

**ADAPTIVE NEURO FUZZY INFERENCE SYSTEM (ANFIS)
UNTUK PENENTUAN DAERAH YANG RAWAN
TERHADAP DEMAM BERDARAH DENGUE**

SKRIPSI

Oleh :
KHORI WIDAYANTI
0710963019-96



**PROGRAM STUDI ILMU KOMPUTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2012**

LEMBAR PENGESAHAN SKRIPSI

ADAPTIVE NEURO FUZZY INFERENCE SYSTEM (ANFIS) UNTUK PENENTUAN DAERAH YANG RAWAN TERHADAP DEMAM BERDARAH DENGUE

Oleh:

KHORI WIDAYANTI
0710963019-96

Setelah dipertahankan di depan Majelis Penguji
Pada tanggal 03 February 2012
dan dinyatakan memenuhi syarat untuk memperoleh
gelar Sarjana Komputer dalam bidang Ilmu Komputer

Pembimbing I,

Pembimbing II,

Candra Dewi, S.Kom,MSc
NIP. 197711142003122001

Dewi Yanti L.,S.Kom,M.Kom
NIP. 198111162005012004

Mengetahui,
Ketua Jurusan Matematika
Fakultas MIPA Universitas Brawijaya

Dr. Abdul Rouf Alghofari, M.Sc.
NIP. 196709071992031001

LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Khori Widayanti
NIM : 0710963019-96
Jurusan : Matematika
Program Studi : Ilmu Komputer
Penulis Tugas Akhir berjudul : Adaptive Neuro Fuzzy Inference System (ANFIS) Untuk Penentuan Daerah Yang Rawan Terhadap Demam Berdarah Dengue

Dengan ini menyatakan bahwa :

1. Isi dari tugas akhir yang saya buat adalah benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang termaktub di isi dan tertulis di daftar pustaka dalam Tugas Akhir ini.
2. Apabila dikemudian hari ternyata Tugas Akhir yang saya tulis terbukti hasil jiplakan, maka saya akan bersedia menanggung segala resiko yang akan saya terima.

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 03 February 2012

Yang menyatakan,

(Khori Widayanti)
NIM. 0710963019-96

ADAPTIVE NEURO FUZZY INFERENCE SYSTEM (ANFIS) UNTUK PENENTUAN DAERAH YANG RAWAN TERHADAP DEMAM BERDARAH DENGUE

ABSTRAK

Demam berdarah dengue (DBD) merupakan penyakit endemis di Indonesia dan beberapa negara yang terletak di daerah tropis maupun subtropis. Meningkatnya kejadian demam berdarah dipengaruhi oleh banyak faktor, salah satunya adalah faktor iklim. Pada penelitian ini di gunakan metode *Adaptive Neuro Fuzzy Inference System* (ANFIS) dalam menentukan kerawanan suatu daerah terhadap DBD berdasarkan faktor iklim daerah tersebut. ANFIS adalah gabungan antara metode *fuzzy logic* dengan *neural network* (JST). Parameter-parameter yang digunakan untuk inputan adalah kelembaban, kecepatan angin, suhu dan curah hujan.

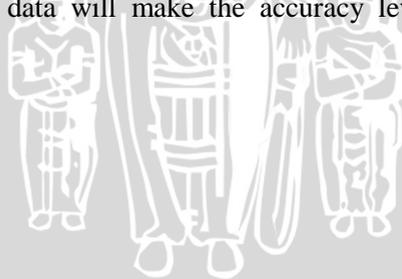
Penelitian ini menggunakan tiga data latih yang berbeda yaitu 40, 75 dan 110 data latih . Data pengujian yang digunakan adalah sebanyak 40 data uji. Menggunakan tiga data latih tersebut didapatkan nilai keakuratan dari penentuan daerah yang rawan terhadap DBD. Hasil uji coba menunjukkan bahwa tingkat akurasi dalam penentuan kerawanan DBD suatu daerah menggunakan ANFIS yang dicoba dengan menggunakan 3 data latih yang berbeda yaitu 40, 75 dan 110 data latih berturut turut adalah 86%, 86%, dan 90%. Perbedaan tingkat akurasi ini dikarenakan data latih yang digunakan berbeda-beda, semakin banyak data latih yang digunakan maka akan semakin baik tingkat akurasi yang didapatkan.

ADAPTIVE NEURO FUZZY INFERENCE SYSTEM (ANFIS) IN DETERMINATING SUSCEPTIBLE AREAS TO DENGUE HAEMORRHAGIC FEVER

ABSTRACT

Dengue haemorrhagic fever (DHF) is an epidemic disease in Indonesia and some countries located in tropical and subtropical areas of the world. The increasing number of DHF cases was caused by many factors, such as climate factor. In this research, Adaptive Neuro Fuzzy Inference System (ANFIS) method is used in determining the vulnerability of a region to dengue haemorrhagic fever based on local climatic factors. ANFIS is a combination of methods between fuzzy logic with neural network (ANN). The parameters used for input are humidity, wind speed, temperature and rainfall.

This research used not only three different number of training data which are 40, 75 and 110 datas but also used 40 datas as it's test data. Using those three training data, the accuracy of the determination of areas vulnerable to dengue haemorrhagic fever is obtained. The trial results showed that the level of accuracy in the determination of a region's DHF vulnerability using ANFIS which was tested by three different training data for 40, 75 and 110 training data consecutively was 86%, 86%, and 90%. The accuracy difference rate was caused by the diverse number of training data, the huge number of training data will make the accuracy level which was obtained better.



KATA PENGANTAR

Alhamdulillah Robbil alamin, puji syukur penulis panjatkan kepada Allah SWT atas segala limpahan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan tugas akhir ini, yang berjudul “Adaptive Neuro Fuzzy Inference System (ANFIS) Untuk Penentuan Daerah Yang Rawan Terhadap Demam Berdarah Dengue”. Sebagai salah satu syarat untuk memperoleh gelar Sarjana dalam bidang Ilmu Komputer

Banyak pihak yang berperan atas terselesaikannya penulisan Skripsi ini. Atas bantuan yang telah diberikan, penulis ingin menyampaikan penghargaan dan ucapan terima kasih yang sedalam-dalamnya kepada:

1. Candra Dewi, S.Kom,MSc, selaku dosen pembimbing utama yang telah meluangkan waktunya dengan sabar membimbing dan memberikan pengarahan kepada penulis.
2. Dewi Yanti Liliana.,S.Kom,M.Kom, selaku dosen pembimbing kedua yang telah membimbing dengan bijaksana dan selalu memberikan masukan kepada penulis.
3. Dr. Abdul Rouf Al-Ghofari, M.Sc, selaku Ketua Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Brawijaya.
4. Drs. Marji, M.T, selaku Ketua Program Studi Ilmu Komputer, Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Brawijaya.
5. Segenap Bapak dan Ibu Dosen yang telah mendidik penulis selama menempuh pendidikan di Program Studi Ilmu Komputer, Jurusan Matematika, Fakultas MIPA, Universitas Brawijaya.
6. Segenap staf dan karyawan di Jurusan Matematika, Fakultas MIPA, Universitas Brawijaya.
7. Kedua orang tua serta adik dan keluarga yang selalu memberikan dukungan materi, doa dan kasih sayang yang tulus kepada penulis dalam menyelesaikan skripsi ini.
8. Bagus ari prasetyo, findi atas segala bantuan dan dukungan semangatnya.
9. Mas rendra, mas fahron, ida, sevi, dita, irine, septafia, yeni, resti, ayu, shela, decy, tri astuti, lia, mbak putri yang telah membantu

penulis dalam menyelesaikan setiap permasalahan pada skripsi ini.

10. Rekan-rekan di Program Studi Ilmu Komputer FMIPA Universitas Brawijaya yang telah banyak memberikan dukungan demi kelancaran pelaksanaan penyusunan skripsi ini.
11. Dan semua pihak yang telah mambantu dan mendukung dalam penulisan skripsi ini yang tidak bisa penulis sebutkan satu persatu.

Semoga skripsi ini bermanfaat bagi pembaca sekalian. Akhirnya, penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan dan mengandung banyak kekurangan, sehingga penulis menghargai kritik dan saran yang membangun dari pembaca.

Malang, 03 February 2012

Penulis



DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
LEMBAR PENGESAHAN SKRIPSI	ii
LEMBAR PERNYATAAN	iii
ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiv
DAFTAR SOURCE CODE	xv
DAFTAR LAMPIRAN	xvi

BAB I PENDAHULUAN

1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan masalah	3
1.4. Tujuan Penelitian	3
1.5. Manfaat Penelitian	3
1.6. Sistematika Penulisan	4

BAB II TINJAUAN PUSTAKA

2.1. Demam Berdarah Dengue (DBD)	5
2.1.1 Vektor Penular DBD	6
2.1.2 Faktor-faktor yang Berhubungan dengan Kejadian Penyakit DBD	6
2.2. Sistem Fuzzy	9
2.2.1 Himpunan Fuzzy	9
2.2.2 Fungsi Keanggotaan	10
2.3. Algoritma cluster K-Means	12
2.3.1 Langkah-langkah Penerapan Algoritma K-Means	12
2.4. Sistem Inferensi Fuzzy	14
2.4.1 Metode Sugeno	15
2.5. Jaringan Syaraf Tiruan	15
2.5.1 Arsitektur Jaringan Syaraf	16
2.5.2 Metode Pelatihan/Pembelajaran	16
2.6. Adaptive Neuro Fuzzy Inference System (ANFIS)	17
2.6.1. Struktur ANFIS	17

2.6.1.1. Leas Square Estimation	20
2.6.1.2. Model Propagasi Error	21
2.6.1.2.1 Error pada lapisan ke-5	21
2.6.1.2.2 Error pada lapisan ke-4	22
2.6.1.2.3 Error pada lapisan ke-3	22
2.6.1.2.4 Error pada lapisan ke-2	22
2.6.1.2.5 Error pada lapisan ke-1	23
2.7. Akurasi Hasil Pengujian	24

BAB III METODOLOGI DAN PERANCANGAN

3.1. Studi Literatur	26
3.2. Sumber Data	26
3.3 Analisis dan Perancangan Sistem	26
3.3.1 Deskripsi Sistem	26
3.3.2 Rancangan Sistem	27
3.3.2.1 Clustering K-Means	32
3.3.2.1.1 Proses Perhitungan Standarisasi Data	34
3.3.2.1.2 Proses Perhitungan Bobot	34
3.3.2.1.3 Proses Perhitungan Jarak Antar Data dan <i>Centroid</i>	35
3.3.2.2 Proses Perhitungan Mean dan Standar Deviasi	36
3.3.2.2.1 Proses Perhitungan Mean	37
3.3.2.2.2 Proses Perhitungan Standar Deviasi.. ..	38
3.3.2.3 Proses Perhitungan Derajat Keanggotaan ...	40
3.3.2.4 Proses Perhitungan Fire Strength.....	41
3.3.2.5 Proses Perhitungan Normalisasi Fire Strength	41
3.3.2.6 Proses Perhitungan Matrik Desain	42
3.3.2.7 Parameter Konsekuensi.....	44
3.3.2.8 Proses Perhitungan Output Jaringan	46
3.3.2.9 Proses Perhitungan Error Jaringan.....	47
3.4 Contoh Perhitungan Manual	49
3.4.1 Clustering K-Means.....	50
3.4.2 Perhitungan Mean dan Standar Deviasi	53
3.4.3 Perhitungan Lapisan 1 (Derajat Keanggotaan).....	54
3.4.4 Perhitungan Lapisan 2 dan 3 (Firing Strenght dan Firing Strenght Ternormalisasi.....	55
3.4.5 Perhitungan Lapisan 4 (Parameter Konsekuensi)	

Menggunakan LSE.....	56
3.4.6 Perhitungan Lapisan 5 (Output Jaringan).....	60
3.4.7 Propagasi Error Menggunakan Metode Steepest Descent	61
3.5 Perancangan Pengujian dan Analisis	63
3.5.1 Pengujian Laju Pembelajaran ANFIS	63
3.5.2 Pengujian Hasil Aktual dan Output Jaringan	64
3.5.3 Pengujian Tingkat Akurasi Kerawanan.....	65
3.6 Rancangan Antar Muka	66

BAB IV IMPLEMENTASI DAN PEMBAHASAN

4.1 Lingkungan Implementasi	69
4.1.1 Lingkungan Perangkat Keras	69
4.1.2 Lingkungan Perangkat Keras	69
4.2 Implementasi Program	69
4.2.1 Tahap Penelitian	69
4.2.1.1 Pengklasteran K-Means Clustering	70
4.2.1.2 Menghitung Mean dan Standar Deviasi	73
4.2.1.3 Menghitung Lapisan Satu	74
4.2.1.4 Menghitung Lapisan Dua	75
4.2.1.5 Menghitung Lapisan Tiga	75
4.2.1.6 Menghitung Lapisan Empat	75
4.2.1.7 Menghitung Parameter Konsekuensi Menggunakan LSE.....	76
4.2.1.8 Menghitung Output Jaringan	77
4.2.1.9 Menghitung Propagasi Error	78
4.2.1.10 Menghitung Nilai Mean dan Standar deviasi	78
4.2.1.11 Proses Iterasi untuk Memperbarui Nilai Mean dan Standar Deviasi	80
4.2.2 Tahap Pengujian	80
4.3 Implementasi Antar Muka	81
4.4 Uji Coba Sistem	85
4.4.1 Uji Laju Pembelajaran	86
4.4.2 Uji Perbandingan Hasil Aktual dan Hasil ANFIS	88
4.4.3 Uji Tingkat Akurasi	93
4.5 Analisa Hasil	94
4.5.1 Analisa Hasil Laju Pembelajaran	94
4.5.2 Analisa Hasil Penentuan Kerawanan DBD	97
4.5.3 Analisa Hasil Tingkat Akurasi Kerawanan DBD.....	99

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan 101
5.2 Saran 101

DAFTAR PUSTAKA 103

UNIVERSITAS BRAWIJAYA



DAFTAR GAMBAR

	Halaman
Gambar 2.1. Inferensi dengan menggunakan metode Sugeno	18
Gambar 2.2. Struktur ANFIS	18
Gambar 3.1. Langkah-langkah penelitian.....	25
Gambar 3.2. Arsitektur Sistem Penentuan Kerawanan Menggunakan ANFIS	28
Gambar 3.3. Proses pelatihan kerawanan menggunakan ANFIS	30
Gambar 3.4. Proses pengujian kerawanan menggunakan ANFIS	31
Gambar 3.5. Proses clustering K-Means	33
Gambar 3.6. Proses perhitungan standarisasi data	34
Gambar 3.7. Proses perhitungan bobot	35
Gambar 3.8. Proses perhitungan jarak antar data dan <i>centroid</i> .	36
Gambar 3.9. Proses perhitungan mean dan standart deviasi	37
Gambar 3.10. Proses perhitungan mean	38
Gambar 3.11. Proses perhiungan standart deviasi	39
Gambar 3.12. Proses perhitungan derajat keanggotaan	40
Gambar 3.13. Proses perhitungan fire strength.....	41
Gambar 3.14. Proses perhitungan normalisasi fire strength.....	42
Gambar 3.15. Proses perhitungan matrik desain	43
Gambar 3.16. Proses perhitungan parameter konsekuens.....	45
Gambar 3.17. Proses perhitungan output jaringan.....	46
Gambar 3.18. Proses perhitungan error jaringan	48
Gambar 3.19. Perancangan antar muka ANFIS pelatihan	66
Gambar 3.20. Perancangan antar muka ANFIS pengujian	67
Gambar 3.21. Perancangan antar muka ANFIS prediksi	68
Gambar 4.1. Tab pelatihan	81
Gambar 4.2. Proses pelatihan ANFIS	82
Gambar 4.3. Tab pengujian	83
Gambar 4.4. Proses pengujian ANFIS.....	84
Gambar 4.5. Tab prediksi	84
Gambar 4.6. Proses prediksi.....	85
Gambar 4.7. Grafik pengaruh laju pembelajaran terhadap RMSE dengan menggunakan 40 data latih	94
Gambar 4.8. Grafik pengaruh laju pembelajaran terhadap RMSE dengan menggunakan 75 data latih	95
Gambar 4.9. Grafik pengaruh laju pembelajaran terhadap RMSE	

	dengan menggunakan 110 data latih	96
Gambar 4.10.	Grafik perbandingan output aktual dan output jaringan menggunakan 40 data latih	97
Gambar 4.11.	Grafik perbandingan output aktual dan output jaringan menggunakan 75 data latih	98
Gambar 4.12.	Grafik perbandingan output aktual dan output jaringan menggunakan 110 data latih	99
Gambar 4.13.	Tingkat akurasi prediksi kerawanan DBD	100

UNIVERSITAS BRAWIJAYA

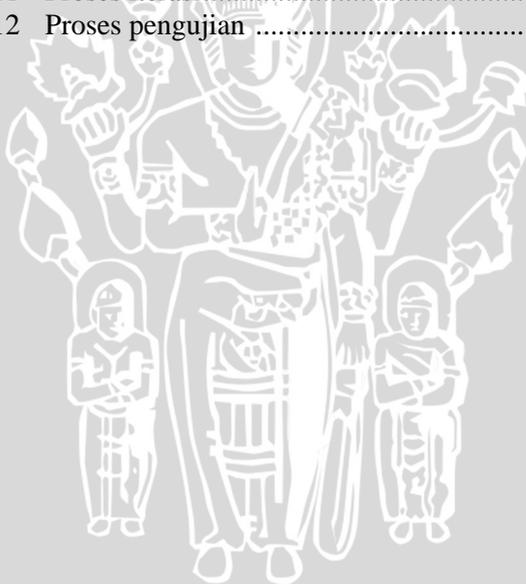


DAFTAR TABEL

	Halaman
Tabel 3.1. Contoh data-data penelitian	49
Tabel 3.2. Standarisasi dan nilai bobot	50
Tabel 3.3. Jarak Euclidean pada iterasi ke-1.....	51
Tabel 3.4. Jarak Euclidean pada iterasi ke-2.....	51
Tabel 3.5. Jarak Euclidean pada iterasi ke-3.....	52
Tabel 3.6. Hasil cluster	53
Tabel 3.7. Output lapisan pertama (derajat keanggotaan)	55
Tabel 3.8. Output lapisan kedua dan ketiga.....	55
Tabel 3.9. Matrik desain.....	57
Tabel 3.10. Nilai f	60
Tabel 3.11. Output jaringan	61
Tabel 3.12. Error pada epoh pertama	62
Tabel 3.13. Contoh tabel laju pembelajaran ANFIS	63
Tabel 3.14. Contoh tabel hasil perbandingan kerawanan.....	64
Tabel 3.15. Contoh tabel hasil pengujian tingkat akurasi	65
Tabel 4.1. Pengujian laju pembelajaran untuk 40 data latih	86
Tabel 4.2. Pengujian laju pembelajaran untuk 75 data latih	87
Tabel 4.3. Pengujian laju pembelajaran untuk 110 data latih ...	88
Tabel 4.4. Pengujian kerawanan dengan 40 data latih.....	88
Tabel 4.5. Pengujian kerawanan dengan 75 data latih.....	90
Tabel 4.6. Pengujian kerawanan dengan 110 data latih.....	92
Tabel 4.7. Pengujian tingkat akurasi kerawanan	93

DAFTAR SOURCECODE

	Halaman
Source code 4. 1 Pengkluster menggunakan K-Mean	70
Source code 4. 2 Perhitungan mean dan standar deviasi	73
Source code 4. 3 Perhitungan lapisan satu	74
Source code 4. 4 Perhitungan lapisan dua	75
Source code 4. 5 Perhitungan lapisan tiga	75
Source code 4. 6 Perhitungan lapisan empat	75
Source code 4. 7 Perhitungan parameter konsekuensi menggunakan LSE	76
Source code 4. 8 Perhitungan output jaringan	77
Source code 4. 9 Perhitungan propagasi error	78
Source code 4.10 Perbaharuan mean dan standar deviasi	78
Source code 4.11 Proses iterasi	80
Source code 4.12 Proses pengujian	81



DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Data latih 40	107
Lampiran 2. Data latih 75	108
Lampiran 3. Data latih 110	110
Lampiran 4. Data uji 40	113
Lampiran 5. Pengujian Laju Pembelajaran Untuk 40 Data Latih	115
Lampiran 6. Pengujian Laju Pembelajaran Untuk 75 Data Latih	115
Lampiran 7. Pengujian Laju Pembelajaran Untuk 110 Data Latih	
.....	116



BAB I

PENDAHULUAN

1.1 Latar Belakang

Penyakit Demam Berdarah Dengue (DBD) adalah penyakit menular yang disebabkan oleh virus dengue dan ditularkan melalui gigitan nyamuk *Aedes aegypti* (Suroso, 1999). Lingkungan merupakan salah satu faktor yang sangat berperan dalam timbul dan penyebaran penyakit DBD, baik lingkungan biologis maupun fisik. Sebuah penelitian di Thailand menyebutkan bahwa suhu dan kelembapan udara serta curah hujan memiliki pengaruh yang signifikan terhadap angka insiden DBD (Prompou dan Jaroensutasinee, 2005). Menurut Soegijanto S. (2003), faktor lingkungan fisik yang berperan terhadap timbulnya penyakit DBD meliputi kelembapan nisbi, cuaca, kepadatan larva dan nyamuk dewasa, lingkungan di dalam rumah, lingkungan di luar rumah dan ketinggian tempat tinggal. Unsur-unsur tersebut saling berperan dan terkait pada kejadian infeksi virus dengue.

Di Negara-negara dengan empat musim, epidemi DBD berlangsung terutama pada musim panas meskipun ditemukan kasus-kasus DBD sporadis pada musim dingin. Di negara-negara di Asia Tenggara, epidemi DBD terutama terjadi pada musim penghujan. Di Indonesia epidemi DBD terjadi beberapa minggu setelah datangnya musim penghujan. Epidemi mencapai angka tertinggi pada sebulan setelah curah hujan mencapai puncak tertinggi untuk kemudian menurun sejalan dengan menurunnya curah hujan (Djunaedi, 2006). Angka kematian yang disebabkan oleh nyamuk *Aedes aegypti* masih cukup tinggi sekarang ini, karena cuaca yang cenderung lembab akibat intensitas hujan yang tinggi sehingga membuat nyamuk *Aedes aegypti* lebih cepat berkembang biak. Pada penelitian sebelumnya yang dilakukan oleh Viani(2009) menjelaskan bahwa salah satu upaya yang dapat dilakukan agar dapat mencegah penyebaran demam berdarah adalah dengan menganalisis pola penyebaran penyakit ini. Analisis ini dilakukan dengan menggunakan metode *spatial statistic*. Metode ini dapat memetakan penyebaran penyakit demam berdarah di Kodya Surabaya dalam kurun waktu tertentu. Hasil pengolahannya dapat memberikan informasi daerah yang telah

terjangkit penyakit demam berdarah, sehingga dapat dilakukan tindakan agar tidak terjadi penyebaran di daerah yang belum terjangkit.

Pada tahun 1992, J.S.R. Jang mengembangkan sistem *Neuro-fuzzy* yang disebut *adaptive neuro fuzzy inference system* (ANFIS). ANFIS adalah jaringan adaptif yang berbasis pada *fuzzy inference system* (sistem inferensi fuzzy). Zhu (2000) dan Shapiro (2002) mengemukakan bahwa ANFIS merupakan metode permodelan terbaik untuk menganalisis data numerik, karena dalam proses training didasarkan minimalisasi nilai kesalahan atau *root mean square error* (RMSE) dari *output*-nya.

Pada penelitian sebelumnya yang dilakukan oleh Muhammad Syafii (2006) menjelaskan bahwa ANFIS dapat digunakan secara selektif untuk mendiagnosa penyakit DBD dan dapat digunakan dengan baik apabila terpenuhi kondisi minimal sesuai aturan yang sudah diketahui. Selain itu ANFIS juga mampu mengembangkan sistem prediksi curah hujan (Ruminta, 2001) dan dibidang kesehatan, ANFIS diaplikasikan untuk menganalisis kondisi ginjal (Subekti, 1999). Menurut penelitian Riyanto et al. (2000), ANFIS dapat memprediksi data deret waktu lebih akurat dibanding metode lainnya, seperti *Back Propagation Multilayer Perceptron* (BPMP) maupun autoregresi. Hasil penelitian lainnya menunjukkan ANFIS dapat mengidentifikasi model curah hujan lebih akurat dibandingkan dengan ARIMA (Ruminta, 2001). Oleh sebab itu penelitian yang akan dilakukan ini memodelkan sistem ANFIS untuk menentukan daerah yang rawan terhadap DBD karena ANFIS merupakan metode yang sesuai untuk kasus ini.

1.2 Rumusan Masalah

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

1. Bagaimana mengimplementasikan ANFIS pada kasus penentuan daerah yang rawan terhadap DBD.
2. Berapa laju pembelajaran yang paling baik pada kasus penentuan daerah yang rawan terhadap DBD.
3. Berapa persentase tingkat akurasi dalam menentukan daerah yang rawan terhadap DBD dengan menggunakan metode ini.

1.3 Batasan Masalah

Dari permasalahan yang ada diatas, akan diberikan batasan-batasan yaitu:

1. Faktor yang digunakan untuk menentukan rawan tidaknya suatu daerah terhadap penyakit DBD adalah faktor iklim yang meliputi suhu, curah hujan, kelembapan dan kecepatan angin.
2. Daerah yang diteliti sesuai dengan sumber data yang telah didapatkan, yaitu di daerah kabupaten Banyuwangi.

1.4 Tujuan

Ada beberapa tujuan yang ingin dicapai dalam melaksanakan penelitian untuk penulisan tugas akhir yaitu,

1. Mengimplementasikan ANFIS sebagai algoritma dalam penentuan daerah rawan penyakit DBD.
2. Mengukur laju pembelajaran yang paling baik untuk penentuan daerah rawan penyakit DBD.
3. Mengukur tingkat akurasi penentuan daerah yang rawan terhadap DBD dari sistem yang dilatih menggunakan ANFIS.

1.5 Manfaat

Sistem ini dapat digunakan sebagai bahan informasi untuk mengevaluasi dan meningkatkan pelayanan kesehatan khususnya bagi program pengendalian dan pemberantasan penyakit demam berdarah dengue (DBD) di wilayah Kabupaten Banyuwangi.

1.6 Sistematika Penulisan

Tugas akhir ini disusun berdasarkan sistematika penulisan sebagai berikut:

1. **BAB I PENDAHULUAN**

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

2. **BAB II TINJAUAN PUSTAKA**

Bab ini berisi teori-teori yang berkaitan dengan topik penulisan skripsi. Teori tersebut meliputi penjelasan tentang penyakit

demam berdarah dengue serta konsep dari *adaptive neuro fuzzy inference system* (ANFIS)

3. **BAB III METODOLOGI DAN PERANCANGAN SISTEM**

Bab ini berisi membahas tentang data penelitian, spesifikasi perangkat lunak, perancangan sistem, perancangan basis data dan perancangan antarmuka.

4. **BAB IV IMPLEMENTASI DAN UJI COBA SISTEM**

Bab ini berisi hasil implementasi perangkat lunak, uji coba sistem dan analisa hasil.

5. **BAB V KESIMPULAN DAN SARAN**

Bab ini berisi kesimpulan dari seluruh rangkaian penelitian serta saran-saran untuk kemungkinan pengembangannya.



BAB II TINJAUAN PUSTAKA

2.1 Demam Berdarah Dengue (DBD)

Dengue adalah penyakit infeksi virus yang ditularkan melalui nyamuk spesies *Aedes* (Hendarwanto 1987). Infeksi virus dengue pada manusia mengakibatkan suatu spektrum manifestasi klinis yang bervariasi. WHO pada tahun 1975 menetapkan 4 kriteria klinis untuk diagnosa DBD sebagai berikut (Hasan 1985) :

1. Demam tinggi dengan mendadak dan terus menerus selama 2-7 hari.
2. Dijumpai manifestasi pendarahan, paling sedikit *rumpel leede test* (uji tornikuet) positif dan terdapat salah satu bentuk pendarahan yaitu pendarahan pada kulit (petekia, purpura, ekimosis), pendarahan hidung (epistaksis), pendarahan gusi, muntah berdarah (hematemesis) dan berak berdarah (melena).
3. Pembesaran hati (hepatomegali) dan
4. Shock yang ditandai dengan nadi lemah, cepat, tekanan darah menurun (tekanan sistolis kurang 80mmHg, normal 120mmHg), kulit dingin dan lembab terutama ujung jari tangan dan kaki, penderita gelisah dan bibir kebiru-biruan (sianosis).

Menurut WHO (dalam Soedarto, 1995), derajat beratnya DBD dibagi menjadi empat tingkatan:

- a. Derajat I: ringan, bila demam mendadak 2-7 hari disertai gejala klinik lain dan manifestasi perdarahan paling ringan yaitu tes tourniquet yang positif.
- b. Derajat II: sedang, dengan gejala lebih berat daripada derajat I, disertai manifestasi perdarahan kulit, epistaksis, perdarahan gusi, hematemesis atau melena. Terdapat gangguan sirkulasi darah perifer yang ringan berupa kulit dingin dan lembab, ujung jari dan hidung dingin.
- c. Derajat III: berat, dengan gejala syok mengikuti gejala-gejala tersebut diatas.
- d. Derajat IV: berat sekali, penderita syok berat, tensi tidak terukur, dan nadi tidak dapat diraba.

2.1.1 Vektor Penular DBD

Vektor penular penyakit DBD adalah nyamuk *Aedes*. Terdapat dua spesies *Aedes* yang menularkan virus dengue, yaitu *Aedes aegypti* dan *Aedes albopictus*. Kedua spesies ini ditemukan hampir diseluruh wilayah Indonesia, kecuali di wilayah dengan ketinggian > 1.000 m diatas permukaan laut. Di Indonesia *Aedes aegypti* merupakan faktor utama yang paling berperan dalam penularan penyakit ini karena vektor tersebut hidup didalam dan sekitar rumah sehingga kesempatan untuk kontak dengan manusia lebih besar, sedangkan *Aedes albopictus* hidup dikebun-kebun.

