

**FUZZY CLUSTERING BACKPROPAGATION NEURAL NETWORKS  
UNTUK PREDIKSI PERKEMBANGAN PENYAKIT DEMAM  
BERDARAH SEBAGAI PENDUKUNG KEPUTUSAN**

**SKRIPSI**

Untuk memenuhi sebagian persyaratan untuk mencapai gelar Sarjana Komputer



Disusun Oleh:

**MOHAMAD ILHAM UBADILLAH**

**NIM. 0910680083**

**PROGRAM STUDI TEKNIK INFORMATIKA  
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER**

**UNIVERSITAS BRAWIJAYA**

**MALANG**

**2013**

## LEMBAR PERSETUJUAN

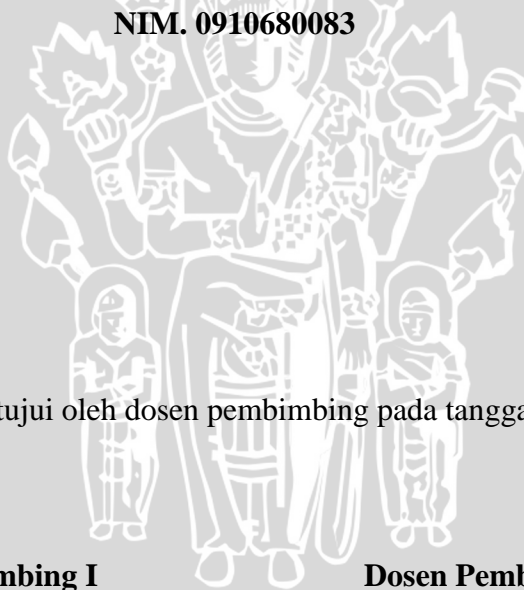
***FUZZY CLUSTERING BACKPROPAGATION NEURAL NETWORKS***  
**UNTUK PREDIKSI PERKEMBANGAN PENYAKIT DEMAM**  
**BERDARAH SEBAGAI PENDUKUNG KEPUTUSAN**

UNIVERSITAS BRAWIJAYA

Disusun Oleh:

**MOHAMAD ILHAM UBaidillah**

**NIM. 0910680083**



Skripsi ini telah disetujui oleh dosen pembimbing pada tanggal 2 Januari 2014

**Dosen Pembimbing I**

**Dosen Pembimbing II**

**Rekyan Regasari M P, ST., MT.**

**Ahmad Afif Supianto, SSi., M.Kom**

**NIP. 77041406120253**

**NIP. 820623 16 1 1 0425**

## LEMBAR PENGESAHAN

### ***FUZZY CLUSTERING BACKPROPAGATION NEURAL NETWORKS*** **UNTUK PREDIKSI PERKEMBANGAN PENYAKIT DEMAM** **BERDARAH SEBAGAI PENDUKUNG KEPUTUSAN**

#### SKRIPSI

Laboratorium Komputasi Cerdas dan Visualisasi

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer

Disusun oleh:

**MOHAMAD ILHAM UBaidillah**

**NIM. 0910680083**

Skripsi ini telah diuji dan dinyatakan lulus tanggal 2 Januari 2014

**Penguji I**

**Penguji II**

**Lailil Muflikhah, S.Kom., M.Sc**

**Imam Cholissodin, S.Si., M.Kom**

**NIP. 19741113 200501 2 001**

**NIP. 850719 16 1 1 0422**

**Penguji III**

**Novanto Yudistira, S.Kom., M.Sc.**

**NIP. 831110 16 1 1 0425**

**Mengetahui**  
**Ketua Program Studi Informatika**

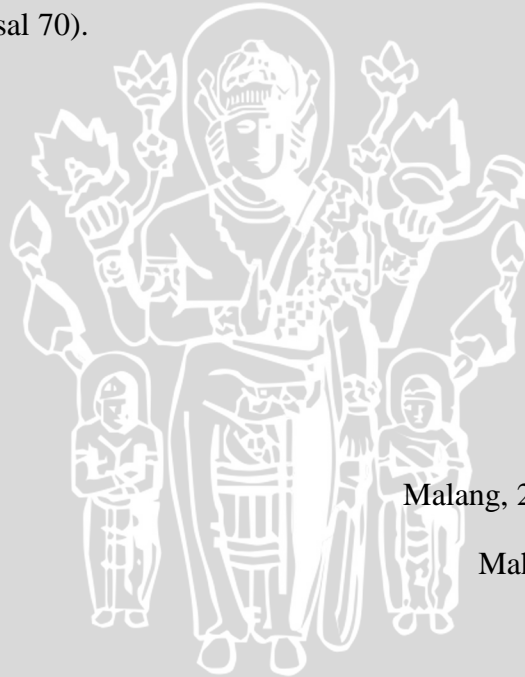
**Drs. Marji, M.T.**

**NIP. 19670801 199203 1 001**

## PERNYATAAN ORISINALITAS SKRIPSI

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

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



Malang, 2 Januari 2014

Mahasiswa,

**Mohamad Ilham Ubaidillah**

**NIM. 0910680083**

## KATA PENGANTAR

*Alhamdulillah rabbil 'alamin.* Puji syukur penulis panjatkan kehadirat Allah SWT, karena atas segala rahmat dan limpahan hidayah-Nya, penulis dapat menyelesaikan tugas akhir yang berjudul FUZZY CLUSTERING BACKPROPAGATION NEURAL NETWORKS UNTUK PREDIKSI PERKEMBANGAN PENYAKIT DEMAM BERDARAH SEBAGAI PENDUKUNG KEPUTUSAN.

Dalam pelaksanaan dan penulisan tugas akhir ini penulis mendapatkan banyak bantuan dari berbagai pihak. Dalam kesempatan ini penulis ingin mengucapkan terima kasih yang sebesar - besarnya kepada :

1. Kedua Orang Tua Penulis, Ahmad Husnul Ibtidak dan Agustini Iriana.
2. Rekyan Regasari MP.,ST.,MT dan Ahmad Afif Supianto, SSi., M.Kom selaku dosen pembimbing skripsi.
3. Semua pihak yang memperlancar penyelesaian skripsi penulis

Penulis menyadari bahwa dalam penyusunan tugas akhir ini masih banyak kekurangan baik format penulisan maupun isinya. Oleh karena itu, saran dan kritik membangun dari para pembaca senantiasa penulis harapkan guna perbaikan bagi tugas akhir selanjutnya. Semoga tugas akhir ini dapat memberikan manfaat bagi semua pihak, Amin.

Malang, Januari 2014

Penulis

## ABSTRAK

**Mohamad Ilham Ubaidillah. 2013. : *Fuzzy Clustering Backpropogation Neural Networks* untuk Prediksi Perkembangan Penyakit Demam Berdarah sebagai Pendukung Keputusan. Skripsi Program Studi Teknik Informatika, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya. Dosen Pembimbing: Rekyan Regasari M.P., ST., M.T., dan Ahmad Afif Supianto S.Si., M.Kom.**

Kasus penderita Demam Berdarah di Indonesia merupakan salah satu masalah kesehatan masyarakat Indonesia yang cenderung meningkat jumlahnya serta semakin luas penyebarannya. Hal itu dikarenakan masih tersebarnya nyamuk *aedes aegypti* (penular penyakit DBD) dan dibuktikan dari terjadinya Kejadian Luar Biasa (KLB) untuk penyakit Demam Berdarah di beberapa daerah. Metode yang digunakan pada penelitian sebelumnya adalah *Fuzzy Clustering Backpropogation Neural Networks*. Perancangan model data pada penelitian ini terdiri dari 3 atribut, dengan 2 atribut sebagai kriteria yaitu Jumlah Hari Hujan (JHH) dan Curah Hujan(CH), dan 1 atribut sebagai output yaitu Kasus (K). Hasil pengujian menunjukkan bahwa FCBPN dapat digunakan dengan baik untuk prediksi kasus demam berdarah dengan parameter model yang tepat.

**Kata kunci** : Prediksi, *Backpropagation Neural Networks*, *Fuzzy Clustering*, Demam Berdarah

## ABSTRACT

**Mohamad Ilham Ubaidillah. 2013. : Fuzzy Clustering Backpropogation Neural Networks for Prediction of Dengue Fever as Decision Support. Undergraduate Thesis of Informatic Study Program, Information Technology and Computer Science Program, Brawijaya University, Malang. Advisor : Rekyan Regasari M.P., ST., M.T., and Ahmad Afif Supianto S.Si., M.Kom.**

Cases of Dengue patients in Indonesia is one of the Indonesian health problems tend to increase in number and more widely spread. That's because Aedes aegypti mosquito (dengue disease transmission) is still spread and evidenced from the occurrence of Extraordinary Events (KLB) for Dengue disease in some areas. The method used in previous studies are Backpropogation Fuzzy Clustering Neural Networks. The design of the data model in this study consists of three attributes, with 2 attributes as criteria: Number of Days Rain (JHH) and Rainfall (CH), and 1 as output attributes are Case (K). Test results show that it can be used with either FCBPN for prediction of dengue cases with appropriate model parameters.

**Keyword** : Prediction, Backpropagation Neural Networks, Fuzzy Clustering, Dengue

## DAFTAR ISI

KATA PENGANTAR .....	i
ABSTRAK .....	ii
ABSTRACT .....	iii
DAFTAR ISI .....	iv
DAFTAR GAMBAR .....	vi
DAFTAR TABEL .....	vii
BAB I PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	2
1.3. Batasan Masalah .....	3
1.4. Tujuan .....	3
1.5. Manfaat .....	3
1.6. Sistematika Pembahasan .....	3
BAB II TINJAUAN PUSTAKA .....	5
2.1. Kajian Pustaka .....	5
2.2. Dasar Teori .....	7
2.2.1. Epidemiologi dan Demam Berdarah .....	7
2.2.2. <i>Fuzzy Backpropagation Neural Network (FCBPN)</i> .....	8
2.2.3. Pengukuran Performa / <i>Performance Measures</i> .....	13
BAB III METODE PENELITIAN .....	15
3.1. Studi Literatur .....	16
3.2. Penyusunan Tinjauan Pustaka .....	16
3.3. Metode Pengambilan Data .....	16
3.4. Analisis Kebutuhan .....	16
3.5. Perancangan Prediksi .....	17
3.6. Implementasi .....	17
3.7. Pengujian dan Analisis .....	17
3.8. Pengambilan Kesimpulan .....	18
BAB IV PERANCANGAN .....	19



4.1. Perancangan Prediksi dengan <i>Fuzzy C-Means backpropagation Neural Network</i> .....	19
4.1.1. Perancangan Model Data .....	19
4.1.2. Perancangan Algoritma <i>Fuzzy C-Means Clustering Back Propagation Neural Networks (FCBPN)</i> .....	19
4.2. Perancangan Manualisasi Metode <i>Fuzzy C-Means Clustering Backpropagation Neural Networks (FCBPN)</i> .....	30
4.2.1. Normalisasi Data .....	31
4.2.2. <i>Clustering dengan Fuzzy C-Means (FCM)</i> .....	31
4.2.3. <i>Degree of Membership Level (MLC<sub>k</sub>)</i> .....	33
4.2.4. <i>Backpropagation Neural Networks</i> .....	33
4.3. Perancangan Skenario Pengujian Metode <i>Fuzzy C-Means Clustering Backpropagation Neural Networks (FCBPN)</i> .....	42
<b>BAB V IMPLEMENTASI</b> .....	43
5.1. Spesifikasi Sistem .....	43
5.1.1. Spesifikasi Perangkat Keras .....	43
5.1.2. Spesifikasi Perangkat Lunak .....	43
5.2. Batasan Implementasi .....	44
5.3. Impelementasi Metode FCBPN .....	44
5.3.1. Normalisasi .....	44
5.3.2. The Degree of Membership Levels (MLC) .....	48
5.3.3. Backpropagation Neural Networks .....	48
5.3.4. Fuzzy Clustering backpropagation Neural Networks .....	54
<b>BAB VI PENGUJIAN DAN ANALISIS</b> .....	56
6.1. Pengujian .....	56
6.2. Analisis .....	69
<b>BAB VII PENUTUP</b> .....	75
7.1. Kesimpulan .....	75
7.2. Saran .....	75
<b>DAFTAR PUSTAKA</b> .....	76



## DAFTAR GAMBAR

Gambar 2.1 Arsitektur dari model FCBPBPN.....	6
Gambar 3.1 Diagram alir metode penelitian.....	15
Gambar 3.2 <i>Block Diagram</i> FCBPBPN.....	17
Gambar 4.1 <i>Block Diagram</i> FCBPBPN.....	21
Gambar 4.2 Diagram Alir Normalisasi.....	22
Gambar 4.3 <i>Fuzzy C-Means Clustering (FCM)</i> .....	23
Gambar 4.4 <i>Generate Matrix</i> .....	24
Gambar 4.5 Diagram Alir Cari Centroid.....	25
Gambar 4.6 Diagram Alir Update Matrix.....	26
Gambar 4.7 Diagram Alir Check Termination.....	27
Gambar 4.8 Diagram Alir Membership Level Cluster.....	28
Gambar 4.9 <i>Block Diagram Backpropagation</i> .....	29
Gambar 4.10 <i>Block Diagram Feedforward</i> .....	30
Gambar 4.11 Model BPN Pada Manualisasi.....	34
Gambar 6.1 Grafik Prediksi kasus Panarukan.....	64
Gambar 6.2 Grafik Prediksi Kasus Situbondo.....	65
Gambar 6.3 Grafik Prediksi Kasus Lokasi Panji.....	66
Gambar 6.4 Grafik Prediksi Kasus Lokasi Asembagus.....	67
Gambar 6.5 Grafik Prediksi Kasus Lokasi Mlandingan.....	68
Gambar 6.6 Grafik Prediksi Kasus Lokasi Besuki.....	68
Gambar 6.7 Grafik MAPE di masing- masing Lokasi.....	69
Gambar 6. 8 Grafik RMSE di masing- masing Lokasi.....	70
Gambar 6. 9 Histogram Kasus Besuki.....	71
Gambar 6.10 Histogram Kasus Asembagus.....	71
Gambar 6.11 Histogram Kasus Panarukan.....	72
Gambar 6.12 Histogram Kasus Panarukan.....	72
Gambar 6.13 Histogram Kasus Panji.....	73
Gambar 6. 14 Histogram Kasus Besuki.....	73

## DAFTAR TABEL

Tabel 2.1 Tabel kriteria.....	5
Tabel 4.1 Tabel Model data .....	19
Tabel 4.2 Tabel Input data training.....	31
Tabel 4.3 Tabel Normalisasi Data Input .....	31
Tabel 4.4 Tabel Inisialisasi <i>Matriks U</i> .....	32
Tabel 4.5 Tabel <i>Centroid</i> iterasi 1 .....	32
Tabel 4.6 Tabel <i>Matriks U</i> iterasi 1 .....	32
Tabel 4.7 Tabel <i>Centroid</i> iterasi 3 .....	32
Tabel 4.8 Tabel <i>Matriks U</i> iterasi 3 .....	33
Tabel 4.9 Tabel <i>MLC</i> .....	33
Tabel 4.10 Tabel <i>MLC</i> .....	33
Tabel 4. 11 Tabel Inisialisasi bobot awal BPN .....	34
Tabel 4.12 Tabel <i>weight</i> BPN iterasi 1 <i>cluster</i> 1 .....	35
Tabel 4.13 Tabel <i>weight</i> BPN iterasi 1 <i>cluster</i> 2 .....	35
Tabel 4.14 Tabel node BPN iterasi 1 <i>cluster</i> 1 .....	35
Tabel 4. 15 Tabel node BPN iterasi 1 <i>cluster</i> 2 .....	36
Tabel 4.16 Tabel <i>weight</i> iterasi 5316 <i>cluster</i> 1 .....	40
Tabel 4.17 Tabel <i>weight</i> iterasi 5316 <i>cluster</i> 2 .....	40
Tabel 4.18 Tabel Data Uji.....	40
Tabel 5. 1 Spesifikasi Perangkat Keras Komputer.....	43
Tabel 5.2 Spesifikasi Perangkat Lunak Komputer.....	43
Tabel 5.3 Source Code Normalisasi.....	44
Tabel 5.4 Table <i>Fuzzy C-Means Clustering</i> .....	45
Tabel 5.5 The Degree of Membership Levels.....	48
Tabel 5.6 Backpropagation Neural Networks .....	48
Tabel 5.7 Fuzzy Clustering backpropagation Neural Networks .....	54
Tabel 6.1 Pengujian RMSE dan MAPE di Panarukan dengan $c = 0.1$ .....	57
Tabel 6. 2 Tabel Pengujian RMSE dan MAPE dengan $a = 50$ .....	58
Tabel 6. 3 Pengujian RMSE dan MAPE di Situbondo dengan $c = 0.1$ .....	58
Tabel 6. 4 Tabel Pengujian RMSE dan MAPE dengan $a = 50$ .....	59
Tabel 6.5 Pengujian RMSE dan MAPE di Panji dengan $c = 0.1$ .....	59
Tabel 6.6 Tabel Pengujian RMSE dan MAPE dengan $a = 40$ .....	60
Tabel 6.7 Pengujian RMSE dan MAPE di Asembagus dengan $c = 0.1$ .....	60
Tabel 6.8 Tabel Pengujian RMSE dan MAPE dengan $a = 70$ .....	61
Tabel 6.9 Pengujian RMSE dan MAPE di Mlandingan dengan $c = 0.1$ .....	62
Tabel 6.10 Tabel Pengujian RMSE dan MAPE dengan $a = 70$ .....	62
Tabel 6. 11 Pengujian RMSE dan MAPE di Besuki dengan $c = 0.1$ .....	63
Tabel 6. 12 Tabel Pengujian RMSE dan MAPE dengan $a = 90$ .....	63
Tabel 6. 13 Tabel hasil pengujian .....	69

## BAB I PENDAHULUAN

### 1.1. Latar Belakang

Kasus penderita Demam Berdarah di Indonesia merupakan salah satu masalah kesehatan masyarakat Indonesia yang cenderung meningkat jumlahnya serta semakin luas penyebarannya. Hal itu dikarenakan masih tersebarnya nyamuk aedes aegypti (penular penyakit DBD) dan dibuktikan dari terjadinya Kejadian Luar Biasa (KLB) untuk penyakit Demam Berdarah di beberapa daerah. Berdasarkan data statistik kasus demam berdarah di Indonesia, pada data terakhir di tahun 2009, jumlah kasus Demam Berdarah di Indonesia sebanyak 158.912 (AI > 55 kasus/100.000 penduduk dengan CFR di atas 1%) [UFA-10].

Kabupaten Situbondo merupakan salah satu daerah dengan tingkat penyebaran penyakit demam berdarah yang tinggi. Hal itu disebabkan oleh banyak faktor. Salah satu faktor yang menyebabkan tingkat penyebaran penyakit demam berdarah menjadi tinggi adalah faktor geografis. Secara geografis wilayah Kabupaten Situbondo merupakan daerah transit dari berbagai kalangan masyarakat dari berbagai daerah yang membuat potensi terjadinya perubahan/evolusi gen virus dengue. Lokasi daerah-daerah terpencil yang tidak dapat terjangkau dalam program penanggulangan dapat memungkinkan terjadinya endemisitas penyakit demam berdarah. Data statistik 3 tahun terakhir di Kabupaten Situbondo menunjukkan kasus demam berdarah masih berkisar antara 100-300 kasus. Faktor keberhasilan penanganan kasus demam berdarah ditentukan dari kecermatan penanganan secara dini mengenai perkembangannya dan pentatalaksanaan, perawatan dalam buku pedoman Pentatalaksanaan DBD[ACL-98].

Beberapa penelitian mengenai penyebaran maupun analisis penyebab penyebaran penyakit demam berdarah sudah dilakukan. Berbagai eksperimen faktor yang diduga terkait dan berpengaruh terhadap akurasi prediksi telah diuji. Semakin tepat faktor akan berpengaruh pada prediksi yang semakin baik. Namun pada kenyataannya tidak semua faktor/kriteria pada penelitian sebelumnya dapat

dipenuhi disuatu daerah (termasuk di Kabupaten Situbondo). Setiap daerah memiliki karakteristik dalam penyebaran penyakit demam berdarah sehingga ada faktor tersebut tidak bisa didapatkan atau dianggap tidak berkorelasi. Pada akhirnya diperlukan analisa lebih spesifik mengenai Epidemiologi penyakit demam berdarah terhadap karakteristik di Kabupaten Situbondo.

Berdasarkan perkembangan demam berdarah yang begitu pesat di Kabupaten Situbondo, maka diperlukan sebuah penelitian dalam memprediksi perkembangan penyakit demam berdarah. Salah satu Penelitian dengan prediksi yang dapat digunakan untuk data time series adalah metode *Fuzzy Clustering Backpropagation Neural Network (FCBPN)*.

Metode FCBPN pada penelitian ini diimplementasikan berdasarkan penelitian sebelumnya yang berjudul 'An improved approach based on fuzzy clustering and Back-propagation Neural Networks with adaptive learning rate for sales forecasting: Case study of PCB industry' karena berdasarkan hasil akurasi yang baik. Metode yang digunakan adalah Fuzzy Clustering dan Backpropagation Neural Networks. Tahap awal sebelum proses prediksi, metode Rescaled Range Analysis akan mendeteksi adanya tren dari pola data time-series di masing masing kriteria pada suatu faktor tersebut. Kemudian Winter Exponential Smoothing akan membantu menghapus tren dengan cara menghasilkan kriteria baru yang nantinya akan membantu memperkuat pola data dalam komputasi prediksi Sistem [AHB-12]. Tahap prediksi menggunakan metode Fuzzy C-Means Clustering Back-Propagation Neural Networks (FCBPN) yang menggabungkan Fuzzy C-Means (FCM) dengan algoritma pembelajaran Backpropagation Neural Network.

## 1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dikemukakan di atas, maka dirumuskan masalah sebagai berikut:

1. Bagaimana menghasilkan parameter  $a$  dan  $c$  yang terbaik pada *Fuzzy Clustering Back-Propagation Neural Networks (FCBPN)* di mana  $a$  dan  $c$  adalah sebuah konstanta dari fungsi *sigmoid* pada tahap *Membership Level Cluster (MLC)*

2. Bagaimana mendapatkan nilai RMSE dan MAPE pada prediksi kasus demam berdarah masing-masing lokasi di Situbondo.

### 1.3. Batasan Masalah

Berdasarkan latar belakang dan rumusan masalah yang telah dikemukakan, penelitian ini mempunyai batasan-batasan masalah sebagai berikut:

1. Data statistik diperoleh dari Dinas Kesehatan Kabupaten Situbondo.
2. *Output* berupa prediksi nilai disuatu waktu.
3. Prediksi yang dihasilkan untuk pembantu penentuan keputusan sehingga keputusan final dikembalikan pada kebijakan pihak – pihak terkait.

### 1.4. Tujuan

Tujuan yang ingin dicapai dalam pembuatan skripsi ini adalah:

1. Membantu penentuan kebijakan dari hasil prediksi sistem mengenai penyebaran penyakit demam berdarah.
2. Membantu prediksi penyebaran penyakit demam berdarah dengan menggunakan data statistik yang dimiliki berupa faktor terkait.

### 1.5. Manfaat

Manfaat yang diharapkan dari penelitian ini adalah membantu pengambilan kebijakan terhadap pengendalian penyakit demam berdarah berdasarkan hasil prediksi metode.

### 1.6. Sistematika Pembahasan

Sistematika penulisan penelitian ditunjukkan untuk memberikan gambaran dan uraian dari penyusunan tugas akhir secara garis besar yang meliputi beberapa bab, sebagai berikut.

## **BAB I : Pendahuluan**

Menguraikan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika pembahasan.

## **BAB II : Tinjauan Pustaka**

Menguraikan tentang dasar teori dan referensi yang mendasari penelitian ini

### **BAB III : Metode Penelitian**

Metode Penelitian menguraikan tentang metode dan langkah kerja yang dilakukan dalam penulisan tugas akhir yang terdiri dari studi literatur, perancangan sistem perangkat lunak, implementasi sistem perangkat lunak, pengujian dan analisis, serta penulisan laporan.

### **BAB IV : Perancangan**

Perancangan berisi tentang perencanaan aplikasi yang dibangun mengacu pada Metodologi Penelitian serta Kajian Pustaka dan Dasar Teori.

### **BAB V : Implementasi**

Membahas implementasi dari metode sesuai dengan perancangan sistem yang telah dibangun.

### **BAB VI : Pengujian dan Analisis**

Memuat hasil pengujian dan analisis terhadap metode yang telah direalisasikan.

### **BAB VII : Penutup**

Memuat kesimpulan yang diperoleh dari pembuatan dan pengujian perangkat lunak yang dikembangkan dalam skripsi ini serta saran – saran untuk pengembangan lebih lanjut.

## BAB II

### TINJAUAN PUSTAKA

Bab ini membahas kajian pustaka dan dasar teori. Dasar teori terdiri dari epidemiologi demam berdarah, *Fuzzy C-Means Clustering*, *Backpropagation Neural Networks*, *Fuzzy C-Means Backpropagation Neural Networks*, dan Pengukuran Performa.

