

PREDIKSI CURAH HUJAN MENGGUNAKAN ADAPTIVE NEURO FUZZY INFERENCE SYTEM (ANFIS)

¹Sekar Kinasih, ² Fhira Nhita, ³Adiwijaya

Ilmu Komputasi Fakultas Informatika Universitas Telkom, Bandung

¹sekarkinasihh@hotmail.com, ² fhiranhita@telkomuniversity.ac.id, ³ adiwijaya@telkomuniveristy.ac.id

ABSTRACT

Weather is the state of the air at certain times and in certain areas that are relatively narrow in short periods of time. The weather is very important and very influential on human activity. One of the weather factor is rainfall, rainfall is uncertain circumstances and required an adaptive method to predict accurately.

One of the science that is often used to predict the weather is Soft Computing (SC). Adaptive Neuro Fuzzy Inference System (ANFIS) is one of the algorithms in the SC and is a combination of Neural Networks and Fuzzy Logic. Algorithm of Adaptive Neuro Fuzzy Inference System (ANFIS) been selected Neural Network has the ability to learn with training based on existing data and Fuzzy Logic capable of performing classification based on linguistic information and fuzziness. This combination is expected to result in a system that is able to learn continuously and is able to provide output with a good level of performance.

The final task is to implement the ANFIS to predict rainfall for the area Soreang with monthly rainfall data. The data is divided into training data and testing data with the proportion 70:30. After research conducted by a combination of parameters obtained results of testing accuracy by 80%.

Keywords: rainfall, prediction, fuzzy, neural networks, Adaptive Neuro Fuzzy Inference System

ABSTRAK

Cuaca adalah keadaan udara pada saat tertentu dan di wilayah tertentu yang relatif sempit pada jangka waktu yang singkat. Cuaca sangat penting dan sangat berpengaruh terhadap aktivitas semua makhluk hidup. Salah satu faktor cuaca adalah curah hujan, curah hujan adalah keadaan yang tidak pasti dan diperlukan sebuah metode yang adaptif untuk memprediksi secara akurat.

Salah satu bidang ilmu yang sering digunakan untuk memprediksi cuaca adalah *Soft Computing* (SC). *Adaptive Neuro Fuzzy Inference System* (ANFIS) merupakan salah satu algoritma didalam SC dan merupakan penggabungan dari Jaringan Syaraf Tiruan dan Logika *Fuzzy*. Algoritma *Adaptive Neuro Fuzzy Inference System* (ANFIS) dipilih karena Jaringan Syaraf Tiruan memiliki kemampuan untuk belajar dengan berdasarkan pelatihan pada data yang ada sebelumnya dan Logika *Fuzzy* mampu melakukan klasifikasi berdasarkan informasi linguistik dan *fuzziness*. Kombinasi ini diharapkan dapat menghasilkan suatu sistem yang mampu belajar secara terus menerus dan mampu memberikan output dengan tingkat performansi yang baik.

Tugas akhir ini mengimplementasikan ANFIS untuk memprediksi curah hujan untuk wilayah Soreang dengan data curah hujan bulanan. Data dibagi menjadi data pelatihan dan data pengujian dengan proporsi 70:30. Setelah dilakukan penelitian dengan kombinasi parameter didapatkan hasil performansi pengujian sebesar 80%.

Kata kunci : curah hujan, prediksi, *fuzzy*, jaringan syaraf tiruan, *Adaptive Neuro Fuzzy Inference System*

I. Pendahuluan

I.1 Latar Belakang

Cuaca sangat penting dan sangat berpengaruh terhadap aktifitas semua makhluk hidup. Curah hujan merupakan salah satu faktor dari cuaca yang juga berpengaruh terhadap aktifitas makhluk hidup. Oleh karena itu, terlihat bahwa penting sekali mengetahui prediksi curah hujan untuk hari tertentu sesuai kebutuhan.