Nyamuk *Aedes aegypti* biasa hidup di dekat manusia dan menyukai tempat-tempat gelap tersembunyi di dalam rumah sebagai tempat istirahatnya. Larva nyamuk ini dapat ditemukan di dalam atau di dekat perumahan, di dalam kaleng, atau tempat-tempat penyimpanan air yang relatif bersih yang digunakan untuk minum atau mandi. Sedangkan nyamuk *Aedes albopictus* berkembangbiak di dalam lubang-lubang pohon, lekukan tanaman, potongan batang bambu, dan buah kelapa yang terbuka. Larvanya dapat hidup di dalam kaleng dan tempat penampungan air lainnya termasuk timbunan sampah ditempat terbuka. Nyamuk tersebut memperoleh makanan dengan menghisap darah berbagai binatang (Soegijanto,2003).

2.1.2 Faktor-faktor Yang Berhubungan Dengan Kejadian Penyakit DBD

Secara ekologi, transmisi dengue dipengaruhi oleh beberapa faktor yaitu *host*, vektor nyamuk, dan virus, dimana lingkungan secara langsung maupun tidak langsung mempengaruhi tiga populasi tersebut dan interaksinya (Gubler, 1988). Faktor manusia yang berhubungan dengan penularan DBD adalah:

- a. Umur
- b. Suku Bangsa/ Ras
- c. Keadaan Sosial Ekonomi
- d. Ketinggian
- e. Iklim

Dari kelima faktor diatas, iklim yang akan di jadikan parameter pada penelitian ini. Iklim adalah keadaan yang mencirikan atmosfer pada suatu daerah dan waktu yang cukup lama (biasanya dalam periode 30 tahun). Unsur-unsur iklim meliputi :

1. Curah hujan

Curah hujan merupakan unsur iklim yang sangat penting bagi kehidupan di bumi. Jumlah curah hujan dicatat dalam inci atau millimeter (1 inci = 25,4 millimeter). Jumlah curah hujan 1 mm menunjukkan tinggi air hujan yang menutupi permukaan 1 mm, jika air tersebut tidak meresap ke dalam tanah atau tidak menguap ke atmosfer.

Di daerah tropis hujannya lebih lebat dari pada di daerah lintang tinggi. Garis yang menghubungkan titik-titik dengan curah hujan yang sama selama periode tertentu disebut *isohyets* (Tjasyono, B, 1995).

Hujan dapat mempengaruhi kehidupan nyamuk dengan dua cara, yaitu : menyebabkan naiknya kelembaban nisbi udara dan menambah tempat dan perindukan. Setiap 1 mm curah hujan menambah kepadatan nyamuk 1 ekor, akan tetapi apabila curah hujan dalam seminggu sebesar 140 mm, maka larva akan hanyut dan mati. Suhu dan kelembaban udara selama musim hujan sangat kondusif untuk kelangsungan hidup nyamuk dewasa, yang juga meningkatkan kemungkinan hidup nyamuk yang terinfeksi. (Gubler, 1988).

2. Suhu udara

Suhu udara merupakan unsur iklim yang sangat penting yang diukur dengan termometer. Suhu adalah unsur iklim yang sulit didefinisikan , karena unsur ini berubah sesuai dengan tempat. Pengukuran suhu udara hanya memperoleh suatu nilai yang menyatakan nilai rata-rata suhu atmosfer (Tjasyono B, 1995).

Suhu berefek langsung terhadap aktivitas dan kemampuan vektor. Virus dengue hanya endemik di wilayah-wilayah tropis, suhu yang tinggi sekitar 30° C cenderung mempercepat replikasi virus. Nyamuk dapat bertahan hidup pada suhu rendah, tetapi metabolismenya menurun atau bahkan berhenti bila suhu turun

sampai di bawah suhu kritis. Pada suhu yang lebih dari 35° C juga mengalami perubahan dalam arti lebih lambat proses-proses fisiologi. Rata-rata suhu optimum untuk pertumbuhan nyamuk adalah 25° C – 27° C. Pertumbuhan nyamuk akan berhenti sama sekali apabila suhu kurang dari 10° C atau lebih dari 40° C. Kecepatan perkembangan nyamuk tergantung dari kecepatan proses metabolisme yang sebagian diatur oleh suhu. Suhu yang tetap lebih dari 27 – 30° C akan mengurangi rata-rata umur populasi nyamuk (Suroso, 2001).

3. Kelembaban udara

Kelembaban udara adalah banyaknya uap air yang terkandung dalam udara yang biasanya dinyatakan dalam persen (%). Umur (*longevity*) nyamuk dipengaruhi oleh kelembaban udara. Pada suhu 20° C kelembaban udara 27% umur nyamuk betina 68 hari dan umur nyamuk jantan 35 hari. Kelembaban 55% umur nyamuk betina 88 hari dan nyamuk jantan 50 hari dan apabila kelembaban 85% umur nyamuk betina 104 hari nyamuk jantan 68 hari. Tanpa makan darah umur nyamuk 104 hari dengan makan darah umur nyamuk mencapai 122 hari. Pada kelembaban kurang dari 60% umur nyamuk akan menjadi pendek, tidak bisa menjadi vektor karena tidak cukup waktu untuk perpindahan virus dari lambung ke kelenjar ludah (Gubler, 1988).

4. Kecepatan angin

Angin ialah gerak udara yang sejajar dengan permukaan bumi. Udara bergerak dari daerah bertekanan tinggi ke daerah bertekanan rendah. Angin diberi nama sesuai dengan arah dari mana angin datang. Misalnya angin timur adalah angin yang datang dari arah timur, angin laut adalah angin yang bertiup dari laut ke darat.

Secara tidak langsung angin akan mempengaruhi penguapan air dan suhu udara. Pengaruh angin yang langsung terhadap nyamuk adalah pada saat terbang. Bila kecepatan angin 11-14 meter per detik atau 25-31 mil/ jam atau 22-28 knot akan menghambat aktifitas terbang nyamuk (Tjasyono, B, 1995).

2.2 Sistem Fuzzy

Sistem *fuzzy* pertama kali diperkenalkan oleh Prof. L. A. Zadeh dari Berkelay pada tahun 1965. Sistem *fuzzy* merupakan penduga numerik yang terstruktur dan dinamis. Sistem ini mempunyai kemampuan untuk mengembangkan sistem intelijen lingkungan yang tidak pasti. Sistem ini menduga suatu fungsi dengan logika *fuzzy*. Dalam logika *fuzzy* terdapat beberapa proses yaitu penentuan himpunan *fuzzy*, penerapan aturan **IF-THEN** dan proses inferensi *fuzzy* (Marimin 2005).

2.2.1 Himpunan Fuzzy

Pada himpunan tegas (*crisp*), nilai keanggotaan suatu item x dalam suatu himpunan A , yang sering ditulis dengan $\mu_A(x)$, memiliki dua kemungkinan, yaitu:

- Satu (1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan, atau
- Nol (0), yang berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan.

Jika pada himpunan *crisp*, nilai keanggotaan hanya ada 2 kemungkinan, yaitu 0 atau 1, pada himpunan *fuzzy* nilai keanggotaannya terletak pada rentang 0 sampai 1. Apabila x memiliki nilai keanggotaan *fuzzy* $\mu_A(x)=1$ berarti x menjadi anggota penuh pada himpunan A . Himpunan *fuzzy* memiliki 2 atribut, yaitu:

- Linguistik, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti: MUDA, PAROBAYA, TUA
- Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variable, seperti: 40, 25, 50, dsb.

Ada beberapa hal yang perlu diketahui dalam memahami sistem fuzzy yaitu:

- Variabel *Fuzzy* merupakan variabel yang akan dibahas dalam sistem *fuzzy* misalnya umur, temperatur, permintaan, dsb.
- Himpunan *Fuzzy* merupakan suatu group yang mewakili suatu kondisi tertentu dalam variabel *fuzzy* misalnya variabel umur dibagi atas 3 himpunan *fuzzy* yaitu muda, parobaya dan tua.
- Semesta Pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel *fuzzy*

misalnya semesta pembicaraan variabel umur adalah 0 sampai 100.

- d. Domain adalah keseluruhan nilai yang diijinkan dalam semesta pembicaraan dan boleh dioperasikan dalam himpunan *fuzzy* misalnya domain umur muda 20 – 45, domain parobaya 25 – 65 dan domain tua 45 – 70 (Kusumadewi 2004).

2.2.2 Fungsi Keanggotaan

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data ke dalam nilai keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Ada beberapa fungsi yang dapat digunakan antara lain:

a. Representasi Linier

Pada representasi linier, pemetaan *input* ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Bentuk ini paling sederhana dan menjadi pilihan yang baik untuk mendekati suatu konsep yang kurang jelas.

Ada 2 keadaan himpunan *fuzzy* yang linier. Pertama, kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol (0) bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi. Persamaan 2.1 merupakan persamaan dari fungsi keanggotaan representasi linier naik.

$$\mu[x] = \begin{cases} 0; & x \leq a \\ \frac{(x-a)}{(b-a)}; & a \leq x \leq b \\ 1; & x \geq b \end{cases} \quad (2.1)$$

Sedangkan pada linier turun, garis lurus dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah dengan fungsi keanggotaan. Persamaan 2.2 merupakan persamaan dari fungsi keanggotaan representasi linier turun.

$$\mu[x] = \begin{cases} \frac{(b-x)}{(b-a)}; & a \leq x \leq b \\ 0; & x \leq a \text{ atau } x \geq b \end{cases} \quad (2.2)$$

b. Representasi Kurva Segitiga

Fungsi keanggotaan segitiga ditandai oleh adanya 3 (tiga) parameter {a,b,c} yang akan menentukan koordinat x dari tiga sudut. Kurva ini pada dasarnya merupakan gabungan antara dua garis (*linier*). Persamaan 2.3 merupakan persamaan dari fungsi keanggotaan segitiga.

$$\mu[x] = \begin{cases} 0; & x \leq a \text{ atau } x \geq c \\ \frac{(x-a)}{(b-a)}; & a \leq x \leq b \\ \frac{(c-x)}{(c-b)}; & b \leq x \leq c \end{cases} \quad (2.3)$$

c. Representasi Kurva Bell

Fungsi keanggotaan generalized bell, disifati oleh parameter {a,b,c}. Persamaan 2.4 merupakan persamaan dari fungsi keanggotaan bell.

$$\text{bell}(x,a,b,c) = \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}} \quad (2.4)$$

parameter b selalu positif, supaya kurva menghadap kebawah

d. Representasi Kurva Bahu

Representasi fungsi keanggotaan *fuzzy* dengan menggunakan kurva bahu pada dasarnya adalah gabungan dari kurva segitiga dan kurva trapesium. Daerah yang terletak di tengah-tengah suatu variabel yang direpresentasikan dalam bentuk segitiga, pada sisi kanan dan kirinya akan naik dan turun. Tetapi terkadang pada salah sisi dari variabel *fuzzy* yang ditinjau ini terdapat nilai yang konstan, yaitu pada himpunan ekstrim kiri dan ekstrim kanan (Kusumadewi 2004).

2.3 Algoritma *clustering* K-Means

K-Means merupakan algoritma *clustering* yang berulang-ulang. Algoritma K-Means dimulai dengan pemilihan secara acak K, K disini merupakan banyaknya *cluster* yang ingin dibentuk. Kemudian tetapkan nilai-nilai K secara random, untuk sementara nilai tersebut menjadi pusat dari *cluster* atau biasa disebut dengan centroid, mean atau “means”. Hitung jarak setiap data yang ada terhadap masing-masing centroid menggunakan rumus Euclidian hingga ditemukan jarak yang paling dekat dari setiap data dengan centroid. Klasifikasikan setiap data berdasarkan kedekatannya dengan centroid. Lakukan langkah tersebut hingga nilai centroid tidak berubah (stabil) (Witten, Ian H. 2005).

2.3.1 Langkah-langkah penerapan algoritma K-Means

Langkah-langkah untuk melakukan pengelompokan dengan tujuan menghasilkan suatu data yang terkelompok adalah sebagai berikut (Santosa, 2007):

1. Standarisasi data yang akan dikelompokan (menentukan bobot dari data mentah yang telah didapatkan). Hal ini dilakukan agar data mempunyai skala yang sama, sehingga pengelompokan akan stabil. Persamaan 2.5 merupakan persamaan untuk penghitungan standarisasi data.

$$X_{ij} = \frac{X_{ij} - XMin_j}{XMax_j - XMin_j}, \quad i = 1, 2, \dots, n; j = 1, 2, \dots, k \quad (2.5)$$

Dimana :

X_{ij} = Data yang distandarisasi

$XMin_j$ = Nilai min pada table ke – j

$XMax_j$ = Nilai max pada table ke – j

Setelah itu menghitung bobot tiap data dengan menggunakan persamaan 2.6, yaitu :

$$\frac{x_1 + x_2 + \dots + x_n}{\sum x_n} \quad (2.6)$$

dimana x_1, x_2, \dots, x_n adalah inputan.

2. Melakukan pengelompokan dengan metode *K-Means Clustering*, langkah-langkahnya sebagai berikut :

- a. Menentukan jumlah *cluster* k .
- b. Menginisialisasi k pusat *cluster* ini bisa dilakukan dengan berbagai cara. Yang paling sering dilakukan adalah dengan cara random. Pusat – pusat *cluster* diberi nilai awal dengan angka-angka random.
- c. Menempatkan setiap data/obyek ke *cluster* terdekat. Kedekatan dua obyek ditentukan jarak antara data dengan pusat *cluster*. Dalam tahap ini perlu dihitung jarak tiap data ke tiap pusat *cluster*. Jarak paling dekat antara satu data dengan data satu *cluster* tertentu akan menentukan suatu data masuk dalam *cluster* mana.

Menentukan ukuran kemiripan atau ketidakmiripan antar data dengan metode jarak Euclidean sesuai dengan persamaan 2.7 :

$$d(x,y) = \|x - y\|^2 = \sqrt{\sum_{i=1}^n (X_i - Y_i)^2} \quad (2.7)$$

dimana :

$d(x,y)$ = ukuran ketidakmiripan

$X = (x_1, x_2, \dots, x_j)$ adalah variabel data

$Y = (y_1, y_2, \dots, y_j)$ adalah variabel pada titik pusat *cluster*.

- d. Menghitung kembali pusat *cluster* dengan keanggotaan *cluster* yang sekarang. Pusat *cluster* adalah rata-rata dari semua data/obyek dalam *cluster* tertentu. Jika dikehendaki bisa juga memakai median dari cluster tersebut. Jika rata-rata (*mean*) bukan satu-satunya ukuran yang bisa dipakai
- e. Menugaskan lagi setiap obyek dengan memakai pusat *cluster* yang baru. Jika pusat *cluster* sudah tidak berubah lagi, maka proses pengklasteran selesai. Atau, kembali lagi ke langkah ketiga sampai pusat *cluster* tidak berubah lagi.
- f. Hasil akhir didapatkan data yang telah tercluster yang akan dimasukkan untuk pencarian mean (\bar{x}) dan standar deviasi (σ), seperti pada persamaan 2.8 dan 2.9:

$$\bar{x} = \frac{(x_1 + x_2 + \dots + x_n)}{\sum n} \quad (2.8)$$

dimana :

\bar{x} = mean

$x = (x_1, x_2, \dots, x_n)$ adalah variabel data

n = jumlah data

$$\sigma = \sqrt{\frac{\sum (x - \bar{x})^2}{(n-1)}} \quad (2.9)$$

dimana :

σ = standar deviasi

\bar{x} = mean

x = adalah variabel data

n = jumlah data

Nilai mean dan standart deviasi ini merupakan nilai a dan c dalam persamaan 2.4.

2.4 Sistem Inferensi Fuzzy

Sistem Inferensi *Fuzzy* merupakan satu kerangka komputasi yang didasarkan pada teori himpunan *fuzzy*, aturan *fuzzy* berbentuk IF-THEN dan penalaran *fuzzy*. Secara garis besar, proses inferensi *fuzzy* dapat dijelaskan sebagai berikut :

System inferensi *fuzzy* menerima input crisp. Input ini kemudian dikirim ke basis pengetahuan yang berisi n aturan *fuzzy* dalam bentuk IF-THEN. *Fire Strength* akan dicari pada setiap aturan. Apabila jumlah aturan lebih dari satu, maka akan dilanjutkan agregasi dari semua aturan. Selanjutnya, pada hasil agregasi akan dilakukan defuzzy untuk mendapatkan nilai *crisp* sebagai output sistem.

System inferensi *fuzzy* yang umum digunakan pada sitem seperti ANFIS, yaitu metode Takagi Sugeno Kang (TSK) atau biasa disebut Sugeno saja (Kusumadewi, 2004).

2.4.1 Metode Sugeno

System inferensi *fuzzy* menggunakan metode Sugeno, memiliki karakteristik yaitu konsekuensi bukan merupakan himpunan *fuzzy*, namun merupakan suatu persamaan linier dengan variabel-variabel sesuai dengan variabel-variabel inputnya. Ada 2 model untuk sistem inferensi *fuzzy* dengan menggunakan metode Sugeno, yaitu orde-0 dan orde-1.

Secara umum bentuk model *fuzzy* Sugeno orde-0 adalah :

IF $(x_1 \text{ is } A_1) \circ (x_2 \text{ is } A_2) \circ \dots \circ (x_n \text{ is } A_n)$ **THEN** $z = k$

dengan A_i adalah himpunan *fuzzy* ke i sebagai anteseden, \circ adalah operator *fuzzy* (seperti AND atau OR), dan k adalah suatu konstanta sebagai konsekuensi.

Sedangkan model *fuzzy* Sugeno orde-1, secara umum berbentuk :

IF $(x_1 \text{ is } A_1) \circ \dots \circ (x_n \text{ is } A_n)$ **THEN** $z = p_1 * x_1 + \dots + p_n * x_n + q$

dengan A_i adalah himpunan *fuzzy* ke i sebagai anteseden, \circ adalah operator *fuzzy* (seperti AND atau OR), dan p_i adalah suatu konstanta ke- i dan q juga merupakan konstanta dalam konsekuensi (Kusumadewi 2004).

2.5 Jaringan Syaraf Tiruan

JST terdiri dari sejumlah elemen pemrosesan informasi yang disebut neuron. Masing-masing neuron dihubungkan oleh saluran penghubung yang memiliki bobot. Bobot ini merepresentasikan informasi yang digunakan untuk pemecahan masalah oleh JST. Setiap neuron memiliki tingkat aktivasi. Umumnya setiap neuron mengirimkan nilai aktivasinya ke beberapa neuron lainnya.

Kemampuan belajar JST direpresentasikan dalam metode pembelajaran. Salah satu metode pembelajaran JST menggunakan algoritma pembelajaran *backpropagation* (propagasi balik). Metoda ini merupakan algoritma pembelajaran terawasi dan biasanya digunakan oleh perceptron dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan neuron yang ada pada lapisan tersembunyi.

Algoritma *backpropagation* menggunakan error output mengubah nilai bobot dalam arah mundur (*backward*). Untuk mendapatkan error ini tahap perambatan maju (*forward*) harus dikerjakan terlebih dahulu. Pada saat penambahan maju, neuron-neuron diaktifkan dengan menggunakan fungsi aktivasi sigmoid (Kusumadewi 2003).

2.5.1 Arsitektur Jaringan Syaraf

Jaringan syaraf tiruan memiliki beberapa arsitektur jaringan yang sering digunakan dalam berbagai aplikasi. Arsitektur JST tersebut, antara lain :

1. Jaringan layar tunggal (*single layer network*)

Jaringan dengan lapisan tunggal terdiri dari 1 layer input dan 1 layer output. Setiap neuro/unit yang terdapat di dalam lapisan/layer input selalu terhubung dengan setiap neuron yang terdapat pada layer output. Jaringan ini hanya menerima input kemudian secara langsung akan mengolahnya menjadi output tanpa harus melalui lapisan tersembunyi.

2. Jaringan layar jamak (*multi layer network*)

Jaringan dengan lapisan jamak memiliki ciri khas tertentu yaitu memiliki 3 jenis layer yakni layer input, layer output, dan juga layer tersembunyi. Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih kompleks dibandingkan jaringan dengan lapisan tunggal. Namun, proses pelatihan sering membutuhkan waktu yang cenderung lama.

3. Jaringan dengan lapisan kompetitif (*competitive layer network*)

Pada jaringan ini sekumpulan neuron bersaing untuk mendapatkan hak menjadi aktif (Kusumadewi 2003).

2.5.2 Metode Pelatihan/ Pembelajaran

Cara berlangsungnya pembelajaran atau pelatihan jaringan syaraf tiruan ada tiga macam, yaitu terawasi (*supervised learning*), tidak terawasi (*unsupervised learning*), dan merupakan gabungan keduanya (*hybrid*). Pada pembelajaran terawasi, kumpulan input yang digunakan outputnya telah diketahui. Pada pembelajaran tak terawasi, jaringan syaraf tiruan mengorganisasi dirinya sendiri untuk

membentuk vector-vektor input yang serupa. Paradigma pembelajaran hibrida merupakan kombinasi dari kedua aturan diatas. Sebagian dari bobot-bobotnya ditentukan melalui pembelajaran terawasi dan sebagian lainnya melalui pembelajaran tak terawasi (Puspitaningrum, 2006).

2.6 Adaptive Neuro Fuzzy Inference System (ANFIS)

Jaringan neural adalah struktur jaringan dimana keseluruhan tingkah laku masukan-keluaran ditentukan oleh sekumpulan parameter-parameter yang dimodifikasi. Sistem *fuzzy* dapat melukiskan suatu sistem dengan pengetahuan linguistik yang mudah dimengerti. Sistem inferensi *fuzzy* dapat digabung dengan algoritma propagasi balik berdasarkan pasangan data masukan-keluaran menggunakan arsitektur jaringan neural. Dengan cara ini memungkinkan sistem *fuzzy* dapat belajar. Gabungan sistem *fuzzy* dengan jaringan neural ini biasa disebut dengan *neuro-fuzzy*.

ANFIS dikembangkan oleh J.S.R Jang pada tahun 1992. Menurut Jang kelas *adaptive network* secara fungsional ekuivalen dengan *fuzzy inference system* . ANFIS adalah arsitektur yang secara fungsional sama dengan *fuzzy rule base* model Sugeno orde satu.

ANFIS adalah sistem inferensi *fuzzy* yang diimplementasikan dalam jaringan adaptif. Pada ANFIS, parameter adalah fungsi keanggotaan premis dan konsekuensi. Pembelajaran ANFIS adalah pengubahan parameter fungsi keanggotaan masukan dan keluaran Kusumadewi (2006).

2.6.1 Struktur ANFIS

Sistem *neuro-fuzzy* berstruktur ANFIS (*Adaptive Neuro Fuzzy Inference Sistem* atau biasa disebut juga *Adaptive Network-based Fuzzy Inference Sistem*) termasuk dalam kelas jaringan neural namun berdasarkan fungsinya sama dengan sistem inferensi fuzzy (Jang, J.-S. R. 1993).

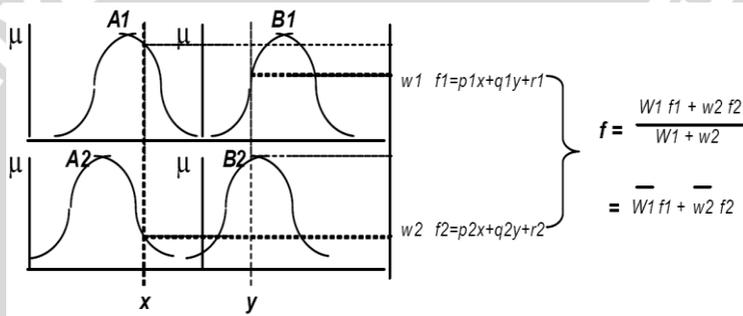
Pada *neuro-fuzzy*, proses belajar pada jaringan neural dengan sejumlah pasangan data berguna untuk memperbaharui parameter-parameter sistem inferensi *fuzzy*. *Fuzzy inference* sistem yang digunakan memiliki dua input yaitu x dan y dan satu output f . Untuk

model *fuzzy* Sugeno orde satu, aturan yang umum dengan dua aturan *fuzzy if-then*, adalah sebagai berikut (Jang, J.-S. R. 1997):

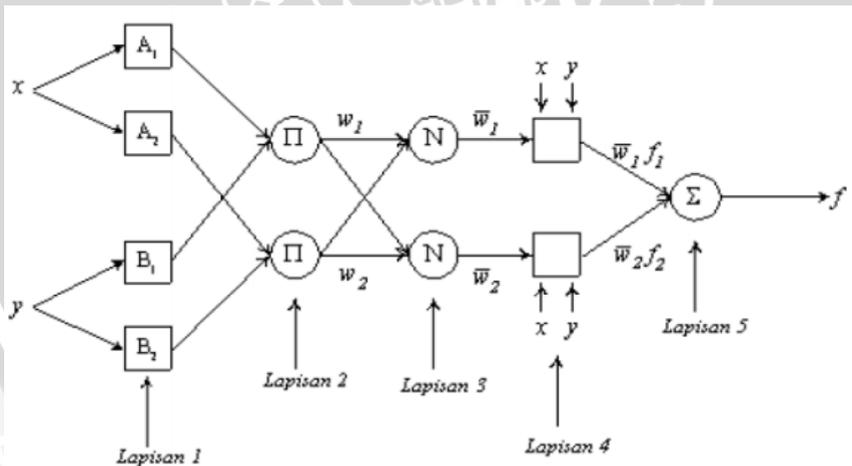
Rule 1 : if x is A_1 and y is B_1 , then $f_1 = p_1x + q_1y + r_1$;

Rule 2 : if x is A_2 and y is B_2 , then $f_2 = p_2x + q_2y + r_2$;

Gambar 2.1 menggambarkan mekanisme pertimbangan untuk model Sugeno ini; struktur ANFIS untuk fungsi yang sama ditunjukkan pada Gambar 2.2, dimana simpul pada lapisan yang sama memiliki fungsi yang sama. Keluaran masing-masing simpul dalam layer l dituliskan O_l, i (Jang, J.-S. R. 1997).



Gambar 2.1 Inferensi dengan menggunakan metode Sugeno



Gambar 2.2 Struktur ANFIS (Jang, J.-S.R., 1997)

Penjelasan pada masing-masing lapisan sebagai berikut:

Lapisan 1:

Setiap simpul i pada lapisan ini adalah simpul adaptif dengan fungsi simpul (Jang, J.-S. R. 1997). Persamaan 2.10 merupakan persamaan derajat keanggotaan:

$$\begin{aligned} O_{1,i} &= \mu A_i(x), \text{ untuk } i = 1, 2, \text{ atau} \\ O_{1,i} &= \mu B_{i-2}(y), \text{ untuk } i = 3, 4 \end{aligned} \quad (2.10)$$

dimana x (atau y) adalah masukan bagi simpul i , dan A_i (atau B_{i-2}) adalah label bahasa (*linguistic label*) seperti misalnya “kecil” atau “luas”, dll. Dengan kata lain, $O_{1,i}$ adalah tingkatan keanggotaan dari himpunan *fuzzy* A ($= A_1, A_2, B_1$ atau B_2) dan menentukan derajat keanggotaan dari masukan x (atau y) yang diberikan. Fungsi keanggotaan parameter dari A dapat didekati dengan fungsi bell (Jang, J.-S. R. 1997) sesuai dengan persamaan 2.11:

$$\mu_{A_i}(x) = \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}} \quad (2.11)$$

Di mana $\{a_i, b_i, c_i\}$ adalah himpunan parameter. Parameter pada lapisan ini disebut parameterparameter premis.

Lapisan 2:

Setiap simpul pada lapisan ini diberi label Π , bersifat non-adaptif (parameter tetap) yang mempunyai keluaran berupa perkalian dari semua sinyal yang masuk (Jang, J.-S. R. 1997). Persamaan 2.12 merupakan persamaan untuk penghitungan firing strength.

$$O_{2,i} = w_i = \mu A_i(x) \mu B_i(y), i = 1, 2, \dots \quad (2.12)$$

Masing-masing keluaran simpul menyatakan derajat pengaktifan dari aturan *fuzzy*. Secara umum beberapa operator T-norm yang dapat mengungkapkan logika *fuzzy* AND dapat digunakan sebagai fungsi simpul pada lapisan ini.

Lapisan 3:

Setiap simpul pada lapisan ini diberi label N , juga bersifat non-adaptif. Masing-masing simpul menampilkan derajat pengaktifan ternormalisasi dengan bentuk (Jang, J.-S. R. 1997). Persamaan 2.13 merupakan persamaan untuk penghitungan normalisasi firing strength.

$$O_{3,i} = \bar{w}_i = \frac{w_i}{w_1 + w_2}, i = 1, 2 \quad (2.13)$$

Apabila dibentuk lebih dari dua aturan, fungsi dapat diperluas dengan membagi w_i dengan jumlah total w untuk semua aturan.

Lapisan 4:

Tiap simpul pada lapisan ini berupa simpul adaptif dengan fungsi simpul (Jang, J.-S. R. 1997). Persamaan 2.14 merupakan persamaan untuk penghitungan matrik desain.

$$O_{4,i} = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i) \quad (2.14)$$

dimana w_i adalah derajat pengaktifan ternormalisasi dari lapisan 3 dan $\{p_i, q_i, r_i\}$ merupakan himpunan parameter dari simpul ini. Parameter di lapisan ini dinamakan parameter-parameter konsekuen.