#### 2.1. Kajian Pustaka

Kajian pustaka pada penelitian ini mengacu pada penelitian yang berjudul ‘*An improved approach based on fuzzy clustering and Back-propagation Neural Networks with adaptive learning rate for sales forecasting: Case study of PCB industry*’. Penelitian tersebut menggunakan studi kasus penjualan di PCB Industri dengan metode yang digunakan adalah *C-Means Fuzzy Backpropagation Neural Network*. Berikut adalah kriteria yang digunakan pada penelitian tersebut:

**Tabel 2.1** Tabel kriteria

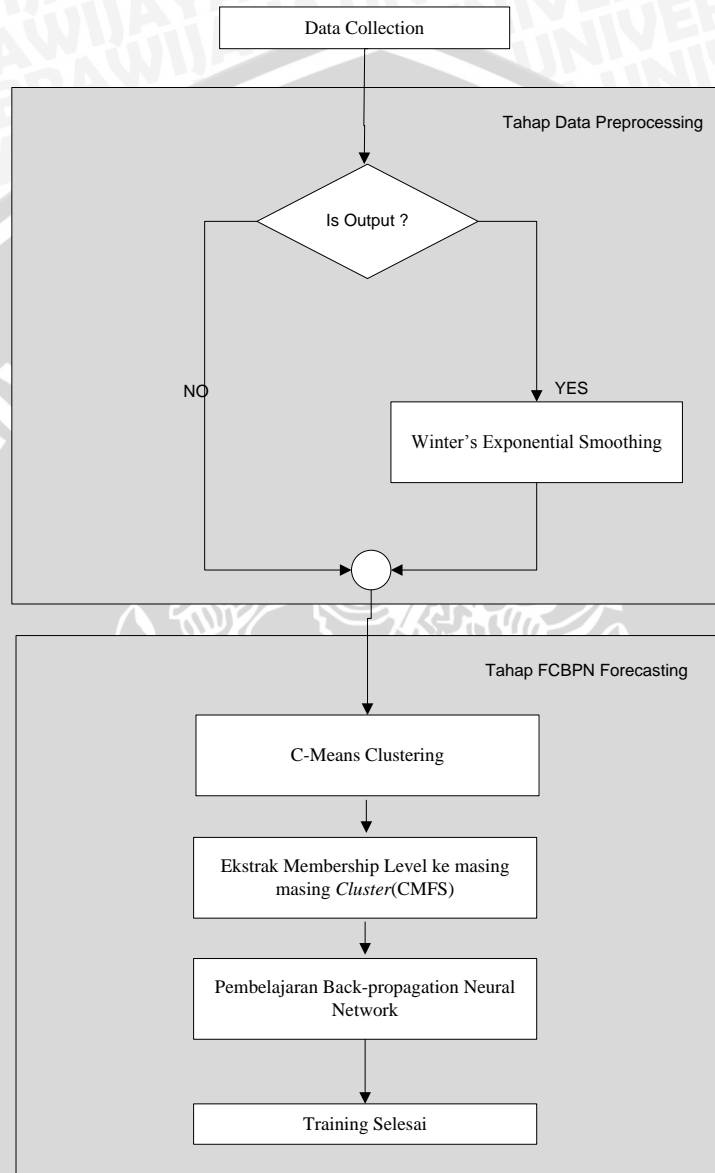
Kriteria	Nama
<b>K1</b>	<i>Consumer price index</i>
<b>N1</b>	<i>Normalized Consumer price index</i>
<b>K2</b>	<i>Liquid crystal element demand</i>
<b>N2</b>	<i>Normalized Liquid crystal element demand</i>
<b>K3</b>	<i>PCB total production value</i>
<b>K4</b>	<i>Preprocessed historical data (WES)</i>
<b>N3</b>	<i>Normalized PCB total production value</i>
<b>N4</b>	<i>Normalized preprocessed historical data (WES)</i>
<b>Y0</b>	<i>Actual historical monthly PCB sales Data</i>
<b>Y</b>	<i>Normalized Actual historical monthly PCB sales Data</i>

Sumber: [AHB-12]

Tahap awal dari model ini adalah *data preprocessing* yaitu penambahan Kriteria baru ( $K_{n+1}$ ). Kriteria baru ini didapatkan dari hasil Prediksi awal dari output dengan menggunakan *Winter's Exponential Smoothing*. Selanjutnya data masukan akan dinormalisasi dan di *cluster* menggunakan *Fuzzy C-Means Clustering* sehingga menghasilkan derajat keanggotaan sebanyak jumlah *cluster* yang ditentukan. Pada langkah akhir data akan diproses secara parallel didalam beberapa *Backpropagation*. Jumlah *Backpropagation* yang digunakan sebanyak



jumlah *cluster* yang dihasilkan oleh FCM. *Adaptive learning rate* akan dibentuk dari kesesuaian tingkat keanggotaan *cluster* ( $MLC_k$ ) dari setiap *records training data set*.



**Gambar 2.1** Arsitektur dari model FCBPN

Sumber: [AHB-12]

Pengujian pada penelitian tersebut dilakukan dengan *data training* dari Januari 1999 sampai Desember 2002 dan data uji di bulan Januari 2003 sampai Desember 2003. Hasil percobaan pada penelitian tersebut menunjukkan bahwa

efektivitas FCBPN melebihi yang penelitian sebelumnya seperti : WES, BPN, RBFNN, KFNN, FNN, WES, BPN dan KGFS. Keuntungan dari FCBPN adalah model ini menggunakan *fuzzy clustering (c-means)* yang memungkinkan setiap *record* data dapat dimiliki oleh masing-masing *cluster* yang direpresentasikan dalam tingkat tertentu. Hal ini berdampak pada kemungkinan kelompok menjadi lebih besar yang berakibat pada meningkatkan keakuratan.

Pada penelitian ini, FCBPN akan dipergunakan dalam memprediksi perkembangan penyakit demam berdarah. Seperti halnya memprediksi penjualan pada PCB Industri, prediksi demam berdarah juga memiliki parameter kriteria yang akan digunakan.

## 2.2. Dasar Teori

Dasar teori pada penelitian ini antara lain adalah Epidemiologi dan Demam Berdarah, *Fuzzy C-Means Backpropagation Neural Networks*, dan Pengukuran Performa.

### 2.2.1. Epidemiologi dan Demam Berdarah

Epidemiologi adalah ilmu yang mempelajari pola kesehatan dan penyakit serta faktor yang terkait di tingkat populasi. Epidemiologi termasuk model *corestone* penelitian kesehatan masyarakat, dan membantu menginformasikan kedokteran berbasis bukti (*evidence based medicine*) untuk mengidentifikasi faktor risiko penyakit serta menentukan pendekatan penanganan yang optimal untuk praktik klinik dan untuk kedokteran preventif.

Saat ini Epidemiologi sudah mengalami perluasan diberbagai bidang. Perluasan tersebut dirasa perlu karena manfaat epidemiologi sangat nyata dirasakan dalam bidang-bidang ilmu tersebut. Pendalaman epidemiologi diantaranya meliputi peramalan berbasis komputer dan pengelolaan agroekosistem.

Demam berdarah (DB) adalah penyakit demam akut yang disebabkan oleh virus *dengue*, yang masuk ke peredaran darah manusia melalui gigitan nyamuk dari genus *Aedes*, misalnya *Aedes aegypti* atau *Aedes albopictus*. Terdapat empat jenis virus dengue berbeda, namun berelasi dekat, yang dapat menyebabkan demam berdarah. Virus *dengue* merupakan virus dari genus *Flavivirus*, famili

*Flaviviridae*. Penyakit demam berdarah ditemukan di daerah tropis dan subtropis di berbagai belahan dunia, terutama di musim hujan yang lembap [ACL-98].

Organisasi Kesehatan Dunia (WHO) memperkirakan setiap tahunnya terdapat 50-100 juta kasus infeksi virus dengue di seluruh dunia. Hingga kini, belum ada vaksin atau obat antivirus bagi penyakit ini. Tindakan paling efektif untuk menekan epidemi demam berdarah adalah dengan mengontrol keberadaan dan sedapat mungkin menghindari vektor nyamuk pembawa virus dengue.

Selain itu, karena nyamuk *Aedes* aktif di siang hari beberapa tindakan pencegahan yang dapat dilakukan adalah menggunakan senyawa anti nyamuk yang mengandung DEET, pikaridin, atau minyak lemon *eucalyptus*, serta gunakan pakaian tertutup untuk dapat melindungi tubuh dari gigitan nyamuk bila sedang beraktivitas di luar rumah. Selain itu, segeralah berobat bila muncul gejala-gejala penyakit demam berdarah sebelum berkembang menjadi semakin parah.

### 2.2.2. Fuzzy Backpropagation Neural Network (FCBPN)

Integrasi *Fuzzy C-Means Clustering* dan *Backpropagation Neural Networks* menjadi sebuah mekanisme metode merupakan model *hybrid* yang digunakan untuk prediksi berdasarkan kriteria dan pembelajaran *output* yang diharapkan.

Arsitektur FCBPN terdiri dari tiga tahap. Tahap pertama yaitu, *output* dari metode *winter's* akan dijadikan input ( $K_4$ ) dalam metode FCBPN untuk menghilangkan efek kecendrungan. Kedua, semua *normalized records* dari data akan dikategorikan ke dalam *cluster* K dengan menggunakan model metode *fuzzy C-means*. Ketiga, *fuzzy distances* dari semua *record data* ( $X_i$ ) ke *cluster centers* ( $C_j$ ) yang sudah ditemukan oleh metode *fuzzy C-means* akan dimasukkan ke dalam jaringan paralel *Back Propagation* (BP). *Fuzzy distances* ini dimasukkan ke dalam BP tergantung dari *learning rate* yang sudah disesuaikan menurut tingkat keanggotaan *cluster* setiap *record data* pelatihan yang telah ditentukan.

### 2.2.2.1. Normalization

Data masukan akan di normalisasi sebelum masuk ke *Fuzzy C-Means Clustering* (FCM) dengan *range* antara 0,1 – 0,9.

$$N_i = 0.1 + 0.8((K_i - \min(K_i))/(\max(K_i) - \min(K_i))), \quad (2-18)$$

di mana :

$i$  = kriteria

$N_i$  = data normalisasi pada kriteria ke  $i$

$K_i$  = data kriteria ke  $i$ .

### 2.2.2.2. Fuzzy C-Means Clustering (FCM)

Dalam *hard clustering*, data dibagi menjadi *clusters* yang berbeda, di mana setiap elemen data hanya termasuk ke dalam satu *cluster*. Dalam *Fuzzy C-means* (FCM), elemen data dapat dikelompokkan lebih dari satu *cluster*. Elemen data tersebut yang berkaitan dengan setiap elemen adalah satu set tingkat keanggotaan. Hal ini didasarkan pada fungsi minimalisasi yang ditunjukkan pada persamaan (2-23) berikut.

$$J_m = \sum_{i=1}^N \sum_{j=1}^C u_{ij}^m \|x_i - c_j\|^2, 1 \leq m < \infty \quad (2-19)$$

Keterangan:

$u_{ij}^m$  = derajat keanggotaan  $x_i$  dalam *cluster*  $j$

$x_i$  = data ke  $i^{th}$

$c_j$  = pusat *cluster* ke  $j^{th}$

Berikut ini adalah langkah-langkah dari algoritma FCM.

1. Inisialisasi derajat keanggotaan matriks secara *random*,

$$U = [u_{ij}], U^{(0)} \quad (2-20)$$

2. Hitung *centroid* dari setiap *cluster*  $C_{(k)} = [c_j]$  dengan

$$U_{(k)}: C_j = \frac{\sum_{i=1}^N u_{ij}^m \cdot x_i}{\sum_{i=1}^N u_{ij}^m} \quad (2-21)$$

3. Untuk setiap titik, perbarui nilai koefisien titik tersebut yang berada di dalam *cluster*

$$(U(k), U(k+1)): u_{ij} = \frac{1}{\sum_{i=1}^C \left( \frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{2/(m-1)}} \quad (2-22)$$

4. Jika  $\|U^{(k+1)} - U^{(k)}\| < \epsilon \cdot 0 < 1$ , (2-23)  
 maka berhenti. Jika sebaliknya, maka kembali ke langkah nomor 2.

Prosedur ini konvergen dengan lokal minimum atau *saddle point* dari  $J_m$ . Parameter yang digunakan dari kombinasi 2 faktor (m dan  $\epsilon$ ) adalah  $m = 2$  dan  $\epsilon = 0.5$

### 2.2.2.3. The degree of Membership levels (MLC<sub>k</sub>)

Pada tahap ini, penelitian ini akan menggunakan fungsi *sigmoid* untuk meningkatkan presisi hasil dan mempercepat proses *training* dari *neural networks*. Setelah itu, *fuzzy distance* antara *record data* ( $X_i$ ) dan pusat *cluster* ( $C_k$ ) ( $AFD_k$ ) akan dimasukkan ke dalam persamaan berikut.

$$AFD_k(X_i) = \text{sigmoid}(\|C_k - X_i\|, [50, 0.5]) \quad (2-24)$$

$$\text{sigmoid}(x, [a, c]) = \frac{1}{1 + e^{-a(x-c)}} \quad (2-25)$$

Tingkat keanggotaan yang termasuk ke dalam *record*  $X_i$  ke  $k^{th}$  dan *cluster* ( $MLC_k(X_i)$ ) berhubungan terbalik dengan jarak dari *records data*  $X_i$  ke pusat *cluster*  $X_i$  ( $AFD_k(X_i)$ ):

$$MLC_k(X_i) = \frac{1}{AFD_k(X_i)} \quad (2-26)$$

*Clusters Membership Fuzzy System* (CMFS) akan mengembalikan tingkat keanggotaan yang termasuk *data record* X ke setiap *cluster*:

$$CMFS(X) = (MLC_1(X), MLC_2(X), MLC_3(X), MLC_4(X)) \quad (2-27)$$

Dengan demikian, kita dapat membuat *training sample* yang baru ( $X_i, MLC_1(X), MLC_2(X), MLC_3(X), MLC_4(X)$ ) untuk evaluasi *adaptive neural networks*.

#### 2.2.2.4. Tahap evaluasi *Adaptive neural networks*

Konsep *Artificial Neural Networks* (ANNs) berasal dari ilmu biologi. Komponen ini terhubung sesuai dengan beberapa pola konektivitas, terkait dengan perbedaan bobot. Bobot dari *neural connection* akan diperbarui dengan *learning*. ANNs memiliki kemampuan untuk mengidentifikasi pola yang bersifat *nonlinear* dengan cara melakukan *learning* dari *data set*. Mungkin metode *Training Back Propagation* (BP) yang paling populer. Struktur dari BP *Neural Networks* terdiri dari *input layer*, *hidden layer*, dan *output layer*. Setiap *layer* berisi *node* I, J dan L yang dinotasikan.  $W_{ij}$  dinotasikan menurut bobot angkat antara *input layer* dan *hidden layer* dan juga  $W_{ij}$  antara *hidden layer* dan *output layers*.

Pada model ini, Sistem Evaluasi *Adaptive Neural Networks* terdiri dari  $n$  *neural networks*. Setiap *cluster*  $K$  terkait dengan  $K^{ith}$  BP *network*. Untuk setiap *cluster*, *training sample* akan dimasukkan ke dalam *Parallel Back Propagation Networks* (BPN) dengan *learning rate* yang sudah disesuaikan dengan tingkat keanggotaan *cluster* ( $MLC_k$ ) dari setiap *records training data set*.

Algoritma *Learning Adaptive Neural Networks* tersusun dari dua prosedur. Prosedur pertama adalah *feed forward step*, dan yang kedua adalah bobot *training back-propagation*. Berikut ini adalah langkah-langkah detail dari kedua prosedur ini.

1. Semua BP *Networks* diinisialisasi dengan bobot *random*.
2. *Feed Forward*

Untuk setiap  $BPN_k$ , diasumsikan untuk masing-masing faktor *input* di dalam *input layer* dinotasikan dengan  $x_i$ ,  $y_j^k$  dan  $o_l^k$  yang mewakili *output* di dalam *hidden layer* dan *output layer*. Rumus  $y_j^k$  dan  $o_l^k$  ditunjukkan pada persamaan berikut.

$$y_j^k = f(X_j^k) = f(w_{oj}^k + \sum_{i=0}^l w_{ij}^k x_i) \quad (2-28)$$

$$o_l^k = f(Y_l^k) = f(w_{ol}^k + \sum_{j=0}^l w_{jl}^k y_j^k) \quad (2-29)$$

$w_{oj}^k$  dan  $w_{ol}^k$  adalah bobot untuk mengatur nilai *threshold*.  $f$  adalah fungsi aktivasi yang digunakan untuk *hidden layer* dan *output layer*. Sedangkan  $X_j^k$  dan  $Y_l^k$  menghitung hasil sementara sebelum memasuki fungsi aktivasi  $f$ . Dalam penelitian ini, fungsi aktivasi yang digunakan adalah fungsi *sigmoid*. Maka, masing-masing *output*  $y_j^k$  dan  $o_l^k$  dalam *hidden layer* dan *output layer* dapat juga ditulis dengan persamaan berikut.

$$y_j^k = f(X_j^k) = \frac{1}{1 + e^{-X_j^k}} \quad (2-30)$$

$$o_l^k = f(Y_l^k) = \frac{1}{1 + e^{-Y_l^k}} \quad (2-31)$$

Fungsi aktivasi  $f$  memberikan efek *nonlinier* ke dalam jaringan dan *maps* yang merupakan hasil dari perhitungan ke *domain* (0,1). Pada penelitian ini, fungsi aktivasi yang digunakan adalah fungsi *sigmoid*.

$$f(t) = \frac{1}{1 + e^{-t}}, f' = (1 - f) \quad (2-32)$$

Rumus *output adaptive neural networks* ditunjukkan oleh persamaan berikut.

$$O_l = \frac{\sum_{k=1}^4 (MLC_k(X_j) \cdot o_l^k)}{\sum_{k=1}^4 MLC_k(X_j)} \quad (2-33)$$

Pada persamaan diatas, efek dari *output*  $o_l^k$  pada *global output*  $O_l$  keduanya sangat tergantung dengan tingkat keanggotaan  $MLC_k$  dari *data record*  $X_j$  ke *cluster*  $k^{ith}$ .

### 3. Back Propagation Weight Training

$$E = \frac{1}{2} \sum_{l=1}^L e_l^2 = \frac{1}{2} \sum_{l=1}^L (t_l - o_l)^2 \quad (2-34)$$

Dengan  $t_k$  merupakan *predefined standart output* (atau output yang diinginkan, atau target nilai) dan  $e_k$  adalah kesalahan dari setiap *output node*. Tujuannya adalah untuk meminimalisasi E, sehingga nilai *weight* pada setiap *link* disesuaikan dan *final output* bisa cocok dengan *output* yang diinginkan.

Kecepatan *learning* dapat ditingkatkan dengan memasukkan *momentum term*. Biasanya, diantara range [0, 1]. Untuk iterasi  $n$  dan untuk  $BPN_k$  (terkait dengan cluster ke  $k^{th}$ ), *adaptive learning rate* pada  $BPN_k$ . Variasi dari *weights*  $\Delta w_k$  ditunjukkan pada persamaan berikut.

$$\eta_k = \frac{MLC_k(X_j)}{\sum_{k=1}^4 MLC_k(X_j)} \times \eta \quad (2-35)$$

$$\Delta w_k(n+1) = \eta_k \times \Delta w_k(n) + \alpha \times \frac{\delta E}{\delta w_k(n)} \quad (2-36)$$

Pada rumus diatas, kita dapat menyimpulkan bahwa variasi  $BPN_k$  *network weights* ( $w_{oj}^k$  dan  $w_{ol}^k$ ) lebih penting selama tingkat keanggotaan ( $MLC_k$ ) dari *data record*  $X_j$  ke *cluster*  $k^{ith}$  bernilai tinggi. Jika nilai dari tingkat keanggotaan ( $MLC_k$ ) dari *data record*  $X_j$  ke *cluster*  $k^{ith}$  bernilai mendekati nol, maka perubahan di dalam  $BPN_k$  *networks weights* sangat *minimal*.

### 2.2.3. Pengukuran Performa / Performance Measures

Untuk mengkoreksi, mengukur, dan menganalisa hasil dari prediksi pada penelitian ini, dilakukan perhitungan sebagai berikut :

a. *Root Means Square Error* (RMSE)

RMSE adalah aturan skoring kuadrat yang mengukur besarnya rata-rata kesalahan. RMSE memberikan bobot yang relatif tinggi untuk kesalahan yang besar. ini berarti RMSE yang paling berguna ketika kesalahan besar sangat tidak diinginkan.

$$RMSE = \sqrt{\frac{1}{N} \sum_{t=1}^N (Y_t - P_t)^2}, \quad (2-37)$$

b. *Means Absolute Percentage Error* (MAPE)

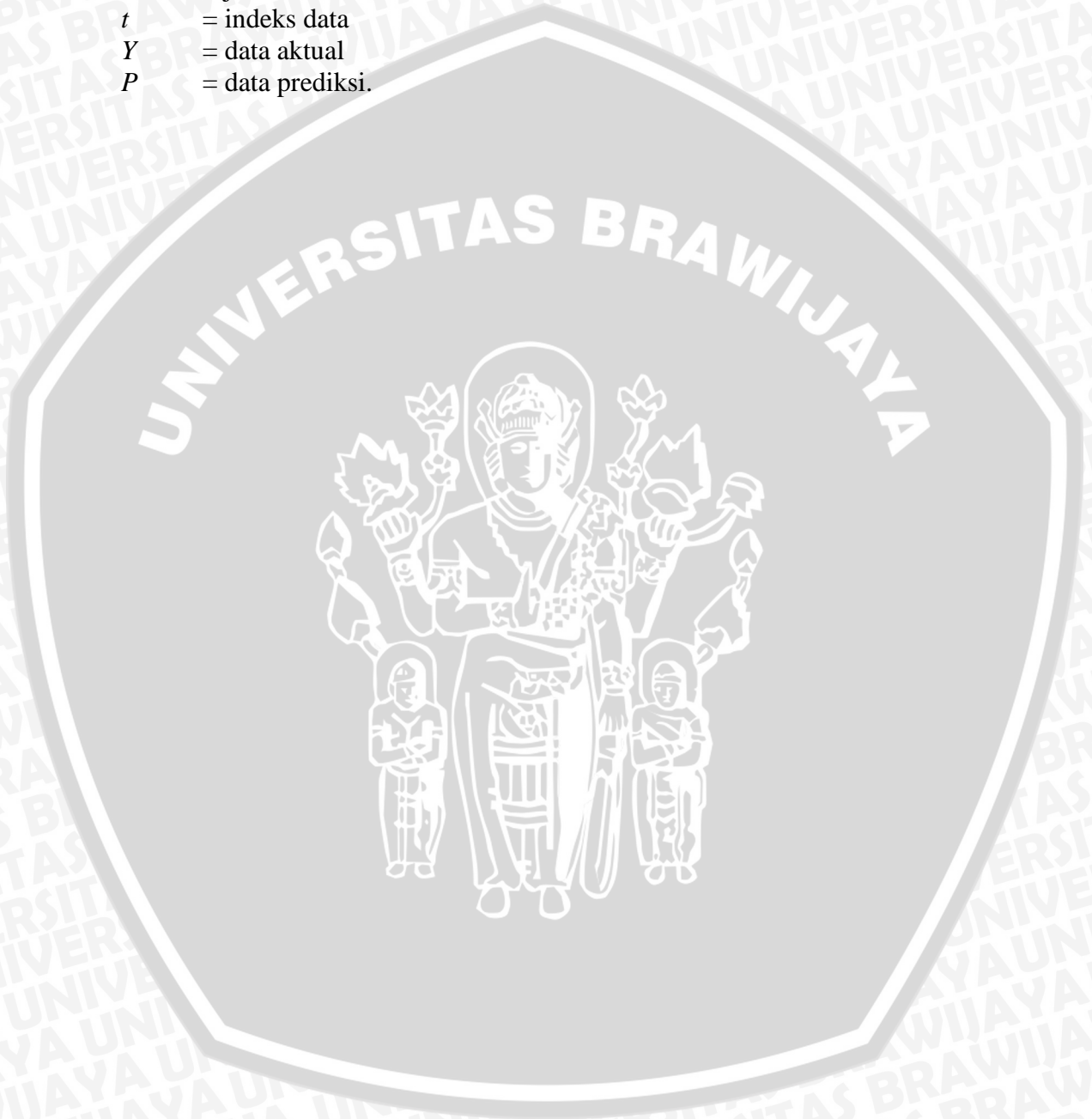
MAPE atau juga dikenal sebagai *Mean Absolute Percentage Deviation* (MAPD), adalah ukuran dari akurasi metode untuk membangun nilai-nilai *time series* pada statistik, khususnya dalam estimasi *trend*. Biasanya akurasi direpresentasikan dalam bentuk persentase.



$$\text{MAPE} = 100 \times \frac{1}{N} \sum_{t=1}^N \frac{|Y_t - P_t|}{Y_t}, \quad (2-38)$$

dimana :

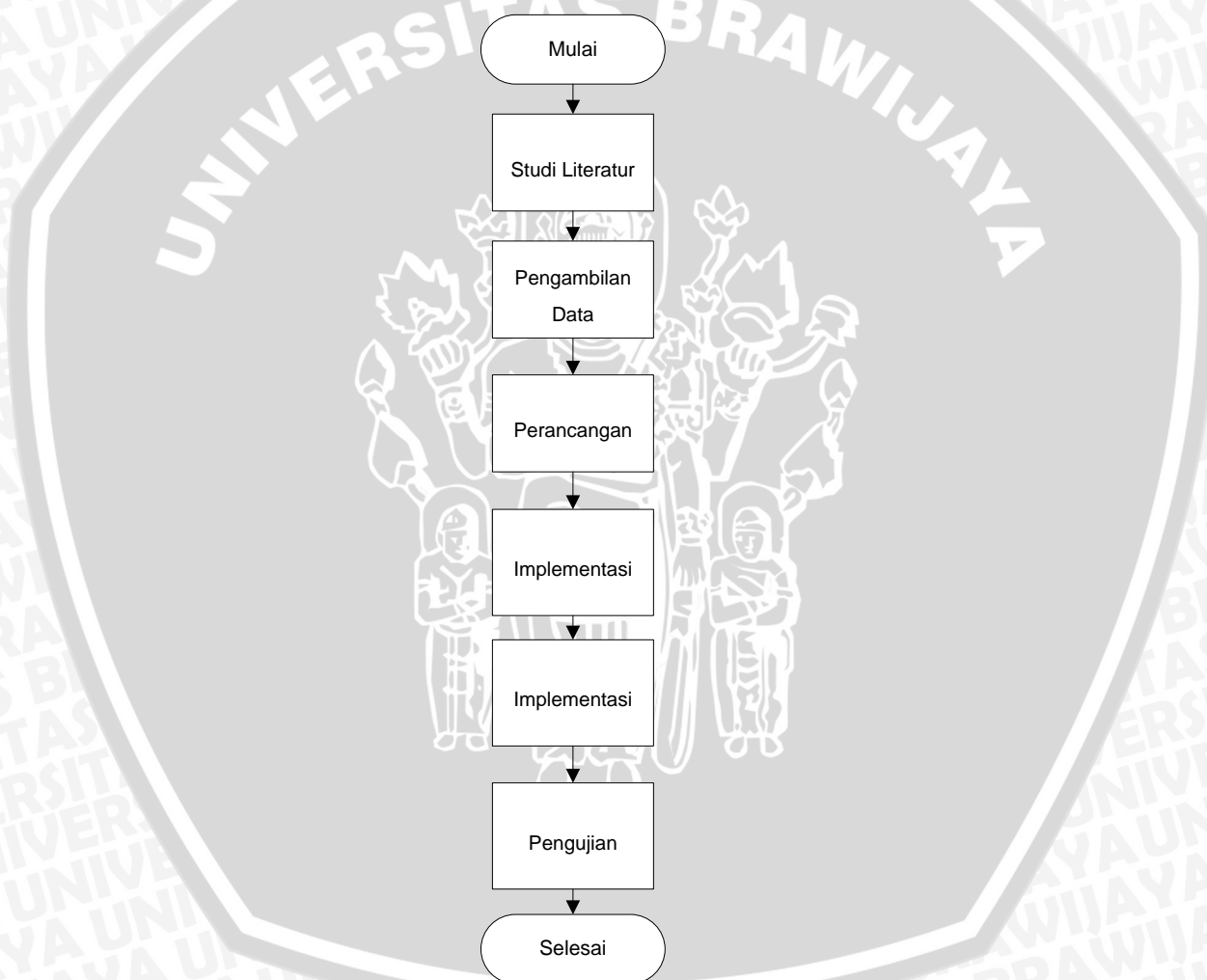
- $N$  = jumlah data
- $t$  = indeks data
- $Y$  = data aktual
- $P$  = data prediksi.