Untuk memprediksi cuaca diperlukan suatu metode yang adaptif dan bisa memprediksi secara akurat. Salah satu bidang ilmu yang sering digunakan untuk memprediksi cuaca adalah *Soft Computing* (SC). Bidang ilmu tersebut sering didefinisikan untuk sekumpulan metodologi yang berkembang secara terus-menerus, yang bertujuan untuk menghasilkan sistem yang memiliki toleransi terhadap ketidakakuratan, ketidakpastian dan kebenaran parsial untuk mencapai ketahanan, bisa ditelusuri dan biaya rendah[2].

Algoritma *Adaptive Neuro Fuzzy Inference System* (ANFIS) termasuk salah satu algoritma *hybrid* yang berada didalam bidang ilmu *Soft Computing* (SC). ANFIS merupakan penggabungan dari Jaringan Syaraf Tiruan dan Logika *Fuzzy*. Algoritma ini dipilih karena Jaringan Syaraf Tiruan memiliki kemampuan untuk belajar berdasarkan pelatihan pada data yang ada sebelumnya dan Logika *Fuzzy* mampu melakukan klasifikasi berdasarkan informasi linguistik. Kombinasi ini diharapkan dapat menghasilkan suatu sistem yang mampu belajar secara terus menerus dan mampu memberikan *output* dengan tingkat performansi yang baik.

I.2 Perumusan Masalah

Berdasarkan latar belakang masalah diatas, maka masalah-masalah yang akan diselesaikan pada tugas akhir ini adalah:

- a. Bagaimana mengimplentasikan *Adaptive Neuro Fuzzy Inference System* untuk kasus prediksi curah hujan ?
- b. Bagaimana performansi sistem peramalan cuaca yang dibuat dengan *Adaptive Neuro Fuzzy Inference System* ?
- c. Bagaimana kompleksitas waktu asimtotik (Big O) algoritma *Adaptive Neuro Fuzzy Inference System* yang dibangun?

I.3 Tujuan

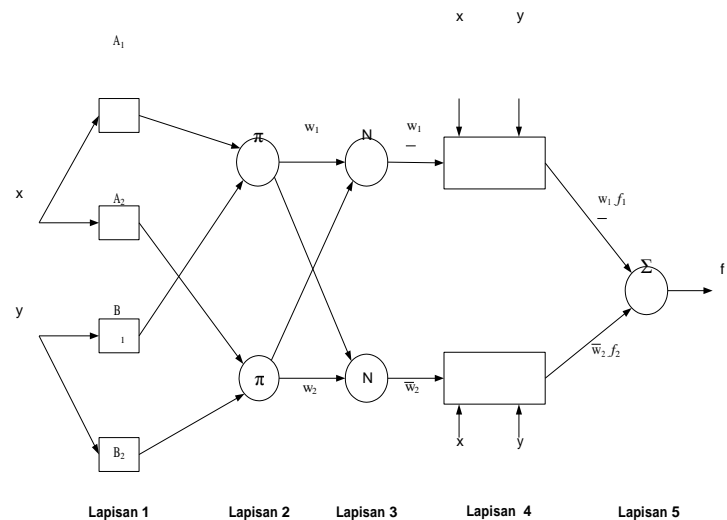
Berdasarkan perumusan masalah, maka tujuan pembuatan tugas akhir ini adalah:

- a. Menganalisis dan mengimplementasikan *Adaptive Neuro Fuzzy Inference System* untuk prediksi curah hujan.
- b. Menganalisis performansi *Adaptive Neuro Fuzzy Inference System* dalam prediksi curah hujan
- c. Menganalisis kompleksitas waktu asimtotik (Big O) dari algoritma *Adaptive Neuro Fuzzy Inference System*.