Lapisan 5:

Simpul tunggal pada lapisan ini diberi label Σ , yang mana menghitung semua keluaran sebagai penjumlahan dari semua sinyal yang masuk (Jang, J.-S. R. 1997). Persamaan 2.15 merupakan persamaan untuk penghitungan output jaringan.

$$O_{5,i} = \sum \bar{w}_i f_i \quad (2.15)$$

2.6.1.1. Least Square Estimation

Ketika nilai parameter-parameter bagian *premise* telah ditentukan, maka total keluaran dapat dinyatakan sebagai kombinasi linier dari parameterparameter *consequent*. Persamaan 2.16 merupakan persamaan untuk penghitungan nilai f .

$$f = \frac{w_1}{w_1+w_2+w_3+w_4} f_1 + \frac{w_2}{w_1+w_2+w_3+w_4} f_2 + \frac{w_3}{w_1+w_2+w_3+w_4} f_3 + \frac{w_4}{w_1+w_2+w_3+w_4} f_4$$

$$= (\bar{w}_1 x_1) p_1 + (\bar{w}_1 x_2) q_1 + (\bar{w}_1 x_3) r_1 + (\bar{w}_1 x_4) s_1 + (\bar{w}_1) t_1 + (\bar{w}_2 x_1) p_2 + (\bar{w}_2 x_2) q_2 + (\bar{w}_2 x_3) r_2 + (\bar{w}_2 x_4) s_2 + (\bar{w}_2) t_2 + (\bar{w}_3 x_1) p_3 + (\bar{w}_3 x_2) q_3 + (\bar{w}_3 x_3) r_3 + (\bar{w}_3 x_4) s_3 + (\bar{w}_3) t_3 + (\bar{w}_4 x_1) p_4 + (\bar{w}_4 x_2) q_4 + (\bar{w}_4 x_3) r_4 + (\bar{w}_4 x_4) s_4 + (\bar{w}_4) t_4 \quad (2.16)$$

di mana linier pada parameter *consequent* $p_1, q_1, r_1, s_1, t_1, p_2, q_2, r_2, s_2, t_2, p_3, q_3, r_3, s_3, t_3$, dan p_4, q_4, r_4, s_4, t_4 . Bentuk sederhananya ada pada persamaan 2.17

$$f = x \theta \quad (2.17)$$

jika matrik X dapat dibalik maka hasilnya adalah persamaan 2.18

$$\theta = x^{-1} f \quad (2.18)$$

Dengan menggunakan metode invers dan dengan mengasumsikan jumlah baris dari pasangan x dan f adalah k , maka diperoleh persamaan untuk teta, yaitu pada persamaan 2.19 :

$$\theta_k = (x^T x)^{-1} x^T f \quad (2.19)$$

2.6.1.2 Model Propagasi Error

Setelah pembelajaran dengan LSE, selanjutnya jaringan adaptif tersebut dapat dilatih untuk mendapatkan parameter a dan c dengan model propagasi error *steepest descent* untuk memperbaiki parameter tersebut (a dan c) yang terdapat pada persamaan 2.4.

Misalkan kita memiliki jaringan adaptif seperti terlihat pada gambar 2.2 yang terdiri atas 5 lapisan, dan kita memiliki sebanyak $N(L)$ neuron pada lapisan ke- L , maka jumlah kuadrat error (SSE) pada lapisan ke- L data ke- p , $1 \leq p \leq N$ adalah sesuai dengan persamaan 2.20 :

$$E_p = \sum_{k=1}^{N(L)} (d_k - X_{L,k})^2 \quad (2.20)$$

2.6.1.2.1 Error pada Lapisan ke-5

Setelah memilih jaringan adaptif yang hanya memiliki 1 neuron pada lapisan output (neuron 13), maka propagasi error yang menuju pada lapisan ke-5 dapat diketahui dengan menggunakan persamaan 2.21:

$$\varepsilon_{13} = \frac{\partial E_p}{\partial x_{13}} = -2(d_{13} - x_{13}) = -2(y_p - y_p) \quad (2.21)$$

Dengan y_p adalah target output data pelatihan ke- p , dan adalah output jaringan pada data pelatihan ke- p .

2.6.1.2.2 Error pada Lapisan ke-4

Propagasi error yang menuju pada lapisan ke-4, yaitu neuron 11 dan neuron 12 dapat diketahui dengan menggunakan persamaan 2.22 :

$$\varepsilon_{11} = \left(\frac{\partial E_p}{\partial x_{13}} \right) \left(\frac{\partial f_{13}}{\partial x_{11}} \right) = \varepsilon_{13} \left(\frac{\partial f_{13}}{\partial x_{11}} \right) = \varepsilon_{13}(1) = \varepsilon_{13}$$

Karena $f_{13} = \overline{w}_1 f_1 + \overline{w}_2 f_2$, maka $\frac{\partial f_{13}}{\partial (\overline{w}_2 f_2)} = 1$ (2.22)

2.6.1.3.3 Error pada Lapisan ke-3

Propagasi error yang menuju pada lapisan ke-3 yaitu neuron 9 dan neuron 10 dapat diketahui dengan menggunakan persamaan 2.23 dan 2.24:

$$\varepsilon_9 = \left(\frac{\partial E_p}{\partial x_{13}} \right) \left(\frac{\partial f_{13}}{\partial x_{11}} \right) \left(\frac{\partial f_{11}}{\partial x_9} \right) = \varepsilon_{11} \left(\frac{\partial f_{11}}{\partial x_9} \right) = \varepsilon_{11} f_1 \quad (2.23)$$

dan

$$\varepsilon_{10} = \left(\frac{\partial E_p}{\partial x_{13}} \right) \left(\frac{\partial f_{13}}{\partial x_{12}} \right) \left(\frac{\partial f_{12}}{\partial x_{10}} \right) = \varepsilon_{12} \left(\frac{\partial f_{12}}{\partial x_{10}} \right) = \varepsilon_{12} f_2 \quad (2.24)$$

2.6.1.3.4 Error pada Lapisan ke-2

Propagasi error yang menuju pada lapisan ke-2, yaitu neuron 7 dan neuron 8 dapat diketahui dengan menggunakan persamaan 2.25 dan 2.26 :

$$\begin{aligned} \varepsilon_7 &= \left(\frac{\partial E_p}{\partial x_{13}} \right) \left(\frac{\partial f_{13}}{\partial x_{11}} \right) \left(\frac{\partial f_{11}}{\partial x_9} \right) \left(\frac{\partial f_9}{\partial x_7} \right) + \left(\frac{\partial E_p}{\partial x_{13}} \right) \left(\frac{\partial f_{13}}{\partial x_{12}} \right) \left(\frac{\partial f_{12}}{\partial x_{10}} \right) \left(\frac{\partial f_{10}}{\partial x_7} \right) \\ &= \varepsilon_9 \left(\frac{\partial f_9}{\partial x_7} \right) + \varepsilon_{10} \left(\frac{\partial f_{10}}{\partial x_7} \right) \\ &= \varepsilon_9 \left(\frac{w_2}{(w_1 + w_2)^2} \right) + \varepsilon_{10} \left(-\frac{w_2}{(w_1 + w_2)^2} \right) \end{aligned}$$

$$= \frac{w_2}{(w_1 + w_2)^2} (\varepsilon_9 - \varepsilon_{10}) \quad (2.25)$$

$$\begin{aligned} \varepsilon_8 &= \left(\frac{\partial E_p}{\partial x_{13}} \right) \left(\frac{\partial f_{13}}{\partial x_{12}} \right) \left(\frac{\partial f_{12}}{\partial x_{10}} \right) \left(\frac{\partial f_{10}}{\partial x_8} \right) + \left(\frac{\partial E_p}{\partial x_{13}} \right) \left(\frac{\partial f_{13}}{\partial x_{11}} \right) \left(\frac{\partial f_{11}}{\partial x_9} \right) \left(\frac{\partial f_9}{\partial x_8} \right) \\ &= \varepsilon_{10} \left(\frac{\partial f_{10}}{\partial x_8} \right) + \varepsilon_9 \left(\frac{\partial f_9}{\partial x_8} \right) \\ &= \varepsilon_{10} \left(\frac{w_1}{(w_1 + w_2)^2} \right) + \varepsilon_9 \left(-\frac{w_1}{(w_1 + w_2)^2} \right) \\ &= \frac{w_1}{(w_1 + w_2)^2} (\varepsilon_{10} - \varepsilon_9) \end{aligned} \quad (2.26)$$

2.6.1.3.5 Error pada Lapisan ke-1

Propagasi error yang menuju pada lapisan ke-1, yaitu neuron 3, 4, 5, dan 6 dapat diketahui dengan menggunakan persamaan 2.27, 2.28, 2.29, 2.30 :

$$\varepsilon_3 = \varepsilon_7 \left(\frac{\partial f_7}{\partial x_3} \right) = \varepsilon_7 \mu_{B1}(x_2) \quad (2.27)$$

$$\varepsilon_4 = \varepsilon_8 \left(\frac{\partial f_8}{\partial x_4} \right) = \varepsilon_8 \mu_{B2}(x_2) \quad (2.28)$$

$$\varepsilon_5 = \varepsilon_8 \left(\frac{\partial f_7}{\partial x_5} \right) = \varepsilon_7 \mu_{A1}(x_1) \quad (2.29)$$

$$\varepsilon_6 = \varepsilon_8 \left(\frac{\partial f_8}{\partial x_6} \right) = \varepsilon_8 \mu_{A1}(x_1) \quad (2.30)$$

Selanjutnya, error tersebut dapat digunakan untuk mencari informasi error terhadap parameter a dan c, sesuai persamaan 2.31 dan 2.32 :

$$\varepsilon_{aik} = \frac{2(x_i - c_{ik})^2}{a_{ik}^3 (1 + \frac{(x_i - c_{ik})^2}{a_{ik}})^2} \quad (2.31)$$

dan

$$\varepsilon_{cik} = \frac{2(x_i - c_{ik})}{a_{ik}^2 (1 + \frac{(x_i - c_{ik})^2}{a_{ik}})^2} \quad (2.32)$$

Dari persamaan 2.31 dan 2.32, perubahan nilai parameter a_{ik} dan c_{ik} (Δa_{ik} dan Δc_{ik}) dapat ditentukan dengan persamaan 2.33 dan 2.34 :

$$\begin{aligned} \Delta a_{ik} &= \eta \varepsilon_{aik} x_i, \text{ dan} \\ \Delta c_{ik} &= \eta \varepsilon_{cik} x_i \end{aligned} \quad (2.34)$$

dengan η adalah laju pembelajaran yang terletak pada interval $[0,1]$, x_i adalah data ke- i . Sehingga nilai a_{ik} dan c_{ik} yang baru dapat dihitung menggunakan persamaan 2.35 dan 2.36 :

$$a_{ik} = a_{ik} (\text{lama}) + \Delta a_{ik}, \text{ dan} \quad (2.35)$$

$$c_{ik} = c_{ik} (\text{lama}) + \Delta c_{ik} \quad (2.36)$$

2.7 Tingkat Akurasi Hasil Pengujian

Sebagai hasil akhir sebuah penelitian, sering dicari tingkat akurasi dari hasil pengujian yang merupakan persentase kebenaran dari hasil percobaan. Tingkat akurasi ini dapat dihitung dengan persamaan 2.37 (Nugraha, 2006):

$$\text{Tingkat akurasi} = \frac{\sum \text{angka benar}}{\sum \text{angka pengujian}} * 100\% \quad (2.37)$$

RMSE adalah rata-rata kuadrat dari perbedaan nilai estimasi dengan nilai observasi suatu variabel. Kelayakan model hasil identifikasi ANFIS diuji dengan menggunakan RMSE yang dinyatakan pada persamaan 2.38. Training ANFIS yang paling baik diperoleh jika RMSE (Root Mean Square Error) mempunyai nilai paling kecil (Salehfar *et al*, 2000).

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (Y'_t - Y_t)^2} \quad (2.38)$$

dimana Y'_t adalah output model ANFIS, Y_t adalah data hasil observasi dan N adalah banyaknya data yang dianalisis.

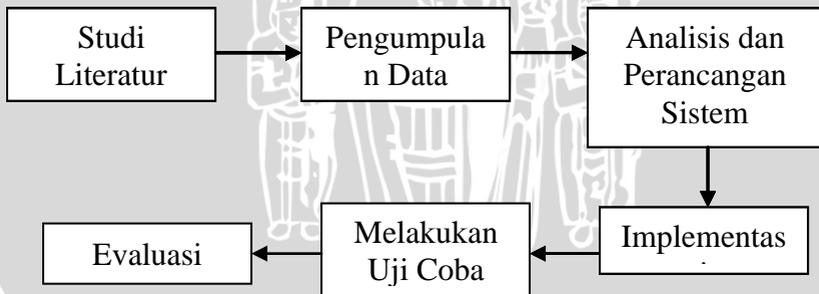
BAB III METODOLOGI DAN PERANCANGAN

Cakupan dari pembahasan bab tiga ini meliputi metode, perancangan dan Langkah-langkah yang dilakukan untuk penentuan daerah yang ralan terhadap demam berdarah menggunakan *Adaptive Neuro Fuzzy Inference System* (ANFIS).

Langkah-langkah yang dijadikan dalam penelitian ini meliputi:

1. Mempelajari literature yang terkait dengan masalah demam berdarah dengue (DBD), faktor-faktor yang mempengaruhi demam berdarah dengue (DBD) dan metode ANFIS.
2. Mengumpulkan data-data yang diperlukan dalam penelitian.
3. Menganalisa dan melakukan perancangan system.
4. Mengimplementasikan hasil perancangan dalam sebuah perangkat lunak untuk penentuan daerah yang rawan terhadap demam berdarah.
5. Melakukan uji coba terhadap perangkat lunak.
6. Mengevaluasi output penentuan daerah yang rawan terhadap demam berdarah yang dihasilkan pada saat uji coba perangkat lunak.

Langkah-langkah yang dilakukan dapat dijelaskan kembali dalam bentuk alur yang ditunjukkan dalam gambar 3.1.



Gambar 3.1 Langkah-langkah penelitian

3.1 Studi Literatur

Pada tahap studi pustaka dilakukan kajian secara teoritis mengenai metode-metode yang dapat mendukung untuk penyelesaian permasalahan dalam penelitian ini. Studi pustaka yang digunakan dapat diperoleh dari buku, jurnal, internet atau penelitian sebelumnya. Tinjauan Pustaka memberikan pengetahuan mendasar bagi peneliti untuk meneliti obyek penelitian, sehubungan dengan itu maka dilakukan studi pustaka mengenai:

1. Demam berdarah dengue (DBD) dan faktor-faktor yang mempengaruhi terjadinya DBD.
2. *Adaptive Neuro-Fuzzy Inference System (ANFIS)* yaitu gabungan antara metode logika fuzzy dengan Jaringan Saraf Tiruan (JST)

3.2 Sumber Data

Data yang digunakan dalam penelitian ini merupakan data iklim daerah Kabupaten Banyuwangi. Data ini memiliki beberapa atribut antara lain temperatur, curah hujan, lembab nisbi dan kecepatan angin. Data tersebut diperoleh dari Badan Meteorologi Klimatologi dan Geofisika Wilayah II, Stasiun Meteorologi Banyuwangi

3.3 Analisis dan Perancangan Sistem

3.3.1 Deskripsi Sistem

Secara umum sistem yang dibangun adalah perangkat lunak untuk penentuan daerah yang rawan terhadap demam berdarah. Sistem ini bertujuan untuk menentukan daerah yang rawan DBD dengan menggunakan inputan iklim yaitu suhu, curah hujan, kelembapan dan kecepatan angin yang nantinya menghasilkan suatu informasi bahwa daerah tersebut rawan atau tidak terhadap demam berdarah dengue (DBD).

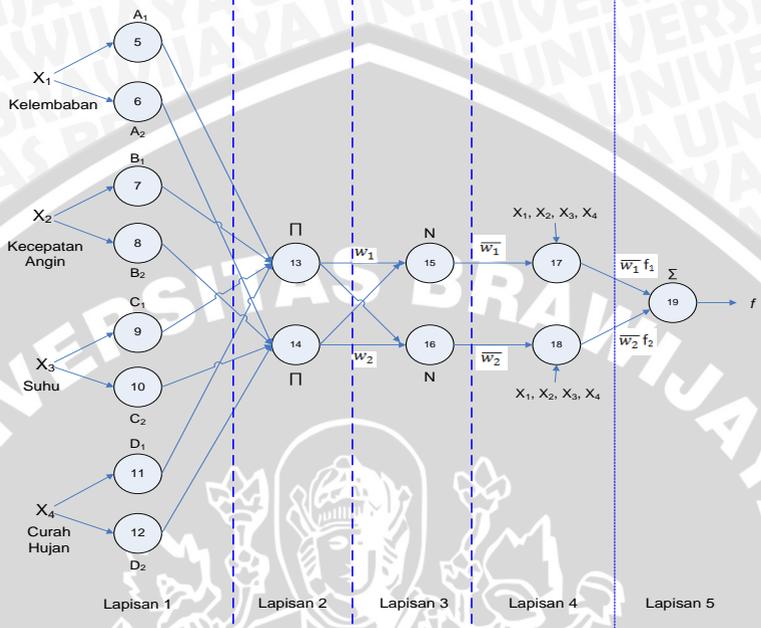
Sistem ini menggunakan salah satu metode yang ada dalam Neuro Fuzzy, yaitu *Adaptive Neuro Fuzzy Inference System (ANFIS)*. Metode ini merupakan gabungan antara penalaran logika fuzzy dan pembelajaran yang dimiliki oleh Jaringan Syaraf Tiruan (JST). Tingkat keberhasilan dari sistem ini diukur dengan menggunakan tingkat keakuratan dan error yang dihasilkan oleh

output sistem. Semakin kecil error yang dihasilkan maka akan semakin besar pula tingkat keberhasilan dari sistem.

Pada Proses penentuan kerawanan ini, sistem harus memperoleh data terlebih dahulu sebagai basis pengetahuan yang nantinya akan disimpan ke dalam Ms. Exel yang kemudian di import ke Ms. Access. Data numerik yang tersimpan dalam Ms. Access akan dipanggil di program untuk dilakukan normalisasi terlebih dahulu. Data yang telah dinormalisasi ini kemudian akan di bagi menjadi *cluster-cluster* sesuai dengan aturan output yang diberikan. Setelah diketahui *cluster* dan derajat keanggotannya maka akan digunakan untuk menghitung lebih lanjut menggunakan ANFIS. Data-data ini akan digunakan untuk menghitung error output yang dihasilkan dan menghitung nilai output.

3.3.2 Rancangan Sistem

Arsitektur jaringan sistem ANFIS ini memiliki 5 lapisan, dimana lapisan pertama adalah perhitungan derajat keanggotaan, lapisan kedua adalah perhitungan *firing strength*, lapisan ketiga adalah perhitungan normalisasi *firing strength*, lapisan keempat adalah perhitungan parameter konsekuensi dan lapisan kelima adalah perhitungan output jaringan. Arsitektur sistem ANFIS pada kasus ini dapat dilihat pada gambar 3.2.



Gambar 3.2 Arsitektur sistem Penentuan Kerawanan menggunakan ANFIS

Input yang digunakan ada empat, yaitu suhu, kelembaban, curah hujan dan kecepatan angin. Input ini akan masuk pada lapisan pertama kemudian akan di *cluster* menjadi dua sesuai aturan yang telah ditetapkan sebelumnya. Setelah diketahui *cluster* nya maka dapat dihitung derajat keanggotaannya. Keluaran dari lapisan pertama ini adalah derajat keanggotaan yang akan digunakan pada lapisan kedua. Pada lapisan kedua akan dihitung nilai dari *firing strength*. keluaran dari lapisan kedua ini adalah nilai *Firing strength* yang akan masuk ke lapisan ketiga. Pada lapisan ketiga *firing strength* akan dinormalisasi dan menghasilkan keluaran *firing strength* ternormalisasi yang akan masuk ke lapisan keempat. Pada lapisan keempat, suhu, kelembaban, curah hujan dan kecepatan angin digunakan untuk menghitung nilai dari f (rule yang telah ditentukan). Setelah diketahui nilai dari f maka nilai tersebut akan dikalikan

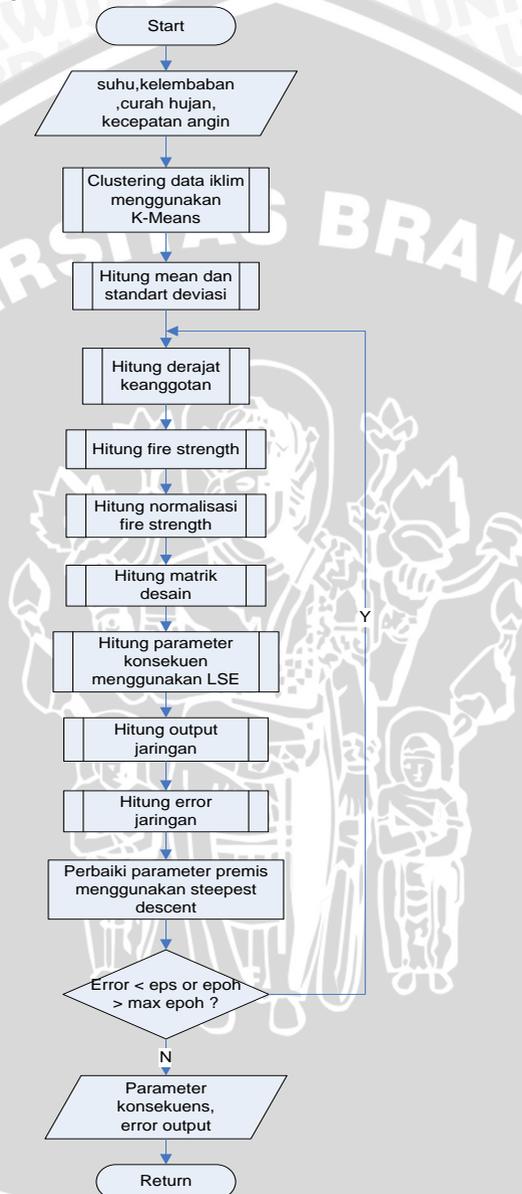
dengan *firing strength* ternormalisasi dan akan menghasilkan output jaringan berupa kerawanan pada lapisan kelima.

Proses penentuan kerawanan menggunakan ANFIS dilakukan dengan pelatihan terhadap data latih terlebih dahulu kemudian melakukan pengujian terhadap data uji. Proses pelatihan terhadap data latih akan menghasilkan nilai parameter premis dan parameter konsekuensi. Parameter konsekuensi ini akan digunakan untuk menghitung kerawanan pada proses pengujian data. Sehingga dapat diketahui nilai dari kerawanan.

Tahapan proses pelatihan yang dilakukan untuk penentuan kerawanan suatu daerah yaitu :

1. Menginputkan data iklim yaitu suhu, kelembaban, curah hujan dan kecepatan angin.
2. Mengelompokkan data inputan menjadi dua kelompok menggunakan clustering K-Means.
3. Input yang telah terkelompok, kemudian dihitung mean dan standart deviasinya.
4. Setelah mendapatkan nilai mean dan standart deviasi, kemudian hitung derajat keanggotaannya menggunakan fungsi bell.
5. Kemudian dihitung nilai *fire strength* dengan mengalikan tiap-tiap derajat keanggotaan yang didapatkan dari tahap sebelumnya.
6. *Fire strength* ini akan dinormalisasi untuk mendapatkan *fire strength* ternormalisasi.
7. Hasilnya akan dilewatkan di lapisan keempat dan dihitung nilai hasil perkalian antara *fire strength* ternormalisasi dan nilai-nilai inputan yang nantinya akan terbentuk matriks desain.
8. Setelah terbentuk matrik desain, maka matrik ini akan digunakan untuk mencari parameter konsekuensi menggunakan LSE.
9. Untuk menghitung output jaringan, langkah awal yang dilakukan adalah menggunakan metode sugeno orde 1 untuk mendapatkan nilai f . Setelah mendapatkan nilai f kemudian menghitung dengan persamaan 2.16.
10. Hasilnya akan digunakan untuk menghitung error jaringan. Jika nilai errornya masih besar dari error yang diharapkan, maka akan dilakukan perbaikan pada parameter konsekuensi dan premis dengan algoritma *Steepest Descent*, hingga error yang didapatkan kurang atau sama dengan error yang diharapkan.

Langkah-langkah diatas dapat dilihat pada flowchart yang terdapat dalam gambar 3.3 :

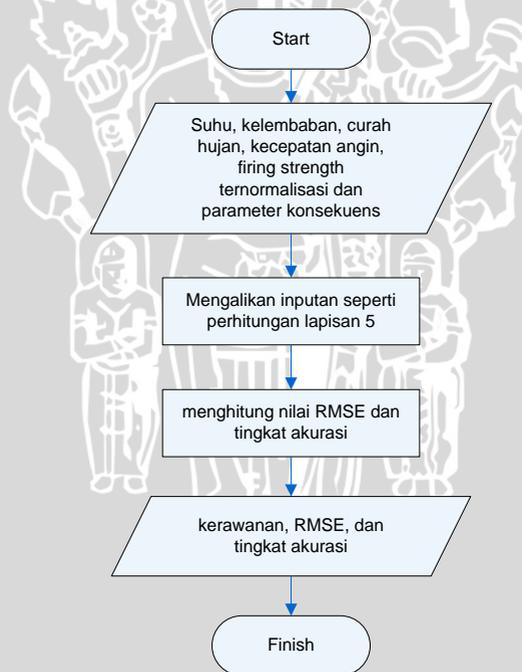


Gambar 3.3 Proses Pelatihan Kerawanon Menggunakan

Setelah dilakukan proses pelatihan maka untuk mendapatkan kerawanan akan dilakukan proses pengujian, dengan tahap sebagai berikut :

1. Menginputkan data iklim yaitu suhu, kelembaban, curah hujan dan kecepatan angin. Data ini merupakan data untuk pengujian.
2. Data yang diinputkan akan langsung dilewatkan pada lapisan keempat sesuai pada gambar 3.2, kemudian akan dihitung nilai perkalian antara data inputan, parameter konsekuen dan *firing strength* ternormalisasi. Nilai yang dihasilkan adalah output jaringan.
3. Setelah mendapatkan output jaringan, kemudian menghitung nilai RMSE dan nilai akurasi dari penentuan kerawanan.
4. Hasil akhir yang dicapai nantinya adalah kerawanan, nilai RMSE, dan tingkat akurasi.

Langkah-langkah diatas dapat dilihat pada flowchart yang terdapat dalam gambar 3.4 :



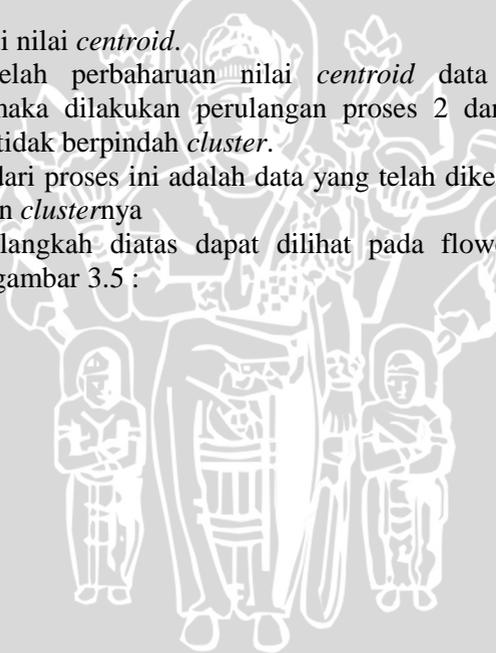
Gambar 3.4 Proses Pengujian Kerawanan Menggunakan

3.3.2.1 Clustering K-Means

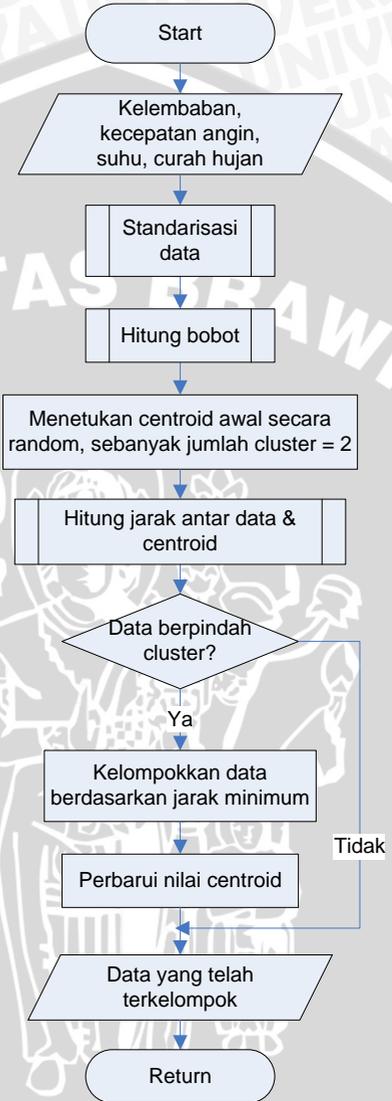
Untuk tahap pengelompokan data menggunakan k-means langkah-langkah yang perlu dilakukan yaitu :

1. Menginputkan dataset iklim
2. Setelah itu dataset iklim distandarisasi dengan menggunakan persamaan 2.5.
3. Menghitung bobot tiap data dengan menggunakan persamaan 2.6.
4. Setelah mendapatkan nilai bobot, tahap selanjutnya adalah menentukan *centroid* awal secara random.
5. Menghitung jarak antar data dan *centroid* sesuai persamaan 2.7
6. Mengelompokkan data yang telah dihitung jarak antar data dan *centroid* nya, berdasarkan jarak minimum atau nilai yang paling kecil.
7. Memperbarui nilai *centroid*.
8. Apabila setelah perbaharuan nilai *centroid* data berpindah clusternya maka dilakukan perulangan proses 2 dan proses 3 sampai data tidak berpindah *cluster*.
9. Hasil akhir dari proses ini adalah data yang telah dikelompokkan sesuai dengan *clusternya*

Langkah-langkah diatas dapat dilihat pada flowchart yang terdapat dalam gambar 3.5 :



Clustering data iklim menggunakan K-Means



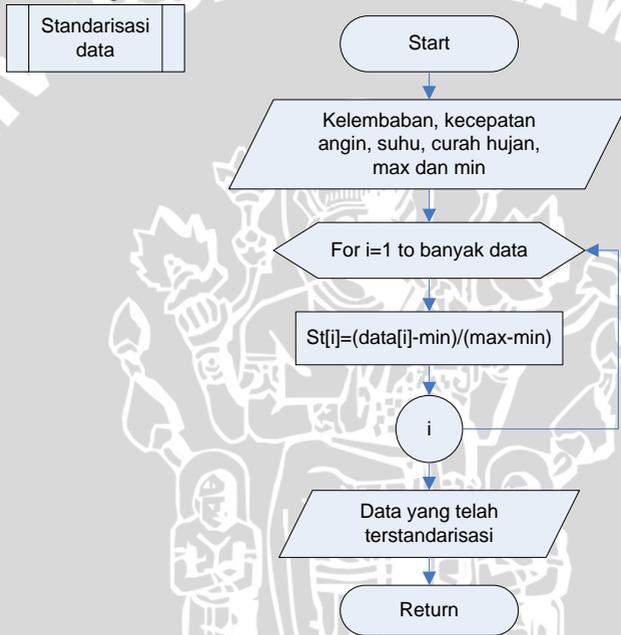
Gambar 3.5 Proses Clustering K-Means

3.3.2.1.1 Proses Perhitungan Standarisasi Data

Setelah mendapatkan input data kemudian akan dilakukan standarisasi dengan langkah sebagai berikut:

1. Menginputkan dataset iklim
2. Setelah itu dataset iklim distandarisi dengan menggunakan persamaan 2.5.
3. Hasil akhir dari proses ini adalah data yang telah terstandarisasi

Langkah-langkah diatas dapat dilihat pada flowchart yang terdapat dalam gambar 3.6 :



Gambar 3.6 Proses Perhitungan Standarisasi Data

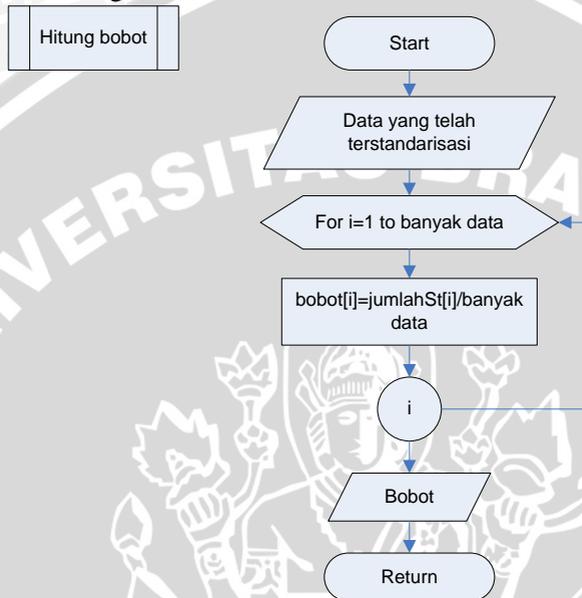
3.3.2.1.2 Proses Perhitungan Bobot

Setelah mendapatkan input data yang telah terstandarisasi kemudian akan dilakukan penghitungan bobot dengan langkah sebagai berikut:

1. Data yang telah terstandarisasi digunakan sebagai input untuk mendapatkan bobot.

2. Menghitung bobot tiap data dengan menggunakan persamaan 2.6.
3. Hasil akhir dari proses ini adalah bobot.

Langkah-langkah diatas dapat dilihat pada flowchart yang terdapat dalam gambar 3.7 :



Gambar 3.7 Proses Perhitungan Bobot

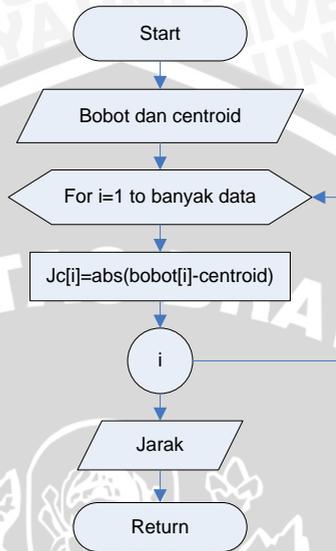
3.3.2.1.3 Proses Perhitungan Jarak Antar Data dan Centroid

Setelah mendapatkan bobot kemudian akan dilakukan penghitungan jarak antar data dan centroid dengan langkah sebagai berikut:

1. Nilai bobot yang telah didapatkan serta nilai *centroid* awal yang didapat secara random digunakan sebagai input pada proses penghitungan jarak.
2. Menghitung jarak antar data dan *centroid* sesuai persamaan 2.7
3. Hasil akhir dari proses ini adalah jarak antar data.