## BAB III

### METODE PENELITIAN

Pada bab ini akan dibahas metode-metode yang akan digunakan dalam penelitian yaitu studi literatur, metode pengambilan data, analisis kebutuhan, perancangan sistem, implementasi, pengujian dan analisis, dan pengambilan kesimpulan. Tahapan dalam proses penelitian ini dapat dilihat secara jelas pada Gambar 3.1 yang menunjukkan rencana atau struktur penelitian yang akan digunakan untuk memecahkan permasalahan dalam penelitian ini.



**Gambar 3.1** Diagram alir metode penelitian

### 3.1. Studi Literatur

Studi literatur menjelaskan dasar teori yang digunakan untuk menunjang penulisan tugas akhir. Teori-teori pendukung tersebut meliputi:

1. Epidemiologi Demam Berdarah
2. *Fuzzy C-Means Clustering*
3. *Backpropagation Neural Networks*

### 3.2. Penyusunan Tinjauan Pustaka

Penyusunan kajian Pustaka berasal dari penelitian yang berjudul '*An improved approach based on fuzzy clustering and Back-propagation Neural Networks with adaptive learning rate for sales forecasting: Case study of PCB industry*'. Pada penelitian tersebut metode FCBPN diterapkan pada peramalan penjualan di Industri PCB. Penelitian ini akan mengadaptasi metode yang digunakan pada penelitian sebelumnya untuk memprediksi kasus demam berdarah berdasarkan kriteria yang telah ditentukan.

Penyusunan dasar teori dilakukan setelah mendapatkan referensi yang tepat untuk mendukung penulisan penelitian ini. Penyusunan dilakukan dengan metode prediksi *Fuzzy C-Means Clustering Backpropagation Neural Networks* (FCBPN).

### 3.3. Metode Pengambilan Data

Data yang digunakan pada penelitian ini adalah data kasus dan faktor demam berdarah. Data tersebut diambil dari data statistik tahun 2010-2012 di Dinas Kesehatan Kabupaten Situbondo. Data faktor yang digunakan untuk prediksi kasus demam berdarah adalah Angka Bebas jentik, Kasus Demam Berdarah, Curah Hujan, dan Jumlah Hari Hujan.

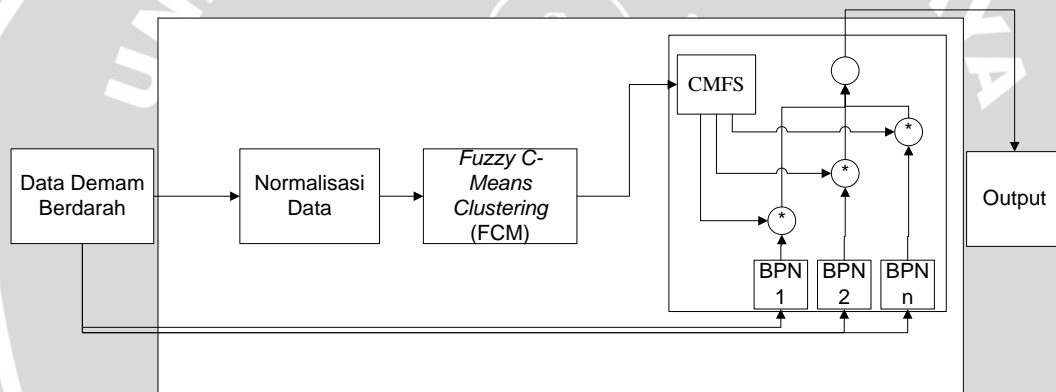
### 3.4. Analisis Kebutuhan

Analisa kebutuhan bertujuan untuk menganalisis dan mendapatkan semua kebutuhan yang diperlukan dalam pengimplementasian metode FCBPN. Analisis kebutuhan disesuaikan dengan lokasi dan variabel penelitian, menentukan kebutuhan data yang akan digunakan, dan mempersiapkan alat dan bahan penelitian. Secara keseluruhan, kebutuhan yang digunakan dalam pembuatan metode FCBPN ini meliputi :

1. Kebutuhan *Hardware*, meliputi:
  - a. Komputer
2. Kebutuhan *Software*, meliputi:
  - a. *XAMPP*
  - b. *PHP Framework*

### 3.5. Perancangan Prediksi

Perancangan menjelaskan bentuk model data dan proses alur metode yang direpresentasikan dalam bentuk diagram dan uji coba data. data yang digunakan adalah data curah hujan dan kasus demam berdarah. Metode yang digunakan adalah *Fuzzy C-Means Clustering Backpropagation Neural Networks*(FCBPN). Secara umum diagram blok dari FCBPN adalah dapat dilihat sebagai berikut :



**Gambar 3.2** Block Diagram FCBPN

### 3.6. Implementasi

Tahap implementasi menjelaskan bagaimana implementasi metode FCBPN.

Tahap-tahap implementasi yang dilakukan antara lain:

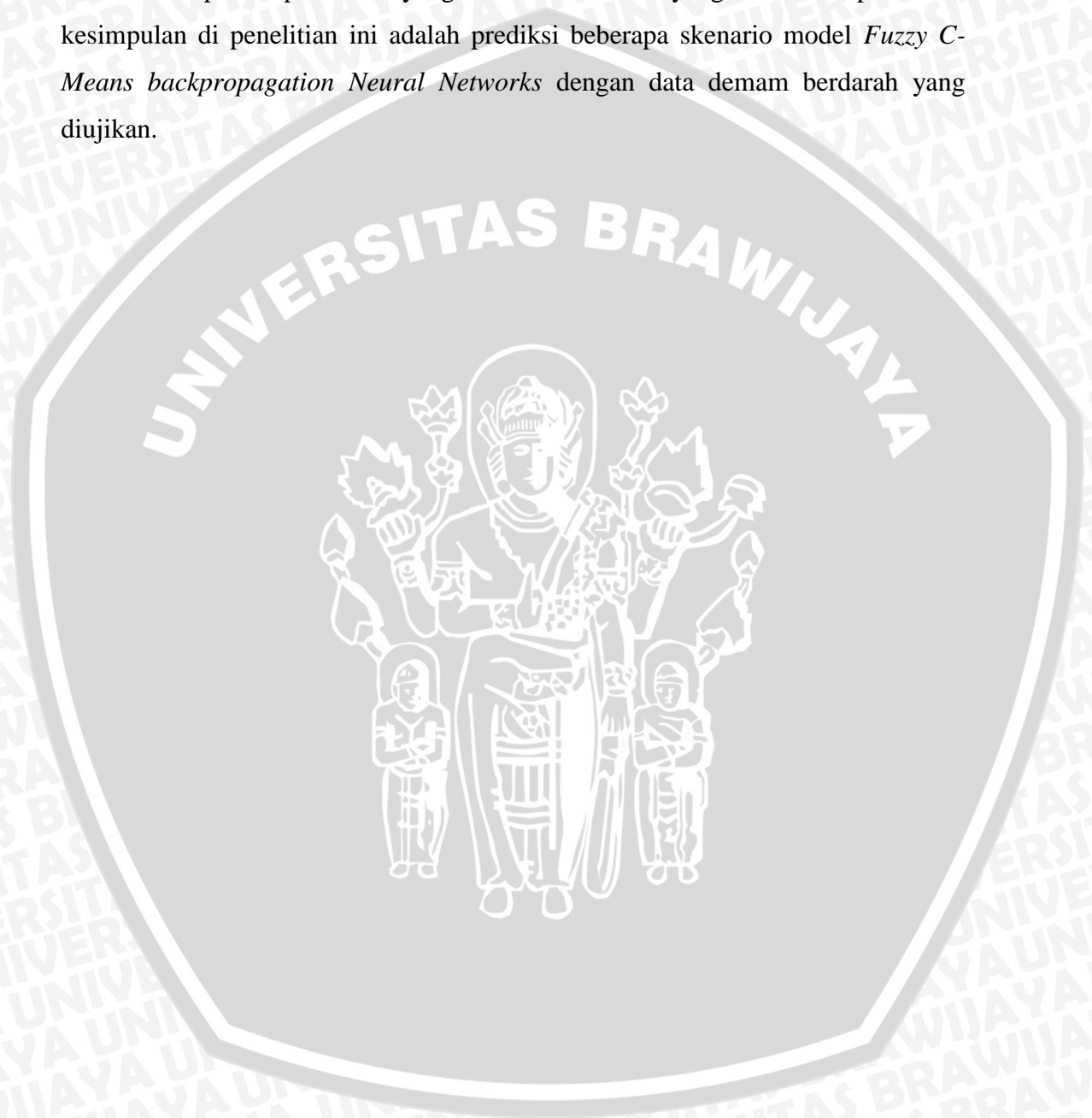
- Membuat basis data dengan *Mysql* sesuai dengan desain basis data dalam perancangan.
- mengimplementasikan metode yang digunakan dengan menggunakan bahasa pemrograman PHP.

### 3.7. Pengujian dan Analisis

Pada tahap ini dilakukan pengujian *Membership Levels Cluster* dan Pengujian Akurasi dengan metode RMSA dan MAPE. Analisis yang dilakukan berdasarkan 2 jenis pengujian tersebut.

### 3.8. Pengambilan Kesimpulan

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan, implementasi dan pengujian sistem telah selesai dilakukan. Kesimpulan diambil dari seluruh proses penelitian yang dilakukan. Acuan yang mendasari penarikan kesimpulan di penelitian ini adalah prediksi beberapa skenario model *Fuzzy C-Means backpropagation Neural Networks* dengan data demam berdarah yang diujikan.



## BAB IV

### PERANCANGAN

Perancangan menjabarkan perancangan metode *Fuzzy C-Means backpropagation Neural Network*, perancangan manualisasi dan perancangan pengujian.

#### 4.1. Perancangan Prediksi dengan *Fuzzy C-Means backpropagation Neural Network*

Perancangan prediksi di penelitian ini terdiri dari perancangan model data dan perancangan metode *Fuzzy C-Means Backpropagation Neural Networks* (FCBPN).

##### 4.1.1. Perancangan Model Data

Perancangan model data pada penelitian ini terdiri dari 3 atribut, dengan 2 atribut sebagai kriteria, dan 1 atribut sebagai output yaitu Jumlah Penderita. Model data diperlihatkan pada Tabel 4.1.

**Tabel 4.1** Tabel Kriteria

Atribut	Keterangan
Curah Hujan (CH)	Angka
Jumlah Hari Hujan (JHH)	Angka

##### 4.1.2. Perancangan Algoritma *Fuzzy C-Means Clustering Back Propagation Neural Networks* (FCBPN)

Data yang digunakan dalam penelitian ini adalah data kasus demam berdarah dan curah hujan Kabupaten Situbondo pada tahun 2010 sampai dengan 2012. Parameter data yang akan memasuki proses pada metode *Fuzzy C-Means Clustering Back Propagation Neural Networks* (FCBPN) adalah Jumlah Kasus Demam Berdarah (K), Curah Hujan (CH) dan Jumlah Hari Hujan (JHH).

Pada tahap awal, parameter data akan memasuki fungsi normalisasi. Tujuannya adalah untuk mengoptimalkan dan mempercepat pada proses *fuzzy*

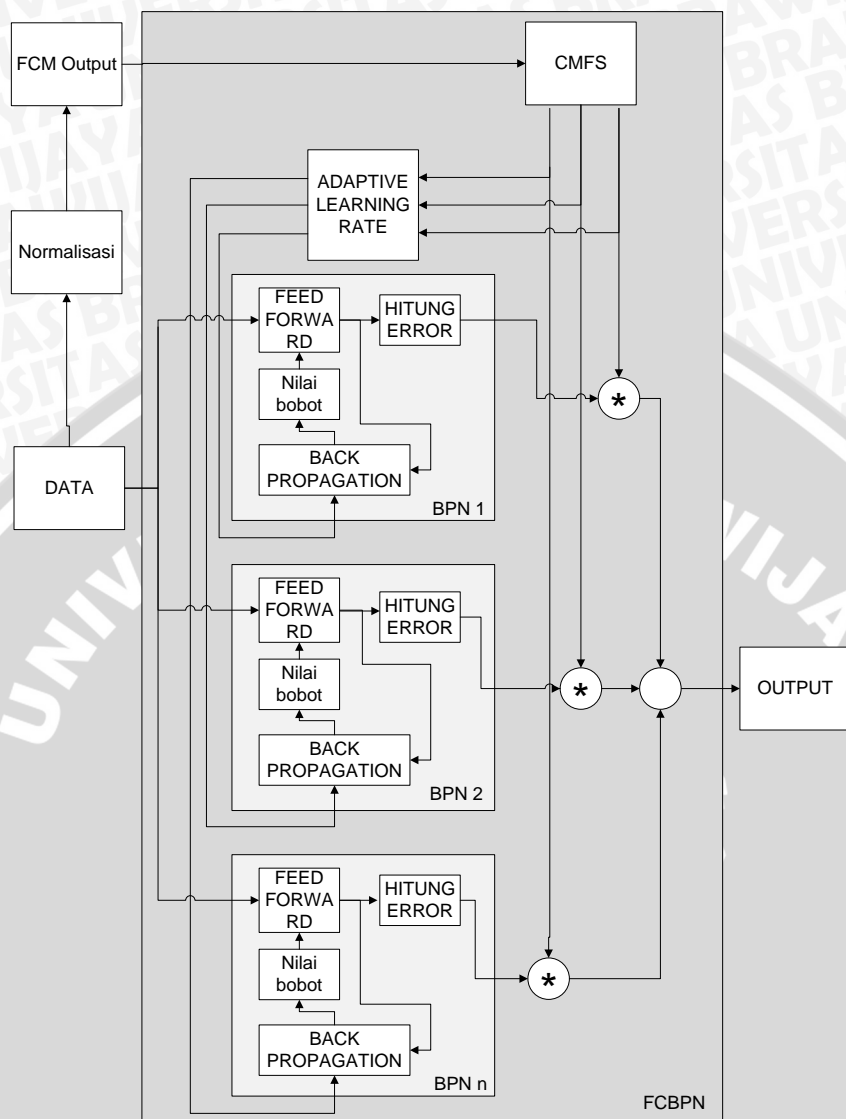
*clustering*. Parameter yang akan memasuki fungsi ini adalah Curah Hujan (CH) dan Jumlah Hari Hujan (JHH). Outputnya berupa nilai 0,1 sampai dengan 0,9.

Data Curah Hujan (CH) dan Jumlah Hari Hujan (JHH) akan memasuki proses *fuzzy clustering*. Tujuannya adalah untuk meng-*cluster* data CH dan JHH. Output pada proses ini berupa pusat *cluster* dan derajat keanggotaan dari masing-masing data.

Langkah selanjutnya setelah memasuki proses *fuzzy clustering*, data CH dan JHH akan memasuki proses *The Degree of Membership Levels*. Proses ini bertujuan untuk mempercepat proses klasifikasi. Inti dari output pada proses ini berupa N *cluster* dengan jumlah N yang sesuai dengan output pada proses sebelumnya yaitu *fuzzy clustering*.

Data *real* pada penelitian ini akan memasuki proses klasifikasi dengan *Back-Propagation Neural Networks* (BPN). Sebanyak N jumlah data yang sama akan memasuki proses BPN. Output dari proses BPN dan proses *The Degree of Membership Levels* akan memasuki proses terakhir yaitu proses yang akan menghasilkan output berupa prediksi jumlah kasus demam berdarah.

Perancangan metode FCBPN terdiri dari beberapa tahap yaitu *rescale* data, normalisasi data, *Membership Level Cluster* (MLC), *Fuzzy Clustering*, dan *Backpropagation Neural Network*. Secara umum arsitektur dari metode FCBPN diperlihatkan pada Gambar 4.1.

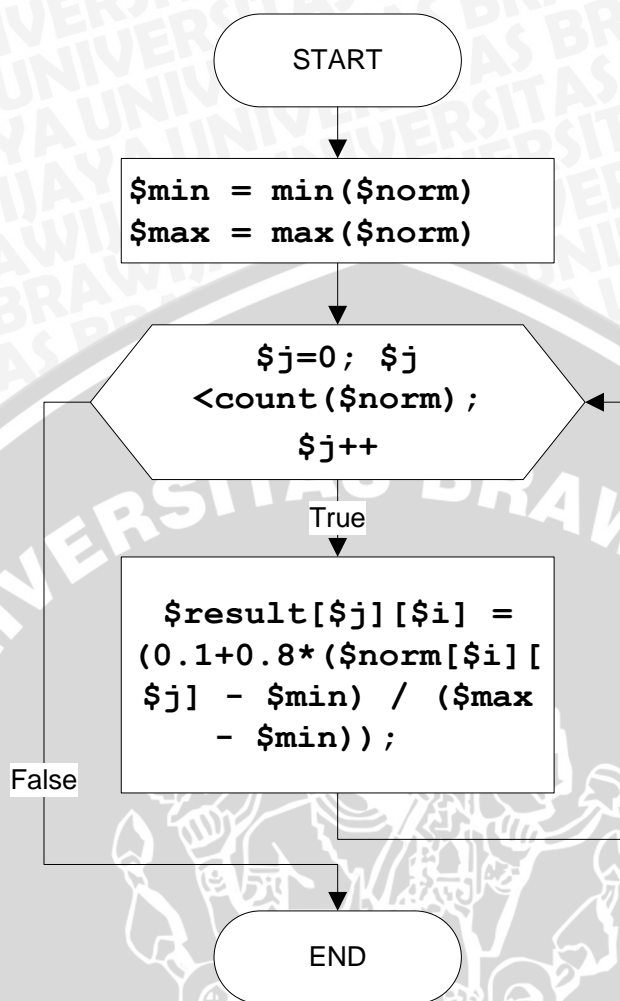


**Gambar 4.1** Block Diagram FCBPN

#### 4.1.2.1. Normalisasi Data

Tahap Normalisasi dilakukan untuk mengubah data menjadi angka diantara 0.1-0.9 seperti yang ditunjukkan pada Gambar 4.7 dibawah ini.

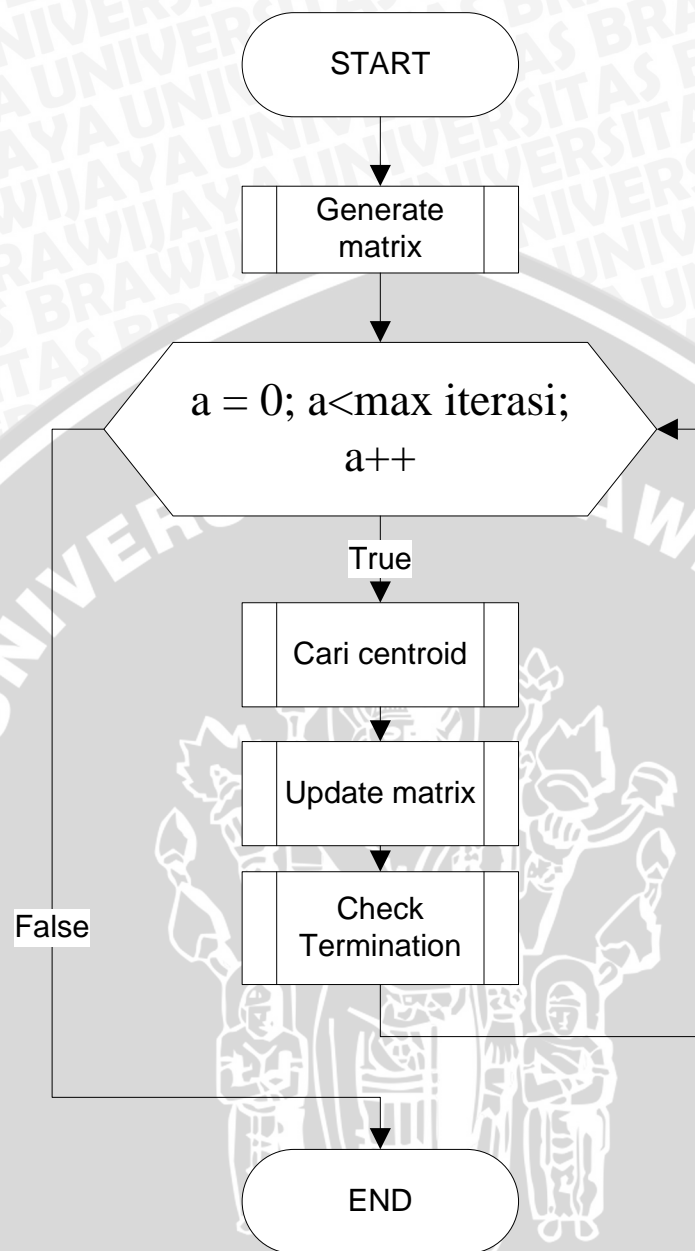




Gambar 4.2 Diagram Alir Normalisasi

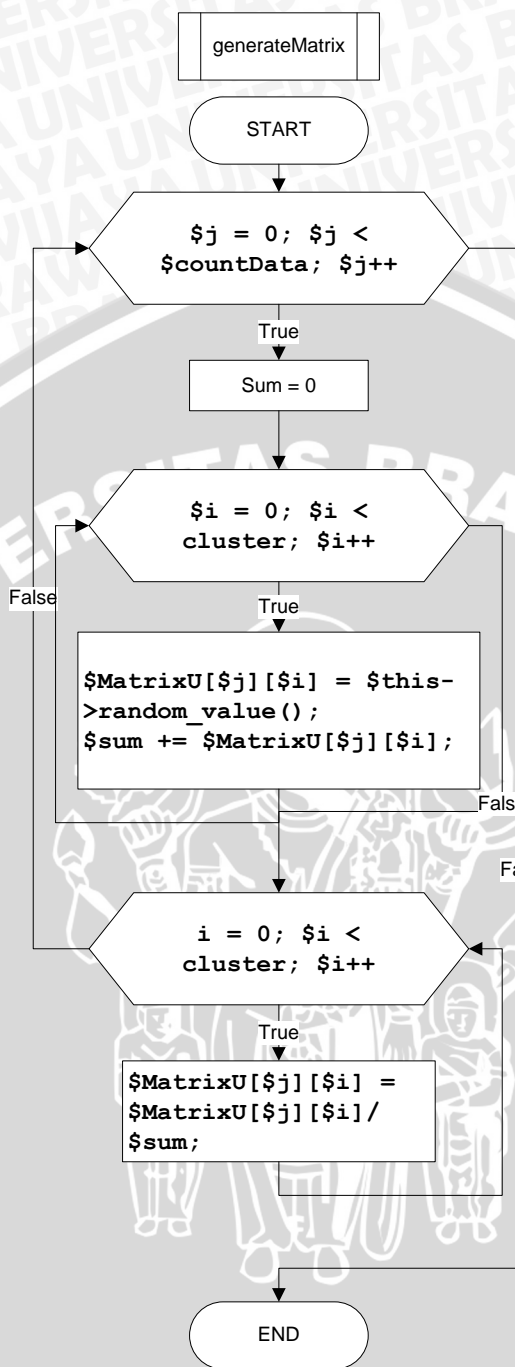
#### 4.1.2.2. Fuzzy C-Means Clustering (FCM)

Tahap *Fuzzy Clustering* digunakan untuk mendapatkan *cluster center* yang diperlukan untuk menentukan *Membership Levels Cluster(MLC)* dari masing masing data. Berikut ini adalah diagram perancangan dari FCM .