II. LANDASAN TEORI

II.1 Arsitektur Adaptive Neuro Fuzzy Inference System (ANFIS)

Adaptive Neuro Fuzzy Inference System (ANFIS) adalah algoritma *hybrid* yang merupakan penggabungan mekanisme *Fuzzy Inference System* yang digambarkan dalam arsitektur jaringan syaraf. ANFIS mempunyai dua parameter yaitu parameter premis dan parameter konsekuen. Pelatihan hybrid dalam algoritma ini dilakukan dengan dua langkah yaitu langkah maju dan langkah mundur. Untuk sederhananya, diasumsikan bahwa *Fuzzy Inference System* yang diperhatikan memiliki dua masukan x dan y dan satu keluaran f. Arsitektur ANFIS dapat digambarkan sebagai berikut :



Gambar 2.1 Arsitektur ANFIS

Pada gambar terlihat bahwa arsitektur ANFIS terdiri atas 5 lapis dan setiap lapis mempunyai fungsi yang berbeda. Terdapat simpul di

setiap lapisan yang memiliki dua bentuk yang berbeda. Simpul adaptif (bersymbol kotak) dan

simpul tetap (bersymbol lingkaran). Fungsi dari setiap lapis adalah sebagai berikut [2] :

Lapisan 1

Semua simpul pada lapisan ini adalah simpul adaptif (parameter dapat berubah) dengan fungsi simpul :

$$O_{1,i} = \mu_{A_i}(x) \quad i=1,2$$

$$O_{1,i} = \mu_{B_{i-2}}(y) \quad i=3,4 \quad (2.1)$$

Dimana x dan y adalah masukan pada simpul i , μ_{A_i} atau $\mu_{B_{i-2}}$ adalah fungsi keanggotaan masing-masing simpul. Simpul $O_{1,i}$ berfungsi untuk menyatakan derajat keanggotaan tiap masukan terhadap himpunan fuzzy A dan B . Fungsi keanggotaan yang dipakai adalah jenis *generalized bell* (gbell) dan berikut ini adalah rumus gbell.

$$\mu_A(x) = \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}} \quad (2.2)$$

Parameter a , b , c pada fungsi keanggotaan gbell dinamakan parameter premis yang adaptif.

Lapisan 2

Semua simpul pada lapisan ini adalah nonadaptif (parameter tetap). Fungsi simpul ini adalah mengalikan setiap sinyal masukan yang datang. Fungsi simpul :

$$O_{2,i} = w_i = \mu_{A_i}(x) \Delta \mu_{B_i}(y), i = 1,2$$

Sehingga,

$$w_1 = \mu_{A_1}(x) \text{AND} \mu_{B_1}(y)$$

$$w_2 = \mu_{A_2}(x) \text{AND} \mu_{B_2}(y) \quad (2.3)$$

Tiap keluaran simpul menyatakan derajat pengaktifan (*firing strength*) tiap aturan fuzzy. Fungsi ini dapat diperluas apabila bagian premis memiliki lebih dari dua himpunan fuzzy. Banyaknya simpul pada lapisan ini menunjukkan banyaknya aturan yang dibentuk.

Fungsi perkalian yang digunakan adalah interpretasi kata hubungan dengan menggunakan operator t-norm.

Lapisan 3

Setiap simpul pada lapisan ini adalah simpul

$$O_{3,i} = \overline{w_i} = \frac{w_i}{w_1 + w_2}, i = 1,2 \quad (2.4)$$

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

Lapisan 4

Semua simpul pada lapisan ini adalah simpul adaptif (parameter dapat berubah) dengan fungsi simpul :

$$O_{4,i} = w_i f_i = w_i (p_i x + q_i y + r_i) \quad (2.5)$$

Dengan adalah derajat pengaktifan ternormalisasi dari lapisan tiga dan parameter p , q , r menyatakan parameter konsekuen yang adaptif.