Langkah-langkah diatas dapat dilihat pada flowchart yang terdapat dalam gambar 3.8 :

Hitung jarak antar data & cluster



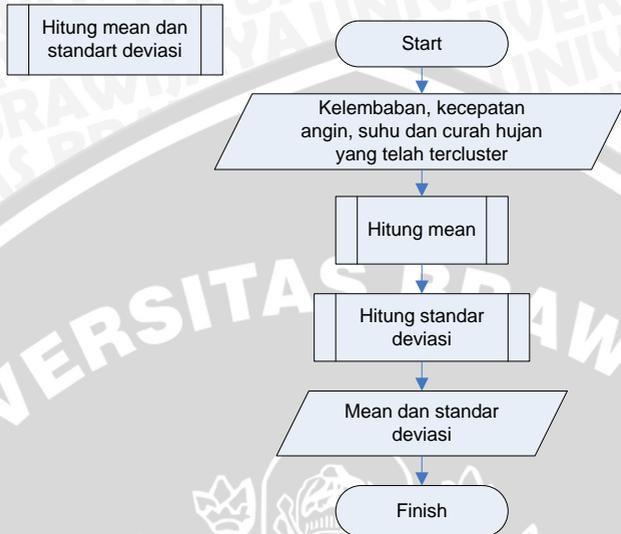
Gambar 3.8 Proses Perhitungan Jarak Antar Data dan

3.3.2.2 Proses Perhitungan Mean dan Standart Deviasi

Tahap awal setelah data tercluster yaitu menghitung mean dan standart deviasi, langkah-langkah yang dilakukan yaitu :

1. Input data iklim yang telah tercluster
2. Menghitung nilai mean sesuai persamaan 2.8 berdasarkan cluster yang terbentuk yaitu 2 cluster dan hingga seluruh data habis.
3. Setelah mendapatkan nilai mean, kemudian menghitung standart deviasi sesuai persamaan 2.9 berdasarkan cluster yang terbentuk yaitu 2 cluster dan hingga seluruh data habis.
4. Hasil akhirnya adalah nilai mean dan standart deviasi

Alur proses perhitungan mean dan standart deviasi dapat dilihat pada gambar 3.9



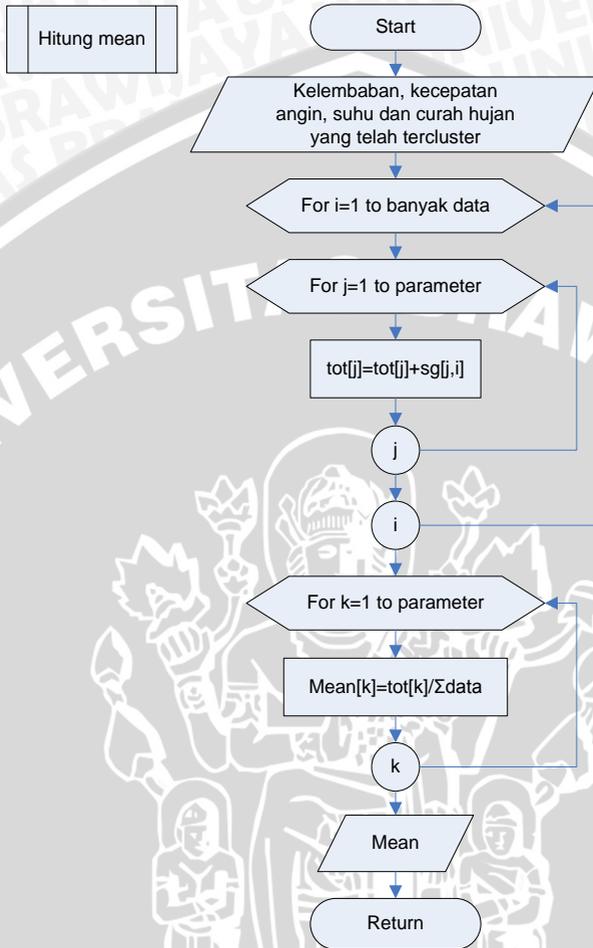
Gambar 3.9 Proses Perhitungan Mean Dan Standart

3.3.2.2.1 Proses Perhitungan Mean

Proses penghitungan mean dilakukan dengan langkah-langkah sebagai berikut :

1. Input data iklim yang telah tercluster
2. Menjumlahkan data tiap parameter pada cluster yang sama
3. Menghitung nilai mean sesuai persamaan 2.8 berdasarkan cluster yang terbentuk yaitu 2 cluster dan hingga seluruh data habis.
4. Hasil akhirnya adalah nilai mean

Alur proses perhitungan mean dan standart deviasi dapat dilihat pada gambar 3.10



Gambar 3.10 Proses Perhitungan Mean

3.3.2.2.2 Proses Perhitungan Standart Deviasi

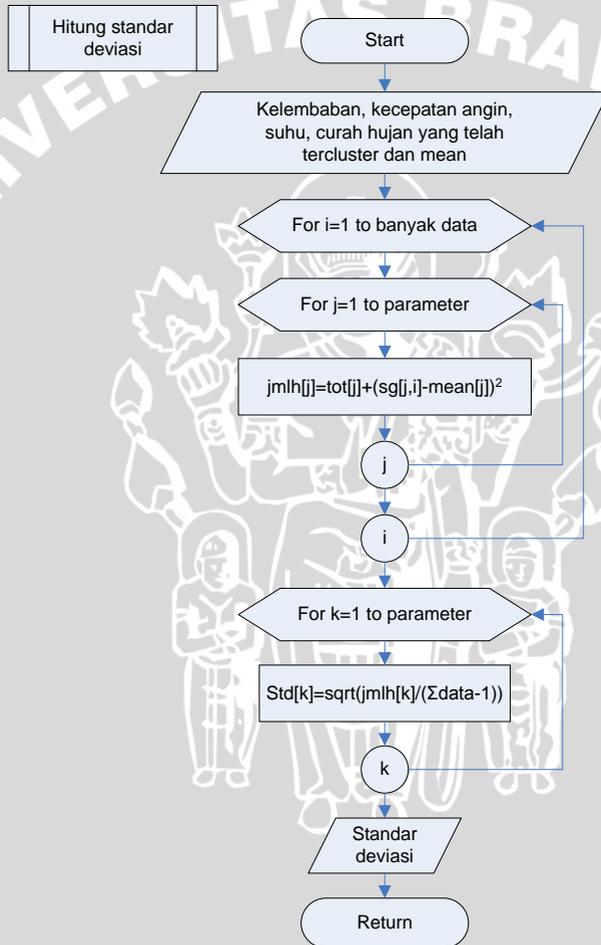
Proses penghitungan standar deviasi dilakukan dengan langkah-langkah sebagai berikut :

1. Input data iklim yang telah tercluster dan mean
2. Menjumlahkan data tiap parameter dikurang mean pada cluster yang sama

3. Menghitung nilai standart deviasi sesuai persamaan 2.9 berdasarkan cluster yang terbentuk yaitu 2 cluster dan hingga seluruh data habis.

4. Hasil akhirnya adalah nilai mean

Alur proses perhitungan mean dan standart deviasi dapat dilihat pada gambar 3.11



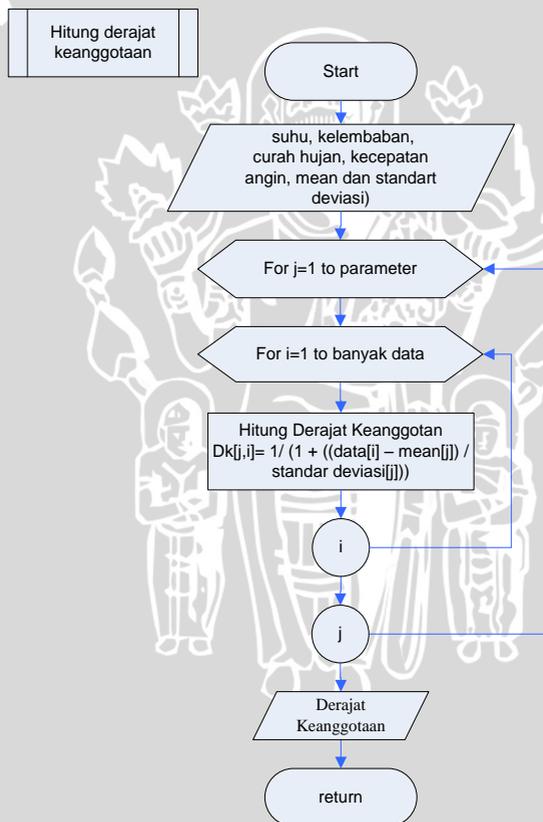
Gambar 3.11 Proses Perhitungan Standar Deviasi

3.3.2.3 Proses Perhitungan Derajat Keanggotaan

Setelah mendapatkan nilai mean dan standart deviasi ini selanjutnya akan menjadi c dan a awal yang nantinya akan digunakan untuk mendapatkan derajat keanggotaan. Langkah-langkah yang dilakukan yaitu :

1. Input nilai mean, standart deviasi dan nilai parameter input x_i
2. Menghitung derajat keanggotaan dengan persamaan 2.12 hingga iterasi yang ditentukan.
3. Hasil akhirnya adalah nilai derajat keanggotaan.

Alur proses perhitungan derajat keanggotaan dapat dilihat pada gambar 3.12



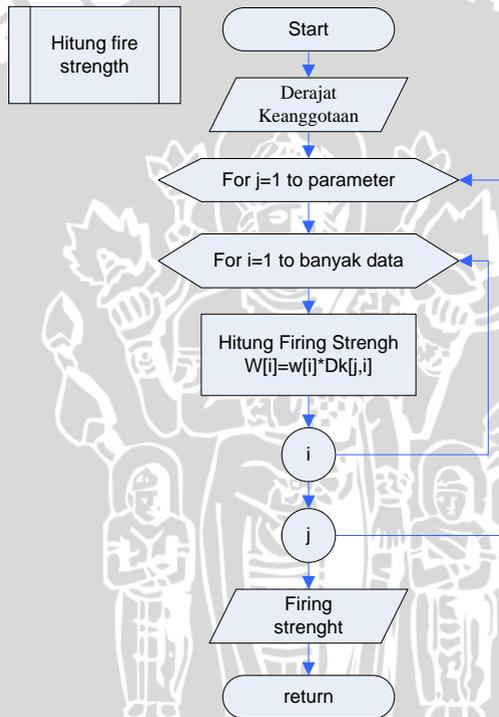
Gambar 3.12 Proses perhitungan derajat

3.3.2.4 Proses Perhitungan Fire Strength

Untuk tahap perhitungan *fire strength*, langkah – langkah yang perlu dilakukan yaitu:

1. Input nilai derajat keanggotaan
2. Menghitung *fire strength* dengan menggunakan persamaan 2.13 hingga iterasi yang telah ditentukan
3. Hasil akhirnya adalah nilai *fire strength*

Alur proses perhitungan *fire strength* dapat dilihat pada gambar 3.13



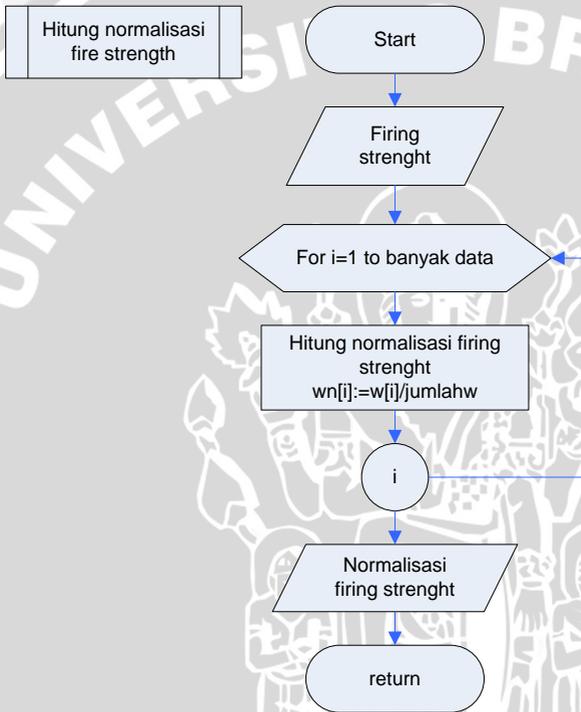
Gambar 3.13 Proses perhitungan *fire*

3.3.2.5 Proses Perhitungan Normalisasi Fire Strength

Untuk tahap perhitungan normalisasi *fire strength*, langkah – langkah yang perlu dilakukan yaitu:

1. Input nilai *fire strength*
2. Menghitung normalisasi *fire strength* dengan menggunakan persamaan 2.14 hingga iterasi yang telah ditentukan
3. Hasil akhirnya adalah nilai *fire strength* yang telah ternormalisasi

Alur proses perhitungan normalisasi *fire strength* dapat dilihat pada gambar 3.14



Gambar 3.14 Proses perhitungan normalisasi *fire*

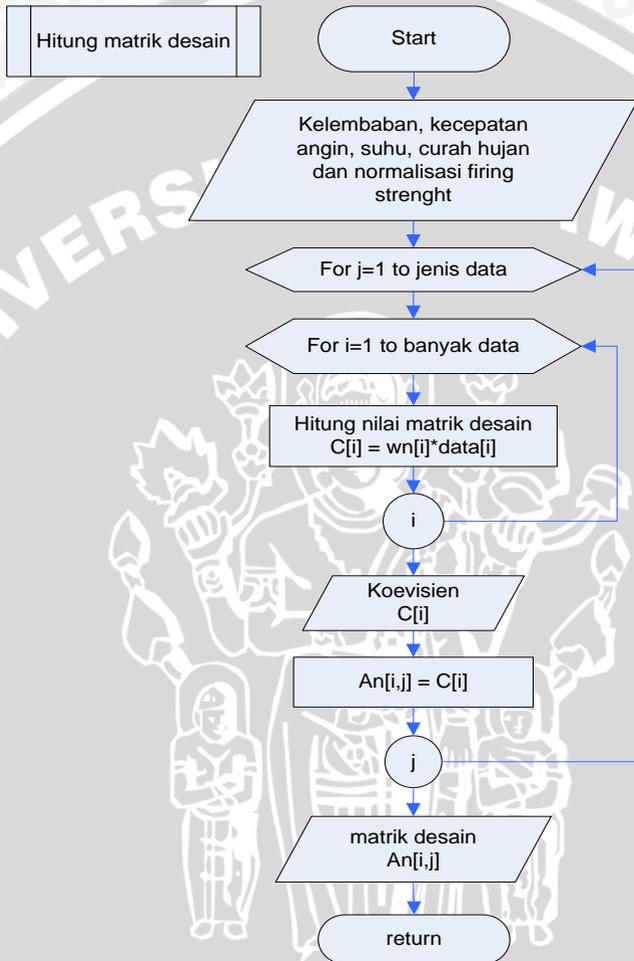
3.3.2.6 Proses Perhitungan Matrik Desain

langkah – langkah yang perlu dilakukan untuk menghitung matrik desain yaitu:

1. Input nilai normalisasi *fire strength* dan nilai parameter input x_i
2. Mengalikan *fire strength* ternormalisasi dengan parameter input x_i , sesuai dengan persamaan 2.15

3. Hasil akhir berupa matrik desain

Alur proses perhitungan matrik desain dapat dilihat pada gambar 3.15



Gambar 3.15 Proses perhitungan matrik desain

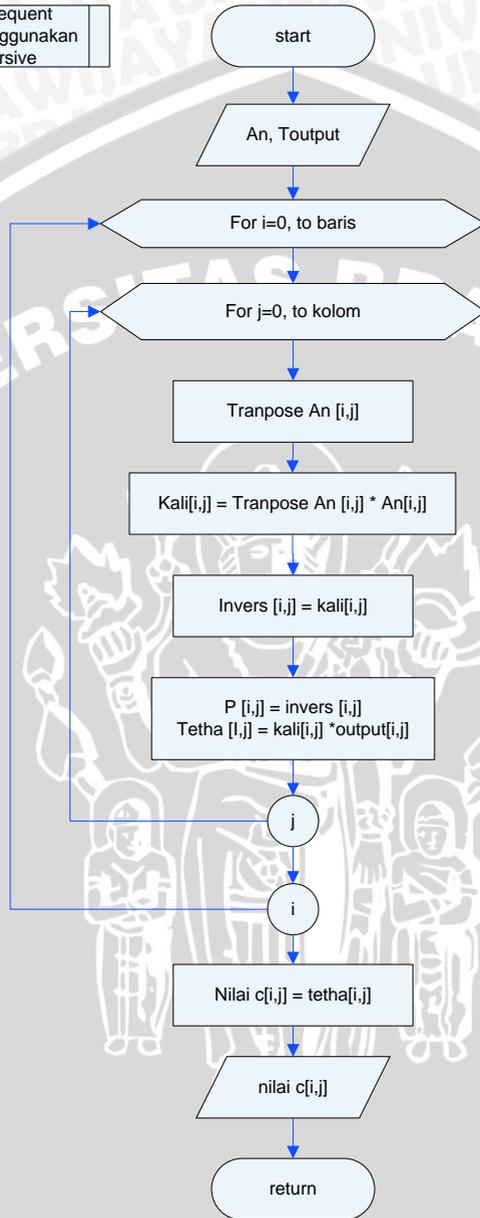
3.3.2.7 Parameter Konsekuens

Menghitung parameter konsekuens menggunakan LSE merupakan sub proses dari proses pelatihan. Proses ini terdapat pada lapisan keempat dari arsitektur jaringan dapat dilihat pada gambar 3.2. Tahapan untuk menghitung parameter konsekuens menggunakan LSE adalah sebagai berikut:

1. Inputan yang digunakan adalah matrik An_{ij} dan output jaringan.
2. Setelah mendapatkan matrik desain yaitu matrik An_{ij} ini digunakan untuk menghitung nilai konsekuens parameter dengan menggunakan LSE.
3. Langkah awal yang dilakukan adalah mentranspose matrik desain.
4. Mengalikan matrik An_{ij} dengan matrik An_{ij} yang telah di transpose. Hasil pengalihan tersebut akan di invers.
5. Hasil yang telah di invers ini disebut dengan P.
6. Mencari nilai tetha dengan mengalikan hasil invers dengan output jaringan.
7. Nilai tetha inilah yang disebut dengan nilai dari parameter-parameter yang adaptif atau c_{ij}
8. Hasil akhir dari perhitungan ini adalah nilai parameter c_{ij} atau disebut dengan parameter konsekuens.

Flowchart proses menghitung parameter konsekuens dapat digambarkan seperti pada Gambar 3.16.

Hitung consequent parameter menggunakan LSE Recursive



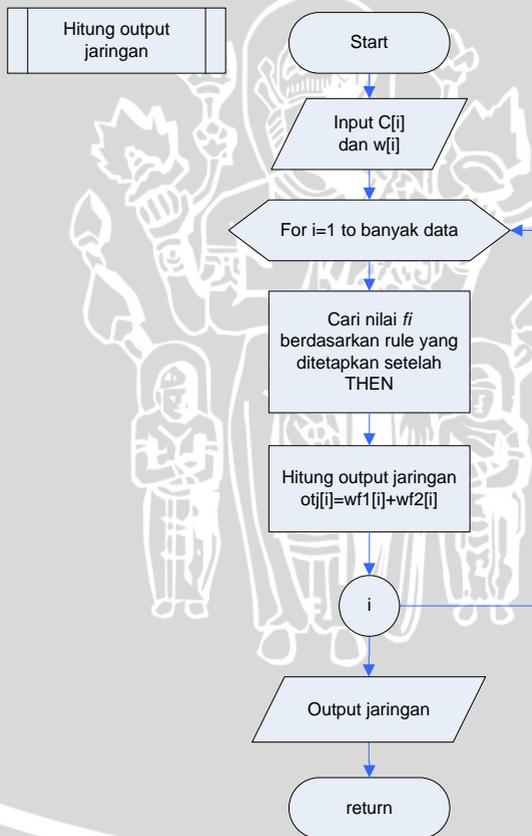
Gambar 3.16 Proses perhitungan parameter

3.3.2.8 Proses Perhitungan Output Jaringan

Setelah mendapatkan parameter konsekuen yang didapatkan dengan metode LSE, langkah selanjutnya adalah menghitung output jaringan dengan langkah-langkah berikut :

1. Input parameter konsekuen dan *fire strength*
2. Mencari nilai *f* dengan menggunakan persamaan setelah THEN berdasarkan rule yang telah terbentuk
3. Setelah mendapatkan nilai *f*, langkah selanjutnya adalah menghitung output jaringan sesuai dengan persamaan 2.16
4. Hasil akhir berupa output kerawanan yang dihasilkan oleh sistem

Alur proses perhitungan output jaringan seperti gambar 3.17



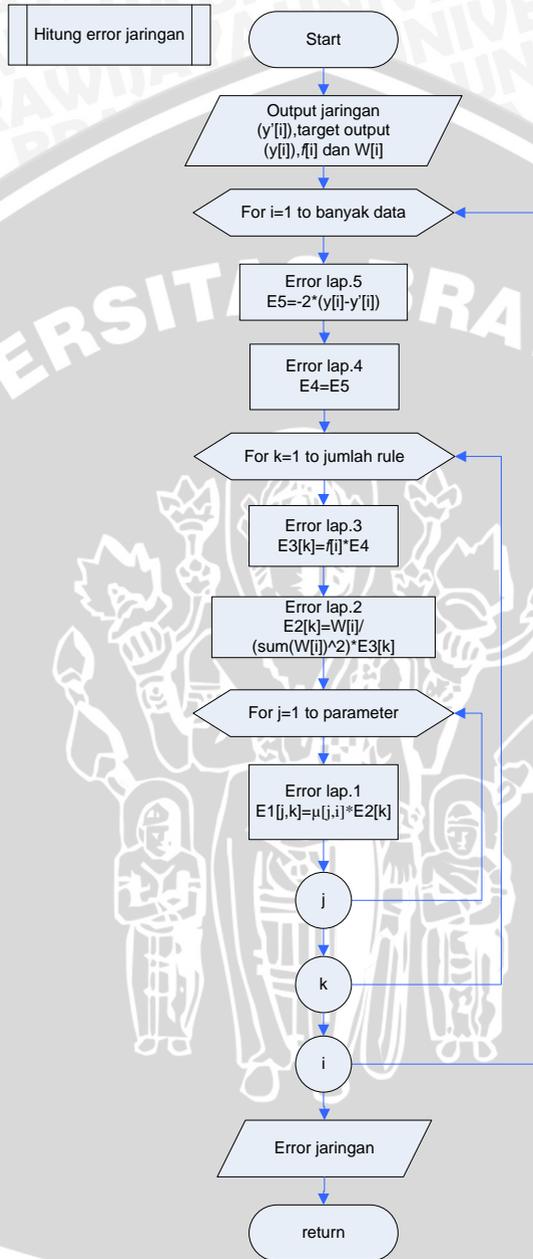
Gambar 3.17 Proses perhitungan output jaringan

3.3.2.9 Proses Perhitungan Error Jaringan

langkah – langkah yang perlu dilakukan untuk menghitung error jaringan yaitu:

1. Input nilai target output, output jaringan dan *fire strength*
2. Menghitung error pada lapisan lima dengan menggunakan persamaan 2.22
3. Setelah mendapatkan nilai error pada lapisan lima, kemudian menghitung nilai error lapisan empat, dimana nilai error pada lapisan empat sama dengan nilai error pada lapisan lima. Persamaan 2.23 adalah persamaan untuk menghitung nilai error lapisan empat
4. Langkah selanjutnya adalah menghitung error lapisan tiga sesuai dengan persamaan 2.24 dan 2.25
5. Selanjutnya adalah menghitung nilai error pada lapisan dua dengan menggunakan persamaan 2.26 dan 2.27
6. Tahap yang terakhir adalah menghitung nilai error pada lapisan satu dengan menggunakan persamaan 2.28 hingga 2.31
7. Perhitungan dilakukan hingga iterasi yang telah ditentukan
8. Hasil akhir merupakan nilai error pada tiap-tiap lapisan

Alur proses perhitungan error jaringan dapat dilihat pada gambar 3.18



Gambar 3.18 Proses perhitungan error jaringan

Data yang digunakan untuk pelatihan ini disimpan di Ms. Access yang nantinya akan panggil di program.

3.4 Contoh Perhitungan Manual

Pada subbab ini akan diberikan simulasi atau contoh perhitungan manual yang menggambarkan bagaimana sistem ANFIS akan bekerja.

Misalkan diberikan pasangan input-output seperti terlihat pada tabel 3.1 (empat variabel input x_1 , x_2 , x_3 , dan x_4 dan satu variabel output f dimana angka 1 menandakan rawan dan angka 2 menandakan tidak rawan):

Tabel 3.1 Contoh data-data pelatihan

No	Kelembaban	Kecepatan Angin	Suhu	Curah Hujan	Target Output
1	70.4	4.5	27.8	8.9	2
2	76.2	2.9	27.6	0.0	2
3	82.7	2.1	27.1	15.3	1
4	72.6	3.4	29.0	0.0	2
5	81.6	1.3	27.2	4.0	1
6	78.3	2.1	27.9	21.1	1
7	79.4	2.2	27.6	5.3	1
8	83.4	1.1	27.2	2.0	1
9	83.9	2.3	27.4	7.2	1
10	81.5	2.8	26.8	9.8	1
11	83.3	2.4	26.3	0.5	1
12	87.6	2.9	26.3	25.4	1

Dari data pada tabel 3.1, akan dibentuk dua basis aturan model sugeno orde satu yaitu:

IF x_1 is A_1 AND x_2 is B_1 AND x_3 is C_1 AND x_4 is D_1 THEN $f_1 = c_{11}x_1 + c_{12}x_2 + c_{13}x_3 + c_{14}x_4 + c_{10}$

IF x_1 is A_2 AND x_2 is B_2 AND x_3 is C_2 AND x_4 is D_2 THEN $f_2 = c_{21}x_1 + c_{22}x_2 + c_{23}x_3 + c_{24}x_4 + c_{20}$

3.4.1 Clustering K-Means

Dari data yang pada tabel 3.1 akan dilakukan pengclustering menggunakan k-means untuk mendapatkan nilai mean dan standart deviasinya. Langkah pertama yaitu melakukan standarisasi data menggunakan persamaan 2.5, kemudian mencari nilai bobot dengan menggunakan persamaan 2.6. Standarisasi data dan nilai bobot dapat dilihat pada tabel 3.2.