**Gambar 4.3** Fuzzy C-Means Clustering (FCM)

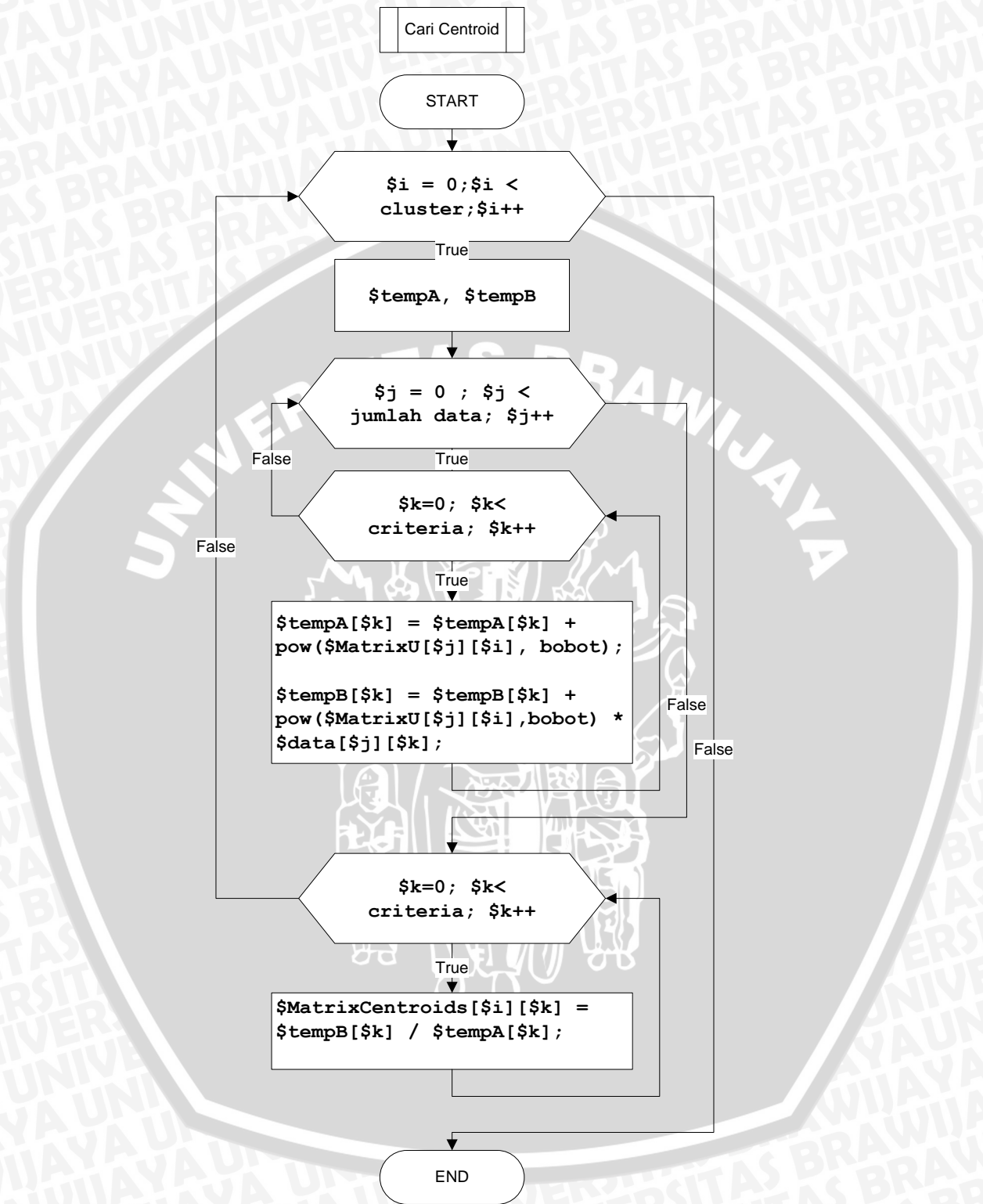
Diagram diatas menjelaskan aliran utama dari FCM. Data akan di iterasi dan dilakukan proses pencarian centroid, pembaharuan matrix (derajat keanggotaan), dan pengecekan kondisi penghentian.



Gambar 4.4 Generate Matrix

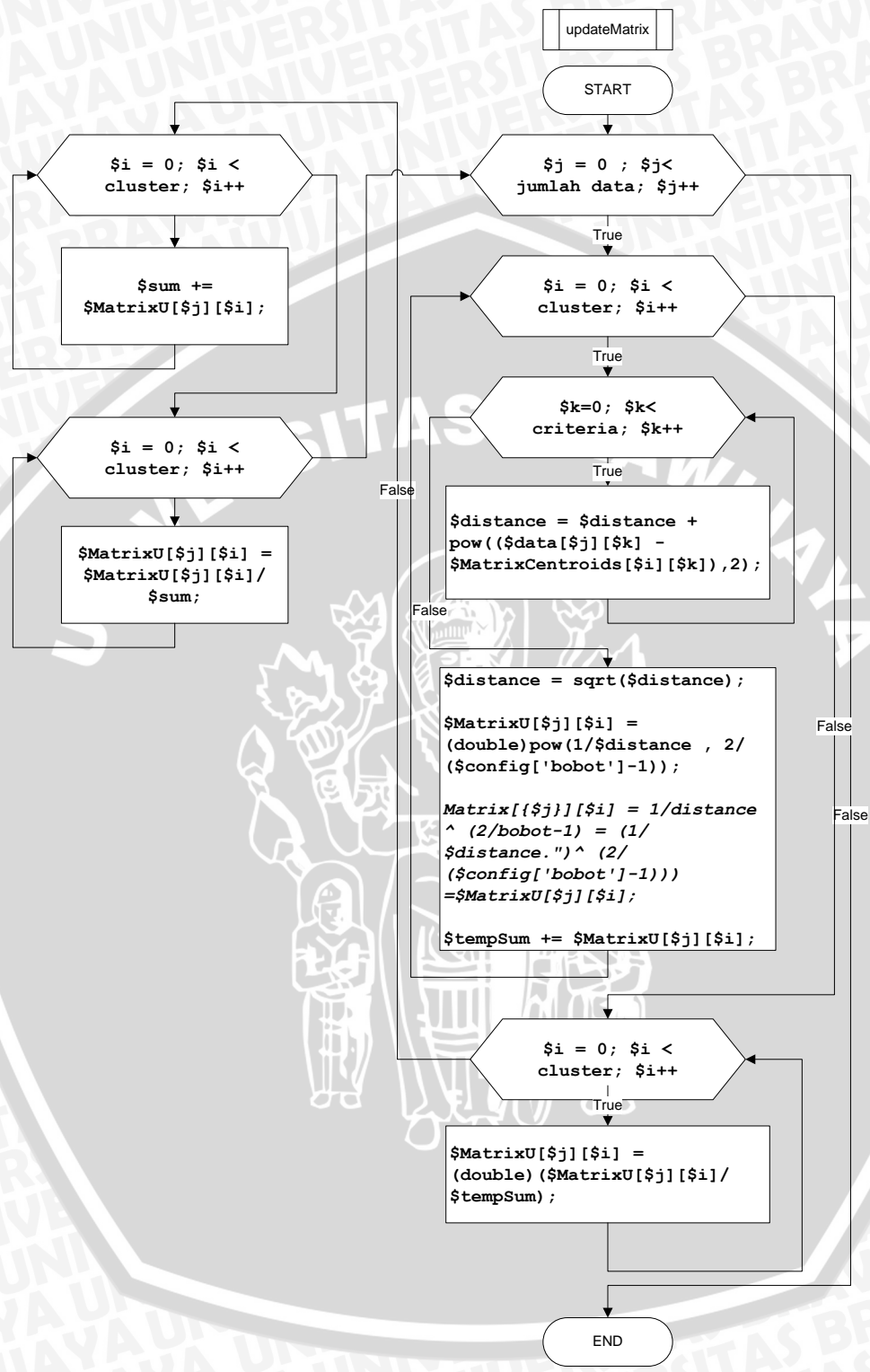
Proses diatas adalah inisialisais derajat keanggotaan dari fuzzy clustering. Data dibuat dari nilai *random* yang kemudian dinormalisasi.





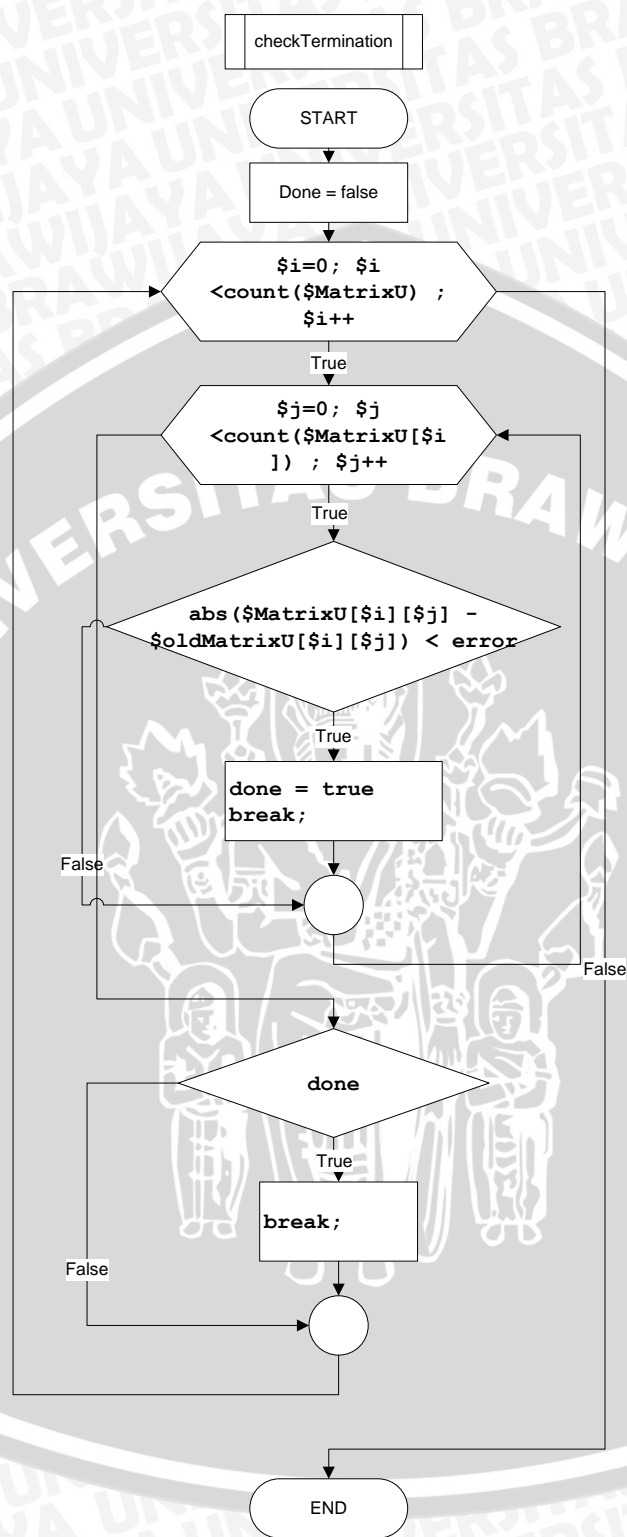
**Gambar 4.5** Diagram Alir Cari Centroid

Proses diatas adalah pencarian cluster center dari FCM. Data diolah sesuai dengan persamaan 2-21.



**Gambar 4.6** Diagram Alir Update Matrix

Proses diatas menjelaskan alur dari pembaharuan derajat keanggotaan dari fuzzy clustering sesuai dengan persamaan 2-22.



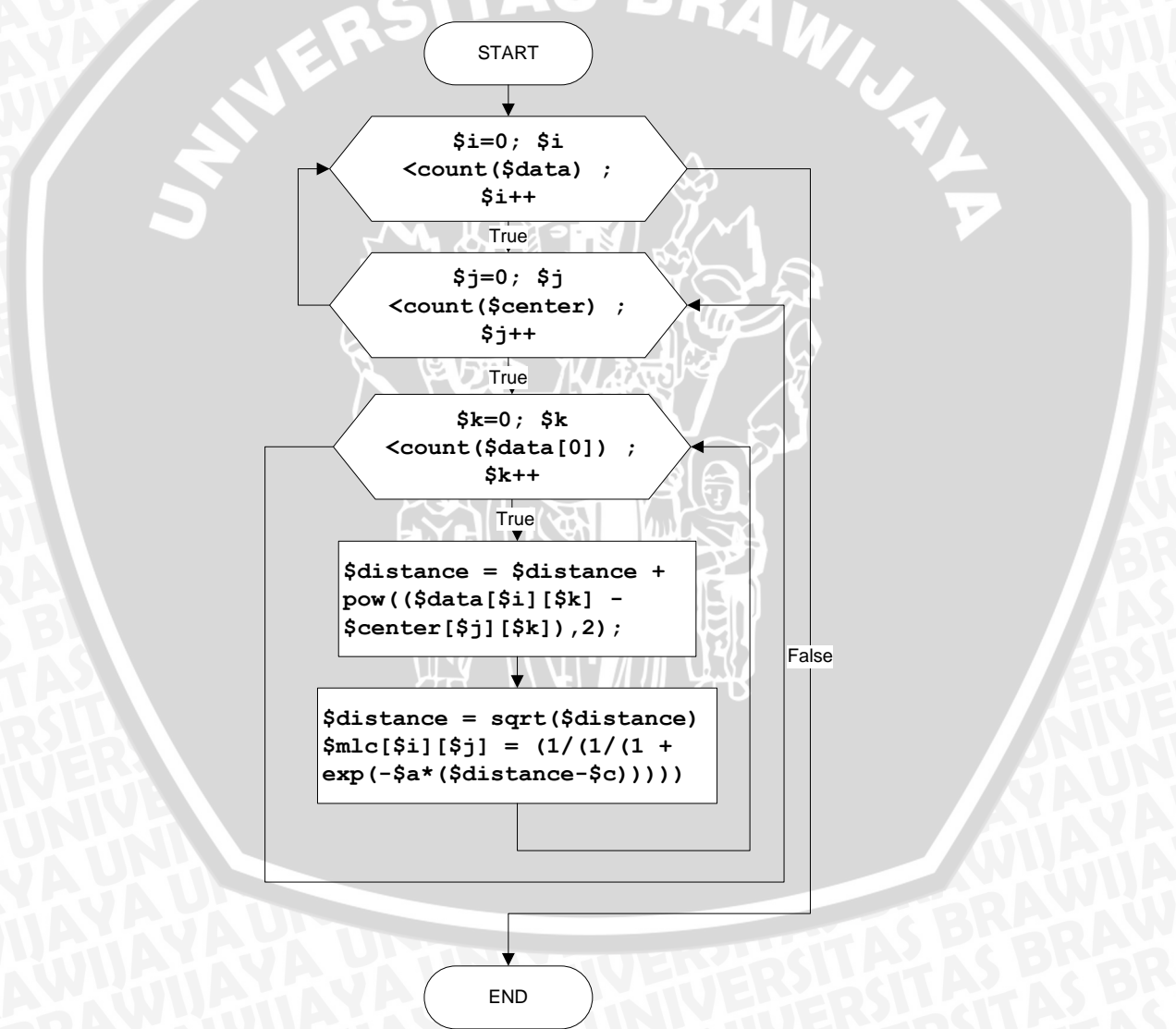
Gambar 4.7 Diagram Alir Check Termination



Proses diatas menjelaskan pengecekan kondisi penghentian dari fuzzy clustering yaitu saat jarak derajat keanggotaan lama dan baru sudah lebih kecil daripada *error* sesuai dengan persamaan 2-13.

**4.1.2.3. Membership Levels Cluster (MLC)**

Tahap Membership Levels Cluster digunakan untuk membantu mempercepat proses training dari Neural Network. Tiap-tiap jarak antara *cluster center* data dan data inputan didapatkan untuk menjadi input dari proses *Neural Network*

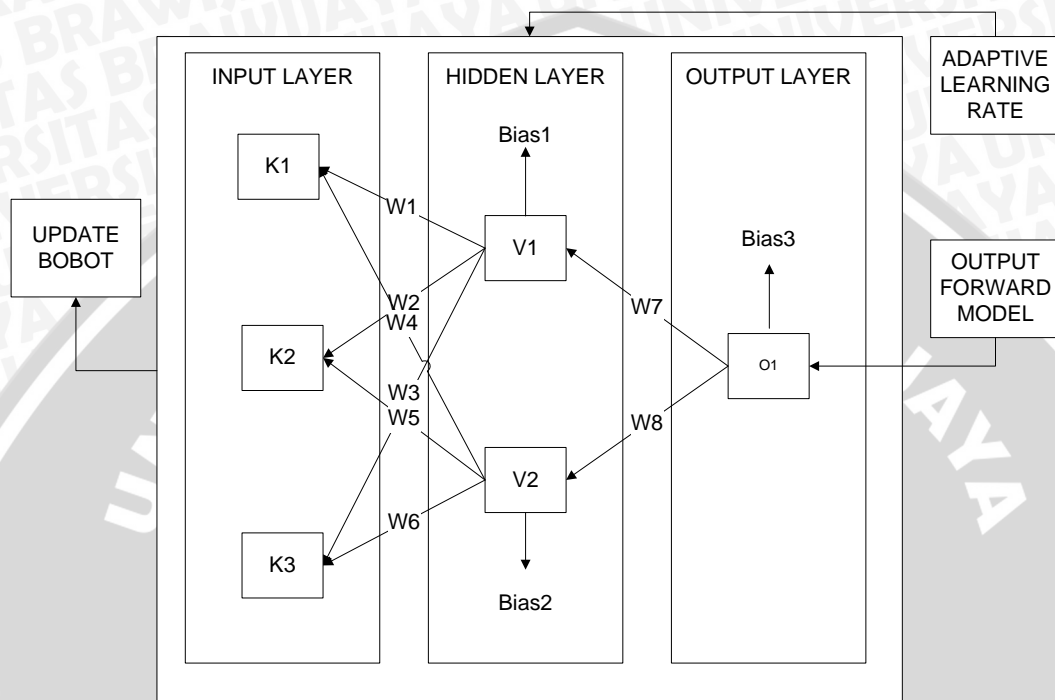


**Gambar 4.8** Diagram Alir Membership Level Cluster



#### 4.1.2.4. Backpropagation Neural Networks (BPN)

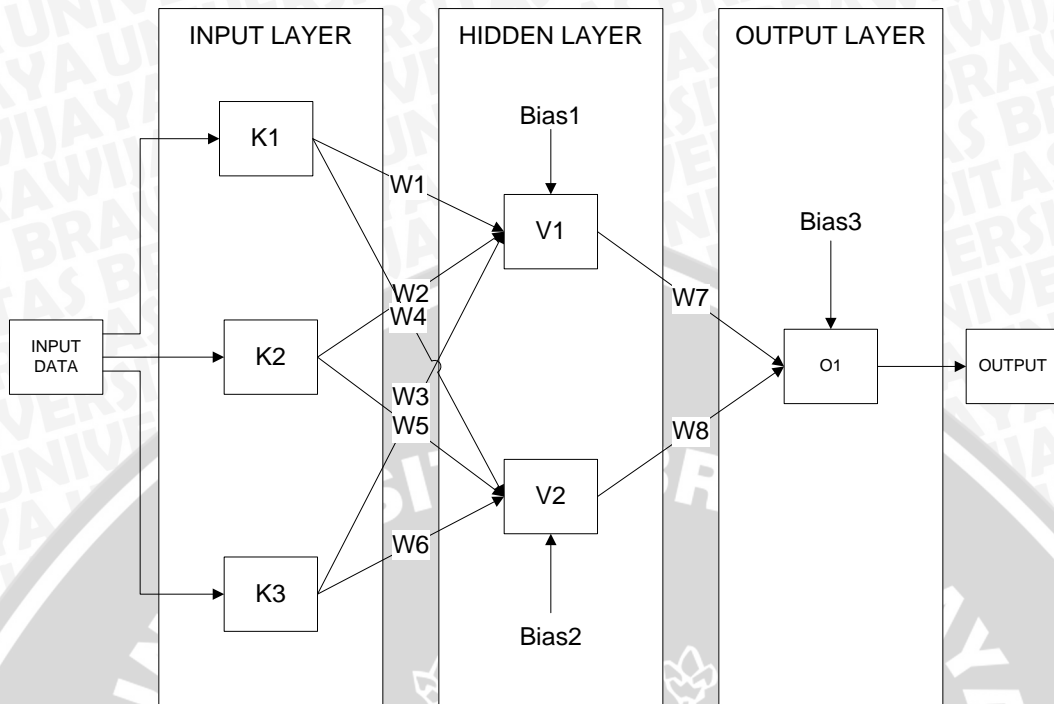
Tahap ini digunakan untuk kalsifikasi yang akan menghasilkan prediksi kasus demam berdarah. Berikut adalah digram blok dari proses BPN



**Gambar 4.9 Block Diagram Backpropagation**

Output  $MLC_k$  akan digunakan sebagai data input pada masing-masing *Backpropagation Neural Network* (BPN ke k). Pada proses Pembelajaran akan dilakukan iterasi untuk pembaharuan bobot yang mengarah pada model dengan nilai kesalahan minimum. Gambar diatas menunjukkan detail arsitektur pembelajaran dari FCBPN.





**Gambar 4.10** Block Diagram Feedforward

Setelah melalui tahap pembelajaran, data uji akan masuk untuk ke masing masing BPN yang bobotnya sudah diperbaharui mengikuti model dari data training. Output pada masing masing BPN akan diproses menghasilkan hasil prediksi.

**4.2. Perancangan Manualisasi Metode Fuzzy C-Means Clustering Backpropagation Neural Networks (FCBPN)**

Perancangan manualisasi menjelaskan perhitungan manual dari metode FCBPN. Pengujian dilakukan dengan menggunakan data set awal seperti pada tabel 4.1 berikut. Data Curah Hujan(CH) dan Jumlah Hari Hujan(JHH) akan menjadi data kriteria sedangkan kasus(K) akan menjadi *output* pada klasifikasi Backpropagation Neural Networks. Awalnya data *output* akan di *rescale* untuk mempercepat proses metode FCBPN.



**Tabel 4.2** Tabel Input data training

No	Curah Hujan(CH)	Jumlah Hari hujan(JHH)	Kasus (K)	Rescale Kasus(K)
1	322	15	18	0.62068965517241
2	74	13	21	0.72413793103448
3	81	9	29	1
4	71	10	18	0.62068965517241

Metode dibagi menjadi 2 bagian penting yaitu mencari nilai *Cluster Membership Fuzzy System*(CMFS) untuk setiap data input dan klasifikasi dengan *Parallel Backpropagation Neural Networks*.

Langkah untuk mencari nilai *Cluster Membership Fuzzy System*(CMFS) dibagi menjadi beberapa tahap yaitu Normalisasi, Fuzzy Clustering, dan *The Degree of Membership Levels Cluster* (MLC).

#### 4.2.1. Normalisasi Data

Data JHH dan CH akan dinormalisasi sesuai dengan persamaan 2-18 sehingga didapatkan hasil pada Tabel 4.3

**Tabel 4.3** Tabel Normalisasi Data Input

No	CH	JHH
1	0.9	0.9
2	0.10956175298805	0.633333333333333
3	0.13187250996016	0.1
4	0.1	0.233333333333333

#### 4.2.2. Clustering dengan Fuzzy C-Means (FCM)

A. Ditentukan konfigurasi parameter FCM sebagai berikut :

- ✓ cluster: 2
- ✓ bobot : 2
- ✓ maksimal Iterasi : 100000
- ✓ kesalahan minimal : 0.0001

B. Inisialisasi derajat keanggotaan/*MatrixU* dengan nilai *random* tanpa aturan

**Tabel 4.4** Tabel Inisialisasi *Matriks U*

No	Cluster 1	Cluster 2
1	0.79889984492495	0.20110015507505
2	0.15316140856133	0.84683859143867
3	0.49384717097827	0.50615282902173
4	0.45337708504238	0.54662291495762

C. Lakukan perhitungan berikut di setiap iterasi

**Tabel 4.5** Tabel *Centroid* iterasi 1

Cluster	Kriteria CH	Kriteria JHH
1	0.56672115740571	0.59544889039697
2	0.13609387842068	0.44639440606714

2. Perbaharui *MatrixU* sesuai rumus 2-22,  $U_{ik} = \frac{1}{\sum_{j=1}^c \frac{|x_{kj} - v_{ij}|}{|x_{kj} - v_{ij}|^{2/(w-1)}}$

**Tabel 4.6** Tabel *Matriks U* iterasi 1

No	Cluster 1	Cluster 2
1	0.79476525999881	0.20523474000119
2	0.14487201851745	0.85512798148255
3	0.21639637226657	0.78360362773343
4	0.11802680239344	0.88197319760656

3. Cek kondisi penghentian sesuai rumus 2-23  $\Delta = |U^t - U^{t-1}|$

Iterasi akan dihentikan apabila telah mencapai maksimal iterasi atau ditemukan  $error < \text{maksimal error}$  antara pengurangan salah satu dari *MatrixU* (iterasi) dengan *MatrixU* (iterasi-1). Kondisi penghentian pada scenario ini terjadi pada iterasi 3. Kondisi penghentian ditunjukkan dengan selisih data ke 1 kriteria 1 iterasi 3 (0.9996864506347) dengan data 1 kriteria 1 iterasi 2 (0.99961714415876) menghasilkan kesalahan 6.9306475936282E-53 yang berarti lebih kecil dari kesalahan minimal yang didefinisikan yaitu 0.0001.