Lapisan 5

Pada lapisan ini hanya terdapat satu simpul tetap

yang fungsinya untuk menjumlahkan semua masukan. Fungsi simpul :

$$O_{5,i} = \sum_i \overline{w_i} f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad (2.6)$$

II.2 Algoritma Pelatihan Hybrid

ANFIS dilatih dengan pelatihan *Hybrid* yang terdiri atas dua langkah, yaitu langkah maju dan langkah mundur. Untuk langkah maju dari algoritma pelatihan *Hybrid* keluaran dari simpul maju hingga mencapai lapisan ke-4 dan parameter-parameter konsekuensi diidentifikasi dengan *Least Square*

Estimator. Pada langkah mundur, sinyal eror

dirambatkan mundur dan parameter premis diperbaharui dengan *Gradien Descent*. Jika nilai dari parameter premis tetap maka keluaran keseluruhannya dapat dinyatakan dengan kombinasi linier dari parameter konsekuen, yaitu[1]:

$$f = \frac{w_1}{w_1 + w_2} f^1 + \frac{w_2}{w_1 + w_2} f^2$$

$f = w_1 f_1 + w_2 f_2$
 nonadaptif mena an si penga terno
 f yang mpilk fung derajat ktifan malisa
 4

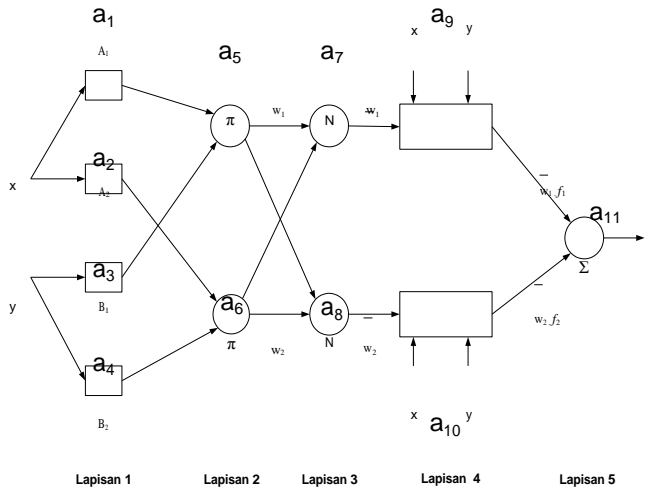
s_i (*normalized firing strength*) yaitu rasio keluaran simpul ke-i pada lapisan sebelumnya terhadap seluruh keluaran lapisan sebelumnya, dengan bentuk fungsi simpul:

(2.7)

$$f = \overline{(w_1 x)} p_1 \overline{(w_1 y)} q_1 \overline{(w_1)} r_1 \overline{(w_2 x)} p_2 \overline{(w_2 y)} q_2 \overline{(w_2)} r_2$$

II.2.1 Langkah maju dengan Least Square

Estimator



Gambar 2.2 Arsitektur pada langkah maju

Mekanisme pada langkah maju adalah sebagai berikut:

Lapisan 1

Pada lapisan ini didefinisikan parameter-parameter fungsi keanggotaan {a,b,c} pada masing-masing simpul. Nilai parameter – parameter ini bersifat adaptif dan akan berubah seiring pembelajaran yang dilakukan oleh jaringan.

Lapisan 2

Pada lapisan ini akan menerapkan *fuzzy* AND untuk semua masukkan yang masuk untuk mendapatkan *firing strength*. Logika *fuzzy* AND akan mencari nilai minimum dari semua inputan yang masuk.

Lapisan 3

Pada lapisan ini akan melakukan normalisasi berdasarkan hasil keluaran dari lapisan 2. Proses normalisasi dilakukan berdasarkan persamaan (2.4).