Tabel 3.2 Standarisasi data dan nilai bobot

No	Kelembaban	Kecepatan Angin	Suhu	Curah Hujan	Bobot
1	0.0000	1.0000	0.5542	0.3519	0.477
2	0.3398	0.5109	0.4908	0.0000	0.335
3	0.7178	0.2810	0.2845	0.6019	0.471
4	0.1319	0.6632	1.0000	0.0000	0.449
5	0.6524	0.0620	0.3468	0.1555	0.304
6	0.4580	0.2774	0.6030	0.8307	0.542
7	0.5234	0.3321	0.4692	0.2067	0.383
8	0.7596	0.0000	0.3310	0.0787	0.292
9	0.7851	0.3358	0.3990	0.2844	0.451
10	0.6470	0.4818	0.2040	0.3873	0.430
11	0.7487	0.3723	0.0000	0.0192	0.285
12	1.0000	0.5130	0.0246	1.0000	0.634

Dari hasil penghitungan bobot didapatkan nilai maksimal dan minimal bobot yaitu 0.634 dan 0.285. Langkah selanjutnya adalah menghitung jarak Euclidean dengan pusat *cluster* awal random antara nilai maksimal dan minimal nilai bobot. Proses ini dilakukan pada iterasi ke-1. Nilai pusat *cluster* random yaitu :

$$C1 = 0.361$$

$$C2 = 0.355$$

Tabel hasil perhitungan jarak Euclidean pada iterasi ke-1 dapat dilihat pada tabel 3.3

Tabel 3.3 Jarak Euclidean pada iterasi ke-1

No	Bobot	C1 (0.361)	C2 (0.355)
1	0.48	0.11543	0.12191
2	0.34	0.02571	0.01923
3	0.47	0.11020	0.11668
4	0.45	0.08767	0.09415
5	0.30	0.05690	0.05042
6	0.54	0.18115	0.18763
7	0.38	0.02175	0.02824
8	0.29	0.06877	0.06229
9	0.45	0.08996	0.09644
10	0.43	0.06890	0.07539
11	0.29	0.07606	0.06958
12	0.63	0.27330	0.27978

Setelah didapatkan jarak Euclidean, tahap selanjutnya adalah iterasi ke-2 yaitu mencari pusat *cluster* yang baru dengan cari mencari nilai rata-rata dari tiap anggota C1 dan C2.

$$C1 = (0.115+0.110+0.088+0.181+0.022+0.09+0.067+0.273)/8 \\ = 0.47965$$

$$C2 = (0.019+0.050+0.062+0.07)/4 \\ = 0.30424$$

Tabel hasil perhitungan jarak Euclidean pada iterasi ke-2 dapat dilihat pada tabel 3.4

Tabel 3.4 Jarak Euclidean pada iterasi ke-2

No	Bobot	C1 (0.480)	C2 (0.304)
1	0.48	0.00312	0.17229
2	0.34	0.14426	0.03115
3	0.47	0.00835	0.16706
4	0.45	0.03088	0.14453
5	0.30	0.17545	0.00004
6	0.54	0.06261	0.23801
7	0.38	0.09679	0.07862

8	0.29	0.18732	0.01191
9	0.45	0.02859	0.14682
10	0.43	0.04964	0.12577
11	0.29	0.19460	0.01920
12	0.63	0.15475	0.33016

Hasil perbandingan pada iterasi ke-1 dan iterasi ke-2 menunjukkan bahwa anggota masing-masing *cluster* berubah, sehingga iterasi dilakukan lagi. Pada iterasi ke-3 mencari pusat *cluster* sama dengan mencari pusat *cluster* pada iterasi ke-2.

$$C1 = (0.003+0.008+0.031+0.063+0.029+0.05+0.155)/7$$

$$= 0.49347$$

$$C2 = (0.031+0.00004+0.079+0.012+0.019)/5$$

$$= 0.31996$$

Tabel hasil perhitungan jarak Euclidean pada iterasi ke-3 dapat dilihat pada tabel 3.5

Tabel 3.5 Jarak Euclidean pada iterasi ke-3

no	bobot	C1 (0.493)	C2 (0.320)
1	0.477	0.017	0.157
2	0.335	0.158	0.015
3	0.471	0.022	0.151
4	0.449	0.045	0.129
5	0.304	0.189	0.016
6	0.542	0.049	0.222
7	0.383	0.111	0.063
8	0.292	0.201	0.028
9	0.451	0.042	0.131
10	0.430	0.063	0.110
11	0.285	0.208	0.035
12	0.634	0.141	0.314

Hasil perbandingan pada iterasi ke-2 dan iterasi ke-3 menunjukkan bahwa anggota masing-masing *cluster* tidak berubah, sehingga iterasi dihentikan. Dengan demikian perhitungan k-means

clustering telah mencapai stabilitasnya dan tidak perlu dilakukan iterasi lagi. Hasil dari *clustering*nya dapat dilihat pada tabel 3.6.

Tabel 3.6 Hasil clustering

No	Kelembaban	Kecepatan Angin	Suhu	Curah Hujan	Cluster
1	70.4	4.5	27.8	8.9	1
2	76.2	2.9	27.6	0.0	2
3	82.7	2.1	27.1	15.3	1
4	72.6	3.4	29.0	0.0	1
5	81.6	1.3	27.2	4.0	2
6	78.3	2.1	27.9	21.1	1
7	79.4	2.2	27.6	5.3	2
8	83.4	1.1	27.2	2.0	2
9	83.9	2.3	27.4	7.2	1
10	81.5	2.8	26.8	9.8	1
11	83.3	2.4	26.3	0.5	2
12	87.6	2.9	26.3	25.4	1

3.4.2 Perhitungan Mean dan Standar Deviasi

Dari data yang ada pada tabel 3.6, maka akan dicari nilai a dan c, dimana nilai tersebut digunakan untuk mencari derajat keanggotaan. Nilai c didapat dari mean dari data sesuai dengan kelasnya, sedangkan nilai a didapat dari standart deviasi dari data, juga sesuai dengan kelasnya. Nilai a dan c yang didapat dari perhitungan ini yaitu :

$$c = \begin{bmatrix} 79.5618 & 2.8380 & 27.4839 & 12.5411 \\ 80.7750 & 1.9750 & 27.1781 & 2.3375 \end{bmatrix}$$

$$a = \begin{bmatrix} 6.1976 & 0.8857 & 0.8772 & 8.6793 \\ 3.0251 & 0.7415 & 0.5417 & 2.2407 \end{bmatrix}$$

3.4.3 Perhitungan Lapisan 1 (Derajat Keanggotaan)

Setelah didapat nilai a dan c, langkah selanjutnya adalah menghitung derajat keanggotaan. Dengan demikian, persamaan 2.10 pada tiap neuron pada lapisan pertama akan menjadi :

$$A_1 = \mu(x_1) = \frac{1}{1 + \left| \frac{x_1 + 76,5618}{6,1976} \right|^2}$$

$$A_2 = \mu(x_1) = \frac{1}{1 + \left| \frac{x_1 + 80,7750}{3,0251} \right|^2}$$

$$B_1 = \mu(x_2) = \frac{1}{1 + \left| \frac{x_2 + 2,8380}{0,8857} \right|^2}$$

$$B_2 = \mu(x_2) = \frac{1}{1 + \left| \frac{x_2 + 1,9750}{0,7415} \right|^2}$$

$$C_1 = \mu(x_3) = \frac{1}{1 + \left| \frac{x_3 + 27,4839}{0,8772} \right|^2}$$

$$C_2 = \mu(x_3) = \frac{1}{1 + \left| \frac{x_3 + 27,1781}{0,5417} \right|^2}$$

$$D_1 = \mu(x_4) = \frac{1}{1 + \left| \frac{x_4 + 12,5411}{8,6793} \right|^2}$$

$$D_2 = \mu(x_4) = \frac{1}{1 + \left| \frac{x_4 + 2,3375}{2,2407} \right|^2}$$

Selanjutnya, output lapisan pertama, berupa derajat keanggotaan setiap data, hasilnya dapat dilihat pada tabel 3.7

Tabel 3.7 Output lapisan pertama (derajat keanggotaan)

Data ke-	Derajat Keanggotaan							
	$\mu A1$	$\mu A2$	$\mu B1$	$\mu B2$	$\mu C1$	$\mu C2$	$\mu D1$	$\mu D2$
1	0.3128	0.0780	0.2161	0.0780	0.8831	0.4289	0.8530	0.1033
2	0.7746	0.3059	0.9998	0.4179	0.9737	0.5917	0.3239	0.4789
3	0.7940	0.7078	0.5660	0.9863	0.8102	0.9541	0.9090	0.0291
4	0.4452	0.1216	0.7338	0.2199	0.2430	0.0786	0.3239	0.4789
5	0.9029	0.9317	0.2521	0.5561	0.9234	0.9905	0.5051	0.6588
6	0.9571	0.5894	0.5582	0.9899	0.7890	0.3372	0.5070	0.0141
7	0.9991	0.8236	0.6851	0.8886	0.9907	0.6579	0.5863	0.3718
8	0.7189	0.5635	0.2062	0.4179	0.8975	0.9997	0.4040	0.9778
9	0.6737	0.4878	0.6941	0.8791	0.9848	0.8833	0.7272	0.1737
10	0.9109	0.9457	0.9902	0.4779	0.6481	0.7166	0.9116	0.0819
11	0.7385	0.5990	0.7854	0.7746	0.3449	0.2646	0.3415	0.5946
12	0.3745	0.1654	0.9995	0.4141	0.3714	0.2960	0.3130	0.0094

3.4.4 Perhitungan Lapisan 2 dan 3 (*Firing Strenght* dan *Firing Strenght* ternormalisasi)

Setelah didapatkan nilai derajat keanggotaan, tahap selanjutnya pada lapisan kedua adalah mencari *firing strength* w_i dengan menggunakan persamaan 2.11 dan pada lapisan ketiga menormalisasi *firing strength* \hat{w}_i dengan menggunakan persamaan 2.12. Hasil perhitungan lapisan dua dan tiga ini dapat dilihat pada tabel 3.8

Tabel 3.8 Output lapisan kedua dan ketiga

Data ke-	Output lapisan 2		Output lapisan 3	
	$w1$	$w2$	$\hat{w}1$	$\hat{w}2$
1	0.0509	0.0003	0.9947	0.0053
2	0.2442	0.0362	0.8708	0.1292
3	0.3310	0.0194	0.9447	0.0553
4	0.0257	0.0010	0.9623	0.0377
5	0.1062	0.3381	0.2390	0.7610

6	0.2137	0.0028	0.9872	0.0128
7	0.3976	0.1790	0.6895	0.3105
8	0.0537	0.2302	0.1893	0.8107
9	0.3349	0.0658	0.8358	0.1642
10	0.5328	0.0265	0.9526	0.0474
11	0.0683	0.0730	0.4834	0.5166
12	0.0435	0.0002	0.9957	0.0043

3.4.5 Perhitungan Lapisan 4 (Parameter Konsekuensi) Menggunakan LSE

Pada lapisan keempat dihitung nilai parameter konsekuensi menggunakan LSE. Langkah yang dilakukan adalah mencari nilai matrik desain terlebih dahulu. Matrik desain yang akan digunakan sebagai *input* pada perhitungan LSE Rekursif, nilainya dapat dicari dengan menghitung nilai $\bar{w}_i x_{ij}$ dengan $i = 1,2$ dan $j = 0,1,2$. Hasilnya dapat dilihat pada tabel 3.9.



Tabel 3.9 Matrik desain

Data ke-	Koefisien dari									
	C11	C12	C13	C14	C10	C21	C22	C23	C24	C20
1	70.0043	4.5012	27.6567	8.8904	0.9947	0.3707	0.0238	0.1465	0.0471	0.0053
2	66.3724	2.4818	24.0590	0.0000	0.8708	9.8463	0.3682	3.5691	0.0000	0.1292
3	78.1476	1.9485	25.5640	14.4427	0.9447	4.5711	0.1140	1.4953	0.8448	0.0553
4	69.9052	3.2444	27.9380	0.0000	0.9623	2.7377	0.1271	1.0941	0.0000	0.0377
5	19.4999	0.3137	6.5079	0.9440	0.2390	62.0938	0.9988	20.7233	3.0060	0.7610
6	77.2500	2.0238	27.5805	20.8303	0.9872	1.0000	0.0262	0.3570	0.2697	0.0128
7	54.7305	1.5428	19.0091	3.6200	0.6895	24.6445	0.6947	8.5596	1.6300	0.3105
8	15.7914	0.2082	5.1455	0.3785	0.1893	67.6461	0.8918	22.0420	1.6215	0.8107
9	70.1042	1.8806	22.8805	6.0388	0.8358	13.7708	0.3694	4.4945	1.1862	0.1642
10	77.6334	2.6195	25.5642	9.3708	0.9526	3.8666	0.1305	1.2733	0.4667	0.0474
11	40.2425	1.1481	12.7012	0.2357	0.4834	43.0075	1.2269	13.5738	0.2518	0.5166
12	87.1912	2.8446	26.2286	25.2898	0.9957	0.3798	0.0124	0.1142	0.1102	0.0043

Setelah didapatkan matrik desain seperti yang ada pada tabel 3.9, maka sebagai proses pembelajaran, dilakukan proses *Least Square Estimation* (LSE). Nilai parameter konsekuen diperoleh dengan menghitung sesuai persamaan 2.18 :

$$X = \begin{pmatrix} 70.004 & 4.501 & 27.657 & 8.890 & 0.995 & 0.371 & 0.024 & 0.146 & 0.047 & 0.005 \\ 66.372 & 2.482 & 24.059 & 0.000 & 0.871 & 9.846 & 0.368 & 3.569 & 0.000 & 0.129 \\ 78.148 & 1.949 & 25.564 & 14.443 & 0.945 & 4.571 & 0.114 & 1.495 & 0.845 & 0.055 \\ 69.905 & 3.244 & 27.938 & 0.000 & 0.962 & 2.738 & 0.127 & 1.094 & 0.000 & 0.038 \\ 19.500 & 0.314 & 6.508 & 0.944 & 0.239 & 62.094 & 0.999 & 20.723 & 3.006 & 0.761 \\ 77.250 & 2.024 & 27.580 & 20.830 & 0.987 & 1.000 & 0.026 & 0.357 & 0.270 & 0.013 \\ 54.730 & 1.543 & 19.009 & 3.620 & 0.690 & 24.645 & 0.695 & 8.560 & 1.630 & 0.310 \\ 15.791 & 0.208 & 5.146 & 0.379 & 0.189 & 67.646 & 0.892 & 22.042 & 1.621 & 0.811 \\ 70.104 & 1.881 & 22.881 & 6.039 & 0.836 & 13.771 & 0.369 & 4.494 & 1.186 & 0.164 \\ 77.633 & 2.620 & 25.564 & 9.371 & 0.953 & 3.867 & 0.130 & 1.273 & 0.467 & 0.047 \\ 40.243 & 1.148 & 12.701 & 0.236 & 0.483 & 43.007 & 1.227 & 13.574 & 0.252 & 0.517 \\ 87.191 & 2.845 & 26.229 & 25.290 & 0.996 & 0.380 & 0.012 & 0.114 & 0.110 & 0.004 \end{pmatrix}$$

$$(X^T X)^{-1} = \begin{pmatrix} 0.041 & 0.120 & 0.181 & -0.003 & -8.492 & -0.096 & -0.259 & -0.431 & -0.003 & 19.981 \\ 0.120 & 0.726 & 0.565 & 0.012 & -27.285 & -0.063 & 0.513 & -0.123 & 0.092 & 8.010 \\ 0.181 & 0.565 & 1.242 & 0.014 & -50.226 & -0.310 & -0.379 & -1.984 & 0.068 & 80.076 \\ -0.003 & 0.012 & 0.014 & 0.007 & -0.248 & 0.027 & 0.244 & 0.246 & -0.011 & -9.180 \\ -8.492 & -27.285 & -50.226 & -0.248 & 2133.358 & 15.732 & 24.881 & 84.396 & -1.830 & -3629.665 \\ -0.096 & -0.063 & -0.310 & 0.027 & 15.732 & 1.466 & 4.067 & 5.354 & 0.146 & -271.958 \\ -0.259 & 0.513 & -0.379 & 0.244 & 24.881 & 4.067 & 21.346 & 24.709 & -0.318 & -1032.308 \\ -0.431 & -0.123 & -1.984 & 0.246 & 84.396 & 5.354 & 24.709 & 39.767 & -1.831 & -1547.580 \\ -0.003 & 0.092 & 0.068 & -0.011 & -1.830 & 0.146 & -0.318 & -1.831 & 0.606 & 36.405 \\ 19.981 & 8.010 & 80.076 & -9.180 & -3629.665 & -271.958 & -1032.308 & -1547.580 & 36.405 & 65668.891 \end{pmatrix}$$

Hasil parameter konsekuen yaitu:

$$\theta_k = (X^T X)^{-1} X^T f =$$

C11	C12	C13	C14	C10	C21	C22	C23	C24	C20
0.0019	0.2426	0.1087	-0.0245	-2.0804	-0.3662	-0.9516	-0.2796	-0.2970	40.7764

3.4.6 Perhitungan Lapisan 5 (Output Jaringan)

Setelah mendapatkan nilai parameter konsekuen, selanjutnya menghitung output jaringan. Langkah awal yang dilakukan adalah mencari nilai f yang didapatkan dari aturan sugeno yang telah ditetapkan. Aturan sugeno yang telah ditetapkan adalah :

IF x_1 is A_1 AND x_2 is B_1 AND x_3 is C_1 AND x_4 is D_1 THEN $f_1 = c_{11}x_1 + c_{12}x_2 + c_{13}x_3 + c_{14}x_4 + c_{10}$

IF x_1 is A_2 AND x_2 is B_2 AND x_3 is C_2 AND x_4 is D_2 THEN $f_2 = c_{21}x_1 + c_{22}x_2 + c_{23}x_3 + c_{24}x_4 + c_{20}$

Hasil dari perhitungan nilai f dapat dilihat pada tabel 3.10

Tabel 3.10 Nilai f

Data ke	f_1	f_2
1	1.957118	0.270539
2	1.761806	2.427824
3	1.147224	-3.584608
4	2.034018	2.848666
5	1.259319	0.860332
6	1.088694	-3.907989
7	1.484319	0.312161
8	1.254313	0.978725
9	1.426649	-1.880133
10	1.420982	-2.111839
11	1.501144	0.538368
12	1.023927	-8.920796

Setelah mendapatkan nilai f , kemudian menghitung output jaringan menggunakan persamaan 2.16. Hasil dari output jaringan terdapat pada tabel 3.11.

Tabel 3.11 Output jaringan

Data ke-	Output Jaringan
1	1.948233694
2	1.847845161
3	0.885737108
4	2.064719374
5	0.955685155
6	1.024838218
7	1.12038493
8	1.030883208
9	0.883734404
10	1.253374473
11	1.003767579
12	0.980796698

3.4.7 Propagasi Error Menggunakan Metode *Steepest Descent*

Jaringan adaptif yang ada, selanjutnya dilatih untuk mendapatkan nilai parameter a dan c , pada persamaan 2.4. Untuk melakukan perbaikan terhadap nilai a dan c , digunakan model propagasi error dengan metode *steepest descent*. Pada epoch atau iterasi pertama didapatkan :

- Propagasi error pada lapisan kelima berdasarkan persamaan 2.22, $\epsilon_{19} = 1.1931 \times 10^{-11}$
- Propagasi error pada lapisan keempat berdasarkan persamaan 2.23, $\epsilon_{17} = 1.1931 \times 10^{-11}$ dan $\epsilon_{18} = 1.1931 \times 10^{-11}$
- Propagasi error pada lapisan ketiga berdasarkan persamaan 2.24 dan 2.25, $\epsilon_{15} = 1.7160 \times 10^{-11}$ dan $\epsilon_{16} = -1.2099 \times 10^{-11}$
- Propagasi error pada lapisan kedua berdasarkan persamaan 2.26 dan 2.27, $\epsilon_{13} = 3.0077 \times 10^{-11}$ dan $\epsilon_{14} = -7.4308 \times 10^{-11}$
- Propagasi error pada lapisan pertama berdasarkan persamaan 2.28 hingga 2.31, $\epsilon_5 = 1.6806 \times 10^{-11}$; $\epsilon_6 = -2.4598 \times 10^{-11}$; $\epsilon_7 = 2.2206 \times 10^{-11}$; $\epsilon_8 = -4.4580 \times 10^{-11}$; $\epsilon_9 = 1.9266 \times 10^{-11}$; $\epsilon_{10} = -4.3967 \times 10^{-11}$; $\epsilon_{11} = 2.1561 \times 10^{-11}$; dan $\epsilon_{12} = -3.9132 \times 10^{-11}$
- Perubahan parameter a dan c (Δa dan Δc) berdasarkan persamaan 2.34 dan 2.35 yaitu :

$$\Delta a_{11} = 1.3684 \times 10^{-11}; \Delta a_{12} = -3.0749 \times 10^{-10}; \Delta a_{21} = 7.2467 \times 10^{-11}; \Delta a_{22} = -3.0895 \times 10^{-10}; \Delta a_{31} = 1.1673 \times 10^{-10}; \Delta a_{32} = -1.8816 \times 10^{-9}; \Delta a_{41} = 3.0852 \times 10^{-11}; \Delta a_{42} = -1.5110 \times 10^{-10}; \Delta c_{11} = 1.6777 \times 10^{-10}; \Delta c_{12} = 1.3144 \times 10^{-9}; \Delta c_{21} = -1.7850 \times 10^{-10}; \Delta c_{22} = -4.5504 \times 10^{-10}; \Delta c_{31} = -8.0360 \times 10^{-10}; \Delta c_{32} = -5.7136 \times 10^{-9}; \Delta c_{41} = -6.2984 \times 10^{-11}; \Delta c_{42} = -5.6882 \times 10^{-11}$$

g. Nilai parameter a dan c yang baru berdasarkan persamaan 2.36 dan 2.37.

$$c = \begin{bmatrix} 79.561801 & 2.837990 & 27.483929 & 12.541071 \\ 80.775000 & 1.975000 & 27.178125 & 2.337500 \end{bmatrix}$$

$$a = \begin{bmatrix} 6.197631 & 0.885693 & 0.877174 & 8.679327 \\ 3.025074 & 0.741462 & 0.541672 & 2.240710 \end{bmatrix}$$

Tabel 3.12 menunjukkan target output, output jaringan dan error yang terjadi antara output jaringan dengan target output.

Tabel 3.12 Error pada epoch pertama

Data ke-	Target Output	Output Jaringan	Error
1	1	1.0677	-0.0677
2	1	1.2308	-0.2308
3	1	1.2845	-0.2845
4	1	0.8622	0.1378
5	1	0.7828	0.2172
6	2	1.6762	0.3238
7	1	1.1089	-0.1089
8	1	0.9717	0.0283
9	1	1.0255	-0.0255
10	1	1.0168	-0.0168
11	1	0.9907	0.0093
12	1	0.9820	0.0180

3.5 Perancangan Pengujian dan Analisis

Pengujian dan analisis yang dilakukan dibagi menjadi tiga yaitu, pengujian laju pembelajaran ANFIS, pengujian kerawanan aktual dan output jaringan dan pengujian tingkat akurasi kerawanan. Hasil ANFIS

3.5.1 Pengujian Laju Pembelajaran ANFIS

Pengujian ini dilakukan untuk mengetahui nilai laju pembelajaran yang paling baik pada kasus penentuan kerawanan suatu daerah. Data yang digunakan untuk pelatihan berbeda-beda. Pada uji coba ini akan membandingkan laju pembelajaran dan nilai RMSE nya. Laju pembelajaran yang diujikan sebanyak 9 percobaan dengan nilai laju pembelajaran antara 0 sampai 1. Sebanyak 9 percobaan itu akan dicari nilai laju pembelajaran yang memiliki nilai RMSE paling kecil.

Perkembangan laju pembelajaran pada tiap pelatihan dicatat untuk mengetahui pengaruh kinerja jaringan ANFIS. Untuk mencatat hasil dari laju pembelajaran dibuat tabel 3.13

Tabel 3.13 Contoh tabel laju pembelajaran

Laju Pembelajaran	RMSE Percobaan Ke-					Rata - Rata RMSE
	1	2	3	4	5	
0.1						
0.2						
0.3						
0.4						
0.5						
0.6						
0.7						
0.8						
0.9						

Keterangan tabel 3.13

- Pada kolom laju pembelajaran, terlihat bahwa pengujian akan dilakukan terhadap 9 laju pembelajaran yang berbeda.

- Kolom RMSE Percobaan ke-, akan menampilkan nilai error sesuai dengan laju pembelajaran yang di tentukan yang dilakukan sebanyak 5 kali
- Kolom Rata-Rata RMSE merupakan penjumlahan hasil RMSE dari banyaknya pengujian pada tiap laju pembelajaran yang sama dan dibagi dengan banyaknya pengujian.

3.5.2 Pengujian Hasil Aktual dan Output Jaringan

Pengujian ini dilakukan untuk membandingkan hasil dari kerawanan aktual dan output jaringan. Hasil keluaran jaringan yang disebut dengan output jaringan ini akan dibandingkan dengan nilai data aktual. Pada pengujian ini, nilai output jaringan yang telah didapat dikembalikan ke dalam cluster yang sebelumnya telah dibentuk. Kerawanan telah dibagi menjadi 2 cluster yaitu rawan dan tidak rawan. Pergerakan rawan dan tidak rawan dari output jaringan akan dibandingkan dengan pergerakan rawan dan tidak rawan dari data aktual. Untuk mencatat hasil dari pengujian ini dibuat tabel 3.14.

Tabel 3.14 Contoh tabel hasil perbandingan kerawanan

Output Aktual	Output Jaringan	Cluster Aktual	Cluster Jaringan

Keterangan tabel 3.14

- Kolom output aktual menampilkan data kerawanan yang sebenarnya.
- Kolom output jaringan akan menampilkan hasil kerawanan, dari output jaringan ANFIS.

- Kolom cluster actual akan menampilkan pengelompokan data yang sebenarnya.
- Kolom cluster jaringan akan menampilkan pengelompokan data yang merupakan pengembalian nilai output jaringan.

Dengan menggunakan tabel 3.14 akan diketahui perbandingan antara data actual dan output jaringan dari sistem ANFIS.

3.5.3 Pengujian Tingkat Akurasi Kerawanan

Pengujian ini dilakukan untuk mengetahui tingkat akurasi dari nilai error yang dihasilkan output jaringan menggunakan ANFIS. Pada uji coba ini, akan dilakukan percobaan dengan menggunakan data latih yang berbeda-beda. Analisis akan dilakukan dengan membandingkan kerawanan yang merupakan output jaringan dari metode ANFIS dengan kerawanan yang merupakan target awal. Perbandingan ini digunakan untuk menghitung tingkat akurasi dengan menggunakan persamaan 2.38. Untuk mencatat hasil dari pengujian ini dibuat tabel 3.15.

Tabel 3.15 Contoh tabel hasil pengujian tingkat akurasi

Data Latih	Persentase Kebenaran (Percobaan ke-)				
	1	2	3	4	5
...					
...					
...					

Keterangan tabel 3.15 diatas adalah :

- Pada kolom data latih menunjukkan akan dilakukan beberapa kali percobaan dengan jumlah data latih yang berbeda-beda.
- Kolom persentase kebenaran menunjukan keakurasian hasil dari pengujian yang telah dilakukan
- Pada kolom di bawah persentase kebenaran menunjukan percobaan akan dilakukan sebanyak 5 kali

Dengan menggunakan tabel 3.15 akan diketahui tingkat kebenaran dari penggunaan sistem ANFIS sebagai penentuan kerawanan DBD.

3.6 Rancangan Antar Muka

Antarmuka (*interface*) untuk sistem ini terdiri dari tiga bagian utama, yaitu:

1. Bagian untuk proses pelatihan

Bagian ini bertujuan untuk mendapatkan parameter konsekuensi yang akan digunakan pada proses penentuan kerawanan. Antarmuka dari bagian ini ditunjukkan pada gambar 3.19.

2. Bagian untuk proses pengujian

Bagian ini bertujuan untuk melakukan pengujian terhadap parameter konsekuensi jaringan yang terbentuk serta kerawanan yang dihasilkan terhadap data aktual. Antarmuka dari bagian ini ditunjukkan pada gambar 3.20.

3. Bagian untuk prediksi

Bagian ini bertujuan untuk melakukan prediksi kerawanan dengan menginputkan satu nilai pada tiap parameter. Antarmuka dari bagian ini ditunjukkan pada gambar 3.21.

PELATIHAN	PENGUJIAN	PREDIKSI
Error harap	<input type="text" value="1"/>	Max <input type="text" value="3"/>
Laju	<input type="text" value="2"/>	<input type="button" value="Proses"/>
<input type="text" value="5"/>		<input type="text" value="8"/>
<input type="text" value="6"/>		
<input type="text" value="7"/>		

Gambar 3.19 Perancangan Antar Muka ANFIS Pelatihan

Pada gambar 3.19, antar muka bagian proses pelatihan terdiri dari :

1. *Text field* untuk menampilkan nilai error harap.
2. *Text field* untuk menampilkan nilai laju pembelajaran.
3. *Text field* untuk menginputkan nilai maksimum error.
4. Tombol proses, untuk melakukan proses ANFIS.
5. *Stringgrid* untuk menampilkan data uji.
6. *Stringgrid* untuk menampilkan nilai parameter premis.
7. *Stringgrid* untuk menampilkan nilai parameter konsekuensi.
8. *Text area* untuk menampilkan proses perubahan parameter premis tiap iterasi.