Berikut ini adalah data *Centroid* dan *MatrixU* di iterasi 3 :

**Tabel 4.7** Tabel *Centroid* iterasi 3

Cluster	Kriteria CH	Kriteria JHH
1	0.88356652556799	0.89385548978634
2	0.11383462859263	0.2972191704405

**Tabel 4.8** Tabel Matriks  $U$  iterasi 3

No	Cluster 1	Cluster 2
1	0.9996864506347	0.00031354936529936
2	0.14487023889278	0.85512976110722
3	0.031771309377863	0.96822869062214
4	0.0040518158887913	0.99594818411121

#### 4.2.3. Degree of Membership Level ( $MLC_k$ )

Lakukan fungsi  $MLC$  untuk setiap data dengan mencari jarak antara *cluster center* dengan data. Perhitungan ditunjukkan dengan persamaan 2-26 dengan konstanta  $a = 50$  dan  $c = 0.5$ .

**Tabel 4.9** Tabel  $MLC$ 

Data	MLC Cluster 1	MLC Cluster 2
1	29949199543.384	1.0000000000222
2	1.0000001329009	3616.3137761791
3	1.0000000000001	3605184.6209126
4	1.0000000000004	2741218013.8885

#### 4.2.4. Backpropagation Neural Networks

Proses prediksi dilakukan dengan metode *Backpropagation Neural Networks*. Inputan dari tahap ini dapat dilihat di Tabel 4.10. Tahapan yang dilakukan di metode ini antara lain adalah tahap *feedforward*, *backpropagation*, dan penentuan *global output*.

**Tabel 4.10** Tabel  $MLC$ 

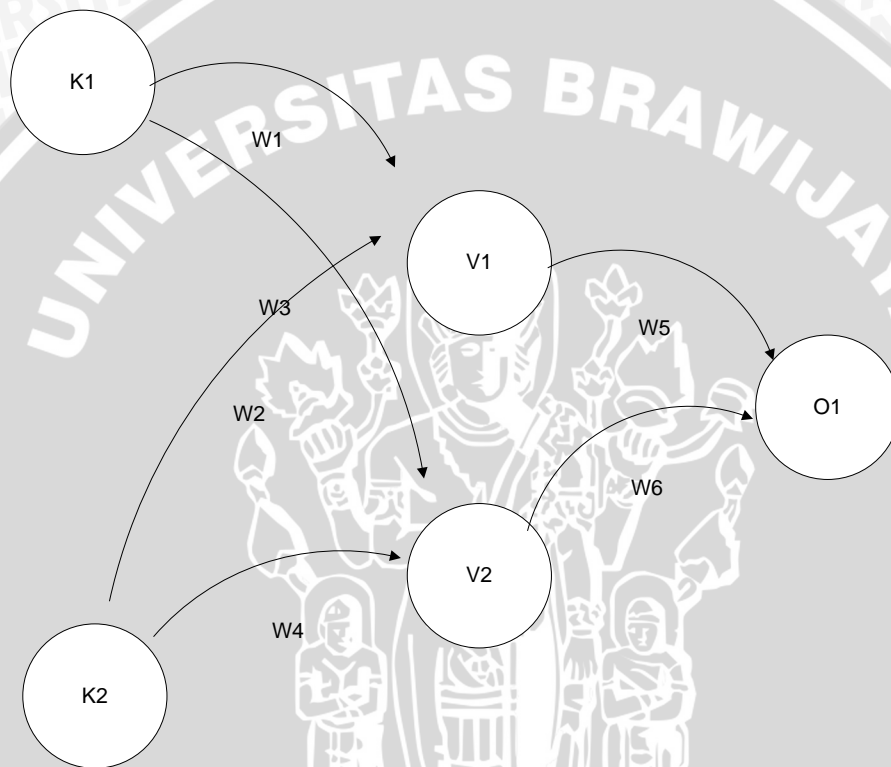
CH	JHH	MLC Cluster 1	MLC Cluster 2	Rescale K
322	15	29949199543.384	1.0000000000222	0.62068965517241
74	13	1.0000001329009	3616.3137761791	0.72413793103448
81	9	1.0000000000001	3605184.6209126	1
71	10	1.0000000000004	2741218013.8885	0.62068965517241

Konfigurasi *Backpropagation* pada kasus uji coba metode ini adalah sebagai berikut :

- ✓ Jumlah BPN = Jumlah Cluster = 2

- ✓  $error = 0.00000001$
- ✓  $max\ epoch = 100000$
- ✓  $learning\ rate = 0.01$
- ✓  $momentum = 0.1$
- ✓  $layers = 2,2,1$
- ✓ *Network Learning Rule : Delta Rule*
- ✓ *Fungsi Aktifasi : Sigmoid*

Arsitektur dari 2 BPN yang dibentuk dari konfigurasi dapat dilihat



**Gambar 4.11** Model BPN Pada Manualisasi

Inisialisasi Bobot awal dengan nilai *random* antara (-1) sampai 1 untuk setiap BPN

**Tabel 4. 11** Tabel Inisialisasi bobot awal BPN

Node Awal	Node Tujuan	Weight
K1	V1	-0.35874507889035
K2	V1	-0.16434217352824
Bias1	V1	-1
K1	V2	-0.93841364787744
K2	V2	0.18497268593402

Bias2	V2	-1
V2	O1	0.70732749412519
V2	O1	0.721549119541
Bias3	O1	-1

#### A. Feedforward

Data input (JHH dan CH) akan diproses sesuai persamaan 2-28 dan 2-29 sehingga didapat *weight* dan nilai node di iterasi 1 seperti ditunjukkan sebagai berikut :

**Tabel 4.12** Tabel *weight* BPN iterasi 1 cluster 1

Node Awal	Node Tujuan	nilai
K1	V1	-0.35874507889035
K2	V1	-0.16434217352824
Bias1	V1	-1
K1	V2	-0.93841364787744
K2	V2	0.18497268593402
Bias2	V2	-1
V2	O1	0.70732749412519
V2	O1	0.721549119541
Bias3	O1	-1

**Tabel 4.13** Tabel *weight* BPN iterasi 1 cluster 2

Node Awal	Node Tujuan	nilai
K1	V1	-0.35874507889035
K2	V1	-0.16434217352824
Bias1	V1	-1
K1	V2	-0.93841364787744
K2	V2	0.18497268593402
Bias2	V2	-1
V2	O1	0.70732749412519
V2	O1	0.721549119541
Bias3	O1	-1

**Tabel 4.14** Tabel node BPN iterasi 1 cluster 1

Node Awal	Nilai
K1	322
K2	15
V1	2.1241608495891E-52

V2	3.4696122864152E-131
O1	0.26894142137

**Tabel 4. 15** Tabel node BPN iterasi 1 *cluster 2*

Node Awal	Nilai
K1	322
K2	15
V1	2.1241608495891E-52
V2	3.4696122864152E-131
O1	0.26894142137

### B. Global Output

Selanjutnya menghitung global output berdasarkan persamaan 2-33,  $O_l =$

$$\frac{\sum_{k=1}^4 (MLC_k(X_j) \cdot o_l^k)}{\sum_{k=1}^4 MLC_k(X_j)}$$

Untuk iterasi 1 didapatkan Global Output sebagai berikut :

$$O_{\text{iterasi 1}} = \frac{((0.26894142137 * 29949199543.384) + (0.26894142137 * 1.0000000000222))}{29949199543.384 + 1.0000000000222}$$

$$= 0.26894142137$$

### C. Means Square Error (MSE)

Setelah didapatkan Global Output, maka error dapat dicari dengan menggunakan *Means Square Error* (MSE) sesuai dengan persamaan 2-34. MSE yang dihasilkan di iterasi 1 adalah 0.061863409991561 yang berarti masih lebih besar daripada minimal kesalahan yang diinginkan (0.00000001) sehingga iterasi dilanjutkan.

### D. Backpropagation

Langkah selanjutnya adalah proses pembelajaran *Backpropagation*. Pada metode ini, pertama-tama dicari terlebih dahulu *adaptive learning rate* (alr) di

masing-masing BPN sesuai persamaan 2-35,  $\eta_k = \frac{MLC_k(X_j)}{\sum_{k=1}^4 MLC_k(X_j)} \times \eta$ .

- alr (iterasi 1 cluster 1) :  $0.1 * 29949199543.384 / 29949199544.384 = 0.099999999996661$

$$- \text{alr (iterasi 1 cluster 2)} : 0.1 * 1.000000000222 / 29949199544.384 = 3.3389874027857E-12$$

Selanjutnya, lakukan proses *backpropagation* sesuai persamaan 2-36 untuk masing masing BPN. Berikut adalah proses di BPN cluster 1:

a. Mencari delta di *Hidden Layer* dan *Output layer*

Nilai *output* (t) untuk data 1 adalah nilai kasus di data 1 sehingga  $t = 0.62068965517241$ .

$$- \text{Delta pada output} = O \times (1-O) \times (t - O) = 0.26894142137 \times (1-0.26894142137) \times (0.62068965517241 - 0.26894142137) = 0.06915790026217$$

- Delta pada *Hidden Layer* :

*Hidden Layer* di Layer ke 1 node ke 1 :

$$\text{sum} = \text{delta}[2][0] \times \text{node}[2][0][0] = 0.06915790026217 \times 0.70732749412519 = 0.048917284291401$$

$$\text{delta}[1][0] = O[1][0] \times (1-O[1][0]) \times \text{sum} = 2.1241608495891E-52 \times (1-2.1241608495891E-52) \times 0.048917284291401 = 1.0390818016001E-53$$

*Hidden Layer* di Layer ke 1 node ke 2 :

$$\text{sum} = \text{delta}[2][0] \times \text{node}[2][0][1] = 0.06915790026217 \times 0.721549119541 = 0.049900822043473$$

$$\text{delta}[1][1] = O[1][1] \times (1-O[1][1]) \times \text{sum} = 3.4696122864152E-131 \times (1-3.4696122864152E-131) \times 0.049900822043473 = 1.7313650526425E-132$$

b. Penggunaan momentum

*Hidden Layer* di Layer ke 1 node ke 1 :

$$\text{weight}[0][0] = \text{momentum} \times \text{weight}(n-1)[0][0] = 0.01 \times 0 = -0.35874507889035$$

$$\text{weight}[0][1] = \text{momentum} \times \text{weight}(n-1)[0][1] = 0.01 \times 0 = -0.16434217352824$$

$$\text{bias} = \text{momentum} \times \text{weight}(n-1)[0][2] = 0.01 \times 0 = -1$$



*Hidden Layer di Layer ke 1 node ke 2 :*

$$\text{weight [1][0]} = \text{momentum} \times \text{weight (n-1)[1][0]} = 0.01 \times 0 = -0.93841364787744$$

$$\text{weight [1][1]} = \text{momentum} \times \text{weight (n-1)[1][1]} = 0.01 \times 0 = 0.18497268593402$$

$$\text{bias} = \text{momentum} \times \text{weight (n-1)[1][2]} = 0.01 \times 0 = -1$$

*Hidden Layer di Layer ke 2 node ke 1 :*

$$\text{weight [0][0]} = \text{momentum} \times \text{weight (n-1)[0][0]} = 0.01 \times 0 = 0.70732749412519$$

$$\text{weight [0][1]} = \text{momentum} \times \text{weight (n-1)[0][1]} = 0.01 \times 0 = 0.721549119541$$

$$\text{bias} = \text{momentum} \times \text{weight (n-1)[0][2]} = 0.01 \times 0 = -1$$

c. *Perbaharui weight*

*Hidden Layer di Layer ke 1 node ke 1 :*

$$\text{weight (n-1) [0][0]} = \text{alr} \times \text{delta[1][0]} \times \text{o[0][0]} = 0.09999999996661 \times$$

$$1.0390818016001\text{E-}53 \times 322 = 3.3458434010407\text{E-}52$$

$$\text{weight[0][0]} = \text{weight[0][0]} + 3.3458434010407\text{E-}52 = -0.35874507889035$$

$$\text{weight (n-1) [0][1]} = \text{alr} \times \text{delta[1][0]} \times \text{o[0][1]} = 0.09999999996661 \times$$

$$1.0390818016001\text{E-}53 \times 15 = 1.5586227023482\text{E-}53$$

$$\text{weight[0][1]} = \text{weight[0][1]} + 1.5586227023482\text{E-}53 = -0.16434217352824$$

$$\text{weight (n-1) [0][2]} = \text{alr} \times \text{delta[1][0]} = 0.09999999996661 \times$$

$$1.0390818016001\text{E-}53 = 1.0390818015654\text{E-}54$$

$$\text{bias[0][2]} = \text{bias[0][2]} + 1.0390818015654\text{E-}54 = -1$$

*Hidden Layer di Layer ke 1 dan node ke 2 :*

$$\text{weight (n-1) [1][0]} = \text{alr} \times \text{delta[1][1]} \times \text{o[0][0]} = 0.09999999996661 \times$$

$$1.7313650526425\text{E-}132 \times 322 = 5.5749954693228\text{E-}131$$

$$\text{weight[1][0]} = \text{weight[1][0]} + 5.5749954693228\text{E-}131 = (-0.93841364787744)$$

$$\text{weight (n-1) [1][1]} = \text{alr} \times \text{delta[1][1]} \times \text{o[0][1]} = 0.09999999996661 \times$$

$$1.7313650526425\text{E-}132 \times 15 = 2.5970475788771\text{E-}132$$

$$\text{weight}[1][1] = \text{weight}[1][1] + 2.5970475788771\text{E-}132 = 0.18497268593402$$

$$\text{weight (n-1) } [1][2] = \text{alr} \times \text{delta}[1][1] = 0.099999999996661 \times$$

$$1.7313650526425\text{E-}132 = 1.7313650525847\text{E-}133$$

$$\text{bias}[1][2] = \text{bias}[1][2] + 1.7313650525847\text{E-}133 = -1$$

*Hidden Layer* di *Layer* ke 2 dan node ke 1 :

$$\text{weight (n-1) } [0][0] = \text{alr} \times \text{delta}[2][0] \times \text{o}[1][0] = 0.099999999996661 \times$$

$$0.06915790026217 \times 2.1241608495891\text{E-}52 = 1.4690250417178\text{E-}54$$

$$\text{weight}[0][0] = \text{weight}[0][0] + 1.4690250417178\text{E-}54 = 0.70732749412519$$

$$\text{weight (n-1) } [0][1] = \text{alr} \times \text{delta}[2][0] \times \text{o}[1][1] = 0.099999999996661 \times$$

$$0.06915790026217 \times 3.4696122864152\text{E-}131 = 2.3995110044429\text{E-}133$$

$$\text{weight}[0][1] = \text{weight}[0][1] + 2.3995110044429\text{E-}133 = 0.721549119541$$

$$\text{weight (n-1) } [0][2] = \text{alr} \times \text{delta}[2][0] = 0.099999999996661 \times 0.06915790026217$$

$$= 0.0069157900259861$$

$$\text{bias}[0][2] = \text{bias}[0][2] + 0.0069157900259861 = -0.99308420997401$$

Dari proses *Backpropagation* iterasi 1 dengan menggunakan data 1 didapatkan pembaruan *weight* sebagai berikut :

**Tabel 4.17** Tabel *weight* iterasi 1 *cluster* 1

Node Awal	Node Tujuan	nilai
K1	V1	-0.35874507889035
K2	V1	-0.16434217352824
Bias1	V1	-1
K1	V2	-0.93841364787744
K2	V2	0.18497268593402
Bias2	V2	-1
V2	O1	0.70732749412519
V2	O1	0.721549119541
Bias3	O1	-1

Iterasi akan dilanjutkan hingga didapatkan MSE < minimal kesalahan. Pada kasus ujicoba, iterasi yang dilakukan sebanyak 5316 dengan MSE = 9.9938043907386E-9. *Weight* yang didapatkan dari iterasi terakhir dapat dilihat sebagai berikut :

**Tabel 4.16** Tabel *weight* iterasi 5316 *cluster 1*

Node Awal	Node Tujuan	nilai
K1	V1	-0.35874507889035
K2	V1	-0.16434217352824
Bias1	V1	-1
K1	V2	-0.93841364787744
K2	V2	0.18497268593402
Bias2	V2	-1
V2	O1	0.70732749412519
V2	O1	0.721549119541
Bias3	O1	0.49187936003401

**Tabel 4.17** Tabel *weight* iterasi 5316 *cluster 2*

Node Awal	Node Tujuan	nilai
K1	V1	-0.35874507898399
K2	V1	-0.16434217354177
Bias1	V1	-1.00000000000012
K1	V2	-0.93841364787744
K2	V2	0.18497268593402
Bias2	V2	-1
V2	O1	0.70732749412341
V2	O1	0.721549119541
Bias3	O1	1.2752296890698

Dengan menggunakan 2 model *weight* diatas. Dilakukan proses *testing* data. data yang digunakan untuk test adalah data training 1 yaitu :

**Tabel 4.18** Tabel Data Uji

No	Curah Hujan(CH)	Jumlah Hujan(JHH)	Hari	Kasus (K)	Rescale Kasus(K)
1	322	15		18	0.62068965517241

### 1. Feedforward

Proses ini seperti proses pada *training* data, tapi tanpa adanya perubahan *weight*.

Untuk BPN 1 :

LAYER : 1

$$\text{data}[0][0] * \text{weight}[0][0] = 322 * -0.35874507889035 = -115.51591540269$$

$$\text{data}[0][1] * \text{weight}[0][1] = 15 * -0.16434217352824 = -2.4651326029236$$

$$\text{output} = -118.98104800562, \text{sigmoid} = 2.1241608495891\text{E-}52$$

$$\text{data}[0][0] * \text{weight}[1][0] = 322 * -0.93841364787744 = -302.16919461654$$

$$\text{data}[0][1] * \text{weight}[1][1] = 15 * 0.18497268593402 = 2.7745902890103$$

$$\text{output} = -300.39460432753, \text{sigmoid} = 3.4696122864152\text{E-}131$$

LAYER : 2

$$\text{data}[1][0] * \text{weight}[0][0] = 2.1241608495891\text{E-}52 * 0.70732749412519 = 1.5024773708587\text{E-}52$$

$$\text{data}[1][1] * \text{weight}[0][1] = 3.4696122864152\text{E-}131 * 0.721549119541 = 2.5034956904115\text{E-}131$$

$$\text{output} = 0.49187936003401, \text{sigmoid} = 0.62054906149793$$

Untuk BPN 2 :

LAYER : 1

$$\text{data}[0][0] * \text{weight}[0][0] = 322 * -0.35874507898399 = -115.51591543285$$

$$\text{data}[0][1] * \text{weight}[0][1] = 15 * -0.16434217354177 = -2.4651326031266$$

$$\text{output} = -118.98104803597, \text{sigmoid} = 2.1241607851049\text{E-}52$$

$$\text{data}[0][0] * \text{weight}[1][0] = 322 * -0.93841364787744 = -302.16919461654$$

$$\text{data}[0][1] * \text{weight}[1][1] = 15 * 0.18497268593402 = 2.7745902890103$$

$$\text{output} = -300.39460432753, \text{sigmoid} => 3.4696122864152\text{E-}131$$

LAYER : 2

$$\text{data}[1][0] * \text{weight}[0][0] = 2.1241607851049\text{E-}52 * 0.70732749412341 = 1.5024773252434\text{E-}52$$

$$\text{data}[1][1] * \text{weight}[0][1] = 3.4696122864152\text{E-}131 * 0.721549119541 = 2.5034956904115\text{E-}131$$

$$\text{output} = 1.2752296890698, \text{sigmoid} = 0.78163666994939$$

## 2. Global Output

$$\begin{aligned} O_{\text{prediksi}} &= \frac{((0.62055 * 29949199543.384) + (0.78163666994939 * 1.0000000000222))}{29949199543.384 + 1.0000000000222} \\ &= 0.6205490615033 \end{aligned}$$

Skalikan kembali Global output untuk mendapatkan nilai Nilai Prediksi

$$\text{Prediksi} = 0.6205490615033 * 29 = 17.995922783596$$

### 4.3. Perancangan Skenario Pengujian Metode *Fuzzy C-Means Clustering Backpropagation Neural Networks (FCBPN)*

Berdasarkan data demam berdarah dan curah hujan yang didapat dari tahun 2009 – tahun 2012, maka untuk merancang pengujian dilakukan beberapa tahap diantaranya adalah sebagai berikut :

1. Pencarian *Cluster* terbaik dari jumlah cluster yang diuji adalah 2,3,4,5. Setelah didapatkan parameter *cluster* terbaik dari hasil nilai RMSE dan MAPE terendah maka nilai *cluster* tersebut akan digunakan sebagai nilai *cluster* untuk tahap selanjutnya.
2. Pencarian koefisien a dan c yang paling optimal dari fungsi sigmoid pada tahap metode *Membership Level Cluster (MLC)*. Pada pengujian ini dilakukan scenario pengujian nilai koefisien a dari 0.1 – 1 dan c dari 1 – 10. Dari hasil pengujian tersebut didapatkan nilai optimal koefisien a dan c yang digunakan sebagai hasil RMSE dan MAPE terbaik.

## BAB V

### IMPLEMENTASI

Pada bab ini dibahas mengenai implementasi berdasarkan hasil yang telah diperoleh dari analisis kebutuhan dan proses perancangan perangkat lunak yang dibuat. Pembahasan terdiri dari penjelasan tentang spesifikasi sistem, batasan-batasan dalam implementasi, implementasi Metode, dan implementasi Sistem.

#### 5.1. Spesifikasi Sistem

Hasil analisis kebutuhan dan perancangan perangkat lunak yang telah diuraikan pada Bab 4 menjadi acuan untuk melakukan implementasi. Spesifikasi sistem diimplementasikan pada spesifikasi perangkat keras dan perangkat lunak.

##### 5.1.1. Spesifikasi Perangkat Keras

Pengembangan sistem ini menggunakan sebuah komputer dengan spesifikasi perangkat keras yang dijelaskan pada Tabel 5.1.

**Tabel 5.1** Spesifikasi Perangkat Keras Komputer

Nama Komponen	Spesifikasi
Prosesor	Intel (R) Core(TM) i3 (4 CPU) @ 2.50GHz
Memori(RAM)	2048 MB
Hardisk	320 GB
VGA	ATI Mobility Radeon HD 5470

##### 5.1.2. Spesifikasi Perangkat Lunak

Pengembangan Sistem ini menggunakan perangkat lunak dengan spesifikasi yang dijelaskan pada Tabel 5.2.

**Tabel 5.2** Spesifikasi Perangkat Lunak Komputer

Nama	Spesifikasi
Sistem Operasi	Microsoft Windows 7 Ultimate 32-bit
Bahasa Pemrograman	HTML, CSS, Javascript, SQL, PHP
<i>Tools</i> pemrograman	Aptana Studio 3.0
<i>Server</i>	Apache Server
Basisdata	MySQL

## 5.2. Batasan Implementasi

Batasan implementasi adalah batasan proses yang bisa dilakukan sistem sesuai dengan perancangan awal sistem. Batasan implementasi ditampilkan agar penelitian ini memiliki ruang lingkup yang jelas dalam mengimplementasikan sistem.

Beberapa batasan dalam mengimplementasikan penelitian ini adalah sebagai berikut:

- a) *Output* yang dihasilkan adalah Prediksi Kasus Demam Berdarah pada satu satuan waktu.
- b) Layer *Output* pada *Neural Network* berjumlah 1.
- c) Fungsi Transformasi *Neural Network* menggunakan fungsi *Sigmoid*.
- d) *Learning Rule* yang digunakan pada *Neural Network* adalah *delta rule*.
- e) Algoritma pembelajaran pada *Backpropagation* menggunakan *Gradient Descent* dengan momentum dan adaptif *learning rate*.

## 5.3. Impelementasi Metode FCBPN

Berikut ini adalah implementasi kode program yang dilakukan dalam Penelitian ini.

