principles Of Artificial Neural Networks*, 2nd
- [HEA-08] Heaton, Jeff. 2008. *Introduction to Neural Networks with Java, Second Edition*. Heaton Research, Inc
- [JUM-09] Jumarwanto, arif. 2009. *Aplikasi Jaringan Saraf Tiruan Bckpropagation untuk memprediksi penyakit THT dirumah sakit Mardi Rahayu kudas*. Jurnal Teknik Elektro Vol. 1 No. 1
- [KLI-01] Klimatologi, Staf Agro. 2001. *Buku Praktikum Klimatologi*. Fakultas Pertanian Universitas Brawijaya Malang.
- [KUS-03] Kusumadewi, S. 2003. *Artificial Intelligence (Teknik dan Aplikasinya)*, Graha Ilmu.
- [MOE-04] Moertini, V.S dan Dhaneswara, G. 2004. *Jaringan Saraf Tiruan Propagasi Balik Untuk Klasifikasi Data*. Jurusan Ilmu Komputer. Universitas Katolik Parahyangan.

- [SAH-12] Sahidi, Toni. T. 2012. *Tugas Akhir : Pengenalan Emosi Suara Dengan Jaringan Syaraf Tiruan Al-Alaoui backpropagation*. Ilmu Komputer-FMIPA-Unibraw. Malang.
- [SET-93] Setiawan,S. 1993. *Artificial Intelligence*. Andi Offset. Yogyakarta edition, World Scientific Publishing Co. Pte. Ltd., Singapore.
- [SIA-05] Siang, Jong Jek. 2005. *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab*. Yogyakarta : ANDI.
- [SUB-10] Subiyanto.2010. *Sistem komputasi Jaringan Syaraf Tiruan Backpropagation.Jurnal teknik elektro Vol.2 No.1*
- [SUR-06] Suroso.2006.*Analisi curah hujan untuk membuat kurva intensity-duration-frequency(IDF) kawasan rawan banjir kabupaten banyumas*. Jurnal Teknik Sipil, Vol.3, no.1
- [PRA-99] Pratama, T. I. B. 1999, “*Metode Peramalan Memakai Jaringan Saraf Buatan Dengan Cara Backpropagation*”, *Jurnal Teknologi Industri*. Vol. III, No. 2, hal 109 – 116.
- [TIT-12] Prasetyo,Tito. 2012. *Peramalan Curah Hujan Dengan Jaringan Syaraf Tiruan Backpropagation Menggunakan Algoritma Pelatihan Quasi Newton BFGS*. Malang: Universitas Brawijaya
- [TJA-04] Tjasyono,Bayong. 2004. *Klimatologi*. ITB. Bandung
- [WAR-87] Waryono, Rivai, Gunawan.1987.*Pengantar Meteorologi Dan Klimatologi*. PT.Bina Ilmu. Surabaya