PELATIHAN	PENGUJIAN	PREDIKSI
<div style="border: 1px solid black; width: 100%; height: 100%; display: flex; align-items: center; justify-content: center;"> 1 </div>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px; text-align: center;">Ambil Data</div> <div style="border: 1px solid black; width: 100%; height: 100%; display: flex; align-items: center; justify-content: center;"> 3 </div>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px; text-align: center;">Proses</div> <div style="border: 1px solid black; width: 100%; height: 100%; display: flex; align-items: center; justify-content: center;"> 5 </div> <div style="margin-top: 10px;"> RMS <input style="text-align: right; color: orange; font-size: 1.2em; margin-left: 5px;" type="text"/> </div> <div style="margin-top: 5px;"> Akurasi <input style="text-align: right; color: orange; font-size: 1.2em; margin-left: 5px;" type="text"/> </div>

Gambar 3.20 Perancangan Antar Muka ANFIS Pengujian

Pada gambar 3.20, antarmuka bagian proses pelatihan terdiri dari :

1. *Stringgrid* untuk menampilkan parameter konsekuensi hasil pelatihan yang akan digunakan pada proses pengujian.
2. Tombol ambil data, untuk mengambil data yang akan digunakan pada proses pengujian.

3. *Stringgrid* untuk menampilkan data yang akan digunakan pada proses pengujian.
4. Tombol proses, untuk menghitung nilai kerawanan yang dihasilkan menggunakan parameter konsekuensi sekaligus menghitung RMSE dan akurasi.
5. *Stringgrid* untuk menampilkan hasil dari proses pengujian.
6. *Text field* untuk menampilkan nilai RMSE.
7. *Text field* untuk menampilkan nilai akurasi.

PELATIHAN	PENGUJIAN	PREDIKSI
Kelembaba	<input type="text"/>	1
Kecepatan	<input type="text"/>	2
Suhu	<input type="text"/>	3
Curah	<input type="text"/>	4
<input type="button" value="Hasil"/>		
		<input type="text"/>
		6

Gambar 3.21 Perancangan Antar Muka ANFIS Prediksi

Pada gambar 3.21, antarmuka bagian proses pelatihan terdiri dari :

1. *Text field* untuk menginputkan nilai kelembaban.
2. *Text field* untuk menginputkan nilai kecepatan angin.
3. *Text field* untuk menginputkan nilai suhu.
4. *Text field* untuk menginputkan nilai curah hujan.
5. Tombol hasil, untuk memproses inputan hingga menghasilkan sebuah kondisi rawan atau tidak rawan.
6. *Text area* untuk menampilkan hasil prediksi.

BAB IV

IMPLEMENTASI DAN PEMBAHASAN

Pada bab ini akan dijelaskan implementasi seluruh proses yang sudah dirancang pada bab sebelumnya disertai dengan analisa data yang dihasilkan.

4.1 Lingkungan Implementasi

Lingkungan implementasi meliputi lingkungan implementasi perangkat keras dan perangkat lunak

4.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan sistem penentuan kerawanan DBD menggunakan ANFIS ini adalah *laptop* dengan spesifikasi :

1. Prosesor Intel Core™ 2 Duo processor T5500 (1.66 GHz, 667 MHz FSB, 2 MB L2 cache)
2. RAM 1 GB DDR2
3. *Harddisk* dengan kapasitas 320 GB

4.1.2 Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan dalam pengembangan sistem penentuan kerawanan DBD menggunakan ANFIS ini adalah :

1. Sistem operasi Windows 7
2. Borland Delphi7
3. Microsoft Access 2007

4.2 Implementasi Program

Dari analisa dan perancangan sistem yang terdapat pada sub bab 3.1, maka pada sub bab ini akan dijelaskan implementasi proses-proses tersebut secara lebih rinci.

4.2.1 Tahap Pelatihan

Pada tahap pelatihan ini sejumlah data iklim yang akan digunakan untuk pelatihan diinputkan dan dicari parameter

konsekuensi yang nantinya akan digunakan pada tahap pengujian. Data latih ini nantinya akan melalui beberapa tahapan pelatihan yang akan dijelaskan pada sub bab berikutnya.

4.2.1.1 Pengklasteran dengan K-Means Clustering

Pada data iklim yang akan dilatih di kelompokkan menjadi 2 kluster yang nantinya digunakan untuk mencari mean. Data iklim yang telah diinputkan distandarisasi, kemudian dicari bobot tiap data. Setelah mendapatkan bobot, tahap selanjutnya adalah menentukan centroid awal yang dibangkitkan secara random, kemudian dihitung jaraknya menggunakan *Euclidean distance* dan dikelompokkan menurut jarak yg terdekat. Proses ini akan terus berulang hingga anggota dalam suatu kelompok sudah tidak berubah. Proses ini dapat dilihat pada sourcecode 4.1.

```
procedure TForm1.Button1Click(Sender: TObject);
var
i, y, c1, c2, itr, j, k, n, m, o, x: Integer;
bobot, stKL, stKC, stSH, stCH, dtKL, dtKC, dtSH, dtCH, minKL, ma
xKL, minKC, maxKC, minSH, maxSH, minCH, maxCH, maxB, minB, maxE
rr, minErr, a, b, RC1, RC2, JC1, JC2, JC3, JC4, n_kel, n_kel2, det
A, z, totEl9, ndt, totf1, totf2, totw1, totw2, totwn1, totwn2, t
otw, tkel1, tkec1, tsuh1, tcurl, tkel2, tkec2, tsuh2, tcur2, to
tkel, totkec, totsuh, totcur, ErrH, Lr: real;
begin
    DM.Q1.Close;
    DM.Q1.SQL.Text:='select * from tabelDataLat';
    DM.Q1.Open;
    DM.Q1.First;
    DM.Q2.Close;
    DM.Q2.SQL.Text:='SELECT MAX(kelembaban) as
max_kel, MIN(kelembaban) as min_kel, MAX(kecepatan_angin)
as max_kec, MIN(kecepatan_angin) as min_kec, MAX(suhu)
as max_sh, MIN(suhu) as min_sh, MAX(curah_hujan) as
max_ch, MIN(curah_hujan) as min_ch FROM tabelDataLat';
    DM.Q2.Open;
    DM.Q2.First;
    i:=1;
    n:=0;
    minB:=100;
    maxB:=0;
    ndt:=dm.Q1.RecordCount;
    setlength(r1, dm.Q1.RecordCount);
```

```

        setlength(r2, dm.Q1.RecordCount);
        StringGrid1.RowCount:=dm.Q1.RecordCount+1;
        StringGrid3.RowCount:=dm.Q1.RecordCount+1;
        StringGrid4.RowCount:=dm.Q1.RecordCount+1;

while (not dm.Q1.Eof) do
    begin
        for i:=1 to dm.Q1.RecordCount do
            begin
                dtKL[i]:=dm.Q1['kelembaban'];
                dtKC[i]:=dm.Q1['Kecepatan_angin'];
                dtSH[i]:=dm.Q1['suhu'];
                dtCH[i]:=dm.Q1['curah_hujan'];
                StringGrid1.Cells[0,i]:=IntToStr(i);
                StringGrid1.Cells[1,i]:=dm.Q1['kelembaban'];
                StringGrid1.Cells[2,i]:=dm.Q1['kecepatan_angin'];
                StringGrid1.Cells[3,i]:=dm.Q1['suhu'];
                StringGrid1.Cells[4,i]:=dm.Q1['curah_hujan'];
                StringGrid1.Cells[5,i]:=dm.Q1['target_output'];
                //mencari nilai max & min per parameter
                maxKL:=dm.Q2['max_kel'];
                minKL:=dm.Q2['min_kel'];
                maxKC:=dm.Q2['max_kec'];
                minKC:=dm.Q2['min_kec'];
                maxSH:=dm.Q2['max_sh'];
                minSH:=dm.Q2['min_sh'];
                maxCH:=dm.Q2['max_ch'];
                minCH:=dm.Q2['min_ch'];
                //standarisasi data
                stKL[i]:=(dtKL[i]-minKL)/(maxKL-minKL);
                stKC[i]:=(dtKC[i]-minKC)/(maxKC-minKC);
                stSH[i]:=(dtSH[i]-minSH)/(maxSH-minSH);
                stCH[i]:=(dtCH[i]-minCH)/(maxCH-minCH);
                //perhitungan bobot
                bobot[i]:=(stKL[i]+stKC[i]+stSH[i]+stCH[i])/4;
                StringGrid1.Cells[7,i]:=FloatToStr(bobot[i]);
                //max & min bobot
                if (bobot[i]<minB) then
                    minB:=bobot[i];
                if (bobot[i]>maxB) then
                    maxB:=bobot[i];
                dm.Q1.Next;
                dm.Q2.Next;
            end;
            Inc(i);
        end;
    end;

```

```

//mencari nilai random
Randomize();
a:=Random; b:=Random;
RC1:=( (maxB-minB)*b)+minB;
RC2:=( (maxB-minB)*a)+minB;
Edit1.Text:=FloatToStr(RC1);
Edit2.Text:=FloatToStr(RC2);

//mengcluster
sama:=false;
while not sama do
begin
inc(n); j:=0; c1:=0; c2:=0; n_kel:=0; n_kel2:=0;
for i:=1 to dm.Q1.RecordCount do
begin
bobot[i]:=StrToFloat(StringGrid1.Cells[7,i]);
JC3[i]:=abs(bobot[i]-StrToFloat(Edit1.Text));
JC4[i]:=abs(bobot[i]-StrToFloat(Edit2.Text));
end;
sama:=true;
for i:=1 to dm.Q1.RecordCount do
begin
if (JC3[i]) < (JC4[i]) then
begin
cluster:='1';
c1:=c1+1;
StringGrid1.Cells[9,i]:=cluster;
n_kel:=n_kel + bobot[i];
end
else begin
cluster:='2';
c2:=c2+1;
StringGrid1.Cells[9,i]:=cluster;
n_kel2:=n_kel2 + bobot[i];
end;
end;
end;
for i:=0 to dm.Q1.RecordCount-1 do
begin
if (r2[i] <> r1[i]) then sama:=False;
end;

```

Sourcecode 4.1 Pengklusteran menggunakan *K-Means clustering*

4.2.1.2 Menghitung Mean dan Standar Deviasi

Setelah data tercluster selanjutnya dihitung mean dan standar deviasi dari masing-masing cluster. Proses ini dapat dilihat pada sourcecode 4.2.

```
for i:=0 to dm.Q1.RecordCount-1 do
  begin
    if (StringGrid1.Cells[9,i+1]='1') then begin
      for j:=1 to 4 do
        begin
          tot1[j]:=tot1[j] +
            StrToFloat(StringGrid1.Cells[j,i+1]);
          inc(npr1[j]);
        end;
      if n=1 then begin
        r1[i]:='0';
        r2[i]:='1';
      end
      else begin
        r1[i]:=r2[i];
        r2[i]:='1';
      end
    end
    else if (StringGrid1.Cells[9,i+1]='2') then begin
      for j:=1 to 4 do
        begin
          tot2[j]:=tot2[j] +
            StrToFloat(StringGrid1.Cells[j,i+1]);
          inc(npr2[j]);
        end;
      if n=1 then begin
        r1[i]:='0';
        r2[i]:='2';
      end
      else begin
        r1[i]:=r2[i];
        r2[i]:='2';
      end;
    end;
  end;
//menghitung rata2
for i:=1 to 4 do
  begin
    rt1[i]:=tot1[i]/npr1[i];
    rt2[i]:=tot2[i]/npr2[i];
  end;
```

```

//ngitung standart deviasi
for i:=0 to dm.Q1.RecordCount-1 do
begin
  if (StringGrid1.Cells[9,i+1]='1') then begin
    for j:=1 to 4 do
      begin
pr1[j]:=pr1[j]+(Power((StrToFloat(StringGrid1.Cells[j,i
+1])-rt1[j]),2));
end;
end
    else if (StringGrid1.Cells[9,i+1]='2') then begin
      for j:=1 to 4 do
        begin
pr2[j]:=pr2[j]+(Power((StrToFloat(StringGrid1.Cells[j,i
+1])-rt2[j]),2));
end;
end;
end;
end;
for i:=1 to 4 do
begin
  std1[i]:=sqrt(pr1[i]/(npr1[i]-1));
  std2[i]:=sqrt(pr2[i]/(npr2[i]-1));
end;

```

Sourcecode 4.2 Perhitungan mean dan standar deviasi

4.2.1.3 Menghitung Lapisan Satu

Lapisan satu merupakan lapisan untuk penghitungan derajat keanggotaan, dimana fungsi yang dipakai adalah fungsi bell pada persamaan 2.4. Sourcecode 4.3 merupakan proses menghitung lapisan satu.

```

for j:=1 to 4 do
  for i:=1 to dm.Q1.RecordCount do
    begin
//perhitungan lapisan 1
kell[j,i]:=1/(1+(Power(((StrToFloat(StringGrid1.Cell
s[j,i])-rt1[j])/(std1[j])),2)));
kel2[j,i]:=1/(1+(Power(((StrToFloat(StringGrid1.Cell
s[j,i])-rt2[j])/(std2[j])),2)));

```

Sourcecode 4.3 Perhitungan lapisan satu

4.2.1.4 Menghitung Lapisan Dua

Pada lapisan dua dilakukan penghitungan *firing strength*. Berikut merupakan proses dari lapisan dua yang dipaparkan pada sourcecode 4.4.

```
//perhitungan lapisan 2  
w1[i]:=w1[i]*kel1[j,i];  
w2[i]:=w2[i]*kel2[j,i];
```

Sourcecode 4.4 Perhitungan lapisan dua

4.2.1.5 Menghitung Lapisan Tiga

Pada lapisan tiga dilakukan penghitungan normalisasi *firing strength*. Berikut merupakan proses dari lapisan tiga yang dipaparkan pada sourcecode 4.5.

```
//perhitungan lapisan 3  
wn1[i]:=w1[i]/(w1[i]+w2[i]);  
wn2[i]:=w2[i]/(w1[i]+w2[i]);
```

Sourcecode 4.5 Perhitungan lapisan tiga

4.2.1.6 Menghitung Lapisan Empat

Pada lapisan keempat terdapat penghitungan matrik desain, tahap ini merupakan tahap yang digunakan untuk dapat menentukan parameter konsekuen yang merupakan output dari lapisan keempat. Proses perhitungan matrik desain dapat dilihat pada sourcecode 4.6.

```
//perhitungan lapisan 4  
c1x0[i]:=wn1[i];  
c1x1[i]:=wn1[i]*StrToFloat(StringGrid1.Cells[1,i]);  
c1x2[i]:=wn1[i]*StrToFloat(StringGrid1.Cells[2,i]);  
c1x3[i]:=wn1[i]*StrToFloat(StringGrid1.Cells[3,i]);  
c1x4[i]:=wn1[i]*StrToFloat(StringGrid1.Cells[4,i]);  
c2x0[i]:=wn2[i];  
c2x1[i]:=wn2[i]*StrToFloat(StringGrid1.Cells[1,i]);  
c2x2[i]:=wn2[i]*StrToFloat(StringGrid1.Cells[2,i]);  
c2x3[i]:=wn2[i]*StrToFloat(StringGrid1.Cells[3,i]);  
c2x4[i]:=wn2[i]*StrToFloat(StringGrid1.Cells[4,i]);
```

Sourcecode 4.6 Perhitungan lapisan keempat

4.2.1.7 Menghitung Parameter Konsekuen Menggunakan LSE

Pada tahap ini dilakukan penghitungan LSE yang tujuannya adalah untuk mendapatkan parameter konsekuen. Parameter konsekuen merupakan output dari lapisan keempat. Ada beberapa tahap yang dilakukan pada metode LSE ini. Proses penghitungan dapat dilihat pada sourcecode 4.7.

```
//matrix desain
SetLength (An,StringGrid4.RowCount-
1,StringGrid4.ColCount-1);
  for i:=1 to StringGrid4.RowCount-1 do
    for j:=1 to StringGrid4.ColCount-1 do
      begin
        An[i-1,j-1]:=StrToFloat (StringGrid4.Cells[j,i]);
        StringGrid4.Cells[j,i]:=FloatToStr (An[i-1,j-1]);
      end;
    //transpose matrix desain
    SetLength (AnT,StringGrid4.colCount-
1,StringGrid4.rowCount-1);
    for i:=1 to StringGrid4.colCount-1 do
      for j:=1 to StringGrid4.rowCount-1 do
        begin
          AnT[i-1,j-1]:=An[j-1,i-1];
        end;
      //perkalian matrix transpose dan matrix desain
      SetLength (AnTxAn,StringGrid4.colCount-
1,StringGrid4.colCount-1);
      for i:=1 to StringGrid4.colCount-1 do
        for j:=1 to StringGrid4.colCount-1 do
          begin
            AnTxAn[i-1,j-1]:=0;
            for k:=1 to StringGrid4.rowCount-1 do
              AnTxAn[i-1,j-1]:=AnTxAn[i-1,j-1]+AnT[j-1,k-
1]*An[k-1,i-1];
            end;
            //perkalian matrix invers dan transpose
            SetLength (BixAnT,StringGrid4.colCount-
1,StringGrid4.rowCount-1);
            for i:=1 to StringGrid4.colCount-1 do
              for j:=1 to StringGrid4.rowCount-1 do
                begin
                  BixAnT[i-1,j-1]:=0;
                  for k:=1 to StringGrid4.colCount-1 do
                    BixAnT[i-1,j-1]:=BixAnT[i-1,j-1]+Bi[i,k]*AnT[k-
1,j-1];
                  end;
                end;
              end;
            end;
```

```

//matrix Yn
SetLength(Yn, StringGrid1.RowCount-1, 1);
for i:=1 to StringGrid1.RowCount-1 do
  for j:=1 to 1 do
    begin
      Yn[i-1, j-1]:=StrToFloat
        (StringGrid1.Cells[j+4, i]);
    end;
//perkalian matrix BixAnT dan Yn
SetLength(Pn, StringGrid4.rowCount-1, 1);
for i:=1 to StringGrid4.colCount-1 do
  for j:=1 to 1 do
    begin
      Pn[i-1, j-1]:=0;
      for k:=1 to StringGrid4.rowCount-1 do
        Pn[i-1, j-1]:=Pn[i-1, j-1]+BixAnT[i-1, k-1]*Yn[k-1, j-1];
      end;
    end;

```

Sourcecode4.7Perhitungan parameter konsekuen menggunakan LSE

4.2.1.8 Menghitung Output Jaringan

Setelah mendapatkan parameter konsekuen, tahap berikutnya adalah mencari output jaringan. Sourcecode 4.8 merupakan proses penghitungan output jaringan.

```

//menghitung output jaringan
f1[i]:=((StrToFloat(StringGrid5.Cells[1,1])*StrToFloat(
StringGrid1.Cells[1,i]))+(StrToFloat(StringGrid5.Cells[
1,2])*StrToFloat(StringGrid1.Cells[2,i]))+(StrToFloat(S
tringGrid5.Cells[1,3])*StrToFloat(StringGrid1.Cells[3,i
]))+(StrToFloat(StringGrid5.Cells[1,4])*StrToFloat(Stri
ngGrid1.Cells[4,i]))+(StrToFloat(StringGrid5.Cells[1,5]
)));
totf1:=totf1+(f1[i]/ndt);
f2[i]:=((StrToFloat(StringGrid5.Cells[1,6])*StrToFloat(
StringGrid1.Cells[1,i]))+(StrToFloat(StringGrid5.Cells[
1,7])*StrToFloat(StringGrid1.Cells[2,i]))+(StrToFloat(S
tringGrid5.Cells[1,8])*StrToFloat(StringGrid1.Cells[3,i
]))+(StrToFloat(StringGrid5.Cells[1,9])*StrToFloat(Stri
ngGrid1.Cells[4,i]))+(StrToFloat(StringGrid5.Cells[1,10
]));
totf2:=totf2+(f2[i]/ndt);
w1f1[i]:=(wn1[i]*f1[i]);
w2f2[i]:=(wn2[i]*f2[i]);
otj[i]:=w1f1[i]+w2f2[i];

```

Sourcecode 4.8 Perhitungan output jaringan

4.2.1.9 Menghitung Propagasi Error

Menghitung propagasi error merupakan cara untuk membuktikan error yang dihasilkan dari perhitungan jaringan. Sesuai dengan arsitektur jaringan yang ada, maka nilai error yang dihitung adalah error dari lapisan ke-5 sampai lapisan ke-1. Proses perhitungan propagasi error dapat dilihat pada sourcecode 4.9.

```
//error lapisan 5
errjar[i]:=(StrToFloat(StringGrid1.Cells[j+4,i])-
otj[i]);
e19[i]:=-2*errjar[i];
totE19:=totE19+(e19[i]/ndt);
//error lapisan 4
e18:=totE19*1;
e17:=totE19*1;
//error lapisan 3
e15:=e17*totf1;
e16:=e18*totf2;
//error lapisan 2
e13:=(totw1/wtot)*(e15-e16);
e14:=(totw2/wtot)*(e16-e15);
//error lapisan 1
e5:=e13*tcurl;
e6:=e14*tcurl2;
e7:=e13*tsuh1;
e8:=e14*tsuh2;
e9:=e13*tkecl;
e10:=e14*tkecl2;
e11:=e13*tkel1;
e12:=e14*tkel2;
```

Sourcecode 4.9 Perhitungan propagasi error

4.2.1.10 Memperbarui Nilai Mean dan Standar Deviasi

Tahap ini merupakan tahap untuk memperbarui nilai mean dan standar deviasi yang nantinya akan digunakan pada iterasi selanjutnya. Proses perbaharuan *mean* dan *standart deviasi* dapat dilihat pada sourcecode 4.10.

```
//mencari nilai mean n stdev yg baru
//menghitung Eaij
ea11:=e5*((2*(Power((totkel-rt1[1]),2)))/
((Power(std1[1],3))*((Power((1+(Power(((totkel-
rt1[1])/std1[1]),2))),2)))));
```

```

ea12:=e6*((2*(Power((totkel-rt2[1]),2)))/
((Power(std2[1],3))*((Power((1+(Power((totkel-
rt2[1])/std2[1]),2)),2)))));
ea21:=e7*((2*(Power((totkec-rt1[2]),2)))/
((Power(std1[2],3))*((Power((1+(Power((totkec-
rt1[2])/std1[2]),2)),2)))));
ea22:=e8*((2*(Power((totkec-rt2[2]),2)))/
((Power(std2[2],3))*((Power((1+(Power((totkec-
rt2[2])/std2[2]),2)),2)))));
ea31:=e9*((2*(Power((totsuh-rt1[3]),2)))/
((Power(std1[3],3))*((Power((1+(Power((totsuh-
rt1[3])/std1[3]),2)),2)))));
ea32:=e10*((2*(Power((totsuh-rt2[3]),2)))/
((Power(std2[3],3))*((Power((1+(Power((totsuh-
rt2[3])/std1[2]),2)),2)))));
ea41:=e11*((2*(Power((totcur-rt1[4]),2)))/
((Power(std1[4],3))*((Power((1+(Power((totcur-
rt1[4])/std1[4]),2)),2)))));
ea42:=e12*((2*(Power((totcur-rt2[4]),2)))/
((Power(std2[4],3))*((Power((1+(Power((totcur-
rt2[4])/std2[4]),2)),2)))));
//menghitung Ecij
ec11:=e5*((2*((totkel-rt1[1])))/
((Power(std1[1],2))*((Power((1+(Power((totkel-
rt1[1])/std1[1]),2)),2)))));
ec12:=e6*((2*((totkel-rt2[1])))/
((Power(std2[1],2))*((Power((1+(Power((totkel-
rt2[1])/std2[1]),2)),2)))));
ec21:=e7*((2*((totkec-rt1[2])))/
((Power(std1[2],2))*((Power((1+(Power((totkec-
rt1[2])/std1[2]),2)),2)))));
ec22:=e8*((2*((totkec-rt2[2])))/
((Power(std2[2],2))*((Power((1+(Power((totkec-
rt2[2])/std2[2]),2)),2)))));
ec31:=e9*((2*((totsuh-rt1[3])))/
((Power(std1[3],2))*((Power((1+(Power((totsuh-
rt1[3])/std1[3]),2)),2)))));
ec32:=e10*((2*((totsuh-rt2[3])))/
((Power(std2[3],2))*((Power((1+(Power((totsuh-
rt2[3])/std1[2]),2)),2)))));
ec41:=e11*((2*((totcur-rt1[4])))/
((Power(std1[4],2))*((Power((1+(Power((totcur-
rt1[4])/std1[4]),2)),2)))));
ec42:=e12*((2*((totcur-rt2[4])))/
((Power(std2[4],2))*((Power((1+(Power((totcur-
rt2[4])/std2[4]),2)),2)))));

```

Sourcecode 4.10 Perbaharuan mean dan standar deviasi

4.2.1.11 Proses Iterasi untuk Memperbarui Nilai Mean dan Standar Deviasi

Pada tahap ini akan dilakukan iterasi sesuai yang ditentukan dengan syarat banyak iterasi dan error yang diharapkan. Hal ini dilakukan untuk mendapatkan nilai parameter konsekuen yang paling bagus. Proses untuk iterasi dapat dilihat pada sourcecode 4.11.

```
while (not error) and (itr<maxitr) do
begin
  inc(itr);
  if (itr>1) then
  begin
    StringGrid2.Cells[1,1]:=StringGrid2.Cells[1,7];
    StringGrid2.Cells[2,1]:=StringGrid2.Cells[2,7];
    StringGrid2.Cells[3,1]:=StringGrid2.Cells[3,7];
    StringGrid2.Cells[4,1]:=StringGrid2.Cells[4,7];
    StringGrid2.Cells[1,2]:=StringGrid2.Cells[1,8];
    StringGrid2.Cells[2,2]:=StringGrid2.Cells[2,8];
    StringGrid2.Cells[3,2]:=StringGrid2.Cells[3,8];
    StringGrid2.Cells[4,2]:=StringGrid2.Cells[4,8];
    StringGrid2.Cells[1,4]:=StringGrid2.Cells[1,10];
    StringGrid2.Cells[2,4]:=StringGrid2.Cells[2,10];
    StringGrid2.Cells[3,4]:=StringGrid2.Cells[3,10];
    StringGrid2.Cells[4,4]:=StringGrid2.Cells[4,10];
    StringGrid2.Cells[1,5]:=StringGrid2.Cells[1,11];
    StringGrid2.Cells[2,5]:=StringGrid2.Cells[2,11];
    StringGrid2.Cells[3,5]:=StringGrid2.Cells[3,11];
    StringGrid2.Cells[4,5]:=StringGrid2.Cells[4,11];
  end;
end;
```

Sourcecode 4.11 Proses iterasi

4.2.2 Tahap Pengujian

Tahap pengujian adalah tahap dimana nilai parameter konsekuen yang dihasilkan dari ANFIS pelatihan akan digunakan untuk dikalikan dengan data pengujian. Hasil dari pengujian merupakan kerawanan jaringan. Kerawanan jaringan ini akan dibandingkan dengan kerawanan aktual. Proses pengujian dapat dilihat pada sourcecode 4.12.

```

cp1:=StrToFloat (StringGrid5.Cells [1,1]);
cp2:=StrToFloat (StringGrid5.Cells [1,2]);
cp3:=StrToFloat (StringGrid5.Cells [1,3]);
cp4:=StrToFloat (StringGrid5.Cells [1,4]);
cp5:=StrToFloat (StringGrid5.Cells [1,5]);
cp6:=StrToFloat (StringGrid5.Cells [1,6]);
cp7:=StrToFloat (StringGrid5.Cells [1,7]);
cp8:=StrToFloat (StringGrid5.Cells [1,8]);
cp9:=StrToFloat (StringGrid5.Cells [1,9]);
cp10:=StrToFloat (StringGrid5.Cells [1,10]);
y1[i]:=(cp1*KL[i])+(cp2*KC[i])+(cp3*SH[i])+
        (cp4*CH[i])+(cp5));
y2[i]:=(cp6*KL[i])+(cp7*KC[i])+(cp8*SH[i])+
        (cp9*CH[i])+(cp10));
w1y1[i]:=(wn1[i]*y1[i]);
w2y2[i]:=(wn2[i]*y2[i]);
HasilJ[i]:=w1y1[i]+w2y2[i];

```

Sourcecode 4.12 Proses pengujian

4.3 Implementasi Antarmuka

Berdasarkan rancangan antarmuka yang terdapat pada sub bab 3.6. Maka pada sub bab ini akan dijelaskan hasil antarmuka untuk penerapan ANFIS dalam penentuan kerawanan DBD pada suatu daerah. Antarmuka ANFIS ini memiliki tiga tab yaitu tab pelatihan, tab pengujian dan tab prediksi.

The screenshot shows a software interface for ANFIS training. The main window is titled "PENENTUAN KERAWANAN DEMAM BERDARAH MENGGUNAKAN ANFIS (ADAFITIVE NEURO FUZZY INFERENCE SYSTEM)". It has three tabs: "PELATIHAN", "PENGUJIAN", and "PENENTUAN KERAWANAN DBD". The "PELATIHAN" tab is selected. In this tab, there are input fields for "Error Harap", "Maximum Iterasi", and "Laju Pembelajaran". There are also radio buttons for "Data Latih" with options 40, 75, and 110. A "Proses" button is located below these fields. To the right is a "Memo1" text area. Below the input fields are two tables: "Parameter Premis" and "Parameter Konsekuensi". The "Parameter Premis" table has columns labeled x1, x2, x3, and x4. The "Parameter Konsekuensi" table has a column labeled "Parameter Konsekuensi".

Gambar 4.1 Tab pelatihan

Gambar 4.1 adalah tampilan tab pelatihan. Pada tab ini pengguna diminta untuk menginputkan nilai error harap, laju pembelajaran, dan maksimal iterasi. Setelah nilai tersebut dimasukkan pengguna harus mengklik tombol proses yang fungsinya adalah untuk memproses inputan yang telah diberikan dan nantinya akan menghasilkan output jaringan, bobot, nilai parameter premis dan parameter konsekuen. Proses pelatihan ANFIS dapat dilihat pada gambar 4.2.