### 5.3.1. Normalisasi

**Tabel 5.3** Source Code Normalisasi

Source Code Normalisasi	
1	<code>public static function normalize(\$data_training) {</code>
2	<code>    \$norm = array();</code>
3	<code>    for (\$i=0; \$i</code>
4	<code>&lt;count(\$data_training[0]) ; \$i++) {</code>
5	<code>        \$norm[\$i] = array();</code>
6	<code>        for (\$j=0; \$j</code>
7	<code>&lt;count(\$data_training) ; \$j++) {</code>
8	<code>            array_push(\$norm[\$i],</code>
9	<code>    \$data_training[\$j][\$i]);</code>
10	<code>    }</code>
11	<code>    }</code>
12	<code>    \$result = array();</code>
13	<code>    for (\$i=0; \$i &lt;count(\$norm) ; \$i++) {</code>
14	<code>        for (\$j=0; \$j &lt;count(\$norm[0]) ;</code>
15	<code>    \$j++) {</code>
16	<code>            \$result[\$j][\$i] =</code>
17	<code>    (0.1+0.8*(\$norm[\$i][\$j] - min(\$norm[\$i]))) /</code>

```

18 (max($norm[$i]) - min($norm[$i]));
19     }
20 }
21     return $result;
22 }
    
```

Penjelasan dari kode program diatas adalah sebagai berikut :

1. Baris 3-15 menjelaskan tentang perulangan data untuk normalisasi
2. Baris 16-17 menjelaskan proses normalisasi

**Tabel 5.4** Table *Fuzzy C-Means Clustering*

Fuzzy C-Means Clustering	
1	<?php class Fcm{
2	private static \$tmpMatrixU;
3	private function random_value() {
4	return (double)(rand()/(32767/2)) - 1;
5	}
6	private function generateMatrix(\$countData,
7	\$config) {
8	...
9	for(\$j = 0; \$j < \$countData; \$j++){
10	\$sum = 0;
11	for(\$i = 0; \$i <
12	\$config['cluster']; \$i++){
13	\$MatrixU[\$j][\$i] = \$this->random_value();
14	\$sum += \$MatrixU[\$j][\$i];
15	}
16	
17	// Normalize Data
18	for(\$i = 0; \$i <
19	\$config['cluster']; \$i++){
20	\$MatrixU[\$j][\$i] = \$MatrixU[\$j][\$i]/\$sum;
21	}
22	}
23	Fcm::\$tmpMatrixU = \$MatrixU;
24	...
25	return \$MatrixU;
26	}
27	private function getCentroids(\$MatrixU,\$data,
28	\$config) {
29	\$countData = count(\$data);
30	\$criteria = count(\$data[0]);
31	for(\$i = 0; \$i < \$config['cluster']; \$i++){
32	...
33	for(\$j = 0 ; \$j < \$countData; \$j++){
34	for (\$k=0; \$k<\$criteria; \$k++) {
35	
36	\$tempA[\$k] = \$tempA[\$k] +
37	pow(\$MatrixU[\$j][\$i], \$config['bobot']);
38	\$tempB[\$k] = \$tempB[\$k] +





```
39     pow($MatrixU[$j][$i],$config['bobot']) * $data[$j][$k];
40         }
41     }
42     }
43     for ($k=0; $k <$criteria; $k++) {
44         $MatrixCentroids[$i][$k] =
45     $tempB[$k] / $tempA[$k];
46     }
47 }
48     return $MatrixCentroids;
49 }
50
51     private function updateMatrix($MatrixCentroids,
52 $data, $config){
53         $countData = count($data);
54         $criteria = count($data[0]);
55         for($j = 0 ; $j<$countData; $j++){
56             $tempSum = 0;
57             for($i = 0; $i <
58 $config['cluster']; $i++){
59                 // Distance 2 points
60                 $distance = 0;
61                 for ($k=0; $k <$criteria ;
62 $k++) {
63                     $distance = $distance
64 + pow(($data[$j][$k] - $MatrixCentroids[$i][$k]),2);
65                 }
66                 $distance =
67 sqrt($distance);
68                 $MatrixU[$j][$i] =
69 (double)pow(1/$distance , 2/($config['bobot']-1));
70 $tempSum += $MatrixU[$j][$i];
71             }
72     for($i = 0; $i < $config['cluster']; $i++){
73         $MatrixU[$j][$i] =
74 (double) ($MatrixU[$j][$i]/$tempSum);
75     }
76     $sum = 0;
77     for($i = 0; $i <
78 $config['cluster']; $i++){
79         $sum += $MatrixU[$j][$i];
80     }
81     for($i = 0; $i <
82 $config['cluster']; $i++){
83         $MatrixU[$j][$i] =
84 $MatrixU[$j][$i]/$sum;
85     }
86 }
87 }
88     return $MatrixU;
89 }
90 }
```

```
91     private function checkTermination($MatrixU,  
92     $oldMatrixU, $config){  
93         $done = false;  
94         for ($i=0; $i <count($MatrixU) ; $i++) {  
95             for ($j=0; $j <count($MatrixU[$i]) ;  
96             $j++) {  
97                 if(abs($MatrixU[$i][$j] -  
98                 $oldMatrixU[$i][$j]) < $config['e']){  
99                     $done = true;  
100                    break;  
101                }  
102            }  
103            if($done) break;  
104        }  
105        return $done;  
106    }  
107  
108    ...  
109  
110    public function run($data, $config){  
111        $countData = count($data);  
112        $criteria = count($data[0]);  
113        $MatrixCentroids = array();  
114        $oldMatrixU = array();  
115        // generate random Matrix  
116        $MatrixU = $this->generateMatrix($countData,  
117        $config);  
118        // iterasi  
119        for($a = 0; $a<$config['maxIterasi']; $a++){  
120            // assign matrix ke n-1  
121            $oldMatrixU = $MatrixU;  
122            // cari centroid  
123            $MatrixCentroids = $this-  
124            >getCentroids($MatrixU,$data, $config);  
125            //Perbaiki MatrixU  
126            $MatrixU = $this-  
127            >updateMatrix($MatrixCentroids, $data, $config);  
128            // check termination  
129            $done = $this-  
130            >checkTermination($MatrixU, $oldMatrixU, $config);  
131            if($done) break;  
132        }  
133        ...  
134    }?>
```

Penjelasan dari kode program diatas adalah sebagai berikut :

1. Baris 6-26 menjelaskan proses inialisasi awal derajat keanggotaan *fuzzy clustering*
2. Baris 27-49 menjelaskan proses pencarian pusat cluster
3. Baris 51-89 menjelaskan proses pembaruan derajat keanggotaan *fuzzy clustering*
4. Baris 91-106 menjelaskan proses pengecekan penghentian iterasi

### 5.3.2. The Degree of Membership Levels (MLC)

**Tabel 5.5** The Degree of Membership Levels

Source Code The Degree of Membership Levels (MLC)	
1	public static function mcfs(\$center, \$clusterto, \$data,
2	\$a=50, \$c=0.5){
3	\$mmlc = array();
4	for (\$i=0; \$i <count(\$data) ; \$i++) {
5	for (\$j=0; \$j <count(\$center) ; \$j++) {
6	\$distance = 0;
7	for (\$k=0; \$k <count(\$data[0]) ;
8	\$k++) {
9	\$distance = \$distance + pow((\$data[\$i][\$k] -
10	\$center[\$j][\$k]),2);
11	}
12	\$distance = sqrt(\$distance);
13	\$mmlc[\$i][\$j] = (1/(1/(1 + exp(-\$a*(\$distance-\$c)))));}}
14	return \$mmlc;
15	}

Penjelasan dari kode program diatas adalah sebagai berikut :

1. Baris 4-12 menjelaskan proses perhitungan jarak antara pusat *cluster* dan data
2. Baris 13 menjelaskan proses perhitungan *Membership Level Cluster*

### 5.3.3. Backpropagation Neural Networks

**Tabel 5.6** Backpropagation Neural Networks

Source Code Backpropagation Neural Networks	
1	<?php
2	...
3	class Bpn{
4	...
5	...
6	private function rando(){
7	return (double)(rand()/(_RAND_MAX/2) - 1;
8	}

```

9      }
10     // sigmoid function
11     private function sigmoid($inputSource){
12         return (double) (1/(1+exp(-$inputSource)));
13     }
14     // mean square error
15     private function mse($target, $actual){
16         return 1/2 * pow($target - $actual,2);
17     }
18     }
19     ...
20
21
22     public function __construct($layers,$beta,$alpha,
23 $cluster){
24         ...
25
26         for($i=1;$i<$this->numLayers;$i++){
27             for($j=0;$j<$this->
28 >layersSize[$i];$j++){
29                 for($k=0;$k < $this->
30 >layersSize[$i-1]+1;$k++){
31                     $rand = $this->
32 >rando();
33                     for ($l=0; $l
34 <$cluster ; $l++) {
35                         $this->
36 >weight[$l][$i][$j][$k]=$rand;
37                     }
38                 }
39                 // bias in the last neuron
40                 for ($l=0; $l <$cluster ;
41 $l++) {
42                     $this->
43 >weight[$l][$i][$j][$this->layersSize[$i-1]] =
44 (double)-1;
45                 }
46             }
47         }
48     }
49     ...
50
51     for($i=1;$i<$this->numLayers;$i++){
52         for($j=0;$j<$this->
53 >layersSize[$i];$j++){
54             for($k=0;$k<$this->layersSize[$i-
55 1]+1;$k++){
56                 for ($l=0; $l <$cluster ;
57 $l++) {
58                     $this->
59 >prevDwt[$l][$i][$j][$k]=(double)0.0;
60                 }
61             }
62         }
63     }
64 }
65

```

```

66     }
67     }
68     }
69     }
70     }
71     }
72     private function fwdParallel($inputSource,
73     $kluster){
74         $this->output[$kluster][0] = $inputSource;
75         for($i=1;$i<$this->numLayers;$i++){
76             for($j=0;$j<$this->
77             >layersSize[$i];$j++){ // layer ke - X
78                 $sum=0.0;
79                 for($k=0;$k<$this->layersSize[$i-
80                 1];$k++){ // layer ke X - 1
81                     $sum += $this->
82                     >output[$kluster][$i-1][$k]*$this->
83                     >weight[$kluster][$i][$j][$k];
84                 }
85                 // Apply bias
86                 $sum+=$this->weight[$kluster][$i][$j][$this->layersSize[$i-1]];
87                 // Apply sigmoid function
88                 $this->output[$kluster][$i][$j] =
89                 $this->sigmoid($sum);
90             }
91         }
92     }
93     }
94     public function bpgtParallel($target, $MLC,$alr,
95     $cluster){
96
97     //////////////////////////////////////
98     /// FIND DELTA FOR OUPUT LAYER (Last Layer) ///
99     //////////////////////////////////////
100     ...
101     $out = $this->output[$cluster][$this->
102     >numLayers-1][0];
103     $this->delta[$cluster][$this->numLayers-
104     1][0] = $out * (1 - $out) * ($target - $out);
105
106     //////////////////////////////////////
107     //////////////////////////////////////
108     /// FIND DELTA FOR HIDDEN LAYERS (From Last
109     Hidden Layer BACKWARDS To First Hidden Layer) ///
110     //////////////////////////////////////
111     //////////////////////////////////////
112
113
114
115
116
117
118
119
120
121

```

```

122 ///////////////////////////////////////////////////
123
124         for($i=$this->numLayers-2; $i > 0; $i--){
125             for($j=0;$j<$this-
126 >layersSize[$i];$j++){
127
128                 $sum=0;
129                 for($k=0;$k<$this-
130 >layersSize[$i+1];$k++){
131                     $sum += $this-
132 >delta[$cluster][$i+1][$k] * $this-
133 >weight[$cluster][$i+1][$k][$j];
134
135                 }
136                 $out = $this-
137 >output[$cluster][$i][$j];
138                 $this->delta[$cluster][$i][$j] =
139 $out * (1 - $out) * $sum;
140
141             }
142         }
143
144         ///////////////////////////////////
145         /// MOMENTUM (Alpha) ///
146         ///////////////////////////////////
147
148         for($i=1;$i<$this->numLayers;$i++){
149             for($j=0;$j<$this-
150 >layersSize[$i];$j++){
151                 for($k=0;$k<$this->layersSize[$i-
152 1];$k++){ // jumlah node dalam layer - 1
153                     $this-
154 >weight[$cluster][$i][$j][$k] += $this->alpha * $this-
155 >prevDwt[$cluster][$i][$j][$k];
156
157                 }
158                 // bias
159                 $this-
160 >weight[$cluster][$i][$j][$this->layersSize[$i-1]] +=
161                 $this->alpha * $this-
162 >prevDwt[$cluster][$i][$j][$this->layersSize[$i-1]];
163             }
164         }
165
166         ///////////////////////////////////
167         /// ADJUST WEIGHTS (Using Steepest Descent) ///
168         ///////////////////////////////////
169
170         for($i=1;$i<$this->numLayers;$i++){
171             for($j=0;$j<$this-
172 >layersSize[$i];$j++){
173

```

```

174         for($k=0;$k<$this->layersSize[$i-
175 1];$k++){
176
177             $this-
178 >prevDwt[$cluster][$i][$j][$k] = $alr * $this-
179 >delta[$cluster][$i][$j] * $this->output[$cluster][$i-
180 1][$k];
181
182             $this->weight[$cluster][$i][$j][$k] += $this-
183 >prevDwt[$cluster][$i][$j][$k];
184
185         }
186         /* --- Apply the corrections -- */
187         $this-
188 >prevDwt[$cluster][$i][$j][$this->layersSize[$i-1]] =
189 $alr * $this->delta[$cluster][$i][$j];
190
191         $this-
192 >weight[$cluster][$i][$j][$this->layersSize[$i-1]] +=
193 $this->prevDwt[$cluster][$i][$j][$this->layersSize[$i-
194 1]];
195     }
196 }
197 }
198 }
199
200 public function Run($data,$output, $MLC, $Thresh,
201 $numEpoch) {
202
203     $MSE = array();
204     $NumPattern = count($data);
205     $NumInput = count($data[0]);
206     $testData = $data;
207
208     //training iteration
209     for($e=0; $e < $numEpoch; $e++){
210
211         for
212 ($k=0;$k<count($MLC[$e*$NumPattern]);$k++) {
213             $this-
214 >ffwdParallel($data[$e*$NumPattern], $k);
215         }
216
217         //global output
218         $sigmaAtas = 0; $sigmaBawah = 0;
219         for ($k=0; $k
220 <count($MLC[$e*$NumPattern]) ; $k++) {
221             $sigmaAtas += $this-
222 >output[$k][2][0] * $MLC[$e*$NumPattern][$k];
223             $sigmaBawah +=
224 $MLC[$e*$NumPattern][$k];
225         }$this->global output[$e] =

```

```
226 $sigmaAtas/$sigmaBawah;
227
228 // hitung error
229 $MSE[$e] = $this->mse($this-
230 >global_output[$e], $output[$e%$NumPattern]);
231
232 //backpropagation
233 for ($k=0; $k
234 <count($MLC[$e%$NumPattern]) ; $k++) {
235     $alr = $this->beta *
236     ($MLC[$e%$NumPattern][$k]/$sigmaBawah);
237     $this-
238 >bpgtParallel($output[$e%$NumPattern],
239 $MLC[$e%$NumPattern][$k], $alr, $k);
240 }
241
242 // break condition
243 if( $MSE[$e] < $Thresh)
244     break;
245 }
246
247 $this->output = array();
248 $testPattern = count($testData);
249 $forecast = array();
250
251 // testing
252 for ($e = 0;$e <$testPattern; $e++ ){
253     for ($k=0; $k <count($MLC[$e]) ; $k++)
254     {
255         $this-
256 >ffwdParallel($testData[$e], $k);
257     }
258     $sigmaAtas = 0; $sigmaBawah = 0;
259     for ($k=0; $k <count($MLC[$e]) ; $k++)
260     {
261         $sigmaAtas += $this-
262 >output[$k][2][0] * $MLC[$e][$k];
263         $sigmaBawah += $MLC[$e][$k];
264     }
265     $forecast[$e] =
266 $sigmaAtas/$sigmaBawah;
267 }
268
269 return
270 array('forecast'=>$forecast, 'error'=>$MSE[count($MSE)-
271 1]);
272 }?>
```



Penjelasan dari kode program diatas adalah sebagai berikut :

1. Baris 11-14 menjelaskan proses fungsi *sigmoid*
2. Baris 21-70 menjelaskan proses inialisasi awal nilai bobot
3. Baris 208 menjelaskan proses iterasi pada BPN
4. Baris 212 menjelaskan proses *Feedforward*
5. Baris 257-266 menjelaskan proses kalkulasi untuk mendapatkan nilai prediksi

### 5.3.4. Fuzzy Clustering backpropagation Neural Networks

**Tabel 5.7** Fuzzy Clustering backpropagation Neural Networks

Source Code Fuzy Clustering Backpropagation Neural Networks	
1	<?php
2	...
3	\$data_no_rescale = Helper::explodeIO(\$data);
4	/* rescale data*/
5	\$data_rescale = Helper::rescale(\$data);
6	/* explode input and output training*/
7	\$straining =
8	Helper::explodeIO(\$data_rescale['result']);
9	\$global['input_training'] = \$straining['input'];
10	\$global['output_training'] = \$straining['output'];
11	/* do NORMALIZATION (0.1 - 0.9) sparately
12	each criteria */
13	\$global['normal_input_training'] =
14	Helper::normalize(\$global['input_training']);
15	/* do FCM */
16	\$global['fcm']['config'] = array(
17	'cluster' => \$config['fcm_cluster'],
18	'bobot' => \$config['fcm_bobot'],
19	'maxIterasi' =>
20	\$config['fcm_max_iterasi'],
21	'e' => \$config['fcm_e'],
22	);
23	\$fcm = new Fcm;
24	\$global['fcm']['data'] = \$fcm-
25	>run(\$global['normal_input_training'],
26	\$global['fcm']['config']);
27	/* do MCFS */
28	\$global['mcfs'] =
29	Helper::mcfs(\$global['fcm']['data']['center'],\$global['f
30	cm']['data']['clusterto'],
31	\$global['normal_input_training']);
32	/* do BPN */
33	\$layers =
34	array(count(\$global['input_training'][0]),2,1);
35	\$BPNconfig = array(

```
36         'thresh'      => $config['bpn_mse'],
37         'max_epoch' =>
38 $config['bpn_max_iterasi'],
39         'beta'       =>
40 $config['bpn_beta'],
41         'alpha'     =>
42 $config['bpn_alpha'],
43         'layers'    => $layers
44     );$global['bpn']['config'] = $BPNconfig;
45     $bpn = new
46 Bpn($BPNconfig['layers'],$BPNconfig['beta'],$BPNconfig['
47 alpha'],count($global['mcfs'][0]));
48     $global['bpn']['data'] = $bpn-
49 >Run($global['input_training'],$global['output_training'
50 ],$global['mcfs'],$BPNconfig['thresh'],
51 $BPNconfig['max_epoch']);
52     $hasil=
53 Helper::derescale($global['bpn']['data']['forecast'],
54 $data_rescale['max'][count($data_rescale['max'])-1]);
...
?>
```

Penjelasan dari kode program diatas adalah sebagai berikut :

1. Baris 5 menjelaskan proses *rescale*
2. Baris 12 menjelaskan proses normalisasi data
3. Baris 16 menjelaskan proses FCM
4. Baris 34 menjelaskan proses BPN

## BAB VI

### PENGUJIAN DAN ANALISIS

Bab ini membahas mengenai tahapan pengujian dan analisis dari metode yang digunakan. Proses pengujian dilakukan menjadi 2 bagian yaitu pengujian *Membership Levels Cluster* untuk mendapatkan parameter yang tepat untuk prediksi dan pengujian akurasi untuk mengukur akurasi metode yang digunakan dengan data yang dipakai.

#### 6.1. Pengujian

Pengujian sistem implementasi *Fuzzy Clustering Backpropogation Neural Networks* untuk Prediksi Perkembangan Penyakit Demam Berdarah sebagai Pendukung Keputusan ini menggunakan data kasus demam berdarah dan curah hujan Kabupaten Situbondo pada tahun 2010 sampai dengan 2012.

Pengujian ini dilakukan dengan 36 bulan data dengan kriteria jumlah kasus demam berdarah, jumlah hari hujan, dan curah hujan. Konfigurasi dari model dapat dilihat sebagai berikut :

- Parameter *Fuzzy C-Means Clustering* :
  - bobot = 2
  - maksimal iterasi = 100000
  - *error* = 0.0001
- Parameter *Backpropagation Neural Network* :
  - *error* = 0.00000001
  - makasimal iterasi = 100000
  - nilai *beta* = 0.1
  - nilai *alpha* = 0.02
  - *layer* = 2,2,1

1. Untuk menguji parameter cluster yang paling cocok digunakan, maka diuji nilai cluster dari 2,3,4 dan 5 di Panarukan.

**Tabel 6. 1** Tabel Pengujian nilai Cluster di Panarukan

No	Jumlah Cluster	RMSE	MAPE
1	2	6.7226813433662	42.65347491334
2	3	6.4226813433662	43.63747491334
3	4	6.3144123594946	42.61895639482
4	5	6.5944420310836	66.267104542967

Berdasarkan Tabel 6.1 didapatkan nilai RMSE dan MAPE terendah ada pada *cluster* 4 sehingga sekenario pengujian selanjutnya digunakan jumlah *cluster* 4. Untuk mengetahui nilai terbaik dari parameter a dan c fungsi *sigmoid* MLC pada persamaan 2-25 di masing-masing lokasi, akan diuji coba beberapa skenario percobaan berikut.

1. Penentuan nilai a dengan nilai awal  $c = 0.1$  diuji dengan nilai a dari 10-100, sehingga didapatkan nilai a terbaik berdasarkan MAPE dan RMSE.
2. Penentuan nilai c dengan nilai a terbaik berdasarkan scenario uji coba pada nomor 1.
3. Nilai a dan c terbaik pada suatu lokasi akan diuji kembali.

Pengujian untuk menentukan nilai a dan c terbaik di data 3 lokasi dengan parameter awal  $c = 0.1$ . Hasil pengujian dengan menggunakan metode RMSE dan MAPE pada kecamatan panarukan dengan parameter  $c = 0.1$ . Pengujian ini dilakukan di 6 kecamatan berbeda. Adapun pengujian ini ditunjukkan dalam Tabel 6.1. Hasil dari pengujian untuk masing-masing daerah diperlihatkan sebagai berikut :

2. Panarukan

**Tabel 6.2** Pengujian RMSE dan MAPE di Panarukan dengan  $c = 0.1$

No	a	RMSE	MAPE
1	10	6.9442222186666	82.01985647388
2	20	6.7226813433662	81.63747491334
3	30	6.7226813433662	81.63747491334



4	40	6.7144123594946	80.61895639482
5	50	<b>6.3944420310836</b>	<b>76.267104542967</b>
6	60	6.7102740463991	99.415252691115
7	70	6.7288764127288	99.415252691115
8	80	6.8455175926376	125.05910569847
9	90	6.8272330494213	131.964969896
10	100	6.7288764127288	99.415252691115

Dari hasil pengujian dalam Tabel 6.2 didapatkan hasil terbaik pada parameter  $a=50$  (pada baris ke 5 dengan nilai dicetak tebal). Kemudian dilakukan pengujian dengan  $a=50$  terhadap parameter  $c$ , seperti ditunjukkan pada Tabel 6.2.

**Tabel 6. 3** Tabel Pengujian RMSE dan MAPE dengan  $a = 50$

No	c	RMSE	MAPE
1	0,1	6.3966136874652	82.756302073831
2	0,2	5.9160797830996	72.463206227574
3	0,3	5.8452259722501	76.997050416591
4	0,4	5.7566386797235	73.161071580612
5	0,5	5.754225500544	72.235145654686
6	0,6	5.8022983951764	70.267553062093
7	0,7	5.6813535163531	59.85750015204
8	0,8	5.6813535163531	59.85750015204
9	0,9	5.6813535163531	59.85750015204
10	1	<b>5.6813535163531</b>	<b>59.85750015204</b>

Pada Tabel 6.3 didapatkan hasil terbaik didapatkan pada parameter  $c = 1$  dan  $a = 50$ .

### 3. Situbondo

**Tabel 6. 4** Pengujian RMSE dan MAPE di Situbondo dengan  $c = 0.1$

No	a	RMSE	MAPE
1	10	6.5213325155046	93.311857629386
2	20	6.4871668186762	91.089635407164
3	30	6.1621605157787	79.503550142918
4	40	5.9976847384822	64.299846439214
5	50	<b>5.7927157323276</b>	<b>64.167464878672</b>

6	60	5.8166427888717	64.400048652922
7	70	5.9278064145929	75.279678282552
8	80	5.823801736552	74.257966916013
9	90	5.7638721552635	75.029571854284
10	100	5.6273143387114	77.094280240832

Dari hasil pengujian dalam Tabel 6.4 didapatkan hasil terbaik pada parameter  $a=50$ . Kemudian dilakukan pengujian dengan  $a=50$  terhadap parameter  $c$ , seperti ditunjukkan pada Tabel 6.4.

Dari tabel diatas, didapat error terkecil  $a = 50$

**Tabel 6. 5** Tabel Pengujian RMSE dan MAPE dengan  $a = 50$

No	c	RMSE	MAPE
1	0,1	5.7686894332923	69.821238520951
2	0,2	5.6470247820325	72.283874597093
3	0,3	5.6837976349933	76.913504226723
4	0,4	5.7855183192366	84.783874597093
5	0,5	<b>5.6223759311443</b>	<b>68.799641184699</b>
6	0,6	5.6297819180182	68.74562883902
7	0,7	5.6297819180182	68.74562883902
8	0,8	5.6297819180182	68.74562883902
9	0,9	5.6297819180182	68.74562883902
10	1	5.6297819180182	68.74562883902

Dari pengujian pada Tabel 6.5 didapatkan hasil parameter terbaik  $a = 50$  dan  $c = 0.5$ .