The screenshot shows the ANFIS training interface with the following data:

Training Parameters:
 Error Harap: 1E-15
 Maximum Iterasi: 9020
 Laju Pembelajaran: 0.8
 Data Lath: 40 (selected), 75, 110

Training Data Table:

No	Kelambaban	Kecepatan Ar	Suhu	Curah Hujan	Target Output	Bob
1	70.375	4.525	27.803125	8.9375	1	0.36
2	76.21875	2.85	27.628125	0	2	0.25
3	82.71875	2.0625	27.059375	15.2875	1	0.42

Parameter Premis:

	x1	x2	x3	x4
Mean 1	78.123463833494	2.41481540529491	27.6536670917816	1.81624149658526
Mean 2	81.0975854260168	2.77361160709225	27.8823660713916	11.1467261904608
SD 1	4.08467255505486	0.53976312464325	1.12131404488688	2.26204272909669
SD 2	4.80337970626592	1.01163461020471	2.7159323113199	7.0674244957916

Parameter Konsekuensi:

Parameter Konsekuensi
-0.0228086213423129
0.349033546797636
0.36926122092939
-0.132913752073988

Output Results:

Iterasi ke-9017:
 MEAN: 78.1234638335059 2.41481540529828 27.6536670917875 1.816241 81.0975854260426 2.7736116070973 27.8823660713957 11.146726
 STANDART DEVIASI: 4.08467255505748 0.539763124643326 1.12131404488724 2.26204 4.80337970625476 1.01163461020345 2.71593231131966 7.067424

Iterasi ke-9018:
 MEAN: 78.1234638334962 2.41481540529554 27.6536670917827 1.816241 81.0975854260217 2.7736116070932 27.8823660713924 11.146726
 STANDART DEVIASI: 4.08467255505535 0.539763124643379 1.12131404488695 2.26204 4.80337970626382 1.01163461020447 2.71593231131966 7.067424

Iterasi ke-9019:
 MEAN: 78.1234638334966 2.41481540529555 27.6536670917827 1.816241 81.0975854260225 2.77361160709336 27.8823660713925 11.14672
 STANDART DEVIASI: 4.08467255505543 0.539763124643434 1.12131404488696 2.2620427 4.803379706263947 1.01163461020443 2.71593231131985 7.067424

Iterasi ke-9020:
 MEAN: 78.123463833494 2.41481540529491 27.6536670917816 1.8162414 81.0975854260168 2.77361160709225 27.8823660713916 11.14672
 STANDART DEVIASI: 4.08467255505486 0.539763124643252 1.12131404488688 2.26204 4.80337970626592 1.01163461020471 2.7159323113199 7.0674244

Gambar 4.2 Proses pelatihan ANFIS

Setelah mendapatkan nilai parameter konsekuen dari pelatihan, langkah selanjutnya adalah melakukan pengujian. Gambar 4.3 adalah tab untuk pengujian.

Form1

PENENTUAN KERAWANAN DEMAM BERDARAH MENGGUNAKAN ANFIS (ADAFIVE NEURO FUZZY INFERENCE SYSTEM)

PELATIHAN | PENGUJIAN | PENENTUAN KERAWANAN DBD

No	Kelembaban	Kecepatan Angin	Suhu	Curah Hujan
1	77.7436655627	1.8690476190425	35.71428571	0
2	78.440684348462	651041666661248125	5.0375	
3	82.54162756692	30729166666125	45	2
4	83.64167804582	84375	25.3375	1.7875
5	75.80748741022	23809523809124	7142857142	0.0142857142
6	80.3554119162	50520833333	25.925	0
7	71.8850995216	2.625	25.0375	0.05
8	74.709441647112	171875	26.175	0
9	73.05271166702	25694444444	26.1	0
10	75.34490772615	1.75520833333	27.3875	0
11	75.17661228791	1.8125	26.925	0.525
12	75.62346750715	1.640625	27.4125	1.0875
13	74.26315475247	1.96428571428	27.7571428571	0.0142857142
14	82.08659460572	1.75	27.2375	5.2625
15	77.079223108812	1.45833333333	28.125	0.2875
16	67.880154187613	5.78125	27.075	0
17	70.5127817138	2.65972222222	28.2333333333	2.7166666666
18	77.47785965011	1.72395833333	27.6625	0.85
19	74.404822910221	1.72916666666	27.9875	0.2875

Proses

	Output Jaringan	Output Target
1	1	1
2	1	1
3	1	1
4	1	1
5	1	1
6	1	2
7	1	2
8	1	2
9	1	2
10	2	2
11	1	1
12	1	1
13	2	2
14	1	1

RMSE 0.385068667978256

Akurasi 95.0%

Benar 34 Salah 6

Gambar 4.4 Proses pengujian ANFIS

Tab berikutnya adalah tab prediksi, tab prediksi ini memiliki fungsi untuk mengetahui kerawanan dengan menginputkan satu nilai pada masing-masing parameter, parameter tersebut antara lain kelembaban, kecepatan angin, suhu dan curah hujan, kemudian tekan tombol hasil maka akan menghasilkan rawan atau tidak rawan untuk nilai yang telah diinputkan. Gambar 4.5 adalah tampilan tab untuk prediksi dan gambar 4.6 adalah proses prediksi

Form1

PENENTUAN KERAWANAN DEMAM BERDARAH MENGGUNAKAN ANFIS (ADAFIVE NEURO FUZZY INFERENCE SYSTEM)

PELATIHAN | PENGUJIAN | PENENTUAN KERAWANAN DBD

Kelembaban range 60 - 90

Kecepatan Angin range 1 - 5

Suhu range 20 - 30

Curah Hujan range 0 - 30

Hasil

Memo2

Memo3

Gambar 4.5 Tab prediksi

Form1

PENENTUAN KERAWANAN DEMAM BERDARAH MENGGUNAKAN ANFIS (ADAFITIVE NEURO FUZZY INFERENCE SYSTEM)

PELATIHAN | PENGUJIAN | PENENTUAN KERAWANAN DBD

Kelembaban	85	range 60 - 90
Kecepatan Angin	2.1	range 1 - 5
Suhu	25.1	range 20 - 30
Curah Hujan	8.6	range 0 - 30

Hal-hal yang harus dilakukan :

- Menguras bak mandi dan tempat penampungan air sekurang-kurangnya seminggu sekali, dikarenakan perkembangan telur nyamuk menetas sekitar 7-10 hari
- Menutup rapat tempat penampungan air. Agar nyamuk tidak menggunakannya sebagai tempat berkembang biak
- Mengganti air pada vas bunga dan tempat minum burung setidaknya seminggu sekali
- Membersihkan perkarangan atau halaman rumah dari barang-barang yang dapat menampung air hujan, karena berpotensi sebagai tempat berkembangnya jentik-jentik nyamuk
- Pengasapan/togging dengan menggunakan malathion dan fenitron yang berguna untuk mengurangi kemungkinan penularan aedes aegypti dengan batas tertentu
- Memberikan bubuk abate (temephos) pada tempat-tempat yang sering menjadi tempat penampungan air

Hasil

Hasil kerawanan : Rawan

Gambar 4.6 Proses prediksi

4.4 Uji Coba Sistem

Dari sistem yang telah dibuat sebelumnya, maka pada sub bab ini akan dijabarkan pembahasan mengenai proses pengujian yang telah dilakukan pada sistem dan hasil proses pengujian tersebut. Sistematika pengujian ini akan dilakukan sebanyak tiga macam. Sistematika pengujian tersebut antara lain:

1. Uji Laju Pembelajaran Untuk Penentuan Kerawanan DBD

Pengujian ini adalah menguji laju pembelajaran yang baik untuk diterapkan pada sistem penentuan kerawanan DBD. Pengujian ini nantinya akan menggunakan tiga data latih yang berbeda yaitu 40 data latih, 75 data latih dan 110 data latih. Pengujian yang dilakukan pertama kali adalah pengujian dengan menggunakan laju pembelajaran dengan membandingkan hasil nilai RMSE nya. Laju pembelaran yang diujikan sebanyak 9 percobaan dengan nilai laju pembelajaran antara 0 sampai 1. Sebanyak 9 percobaan itu akan dicari nilai laju pembelajaran yang memiliki nilai RMSE paling kecil.

2. Uji perbandingan hasil aktual dan hasil jaringan ANFIS

Pengujian ini adalah menguji nilai output jaringan ANFIS. Nilai output jaringan ANFIS ini akan dibandingkan dengan nilai output aktual. Pada pengujian ketiga ini, nilai output jaringan ANFIS yang telah didapat dikembalikan ke dalam cluster yang sebelumnya telah dibentuk. Sebelumnya kerawanan DBD telah dibagi menjadi dua cluster yaitu rawan dan tidak rawan. Pergerakan rawan dan tidak rawan dari output jaringan ANFIS akan dibandingkan dengan pergerakan rawan dan tidak rawan dari output aktual. Pengujian ini dilakukan sebanyak tiga kali dengan menggunakan data latih yang berbeda-beda yaitu 40, 75 dan 110 data latih.

3. Uji tingkat akurasi

Pengujian yang terakhir adalah menguji tingkat akurasi dari sistem. Pengujian ini dilakukan 5 kali dengan menggunakan data latih yang berbeda-beda yaitu 40, 75 dan 110 data latih. Hasil dari pengujian ini adalah tingkat akurasi dari percobaan ke-1 sampai percobaan ke-5. Tingkat akurasi ini nantinya akan di rata-rata sehingga hasilnya merupakan tingkat akurasi dari sistem.

4.4.1 Uji Laju Pembelajaran

Data latih pertama yang digunakan untuk uji coba adalah 40 data latih. Sesuai dengan sistematika pengujian, pengujian dilakukan sebanyak 9 kali dimulai dengan laju pembelajaran terkecil sampai laju pembelajaran terbesar yaitu 0,1 – 0,9. Hasil dari pengujian laju pembelajaran menggunakan 40 data latih dapat dilihat pada tabel 4.1.

Tabel 4.1 Pengujian Laju Pembelajaran Untuk 40 Data Latih

Laju Pembelajaran	Rata-rata RMSE
0.1	0.401847678526
0.2	0.403305108493
0.3	0.403305108493

0.4	0.401361058222
0.5	0.401361058216
0.6	0.403305108498
0.7	0.401361058202
0.8	0.321283008500
0.9	0.403305108499

Pada tabel dapat dilihat bahwa nilai RMSE yang paling kecil yang dihasilkan oleh pengujian ini adalah 0.321283008500 pada saat laju pembelajaran adalah 0,8. Sedangkan nilai RMSE yang paling besar adalah 0.403305108499 pada saat laju pembelajaran 0,9. Hasil dari pengujian laju pembelajaran untuk 75 data latih dapat dilihat pada tabel 4.2.

Tabel 4.2 Pengujian Laju Pembelajaran Untuk 75 Data Latih

Laju Pembelajaran	Rata-rata RMSE
0.1	0.376099911555
0.2	0.375655294872
0.3	0.375655294873
0.4	0.375075421613
0.5	0.375075421612
0.6	0.375075421613
0.7	0.375655294872
0.8	0.299712413334
0.9	0.375075421613

Pada tabel dapat dilihat bahwa nilai RMSE yang paling kecil yang dihasilkan oleh pengujian ini adalah 0.29971241333 pada saat laju pembelajaran adalah 0,8. Sedangkan nilai RMSE yang paling besar adalah 0.37609991155 pada saat laju pembelajaran 0,1. Hasil dari pengujian laju pembelajaran untuk 110 data latih dapat dilihat pada tabel 4.3.

Tabel 4.3 Pengujian Laju Pembelajaran Untuk 110 Data Latih

Laju Pembelajaran	Rata-rata RMSE
0.1	0.3727075582402
0.2	0.3727075582401
0.3	0.3727075582413
0.4	0.3727075582383
0.5	0.3727075582359
0.6	0.3727075582357
0.7	0.3727075582342
0.8	0.2981660465901
0.9	0.3727075582380

Pada tabel dapat dilihat bahwa nilai RMSE yang paling kecil yang dihasilkan oleh pengujian ini adalah 0.2981660465901 pada saat laju pembelajaran adalah 0,8. Sedangkan nilai RMSE yang paling besar adalah 0.3727075582413 pada saat laju pembelajaran 0,3.

4.4.2 Uji Perbandingan Hasil Aktual dan Hasil ANFIS

Data latih pertama yang digunakan adalah 40 data latih. 40 data latih ini akan diuji sesuai dengan sistematika pengujian yang sebelumnya telah dipaparkan. Data latih ini diuji dan hasil output jaringan ANFISnya dibandingkan dengan nilai output aktual yang telah diperoleh. Hasil pengujian dengan menggunakan 40 data latih dapat dilihat pada tabel 4.4.

Tabel 4.4 Pengujian Kerawanan dengan 40 data latih

Output Aktual	Output ANFIS	Cluster Aktual	Cluster ANFIS
1	1	Rawan	Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan

1	1	Rawan	Rawan
1	1	Rawan	Rawan
2	1	Tidak Rawan	Rawan
2	1	Tidak Rawan	Rawan
2	1	Tidak Rawan	Rawan
2	2	Tidak Rawan	Tidak Rawan
2	2	Tidak Rawan	Tidak Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan
2	2	Tidak Rawan	Tidak Rawan
1	1	Rawan	Rawan
2	2	Tidak Rawan	Tidak Rawan
2	2	Tidak Rawan	Tidak Rawan
2	2	Tidak Rawan	Tidak Rawan
2	1	Tidak Rawan	Rawan
2	2	Tidak Rawan	Tidak Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan
2	2	Tidak Rawan	Tidak Rawan
2	2	Tidak Rawan	Tidak Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan
2	2	Tidak Rawan	Tidak Rawan
2	2	Tidak Rawan	Tidak Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan

2	1	Tidak Rawan	Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan
2	2	Tidak Rawan	Tidak Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan

Data yang diujikan sebanyak 40 data. Pada sistem telah ditentukan bahwa data akan dikelompokkan menjadi dua, yaitu rawan dan tidak rawan. Pada hasil pengujian dengan menggunakan 40 data latih, kesamaan pada pengelompokan data aktual dan pengelompokan output jaringan adalah 87,5%. Dimana data yang sama sebanyak 35 data dan yang tidak sama sebanyak 5 data.

Hasil pengujian kerawanan dengan data latih sebanyak 75 dapat dilihat pada tabel 4.5.

Tabel 4.5 Pengujian Kerawanan dengan 75 data latih

Output Aktual	Output ANFIS	Cluster Aktual	Cluster ANFIS
1	1	Rawan	Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan
2	1	Tidak Rawan	Rawan
2	1	Tidak Rawan	Rawan
2	1	Tidak Rawan	Rawan
2	2	Tidak Rawan	Tidak Rawan
2	2	Tidak Rawan	Tidak Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan
2	2	Tidak Rawan	Tidak Rawan
1	1	Rawan	Rawan

2	2	Tidak Rawan	Tidak Rawan
2	2	Tidak Rawan	Tidak Rawan
2	2	Tidak Rawan	Tidak Rawan
2	1	Tidak Rawan	Rawan
2	2	Tidak Rawan	Tidak Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan
2	2	Tidak Rawan	Tidak Rawan
2	2	Tidak Rawan	Tidak Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan
2	2	Tidak Rawan	Tidak Rawan
2	2	Tidak Rawan	Tidak Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan
2	1	Tidak Rawan	Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan
2	2	Tidak Rawan	Tidak Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan

Data yang diujikan sebanyak 40 data. Pada sistem telah ditentukan bahwa data akan dikelompokkan menjadi dua, yaitu rawan dan tidak rawan. Pada hasil pengujian dengan menggunakan 75 data latih, kesamaan pada pengelompokan data aktual dan pengelompokan output jaringan adalah 87.5%. Dimana data yang sama sebanyak 35 data dan yang tidak sama sebanyak 5 data.

Hasil pengujian kerawanan dengan data latih sebanyak 110 dapat dilihat pada tabel 4.6.

Tabel 4.6 Pengujian Kerawanan dengan 110 data latih

Output Aktual	Output ANFIS	Cluster Aktual	Cluster ANFIS
1	1	Rawan	Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan
2	2	Tidak Rawan	Tidak Rawan
2	2	Tidak Rawan	Tidak Rawan
2	2	Tidak Rawan	Tidak Rawan
2	2	Tidak Rawan	Tidak Rawan
2	2	Tidak Rawan	Tidak Rawan
1	2	Rawan	Tidak Rawan
1	2	Rawan	Tidak Rawan
2	2	Tidak Rawan	Tidak Rawan
1	1	Rawan	Rawan
2	2	Tidak Rawan	Tidak Rawan
2	2	Tidak Rawan	Tidak Rawan
2	2	Tidak Rawan	Tidak Rawan
2	2	Tidak Rawan	Tidak Rawan
2	2	Tidak Rawan	Tidak Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan
2	2	Tidak Rawan	Tidak Rawan

2	2	Tidak Rawan	Tidak Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan
2	2	Tidak Rawan	Tidak Rawan
2	2	Tidak Rawan	Tidak Rawan
1	1	Rawan	Rawan
1	2	Rawan	Tidak Rawan
2	1	Tidak Rawan	Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan
2	2	Tidak Rawan	Tidak Rawan
1	1	Rawan	Rawan
1	1	Rawan	Rawan

Data yang diujikan sebanyak 40 data. Pada sistem telah ditentukan bahwa data akan dikelompokkan menjadi dua, yaitu rawan dan tidak rawan. Pada hasil pengujian dengan menggunakan 110 data latih, kesamaan pada pengelompokan data aktual dan pengelompokan output jaringan adalah 90.0%. Dimana data yang sama sebanyak 36 data dan yang tidak sama sebanyak 4 data.

4.4.3 Uji Tingkat Akurasi

Seperti yang telah dipaparkan dalam sistematika pengujian, data latih yang digunakan adalah 40, 75 dan 110 data latih dan data pengujian sebanyak 40 data. Hasil pengujian tingkat akurasi kerawanan DBD dapat dilihat pada tabel 4.7.

Tabel 4.7 Pengujian Tingkat Akurasi Kerawanan

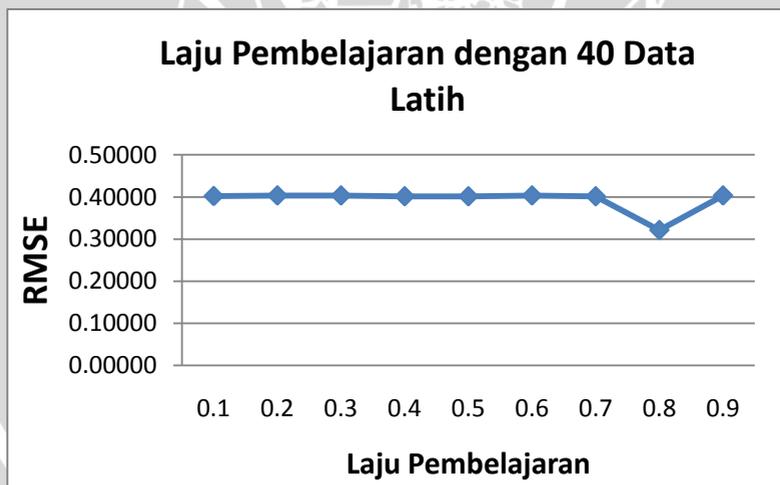
Data Latih	Pengujian ke-					Rata-Rata
	1	2	3	4	5	
40	85.0%	87.5%	85.0%	85.0%	87.5%	86.00%
75	87.5%	87.5%	85.0%	85.0%	85.0%	86.00%
110	90.0%	90.0%	90.0%	90.0%	90.0%	90.00%

Menggunakan 5 kali percobaan pengujian didapatkan nilai yang berbeda-beda. Pengujian tingkat akurasi yang dilakukan pada 40 data latih memiliki nilai rata-rata 86% dengan nilai tingkat akurasi yang berbeda-beda dari percobaan ke-1 sampai ke-5. Sedangkan untuk 75 data latih memiliki nilai rata-rata 86% dengan tingkat akurasi yang berbeda-beda dari percobaan ke-1 sampai ke-5. Pengujian yang terakhir adalah menggunakan 110 data latih, dimana nilai rata-rata tingkat akurasi adalah 90%. Tingkat akurasi dari 110 data latih ini memiliki nilai yang sama dari percobaan ke-1 sampai ke-5.

4.5 Analisa Hasil

4.5.1 Analisa Hasil Laju Pembelajaran

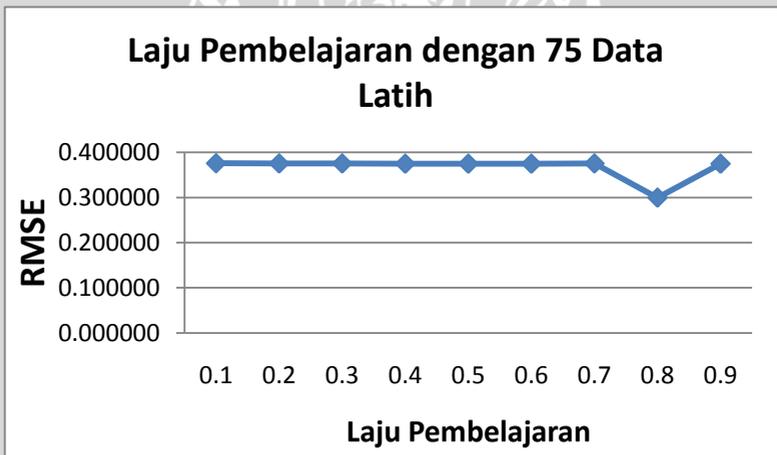
Pada uji coba pertama yang menggunakan 40 data latih diketahui bahwa nilai RMSE yang dihasilkan oleh tiap-tiap laju pembelajaran memiliki nilai yang berbeda-beda. Nilai RMSE yang paling kecil berada pada saat laju pembelajaran 0,8. Sedangkan nilai RMSE yang paling besar terdapat pada saat laju pembelajaran 0,9. Grafik pengaruh laju pembelajaran terhadap nilai RMSE dengan 40 data latih dapat dilihat pada gambar 4.7.



Gambar 4.7 Grafik Pengaruh Laju Pembelajaran terhadap RMSE dengan Menggunakan 40 Data Latih

Pada gambar 4.7 ditunjukkan bahwa nilai laju pembelajaran yang kecil, belum tentu memiliki nilai error yang kecil pula. Grafik nilai error ini memiliki proses kenaikan dan penurunan yang cenderung stabil saat pembelajaran 0.1 sampai 0.7 dan saat 0.8 terjadi penurunan yang cukup drastis, kemudian pada 0.9 terjadi kenaikan kembali. Pada data latih 40 ini, nilai pembelajaran yang relative besar yaitu 0,8 memiliki nilai RMSE yang paling kecil yaitu 0.321283008500. Sehingga untuk memaksimalkan penggunaan sistem penentuan kerawanan, maka pada saat melakukan prediksi digunakan laju pembelajaran 0,8.

Pada percobaan kedua yang menggunakan 75 data latih. Nilai RMSE yang paling kecil berada pada saat laju pembelajaran 0,8. Sedangkan nilai RMSE yang paling besar terdapat pada saat laju pembelajaran 0,1. Grafik pengaruh laju pembelajaran terhadap RMSE dengan 75 data uji dapat dilihat pada gambar 4.8.

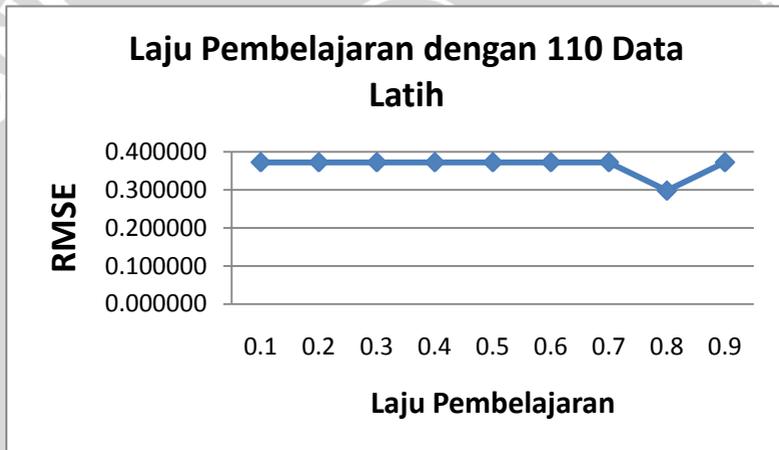


Gambar 4.8 Grafik Pengaruh Laju Pembelajaran terhadap RMSE dengan Menggunakan 75 Data Latih

Pada gambar 4.8 ditunjukkan bahwa nilai laju pembelajaran yang kecil, belum tentu memiliki nilai error yang kecil pula. Grafik nilai error ini memiliki proses kenaikan dan penurunan yang cenderung stabil saat pembelajaran 0.1 sampai 0.7 dan saat 0.8 terjadi penurunan yang cukup drastis, kemudian pada 0.9 terjadi

kenaikan kembali. Pada data latih 75 ini, memiliki pola grafik yang tidak jauh berbeda dengan grafik laju pembelajaran yang menggunakan 40 data latih. Nilai pembelajaran untuk 75 data latih, relative besar yaitu 0,8 yang memiliki nilai RMSE yang paling kecil yaitu 0.29971241333. Sehingga untuk memaksimalkan penggunaan sistem penentuan kerawanan, maka pada saat melakukan prediksi digunakan laju pembelajaran 0,8.

Pada percobaan ketiga menggunakan data latih sebanyak 110 data. Nilai RMSE yang paling kecil berada pada saat laju pembelajaran 0,8. Sedangkan nilai RMSE yang paling besar terdapat pada saat laju pembelajaran 0,3. Grafik pengaruh laju pembelajaran terhadap RMSE dengan 110 data latih dapat dilihat pada gambar 4.9.



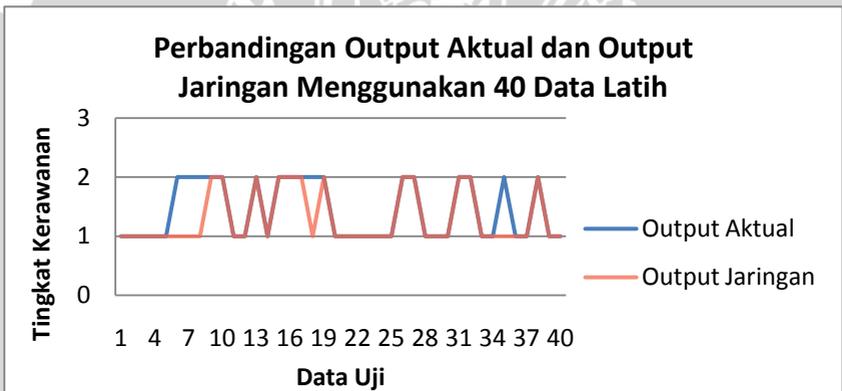
Gambar 4.9 Grafik Pengaruh Laju Pembelajaran terhadap RMSE dengan Menggunakan 110 Data Latih

Pada gambar 4.9 ditunjukkan bahwa nilai laju pembelajaran yang kecil, belum tentu memiliki nilai error yang kecil pula. Grafik nilai error ini memiliki proses kenaikan dan penurunan yang cenderung stabil saat pembelajaran 0.1 sampai 0.7 dan saat 0.8 terjadi penurunan yang cukup drastis, kemudian pada 0.9 terjadi kenaikan kembali. Pada data latih 110 ini, memiliki pola grafik yang tidak jauh berbeda dengan grafik laju pembelajaran yang menggunakan 40 data latih dan 75 data latih. Nilai pembelajaran untuk 75 data latih, relative besar yaitu 0.8 yang memiliki nilai

RMSE yang paling kecil yaitu 0.2981660465901. Sehingga untuk memaksimalkan penggunaan sistem penentuan kerawanan, maka pada saat melakukan prediksi digunakan laju pembelajaran 0.8.

4.5.2 Analisa Hasil Penentuan Kerawanan DBD

Pengujian hasil prediksi kerawanan dengan menggunakan data latih 40, 75 dan 110. Laju pembelajaran yang digunakan adalah laju pembelajaran yang paling baik pada tiap-tiap data latih, yaitu laju pembelajaran 0,8. Data pengujian yang digunakan adalah sebanyak 40 data dengan membandingkan nilai output jaringan dan data aktual. Dari pengujian didapatkan bahwa semakin banyak data pelatihan yang digunakan, maka semakin banyak kesamaan antara output jaringan dengan target output yang diberikan. Grafik hasil prediksi kerawanan dengan menggunakan 40 data latih dapat dilihat pada gambar 4.10

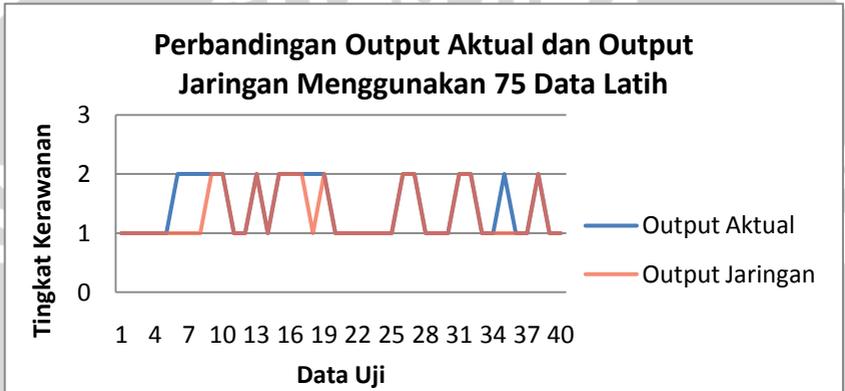


Gambar 4.10 Grafik Perbandingan Output Aktual dan Output Jaringan Menggunakan 40 Data Latih

Pada gambar 4.10 ditunjukkan bahwa output jaringan masih belum bisa mengikuti pola dari naik dan turunnya output aktual. Hal ini terlihat pada beberapa data yang berbeda antara nilai kerawanan output aktual dan nilai kerawanan output jaringan. Pada output aktual pertama yang memiliki nilai kerawanan 1 mengalami perbedaan dengan output jaringannya yang memiliki nilai kerawanan 2, tetapi pada output aktual kedua nilai kerawanan sama dengan nilai

kerawanan output jaringan. Pada data terakhir, output jaringan mengikuti nilai dari data aktual. Perbedaan nilai kerawanan antara output aktual dan output jaringan berada pada data uji ke-1, ke-5, ke-6, ke-18 dan ke-36, sehingga kebenaran rawan dan tidak rawan sebesar 87,5%. Dimana data yang sama sebanyak 35 data dan yang tidak sama sebanyak 5 data.

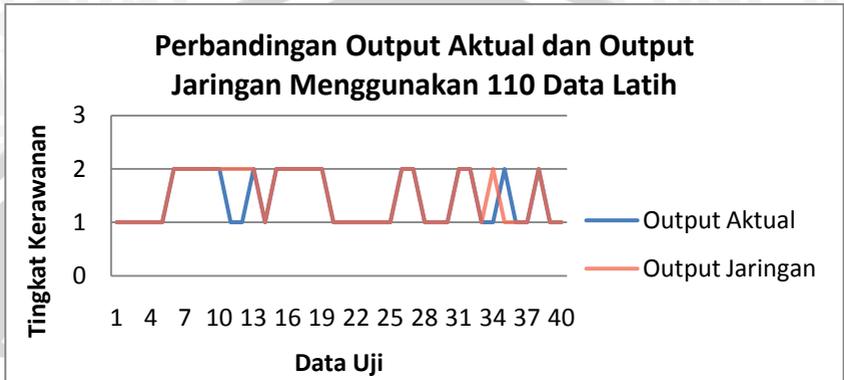
Grafik hasil prediksi kerawanan dengan menggunakan 75 data latih dapat dilihat pada gambar 4.11



Gambar 4.11 Grafik Perbandingan Output Aktual dan Output Jaringan Menggunakan 75 Data Latih

Pada gambar 4.11 ditunjukkan bahwa output jaringan masih belum bisa mengikuti pola dari naik dan turunnya output aktual. Hal ini terlihat pada beberapa data yang berbeda antara nilai kerawanan output aktual dan nilai kerawanan output jaringan. Pada output aktual pertama yang memiliki nilai kerawanan 1 mengalami perbedaan dengan output jaringannya yang memiliki nilai kerawanan 2, tetapi pada output aktual kedua nilai kerawanan sama dengan nilai kerawanan output jaringan. Pada data terakhir, output jaringan mengikuti nilai dari data aktual. Perbedaan nilai kerawanan antara output aktual dan output jaringan berada pada data uji ke-1, ke-5, ke-6, ke-18 dan ke-36, sehingga kebenaran rawan dan tidak rawan sebesar 87,5%. Dimana data yang sama sebanyak 35 data dan yang tidak sama sebanyak 5 data.

Grafik hasil prediksi kerawanan dengan menggunakan 110 data latih dapat dilihat pada gambar 4.1

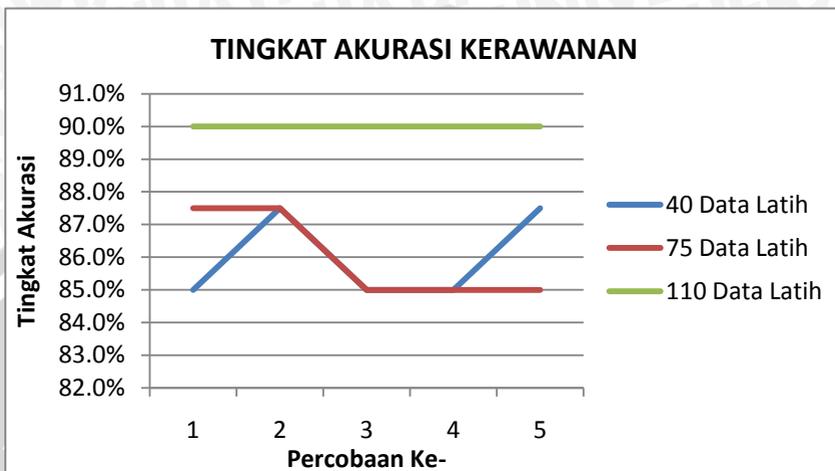


Gambar 4.12 Grafik Perbandingan Output Aktual dan Output Jaringan Menggunakan 110 Data Latih

Pada gambar 4.12 ditunjukkan bahwa output jaringan bisa mengikuti pola dari naik dan turunnya output aktual. Hal ini terlihat pada beberapa data yang memiliki kesamaan nilai kerawanan output aktual dan nilai kerawanan output jaringan. Pada output aktual pertama nilai kerawanannya sama antara output aktual dan output jaringan yaitu 1, tetapi masih terdapat perbedaan antara nilai kerawanan yang dihasilkan output aktual dengan nilai kerawanan output jaringan. Pada data terakhir, output jaringan mengikuti nilai dari data actual yaitu 1. Kebenaran rawan dan tidak rawan sebesar 90%. Dimana data yang sama sebanyak 36 data dan yang tidak sama sebanyak 4 data.

4.5.3 Analisa Hasil Tingkat Akurasi Kerawanan DBD

Pengujian tingkat akurasi kerawanan dengan cara melakukan pengujian sebanyak 5 kali pada masing-masing data latih. Grafik tingkat akurasi kerawanan terlihat pada gambar 4.13.



Gambar 4.13 Tingkat Akurasi Prediksi Kerawanan DBD

Pada gambar 4.13 terlihat bahwa tingkat akurasi prediksi kerawanan dengan menggunakan 40 data latih dan 75 data latih mengalami perubahan pada tiap percobaan. Sehingga rata-rata tingkat akurasinya sama yaitu 86% dan 86%. Perubahan ini terjadi karena pengambilan nilai centroid untuk clustering K-Mean dibangkitkan secara random, sehingga pada saat melakukan pengujian, nilai kebenarannya dapat berubah-ubah yang mempengaruhi tingkat akurasi. Pada data latih 110 tingkat akurasi tidak mengalami berubah dari percobaan pertama hingga terakhir, sehingga rata-rata tingkat akurasinya yaitu 90%.

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang diperoleh dari skripsi ini adalah:

1. Metode ANFIS dapat diimplementasikan untuk penentuan kerawanan. Proses awal yang dilakukan adalah mengcluster data latih dengan menggunakan metode menclusteran K-Means, kemudian melakukan metode ANFIS dengan menggunakan rule fuzzy berbasis metode Sugeno orde satu dan diimplementasikan dalam bentuk jaringan syaraf tiruan dengan 5 layer yang terdiri dari 3 layer tetap dan 2 layer adaptif. Layer 1 parameter premis diperbaiki menggunakan *steepest descent* dan layer 4 parameter konsekuensi yang nilainya diperbaiki menggunakan *LSE*. Hasil parameter konsekuensi yang telah melalui tahap pengoptimalan yang dihasilkan akan digunakan untuk menghitung kerawanan DBD.
2. Laju pembelajaran yang paling optimal yang telah dicoba dengan menggunakan tiga data latih yang berbeda-beda, yaitu laju pembelajaran 0,8. Sehingga untuk kasus penentuan kerawanan DBD yang berbeda dapat digunakan laju pembelajaran 0,8 agar mendapatkan hasil yang optimal dan error yang kecil.
3. Tingkat akurasi yang diuji coba dengan menggunakan 40 data uji dan tiga data latih yang berbeda-beda yaitu 40 data, 75 data dan 110 data, memiliki tingkat akurasi sebesar 86.0%, 86.0%, dan 90%.

5.2 Saran

Untuk pengembangan lanjut perangkat lunak maka ada beberapa saran yang dapat diberikan :

1. Untuk pengembangan selanjutnya dapat digunakan metode cluster lainnya selain cluster K-Mean karena K-Means sangat bergantung pada pemilihan nilai awal centroid, apabila saat

pemilihan awal mendapatkan random yang bagus maka pada jaringan ANFIS menghasilkan nilai error yang kecil.

2. Untuk meningkatkan tingkat akurasi, dapat melakukan penelitian lebih lanjut terhadap faktor-faktor lain yang mempengaruhi kerawanan DBD pada suatu daerah. Faktor-faktor tersebut dapat digunakan sebagai parameter dari ANFIS.

UNIVERSITAS BRAWIJAYA



DAFTAR PUSTAKA

- Djunaedi, Djoni. 2006. *Demam Berdarah Dengue*. UMM-Press, Malang
- Gubler, D.J. 1988. *The arbovirus epidemiology and ecology*. Dengue Newsletter 2 : 45 – 52
- Hasan, R. 1985. *Buku Ajar Ilmu Kesehatan Anak*. Bagian Ilmu Kesehatan anak Fakultas Kedokteran Universitas Indonesia. Jakarta.
- Hendarwanto.1987. *Buku Ajar Ilmu Penyakit Dalam*. Balai Penerbit Fakultas Kedokteran Universitas Indonesia. Jakarta.
- Jang J.-S.R., Sun C.-T., Mizutani E .1997. *Neuro-Fuzzy And Soft Computing: A Computational Approach To Learning And Machine Intelligence*. Prentice-Hall, Inc.
- Kusumadewi. S dan H. Purnomo. 2004. *Aplikasi Logika Fuzzy Untuk Mendukung Keputusan*. Graha Ilmu, Yogyakarta.
- Kusumadewi. S. 2003. *Artificial Intelligence (Tehnik dan aplikasinya)*. Graha Ilmu, Yogyakarta.
- Kusumadewi, Sri dan Sri Hartati. 2006. *NEURO FUZZY: Integrasi Sistem Fuzzy & Jaringan Syaraf*. Graha Ilmu. Yogyakarta
- Marimin. 2005. *Teori dan aplikasi sistem pakar dalam tehnologi manajerial*. IPB –Press, Bogor.
- Nugraha, Dany, dkk. 2006. *Diagnosis Gangguan Sistem Urinari pada Anjing dan Kucing Menggunakan VFI 5*. Institut Pertanian Bogor

- Prompou, S., M. Jaroensutasinee, dan K. Jaroensutasinee. 2005. *Climatic Factors Affecting Dengue Haemorrhagic Fever Incident in Southern Thailand*. Dengue Bulletin, [online], Vol. 29, pp 41-48. Dari: [http://who.int/LinkFiles/Dengue Bulletin/](http://who.int/LinkFiles/Dengue_Bulletins/). Diakses tanggal: 13 Mei 2011.
- Puspitaningrum, Diyah. 2006. *Jaringan Syaraf Tiruan*. Penerbit ANDI. Yogyakarta.
- Riyanto, B., Febrianto, F., and Machbub, C. 2000. *Adaptive network based Fuzzy Inference System for forecasting daily gasoline demand. Proceeding of the Sixth AEESEAP Triennial conference*. Kuta, Bali, Indonesia. August 23-25, 2000
- Ruminta. 2001. *Pendugaan Curah Hujan Di Wilayah Sumatra Dengan Menggunakan ANFIS*. Tesis Program Master, Institut Teknologi Bandung.
- Salehfar, H., Bengiamin, N., and Huang, J. 2000. *A Systematic approach to linguistic fuzzy modeling based on input-output data*. Proceedings of the 2000 Winter Simulation Conference. J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, Eds., University of North Dakota, U.S.A.
- Santosa, B. 2007. *Data Mining: Teknik pemanfaatan data untuk keperluan bisnis*. Graha Ilmu: Jogjakarta.
- Shapiro, A F. 2002 *From Neural Networks, Fuzzy Logic, and Genetic Algorithms to ANFIS and Beyond*. A Proposal for the American Risk and Insurance Association 2002 Annual Meeting. Universitas Park, USA.
- Soedarto. 1995. *Penyakit-Penyakit Infeksi Di Indonesia*. Widya Medika, Surabaya
- Soegijanto S. 2003. Prospek *Pemanfaatan Vaksin Dengue untuk menurunkan prevalensi di masyarakat*. Dipresentasikan di

Peringatan 90 tahun Pendidikan Dokter di FK Unair.Surabaya.

- Subekti, M. R., Achmad, B dan Suyitno, G. 1999. *Analisi Kondisi Ginjal Pasien Menggunakan Metode Jaringan Saraf Tiruan*. Proceeding Lokakarya Komputasi dan Sains Nuklir X, Batan.
- Suroso, T. 2001. *Perubahan Iklim dan Kejadian Penyakit Yang Ditularkan Vektor*. Makalah Pada Semiloka Perubahan iklim dan Keasehatan 27-29 Maret 2001 di Ciloto.
- Suroso. T. 1999. *Epidemiological situation of dengue hemorrhagic fever and its control in Indonesi*. Seminar on Dengue Fever/ Dengue Hemorrhagic Fever, Surabaya
- Syafii, Muhammad. 2006. *Adaptive neuro fuzzy inference system (anfis) untuk diagnosa dan tatalaksana Penyakit demam berdarah dengue*. Tesis S2. Departemen Ilmu Komputer Institut Pertanian Bogor. Bogor. Dipublikasikan.
- Tjasyono B. 1995. *Klimatologi Umum*. Penerbit ITB, Bandung.
- Viani Srituka. 2009. *Penelitian Penyebaran Penyakit Demam Berdarah di Wilayah Kodya Surabaya dengan Menggunakan Spatial Statistics*. Skripsi S1. Universitas Petra. Surabaya. Dipublikasikan.
- Witten, Ian H. dan Frank, Eibe. 2005. *Data Mining Practical Machine Learning Tools and Techniques, Second Edition*. Morgan Kaufmann, San Fransisco.
- Zhu, Y. 2000. *ANFIS : Adaptive Neuro Fuzzy Inference System*. EE Dept. Univ. of Missouri, Rolla.

LAMPIRAN

Lampiran 1. Data latihan 40

Kelembaban	Kecepatan Angin	Suhu	Curah Hujan	Kerawanan
70	4.5	27.8	8.9	Rawan
76	2.9	27.6	0.0	Tidak rawan
83	2.1	27.1	15.3	Rawan
73	3.4	29.0	0.0	Tidak rawan
82	1.3	27.2	4.0	Rawan
78	2.1	27.9	21.1	Rawan
79	2.2	27.6	5.3	Rawan
83	1.1	27.2	2.0	Rawan
84	2.3	27.4	7.2	Rawan
82	2.8	26.8	9.8	Rawan
83	2.4	26.3	0.5	Rawan
88	2.9	26.3	25.4	Rawan
77	2.2	28.1	2.3	Tidak rawan
73	2.6	28.0	0.0	Tidak rawan
75	2.7	28.5	0.2	Tidak rawan
74	2.9	28.9	0.0	Tidak rawan
75	2.2	29.2	0.2	Tidak rawan
71	2.3	29.2	0.6	Tidak rawan
73	2.4	29.5	3.1	Tidak rawan
75	2.3	29.0	0.0	Tidak rawan
76	1.9	29.1	1.8	Tidak rawan
82	1.6	27.0	7.7	Rawan
88	1.8	26.6	8.4	Rawan
84	2.1	26.4	10.5	Rawan
78	2.4	27.2	14.4	Rawan
84	2.3	27.1	2.2	Rawan
82	4.9	27.5	9.7	Rawan
82	2.1	26.9	2.7	Rawan
80	2.0	27.7	0.4	Tidak rawan
79	2.0	28.0	5.1	Tidak rawan
84	3.1	25.4	0.3	Tidak rawan

84	2.6	26.5	0.2	Tidak rawan
81	2.9	26.2	0.9	Rawan
79	2.3	26.5	0.0	Tidak rawan
74	3.4	26.5	0.0	Tidak rawan
75	2.7	26.4	3.3	Rawan
81	3.5	27.2	0.7	Rawan
78	2.8	28.3	0.1	Tidak rawan
79	2.8	28.1	3.3	Tidak rawan
79	2.4	27.5	7.0	Rawan

Lampiran 2. Data latihan 75

Kelembaban	Kecepatan Angin	Suhu	Curah Hujan	Kerawanan
70	4.5	27.8	8.9	Rawan
76	2.9	27.6	0.0	Tidak rawan
83	2.1	27.1	15.3	Rawan
73	3.4	29.0	0.0	Tidak rawan
82	2.8	26.8	9.8	Rawan
83	2.4	26.3	0.5	Rawan
88	2.9	26.3	25.4	Rawan
77	2.2	28.1	2.3	Tidak rawan
73	2.6	28.0	0.0	Tidak rawan
75	2.7	28.5	0.2	Tidak rawan
74	2.9	28.9	0.0	Tidak rawan
75	2.2	29.2	0.2	Tidak rawan
71	2.3	29.2	0.6	Tidak rawan
73	2.4	29.5	3.1	Tidak rawan
75	2.3	29.0	0.0	Tidak rawan
76	1.9	29.1	1.8	Tidak rawan
82	1.6	27.0	7.7	Rawan
88	1.8	26.6	8.4	Rawan
84	2.1	26.4	10.5	Rawan
78	2.4	27.2	14.4	Rawan
84	2.3	27.1	2.2	Rawan
82	4.9	27.5	9.7	Rawan
82	2.1	26.9	2.7	Rawan
80	2.0	27.7	0.4	Tidak rawan

79	2.0	28.0	5.1	Tidak rawan
84	3.1	25.4	0.3	Tidak rawan
84	2.6	26.5	0.2	Tidak rawan
81	2.9	26.2	0.9	Rawan
79	2.3	26.5	0.0	Tidak rawan
74	3.4	26.5	0.0	Tidak rawan
75	2.7	26.4	3.3	Rawan
81	3.5	27.2	0.7	Rawan
78	2.8	28.3	0.1	Tidak rawan
79	2.8	28.1	3.3	Tidak rawan
79	2.4	27.5	7.0	Rawan
85	2.3	25.9	0.0	Tidak rawan
78	2.9	25.6	0.4	Tidak rawan
79	3.6	25.1	1.5	Rawan
78	2.5	24.6	0.0	Rawan
86	2.0	26.6	5.2	Rawan
79	1.9	26.7	1.1	Rawan
80	3.2	26.6	0.0	Tidak rawan
81	1.4	26.9	2.3	Rawan
84	1.3	26.5	11.5	Rawan
83	1.9	26.3	7.7	Rawan
86	2.0	26.6	5.2	Rawan
81	1.8	26.4	17.6	Rawan
78	2.1	26.7	9.7	Rawan
80	2.0	27.3	1.9	Rawan
83	2.9	25.4	3.2	Rawan
82	3.0	25.3	2.1	Rawan
84	3.1	25.4	0.2	Tidak rawan
81	2.1	24.2	1.7	Rawan
82	2.4	25.1	4.1	Rawan
70	2.5	25.4	0.0	Tidak rawan
80	2.3	27.3	2.8	Rawan
76	1.4	27.4	2.0	Rawan
76	2.2	27.3	1.0	Rawan
78	1.9	27.9	6.1	Tidak rawan
78	1.8	27.7	14.9	Rawan

76	1.7	27.5	1.8	Rawan
79	2.1	27.0	2.5	Rawan
82	1.3	27.2	4.0	Rawan
78	2.1	27.9	21.1	Rawan
79	2.2	27.6	5.3	Rawan
83	1.1	27.2	2.0	Rawan
84	2.3	27.4	7.2	Rawan
77	2.1	28.1	0.3	Tidak rawan
68	3.6	27.1	0.0	Tidak rawan
71	2.7	28.2	2.7	Tidak rawan
77	1.7	27.7	0.9	Tidak rawan
83	1.7	26.6	5.3	Rawan
74	2.0	26.2	1.3	Rawan
72	2.5	27.2	0.2	Tidak rawan
74	1.7	28.0	0.3	Tidak rawan

Lampiran 3. Data latihan 110

Kelembaban	Kecepatan Angin	Suhu	Curah Hujan	Kerawanan
83	1.6	26.5	3.2	Rawan
80	2.3	27.3	2.8	Rawan
82	2.0	27.1	3.8	Rawan
84	1.3	26.5	11.5	Rawan
81	1.8	26.4	17.6	Rawan
74	2.0	28.4	0.0	Tidak rawan
82	1.6	27.0	7.7	Rawan
88	1.8	26.6	8.4	Rawan
84	2.1	26.4	10.5	Rawan
78	2.4	27.2	14.4	Rawan
84	2.3	27.1	2.2	Rawan
82	4.9	27.5	9.7	Rawan
82	2.1	26.9	2.7	Rawan
80	2.0	27.7	0.4	Tidak rawan
79	2.0	28.0	5.1	Tidak rawan
74	2.2	27.8	0.4	Tidak rawan
81	1.5	27.3	11.5	Rawan
75	1.9	27.1	0.9	Rawan

80	1.6	27.0	1.2	Rawan
83	1.6	27.4	5.4	Rawan
80	2.2	26.8	0.4	Tidak rawan
80	3.0	25.6	4.9	Rawan
83	2.9	25.4	3.2	Rawan
82	3.0	25.3	2.1	Rawan
84	3.1	25.4	0.2	Tidak rawan
81	2.1	24.2	1.7	Rawan
82	2.4	25.1	4.1	Rawan
82	2.6	25.1	1.9	Rawan
72	3.3	25.4	0.0	Tidak rawan
76	2.6	25.3	0.4	Tidak rawan
79	2.5	26.0	0.2	Tidak rawan
76	2.4	25.4	0.1	Tidak rawan
70	2.5	25.4	0.0	Tidak rawan
77	2.7	26.3	0.2	Tidak rawan
72	3.1	26.2	0.0	Tidak rawan
69	3.7	26.5	0.0	Tidak rawan
73	3.0	27.1	0.0	Tidak rawan
71	2.4	27.3	4.3	Rawan
77	2.2	28.1	2.3	Tidak rawan
73	2.6	28.0	0.0	Tidak rawan
75	2.7	28.5	0.2	Tidak rawan
74	2.9	28.9	0.0	Tidak rawan
75	2.2	29.2	0.2	Tidak rawan
71	2.3	29.2	0.6	Tidak rawan
73	2.4	29.5	3.1	Tidak rawan
75	2.3	29.0	0.0	Tidak rawan
76	1.9	29.1	1.8	Tidak rawan
83	2.5	27.3	15.8	Rawan
70	4.5	27.8	8.9	Tidak rawan
76	2.9	27.6	0.0	Tidak rawan
83	2.1	27.1	15.3	Rawan
73	3.4	29.0	0.0	Tidak rawan
82	1.3	27.2	4.0	Rawan
78	2.1	27.9	21.1	Rawan

79	2.2	27.6	5.3	Rawan
83	1.1	27.2	2.0	Rawan
84	2.3	27.4	7.2	Rawan
82	2.8	26.8	9.8	Rawan
83	2.4	26.3	0.5	Rawan
88	2.9	26.3	25.4	Rawan
79	2.2	27.3	8.2	Rawan
83	1.5	27.4	13.8	Rawan
80	1.8	27.1	0.3	Tidak rawan
80	2.6	27.3	0.0	Tidak rawan
78	2.3	27.6	0.1	Tidak rawan
80	1.9	26.9	0.0	Tidak rawan
82	1.8	27.1	2.9	Rawan
80	1.6	26.9	0.3	Tidak rawan
82	1.8	27.3	1.2	Rawan
78	3.5	26.2	0.0	Tidak rawan
85	2.0	25.5	7.2	Rawan
86	2.1	25.9	7.9	Rawan
81	2.5	25.3	0.4	Tidak rawan
82	2.1	25.6	0.3	Tidak rawan
83	2.2	26.0	0.0	Tidak rawan
79	3.0	27.0	4.3	Rawan
78	2.2	27.2	4.2	Rawan
86	1.8	26.4	15.5	Rawan
81	1.8	26.8	14.1	Rawan
81	2.5	26.7	7.7	Rawan
82	1.3	27.2	4.0	Rawan
83	2.5	25.6	0.7	Rawan
84	3.1	25.4	0.3	Tidak rawan
84	2.6	26.5	0.2	Tidak rawan
81	2.9	26.2	0.9	Rawan
79	2.3	26.5	0.0	Tidak rawan
74	3.4	26.5	0.0	Tidak rawan
75	2.7	26.4	3.3	Rawan
81	3.5	27.2	0.7	Rawan
78	2.8	28.3	0.1	Tidak rawan

79	2.8	28.1	3.3	Tidak rawan
79	2.4	27.5	7.0	Rawan
86	2.0	26.6	5.2	Rawan
72	2.5	27.2	0.2	Tidak rawan
76	3.1	27.3	0.0	Tidak rawan
76	1.4	27.4	2.0	Rawan
81	1.4	26.9	2.3	Rawan
79	2.1	27.0	2.5	Rawan
78	1.9	27.9	6.1	Tidak rawan
82	2.5	26.3	9.3	Rawan
82	2.6	26.9	2.6	Rawan
82	2.2	27.0	10.8	Rawan
78	1.7	27.5	1.6	Rawan
84	1.4	26.7	11.8	Rawan
81	2.4	27.0	6.1	Rawan
85	1.6	26.1	12.6	Rawan
83	2.0	26.1	7.1	Rawan
80	2.0	26.1	7.5	Rawan
83	2.4	25.8	17.3	Rawan
83	1.9	26.3	7.7	Rawan

Lampiran 4. Data uji 40

Kelembaban	Kecepatan Angin	Suhu	Curah Hujan	Kerawanan
78	1.9	25.4	0.0	Rawan
78	2.7	24.8	5.0	Rawan
83	2.3	25.5	2.0	Rawan
84	2.8	25.3	1.8	Rawan
76	2.2	24.7	0.0	Rawan
80	2.5	25.9	0.0	Tidak rawan
72	2.6	25.0	0.1	Tidak rawan
75	2.2	26.2	0.0	Tidak rawan
73	2.3	26.1	0.0	Tidak rawan
75	1.8	27.4	0.0	Tidak rawan
75	1.8	26.9	0.5	Rawan
76	1.6	27.4	1.1	Rawan

74	2.0	27.8	0.0	Tidak rawan
82	1.8	27.2	5.3	Rawan
77	2.1	28.1	0.3	Tidak rawan
68	3.6	27.1	0.0	Tidak rawan
71	2.7	28.2	2.7	Tidak rawan
77	1.7	27.7	0.9	Tidak rawan
74	1.7	28.0	0.3	Tidak rawan
83	1.7	26.6	5.3	Rawan
77	2.9	26.5	4.7	Rawan
77	2.0	27.2	2.5	Rawan
78	1.7	27.5	1.6	Rawan
81	2.4	27.0	6.1	Rawan
86	2.0	26.6	5.2	Rawan
72	2.5	27.2	0.2	Tidak rawan
76	3.1	27.3	0.0	Tidak rawan
81	1.4	26.9	2.3	Rawan
79	2.1	27.0	2.5	Rawan
86	2.0	26.6	5.2	Rawan
72	2.5	27.2	0.2	Tidak rawan
76	3.1	27.3	0.0	Tidak rawan
76	1.4	27.4	2.0	Rawan
76	2.2	27.3	1.0	Rawan
78	1.9	27.9	6.1	Tidak rawan
76	1.7	27.5	1.8	Rawan
86	2.0	26.6	5.2	Rawan
74	1.7	28.0	0.3	Tidak rawan
73	5.5	27.5	7.6	Rawan
79	1.9	26.7	1.1	Rawan

Lampiran 5. Pengujian Laju Pembelajaran Untuk 40 Data Latih

Laju P.	Percobaan					Rata-rata
	1	2	3	4	5	
0.1	0.400390248556	0.410110499945	0.400390248556	0.400390248552	0.397957147019	0.401847678526
0.2	0.397957146990	0.400390248559	0.410110499942	0.397957147029	0.410110499947	0.403305108493
0.3	0.410110499945	0.397957147023	0.410110499944	0.400390248530	0.397957147022	0.403305108493
0.4	0.397957147012	0.397957147043	0.400390248556	0.410110499944	0.400390248555	0.401361058222
0.5	0.400390248548	0.397957147020	0.397957147027	0.400390248549	0.410110499938	0.401361058216
0.6	0.397957147042	0.397957147019	0.400390248550	0.410110499935	0.410110499945	0.403305108498
0.7	0.400390248560	0.410110499941	0.397957146991	0.397957147020	0.400390248498	0.401361058202
0.8	0.400390248542	0.410110499950	0.397957147002	0.400390248572	0.397957146978	0.321283008500
0.9	0.397957147056	0.397957146988	0.410110499945	0.400390248543	0.410110499961	0.403305108499

Lampiran 6. Pengujian Laju Pembelajaran Untuk 75 Data Latih

Laju P.	Percobaan					Rata-rata
	1	2	3	4	5	
0.1	0.373915675092	0.373915675092	0.376815041393	0.379038124804	0.376815041393	0.376099911555
0.2	0.376815041393	0.373915675092	0.373915675092	0.376815041393	0.376815041393	0.375655294872
0.3	0.373915675093	0.376815041393	0.376815041393	0.373915675093	0.376815041393	0.375655294873

0.4	0.373915675092	0.376815041393	0.376815041393	0.373915675094	0.373915675092	0.375075421613
0.5	0.376815041393	0.373915675091	0.376815041393	0.373915675091	0.373915675091	0.375075421612
0.6	0.373915675092	0.376815041393	0.376815041392	0.373915675094	0.373915675092	0.375075421613
0.7	0.373915675090	0.376815041393	0.376815041392	0.373915675093	0.376815041392	0.375655294872
0.8	0.373915675093	0.376815041393	0.373915675093	0.373915675093	0.373915675093	0.299712413334
0.9	0.373915675092	0.373915675093	0.376815041393	0.376815041394	0.373915675093	0.375075421613

Lampiran 7. Pengujian Laju Pembelajaran Untuk 110 Data Latih

Laju P.	Percobaan					Rata-rata
	1	2	3	4	5	
0.1	0.372707558240	0.372707558240	0.372707558240	0.372707558240	0.372707558240	0.3727075582402
0.2	0.372707558240	0.372707558240	0.372707558240	0.372707558240	0.372707558240	0.3727075582401
0.3	0.372707558241	0.372707558242	0.372707558241	0.372707558241	0.372707558242	0.3727075582413
0.4	0.372707558236	0.372707558241	0.372707558241	0.372707558236	0.372707558236	0.3727075582383
0.5	0.372707558237	0.372707558235	0.372707558237	0.372707558237	0.372707558235	0.3727075582359
0.6	0.372707558236	0.372707558235	0.372707558235	0.372707558236	0.372707558236	0.3727075582357
0.7	0.372707558234	0.372707558234	0.372707558234	0.372707558234	0.372707558234	0.3727075582342
0.8	0.372707558233	0.372707558242	0.372707558242	0.372707558233	0.372707558233	0.2981660465901
0.9	0.372707558239	0.372707558237	0.372707558239	0.372707558237	0.372707558239	0.3727075582380

UNIVERSITAS BRAWIJAYA