### 3. Panji

Hasil pengujian dengan menggunakan metode RMSE dan MAPE pada kecamatan Panji dengan parameter  $c = 0.1$  ditunjukkan dalam Tabel 6.6

**Tabel 6.6** Pengujian RMSE dan MAPE di Panji dengan  $c = 0.1$

No	a	RMSE	MAPE
1	10	8.5081660120668	51.621381667113
2	20	8.5049005481154	54.399159444891
3	30	8.2965856431024	50.463974259706
4	40	<b>7.9930525388545</b>	<b>41.569743490475</b>

5	50	8.286535263104	59.75265418863
6	60	8.1069243380102	54.521172707148
7	70	8.0363756341608	52.909705470681
8	80	7.8740078740118	48.452033513009
9	90	7.9337534902746	51.69277425375
10	100	7.9355039047449	52.248329809305

Dari hasil pengujian dalam Tabel 6.6 didapatkan hasil terbaik pada parameter  $a=40$ . Kemudian dilakukan pengujian dengan  $a=40$  terhadap parameter  $c$ , seperti ditunjukkan pada Tabel 6.6.

**Tabel 6.7** Tabel Pengujian RMSE dan MAPE dengan  $a = 40$

No	c	RMSE	MAPE
1	0,1	7.9982637004679	83.502734844198
2	0,2	<b>7.6594168620507</b>	<b>65.027801354021</b>
3	0,3	7.8066922857536	61.290352601328
4	0,4	7.7853138094178	78.579388655608
5	0,5	8.0173423138705	80.53010562157
6	0,6	7.9056941504209	82.514573840793
7	0,7	7.9056941504209	82.514573840793
8	0,8	7.9056941504209	82.514573840793
9	0,9	7.9056941504209	82.514573840793
10	1	7.9056941504209	82.514573840793

Dari pengujian pada Tabel 6.7 didapatkan hasil parameter terbaik  $a = 20$  dan  $c = 0.2$ .

#### 4.Asembagus

Hasil pengujian dengan menggunakan metode RMSE dan MAPE pada kecamatan Asembagus dengan parameter  $c = 0.1$  ditunjukkan dalam Tabel 6.7

**Tabel 6.8** Pengujian RMSE dan MAPE di Asembagus dengan  $c = 0.1$

No	a	RMSE	MAPE
1	10	2.4494897427832	34.626022126022



2	20	2.4494897427832	34.626022126022
3	30	2.4720661623652	29.070466570467
4	40	2.6509956200978	38.977873977874
5	50	2.5331140255951	27.866762866763
6	60	2.5331140255951	27.866762866763
7	70	<b>2.4494897427832</b>	<b>27.588985088985</b>
8	80	2.4551533104427	30.366762866763
9	90	2.5	41.477873977874
10	100	2.4551533104427	30.366762866763

Dari hasil pengujian dalam Tabel 6.8 didapatkan hasil terbaik pada parameter  $a=70$ . Kemudian dilakukan pengujian dengan  $a=70$  terhadap parameter  $c$ , seperti ditunjukkan pada Tabel 6.9.

Terkecil di  $a = 70$

**Tabel 6.9** Tabel Pengujian RMSE dan MAPE dengan  $a = 70$

No	c	RMSE	MAPE
1	0,1	<b>2.5221243250703</b>	<b>24.905002405002</b>
2	0,2	2.5495097567964	37.867965367965
3	0,3	2.5766041389567	37.867965367965
4	0,4	2.5766041389567	38.423520923521
5	0,5	2.5766041389567	38.423520923521
6	0,6	2.5766041389567	38.423520923521
7	0,7	2.5927248643507	41.201298701299
8	0,8	2.5927248643507	41.201298701299
9	0,9	2.5927248643507	41.201298701299
10	1	2.5927248643507	41.201298701299

Dari pengujian pada Tabel 6.9 didapatkan hasil parameter terbaik  $a = 70$  dan  $c = 0.1$ .

##### 5. Mlandingan

Hasil pengujian dengan menggunakan metode RMSE dan MAPE pada kecamatan

Mlandingan dengan parameter  $c = 0.1$  ditunjukkan dalam Tabel 6.9



**Tabel 6.10** Pengujian RMSE dan MAPE di Mlandingan dengan  $c = 0.1$ 

No	a	RMSE	MAPE
1	10	1.6158932858054	21.875
2	20	1.6414763002994	19.0972222222222
3	30	1.6329931618555	21.875
4	40	1.6329931618555	21.875
5	50	1.5365907428821	21.412037037037
6	60	1.5545631755148	23.726851851852
7	70	<b>1.4142135623731</b>	<b>20.949074074074</b>
8	80	1.6499158227686	10.7638888888889
9	90	1.6583123951777	16.3194444444444
10	100	1.6666666666667	19.0972222222222

Dari hasil pengujian dalam Tabel 6.10 didapatkan hasil terbaik pada parameter  $a=70$ . Kemudian dilakukan pengujian dengan  $a=70$  terhadap parameter  $c$ , seperti ditunjukkan pada Tabel 6.10.

Didapatkan  $A = 70$

**Tabel 6.11** Tabel Pengujian RMSE dan MAPE dengan  $a = 70$ 

No	c	RMSE	MAPE
1	0,1	1.5545631755148	18.634259259259
2	0,2	1.5365907428821	21.412037037037
3	0,3	1.5275252316519	18.634259259259
4	0,4	1.569295517644	19.560185185185
5	0,5	1.6329931618555	13.078703703704
6	0,6	1.6329931618555	13.078703703704
7	0,7	<b>1.5072751268322</b>	<b>13.078703703704</b>
8	0,8	1.5811388300842	7.0601851851852
9	0,9	1.5811388300842	15.393518518519
10	1	1.5811388300842	15.393518518519

Dari pengujian pada Tabel 6.11 didapatkan hasil parameter terbaik  $a = 70$  dan  $c = 0.7$ .

## 6. Besuki

Hasil pengujian dengan menggunakan metode RMSE dan MAPE pada kecamatan Besuki dengan parameter  $c = 0.1$  ditunjukkan dalam Tabel 6.12

**Tabel 6. 2** Pengujian RMSE dan MAPE di Besuki dengan  $c = 0.1$

No	a	RMSE	MAPE
1	10	7.1995370221517	79.069824511001
2	20	7.0099136149377	78.143898585075
3	30	6.6541549258924	71.034110225287
4	40	6.6122781416258	58.99707318825
5	50	6.5933468149508	53.441517632694
6	60	6.6957698088669	58.071147262324
7	70	6.6999170807473	63.626702817879
8	80	6.6999170807473	63.626702817879
9	90	<b>6.5680201650793</b>	<b>46.719905911082</b>
10	100	6.6936952093411	60.848925040102

Dari hasil pengujian dalam Tabel 6.11 didapatkan hasil terbaik pada parameter  $a=90$ . Kemudian dilakukan pengujian dengan  $a=90$  terhadap parameter  $c$ , seperti ditunjukkan pada Tabel 6.12.

**Tabel 6. 13** Tabel Pengujian RMSE dan MAPE dengan  $a = 90$

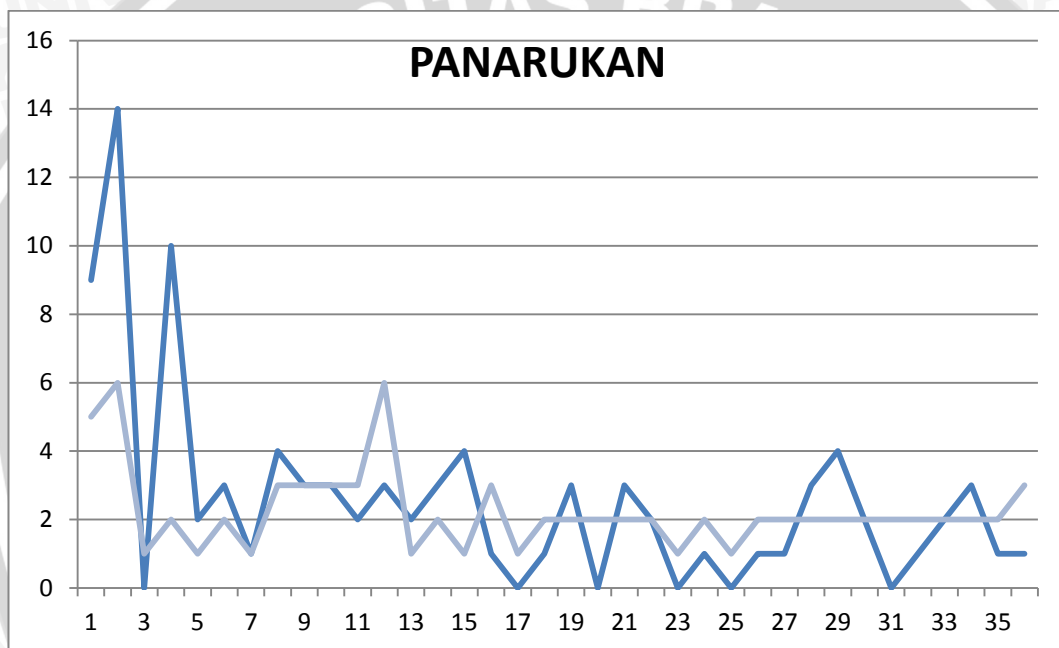
No	c	RMSE	MAPE
1	0,1	7.1472605474639	45.569965165553
2	0,2	7.1161006800697	49.086448682037
3	0,3	<b>7.1121486568043</b>	<b>40.308670904259</b>
4	0,4	7.112196097784	43.993856089444
5	0,5	7.1141486568043	46.308670904259
6	0,6	7.1141486568043	46.308670904259
7	0,7	7.1141486568043	46.308670904259
8	0,8	7.1141486568043	46.308670904259
9	0,9	7.1141486568043	46.308670904259
10	1	7.1141486568043	46.308670904259

Dari pengujian pada Tabel 6.13 didapatkan hasil parameter terbaik  $a = 90$  dan  $c = 0.3$ .

Dari hasil pengujian untuk menentukan parameter terbaik, data di masing-masing lokasi diuji dengan parameter tersebut. Berikut adalah pengujian berdasarkan parameter  $a$  dan  $c$  fungsi *sigmoid* terbaik.

### 1. Panarukan

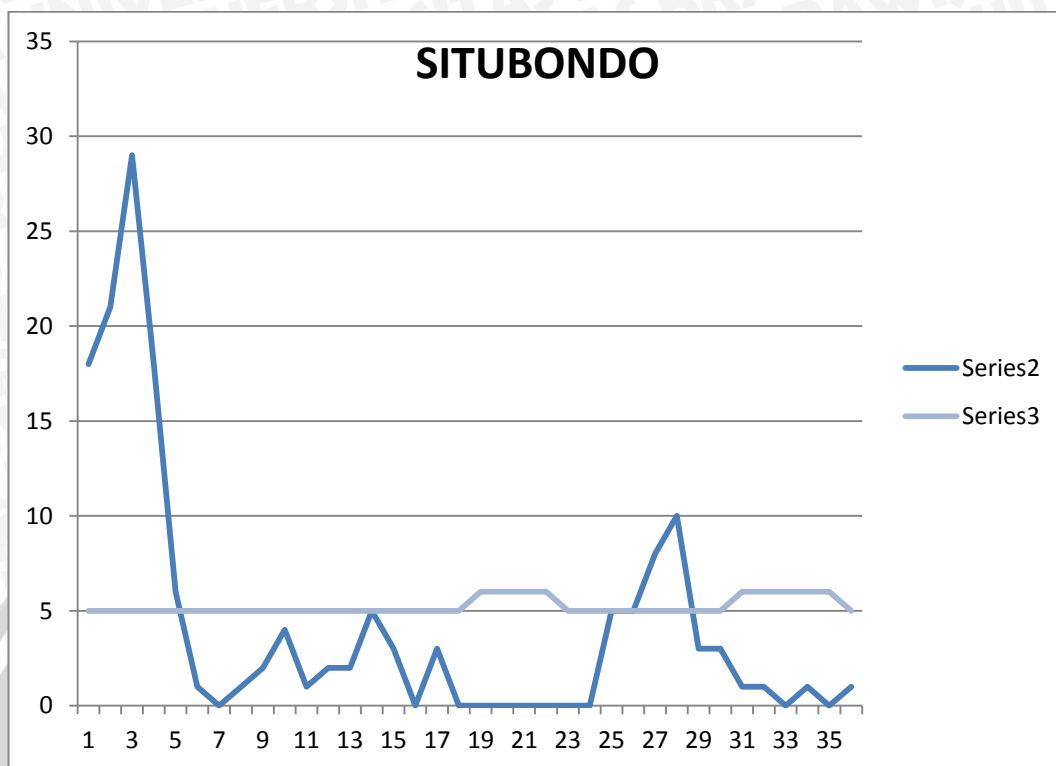
Untuk Pengujian pada lokasi Panarukan didapat hasil  $a = 50$ ,  $c = 1$ , MAPE = 53.710317460317 dan RMSE = 2.8136571693557. Adapun hasil dari pengujian ini ditunjukkan pada Gambar 6.1.



**Gambar 6.1** Grafik Prediksi kasus Panarukan

### 2. Situbondo

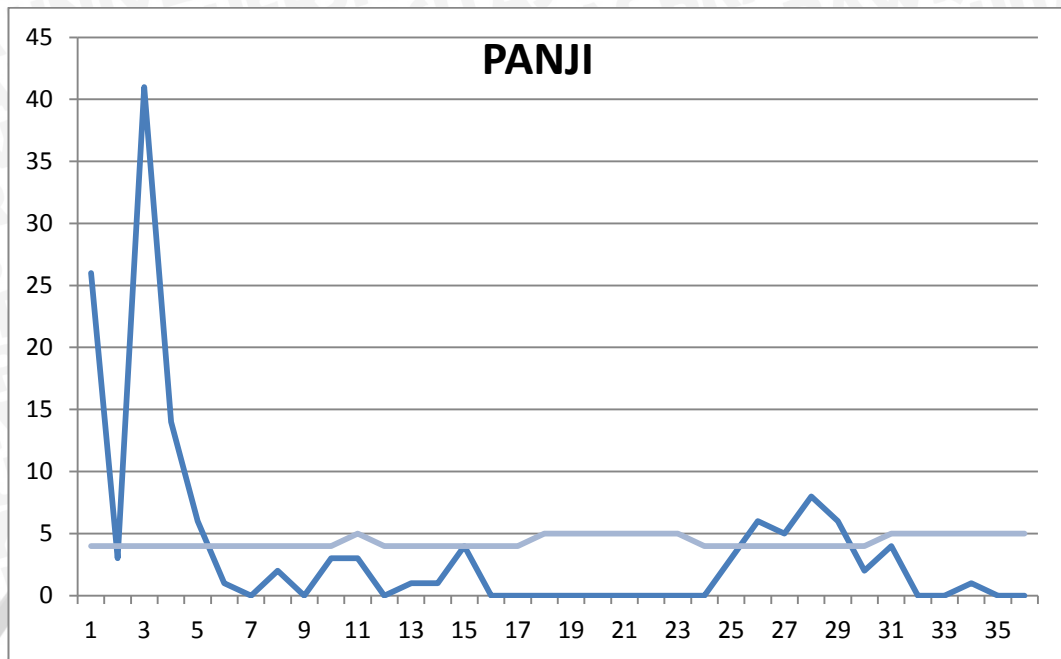
Untuk Pengujian pada lokasi Situbondo didapat hasil  $a = 50$ ,  $c = 0,5$ , MAPE = 81.563963388676 dan RMSE = 5.9744827763116. Adapun hasil dari pengujian ini ditunjukkan pada Gambar 6.2.



Gambar 6.2 Grafik Prediksi Kasus Situbondo

3. Panji

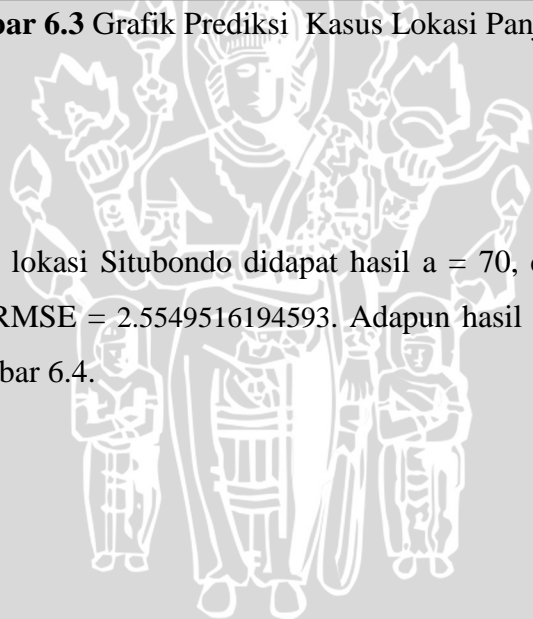
Untuk Pengujian pada lokasi Situbondo didapat hasil  $a = 20$ ,  $c = 0,2$ ,  $MAPE = 66.796981248201$  dan  $RMSE = 7.7531355664087$ . Adapun hasil dari pengujian ini ditunjukkan pada Gambar 6.3.

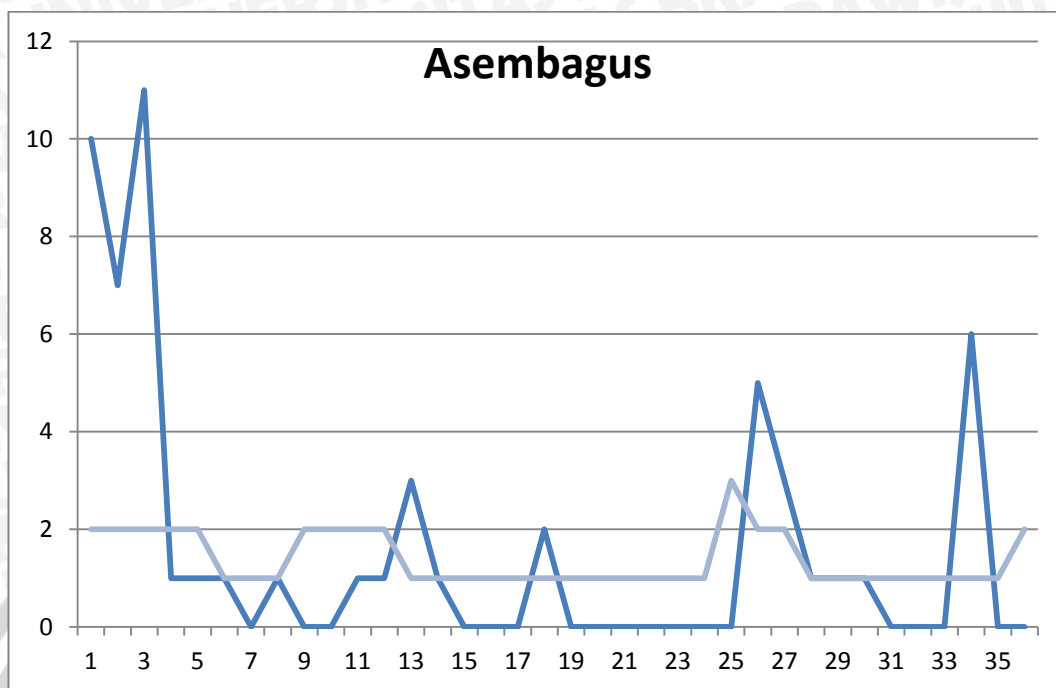


**Gambar 6.3** Grafik Prediksi Kasus Lokasi Panji

4. Asembagus

Untuk Pengujian pada lokasi Situbondo didapat hasil  $a = 70$ ,  $c = 0,1$ ,  $MAPE = 29.348244348244$  dan  $RMSE = 2.5549516194593$ . Adapun hasil dari pengujian ini ditunjukkan pada Gambar 6.4.

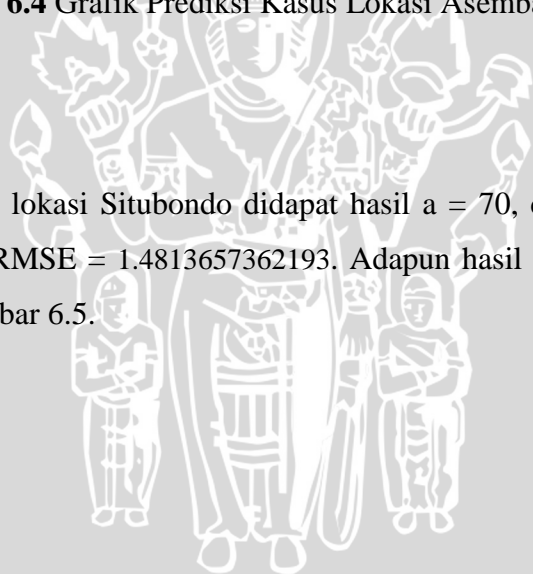


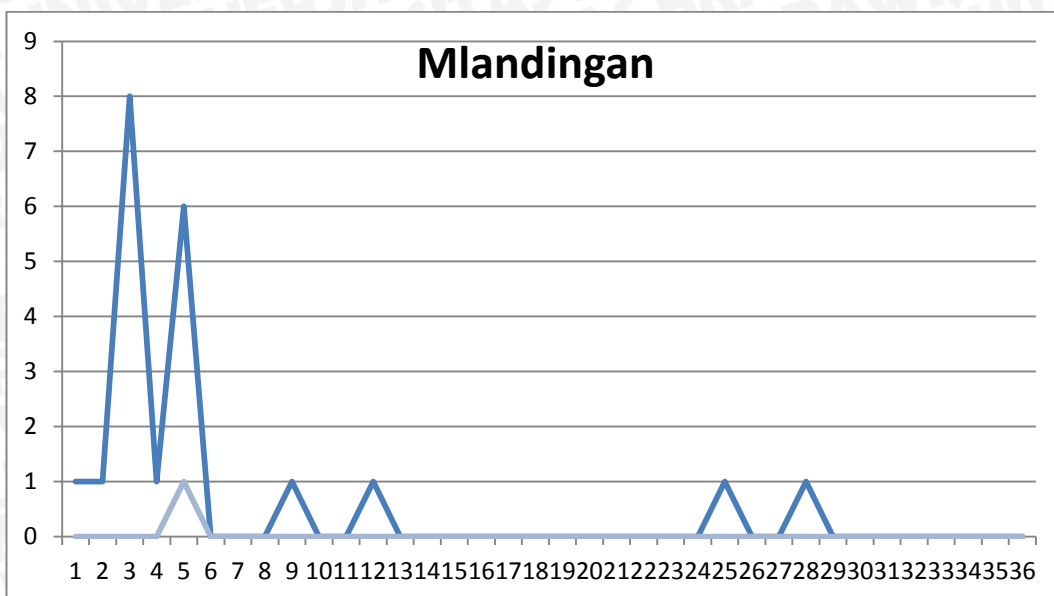


Gambar 6.4 Grafik Prediksi Kasus Lokasi Asembagus

### 5. Mlandingan

Untuk Pengujian pada lokasi Situbondo didapat hasil  $a = 70$ ,  $c = 0,7$ ,  $MAPE = 10.300925925926$  dan  $RMSE = 1.4813657362193$ . Adapun hasil dari pengujian ini ditunjukkan pada Gambar 6.5.

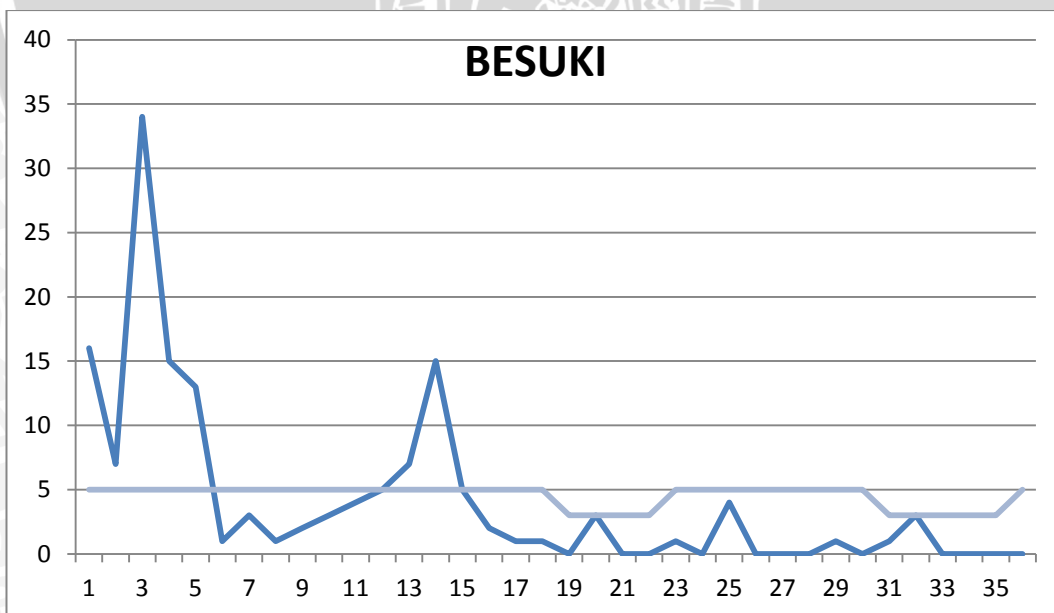




**Gambar 6.5** Grafik Prediksi Kasus Lokasi Mlandingan

3. Besuki

Untuk Pengujian pada kabupaten Besuki didapat hasil  $a = 90$ ,  $c = 0,3$ ,  $MAPE = 65.908867820633$  dan  $RMSE = 6.0415229867973$ . Adapun hasil dari pengujian ini ditunjukkan pada Gambar 6.6.



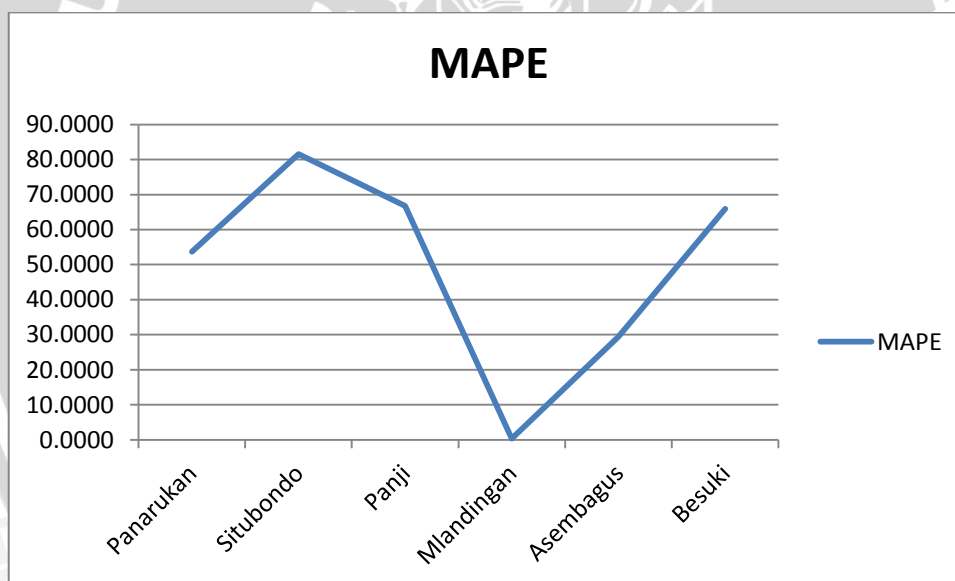
**Gambar 6.6** Grafik Prediksi Kasus Lokasi Besuki

**6.2. Analisis**

Dari hasil pengujian, didapatkan MAPE dan RMSE sesuai persamaan 2-37 dan 2-38 sebagai berikut :

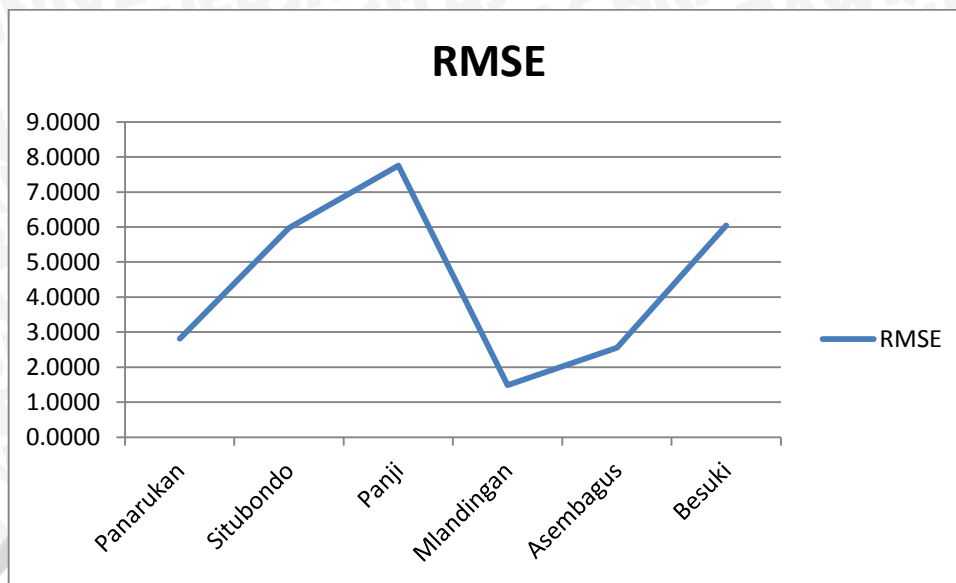
**Tabel 6. 1** Tabel hasil pengujian

Lokasi	a	c	MAPE	RMSE
Panarukan	50	1	53.7103	2.8136
Situbondo	50	0.5	81.5639	5.9744
Panji	20	0.2	66.7969	7.7531
Mlandingan	70	0.7	0.3009	1.4813
Asembagus	70	0.1	29.3482	2.5549
Besuki	90	0.3	65.9088	6.0415
<b>Rata-rata</b>			<b>49.6048</b>	<b>4.4365</b>



**Gambar 6.7** Grafik MAPE di masing- masing Lokasi



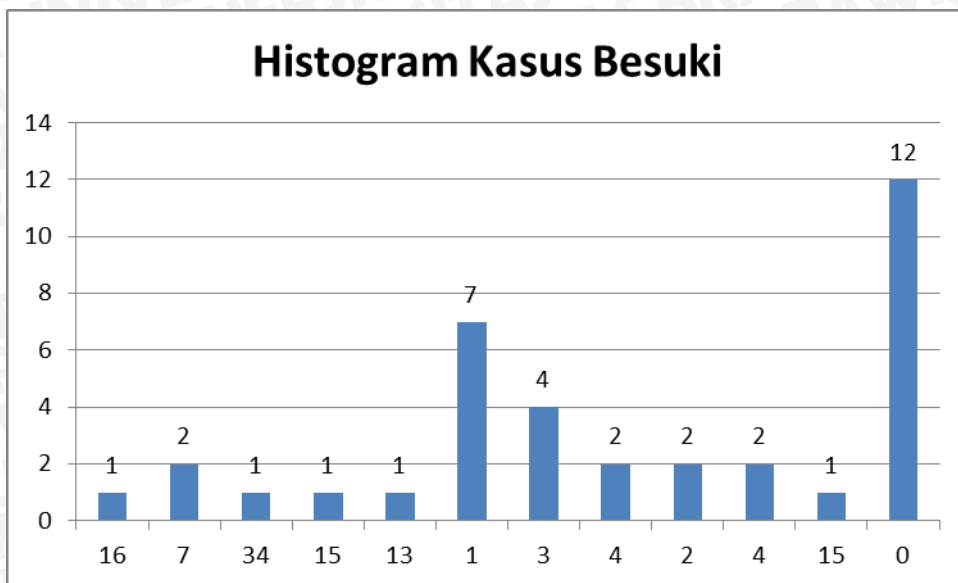


**Gambar 6.8** Grafik RMSE di masing- masing Lokasi

Nilai terendah pada lokasi Mlandingan dengan MAPE = **0.3009** dan RMSE = **1.4813** karena prediksi yang dilakukan paling tepat diantara lainnya.

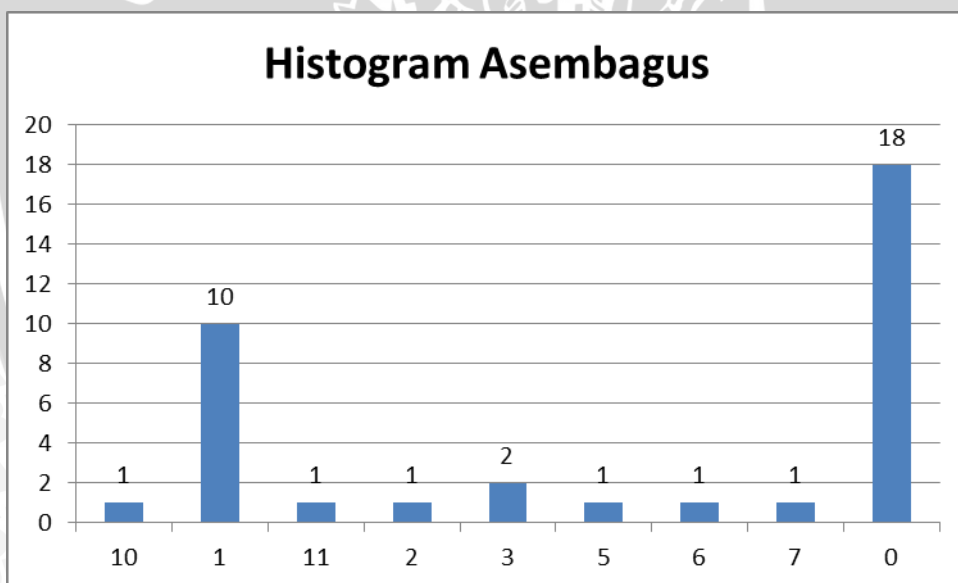
Data tertinggi didapat di lokasi Panji dengan MAPE = **66.7969** dan RMSE = **7.7531**.

Dari hasil pengujian, nilai  $a$  dan  $c$  optimal yang didapatkan berbeda-beda sehingga tidak dapat diimplementasikan disemua lokasi. Adapun gambar-gambar berikut di bawah ini adalah data hasil Histogram pada masing-masing lokasi.



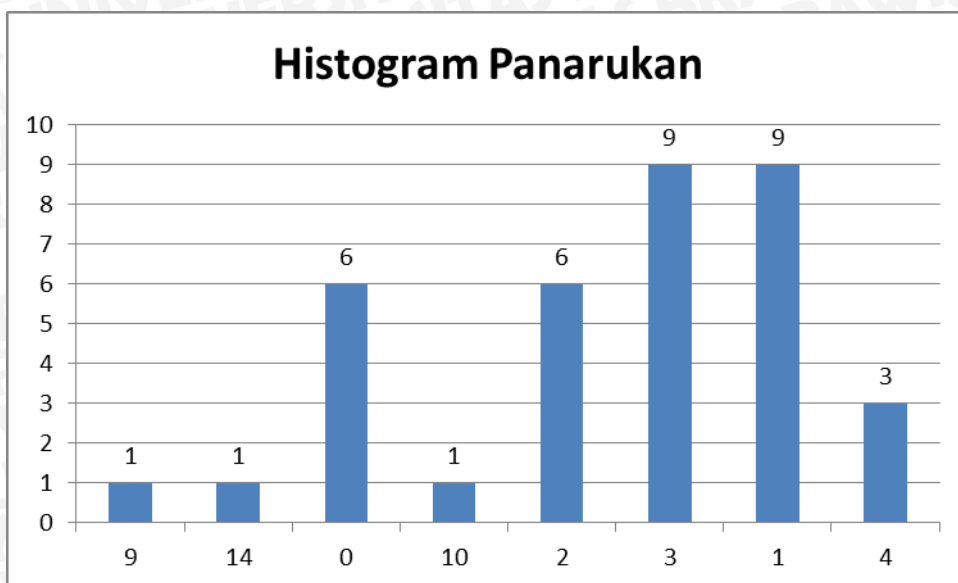
Nilai variansi untuk kasus Besuki adalah 48.551260504202

**Gambar 6.9** Histogram Kasus Besuki



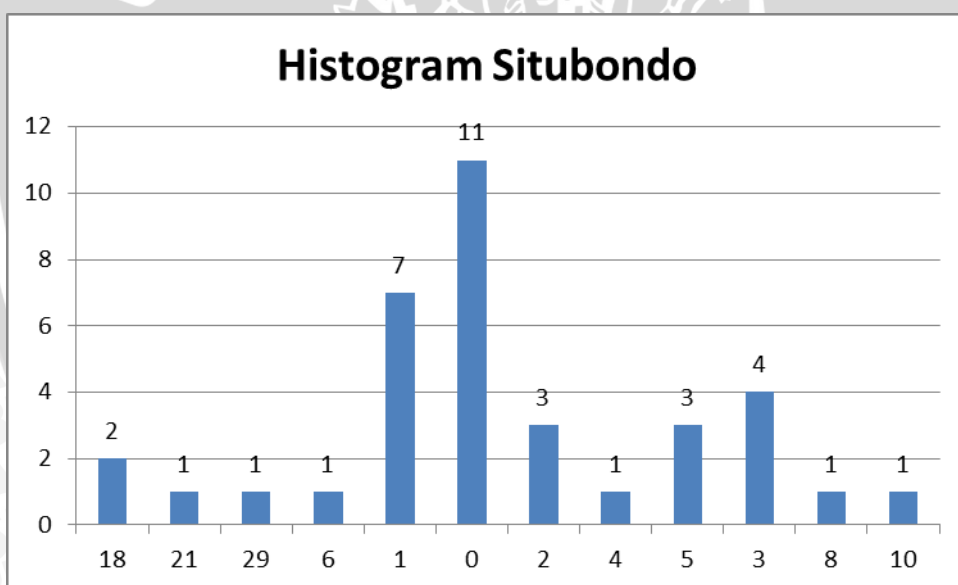
Nilai variansi untuk kasus Asembagus adalah 7.6306306306306

**Gambar 6.10** Histogram Kasus Asembagus



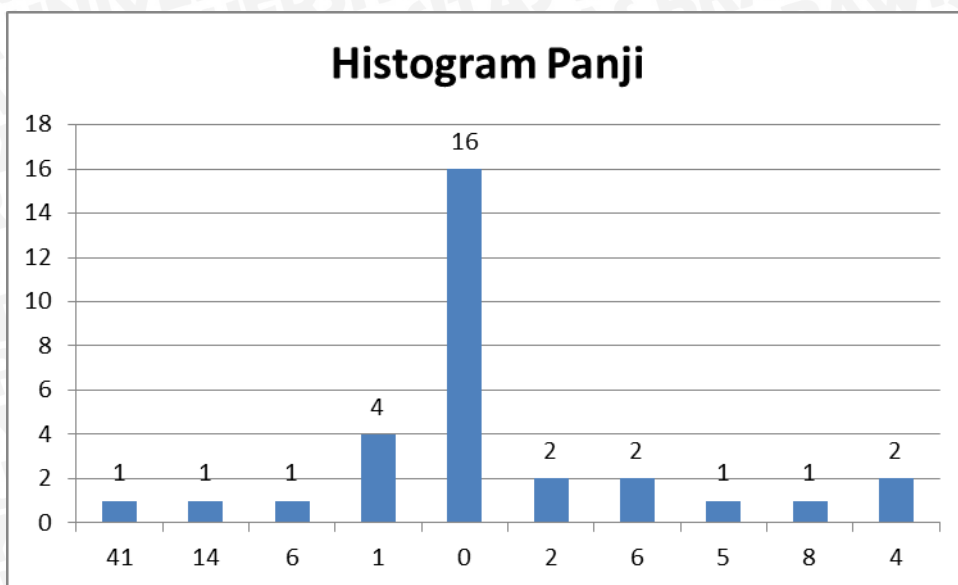
Nilai variansi untuk kasus Panarukan adalah 8.5357142857143

**Gambar 6.11 Histogram Kasus Panarukan**



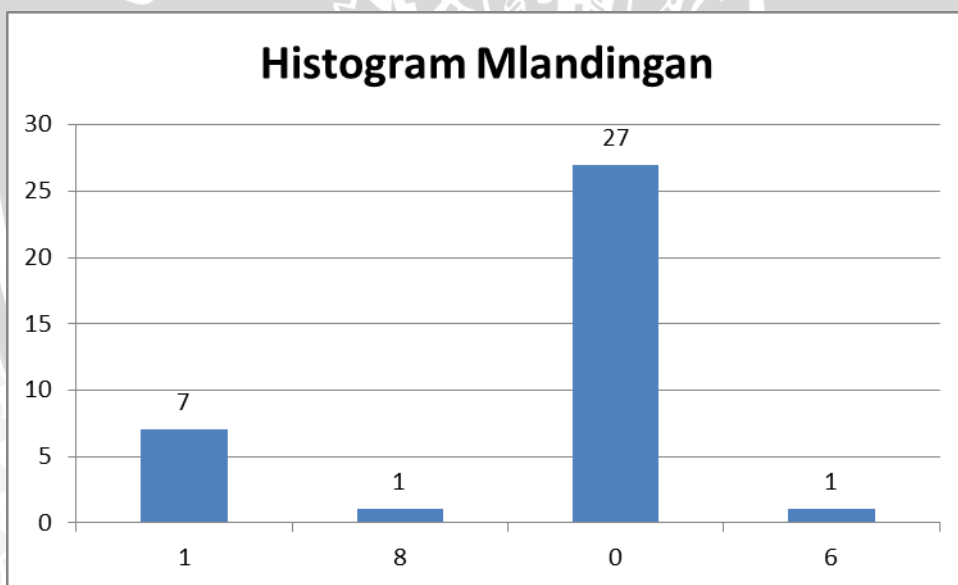
Nilai variansi untuk kasus Situbondo adalah 46.206349206349

**Gambar 6.12 Histogram Kasus Situbondo**



Nilai variansi untuk kasus Panji adalah 65.369047619048

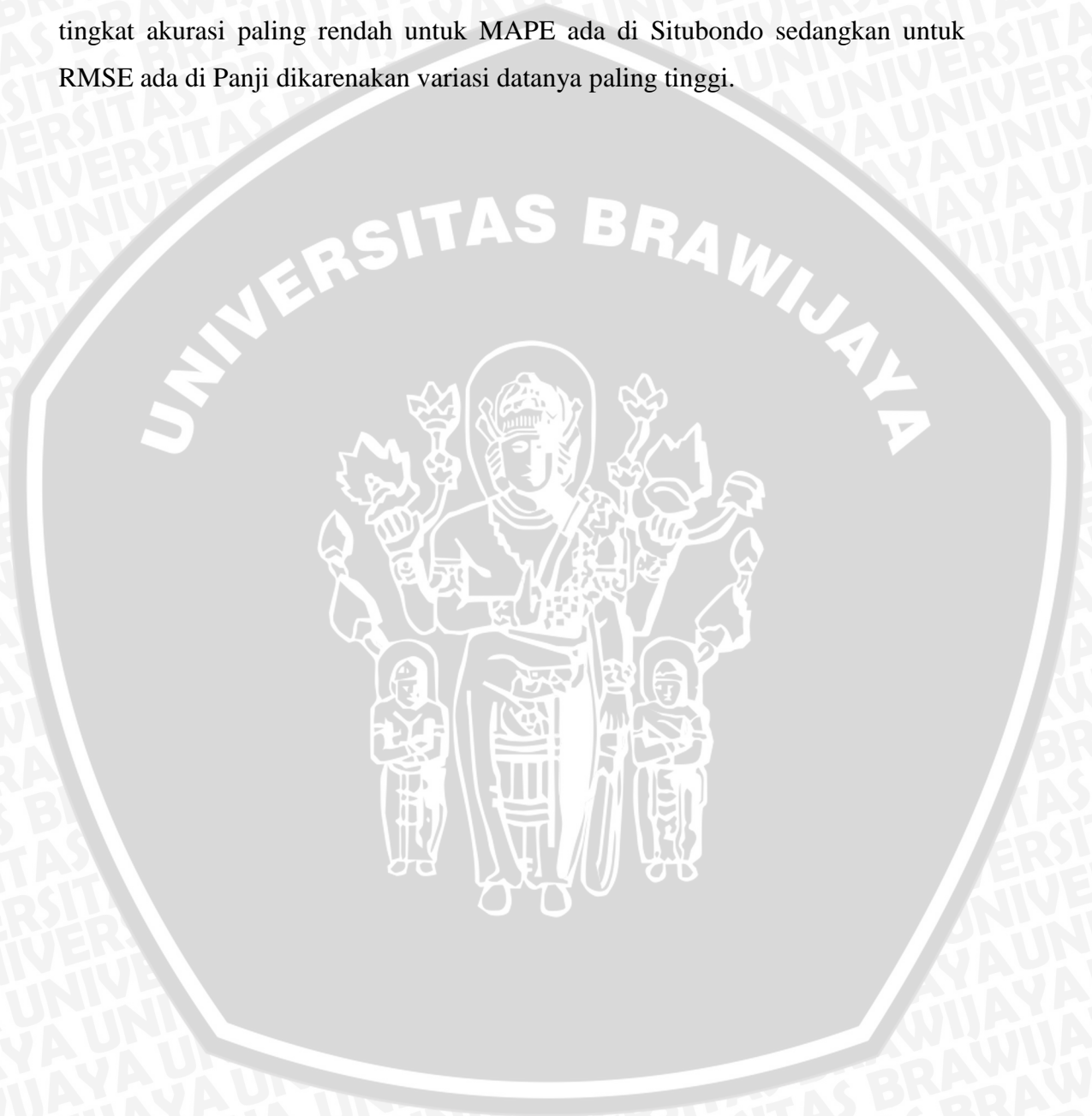
**Gambar 6.13 Histogram Kasus Panji**



Nilai variansi untuk kasus Mlandingan adalah 1.8022727272727

**Gambar 6. 14 Histogram Kasus Mlandingan**

Berdasarkan dari histogram di atas dapat kita simpulkan bahwa tingkat akurasi paling tinggi terdapat pada lokasi Mlandingan karena variansi datanya paling rendah dan terdapat satu data dengan nilai paling tinggi yaitu 0 yang membuat pembelajaran pada fase *training* itu lebih bagus. Sebaliknya data dengan tingkat akurasi paling rendah untuk MAPE ada di Situbondo sedangkan untuk RMSE ada di Panji dikarenakan variasi datanya paling tinggi.



## BAB VII

### PENUTUP

Berdasarkan perancangan, implementasi dan hasil pengujian dari Penelitian, maka didapatkan kesimpulan dan saran sebagai berikut:

#### 7.1. Kesimpulan

Berdasarkan penelitian didapatkan kesimpulan dari segi metode sebagai berikut:

1. Penentuan parameter  $a$  dan  $c$  untuk tiap lokasi tidak sama karena tidak didapatkan nilai yang sama untuk hasil pengujian terbaik dari masing-masing lokasi
2. Nilai terendah dari pengujian ada di lokasi Mlandingan dengan MAPE : 49.604883365333 dan RMSE : 4.436519309092. Nilai tertinggi dari pengujian ada di lokasi Panji dengan MAPE : 66.796981248201 dan RMSE : 7.7531355664087
3. Rata-rata terendah dari seluruh pengujian adalah dengan MAPE : 49.604883365333 dan RMSE : 4.436519309092

#### 7.2. Saran

Saran yang dapat diberikan setelah pengerjaan tugas akhir ini adalah :

1. Untuk penelitian selanjutnya diharapkan melakukan pengujian dengan nilai parameter FCBPN yang lebih bervariasi.
2. Pengujian dapat dilakukan dengan membandingkan dengan metode prediksi lain seperti *Backpropagation* dan ANFIS

**DAFTAR PUSTAKA**

- [AHB-12] Attarius Hicham, Bouhorma Mohammed , dan Sofi Anas. 2012. An improved approach based on fuzzy clustering and Back-propagation Neural Networks with adaptive learning rate for sales forecasting: Case study of PCB industry
- [AHE-12] Attarius Hicham, Bouhorma Mohamed, El Fallahi Abdellah. (2012). An improved approach based on fuzzy clustering and Back-Propagation Neural Networks with adaptive learning rate for sales forecasting: Case study of PCB industry
- [SPD-12] Supriyadi, Didi. 2012. Sistem Informasi Penyebaran Penyakit Demam Berdarah menggunakan Metode Jaringan Syaraf Tiruan *Backpropagation*.
- [CHA-05] Chang, P.-C, & Lai, C.-Y. 2005. A hybrid system combining self-organizing maps with case-based reasoning in wholesalerbs new-release book forecasting. *Expert Systems with Applications*, 29(1),183 192.
- [CHA-06] Chang, P.-C, & Wang, Y.-W. 2006. Fuzzy Delphi and back-propagation model for sales forecasting in PCB industry. *Expert Systems with Applications*, 30(4).
- [CHW-07] Chang, P.-C, Wang, Y.-W., & Liu, C.-H. 2007. The development of a weighted evolving fuzzy neural network for PCB sales forecasting. *Expert Systems with Applications*, 32(1),86 96
- [CHL-09] Chang, P.-C, Liu, C.-H., & Fan, C. Y. 2009. Data clustering and fuzzy neural network for sales forecasting: A case study in printed circuit board industry. *Knowledge Based Systems*, 22(5), 344 355.



- [CHY-07] Chang, P.-C, Yen-Wen Wang, Chen-HaoLiu. 2007. The development of a weighted evolving fuzzy neural network for PCB sales forecasting. *Expert Systems with Applications*
- [SKS-10] Sri Kusumadewi, dan Sri Hartati. 2010. *Neuro-Fuzzy. Integrasi Sistem Fuzzy & Jaringan Syaraf*
- [AGA-98] A.G. Andjaparidze. Guidelines for treatment of Dengue Fever/Dengue Haemorrhagic Fever in Small Hospital. WHO Regional Office for South-Eash Asia. September 1998.
- [ACL-98] Alex Chaerulfalah. Pengelolaan kasus Demam Berdarah Dengue di Bagian Ilmu Kesehatan Anak RS. Dr. Hasan Sadikin/FK. UNPAD, Bandung. Dipresentasikan pada Lokakarya Tatalaksana DBD. Subdit Arbovirosis Dit.Jen. PPM & PLP Depkes RI. Cibogo, September 1998
- [UFA-10] Prof. dr. Umar Fahmi Achmadi MPH,Ph.d. 2010.Bulletin Jendela Epidemiologi Demam Berdarah, Manajemen Demam berdarah berbasis Wilayah. Jakarta
- [AUF-10] Achmadi, Umar fahmi.2010 .Kusumadewi, Sri dan Purnomo Hari. 2010. *Aplikasi Logika Fuzzy Untuk Pendukung Keputusan*, Edisi 2. Yogyakarta: Graha Ilmu
- [SYT-11] Suyanto. 2011. *Artificial Intelligence*. Edisi Revisi. Bandung: Informatika Bandung