Lapisan 4

Pada lapisan ini digunakan metode *Least Square Estimator* (LSE) untuk mendapatkan nilai parameter {p,q,r}. Berdasarkan persamaan (2.5) dan berdasarkan ilustrasi pada gambar 2.2, nilai yang masuk pada lapisan ini didefinisikan sebagai matriks A, yaitu:

$$\theta = (A^T A)^{-1} A^T y \tag{2.8}$$

Dimana, y adalah nilai aktual

$$\theta = [p_1 \ q_1 \ r_1 \ p_2 \ q_2 \ r_2] \tag{2.9}$$

Kemudian didapatkan:

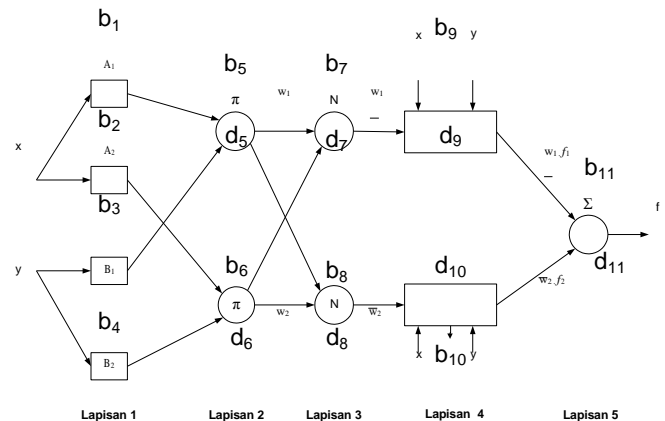
$$f_1 = p_1 x + q_1 y + r_1$$

$$f_2 = p_2 x + q_2 y + r_2 \tag{2.10}$$

Lapisan 5

Simpul pada lapisan ini berfungsi untuk menghitung semua *output* dari lapisan 4.

II.2.2 Langkah mundur dengan Gradient Descent



Gambar 2.3 Arsitektur pada langkah mundur

Dari arsitektur *Adaptive Neuro-Fuzzy Inference System* (ANFIS), dimisalkan bahwa himpunan data pelatihan mempunyai P pasangan *input-output* yang diinginkan, kemudian jumlah kuadrat kesalahan yang merupakan selisih hasil prediksi dan data aktual dapat dijabarkan sebagai berikut[1] :

$$E_p = \sum_{k=1}^N (d_{p,k} - f_{p,k})^2 \tag{2.11}$$

Dimana,

2. $d_{p,k}^A = [(a_7 * x) \text{ hasil output ke } k]$
3. $f_{p,k}$ Penerapan LSE: nilai data yang sebenarnya ke k

1. $\sum_{k=1}^N$: komponen ke- k ($1 \leq k \leq N$)

Untuk perhitungan *error* pada langkah mundur untuk masing-masing lapisan adalah sebagai berikut :

Error pada lapisan 5

Pada lapisan ini akan menghitung selisih keluaran dengan data aktual, berdasarkan persamaan (2.11) maka perhitungan error untuk lapisan ini adalah:

$$\varepsilon_{11} = \frac{\partial E_p}{\partial x_{i,k}} = -2(y - a_{11}) \tag{2.12}$$

Dimana,

- y = nilai aktual
- a_{11} = nilai *output* pada langkah maju

Sehingga $b_{11} = \varepsilon_{11}$.

Error pada lapisan 4

Nilai pada simpul $b_{11} = b_9 + b_{10}$ karena pada langkah maju $a_{11} = a_9 + a_{10} = \overline{w_1 f_1} + \overline{w_2 f_2}$

Sehingga,

$$- d_{11} = \overline{w_1 f_1} + \overline{w_2 f_2} = a_{11}$$

$$- b_9 = \left| \frac{\partial \varepsilon_p}{\partial a_{11}} \right| \left| \frac{\partial d_{11}}{\partial a_9} \right| = b_{11} \left| \frac{\partial d_{11}}{\partial a_9} \right| = b_{11}(1),$$

sehingga $b_9 = b_{11}$ (2.13)

$$- b_{10} = \left| \frac{\partial \varepsilon_p}{\partial a_{11}} \right| \left| \frac{\partial d_{11}}{\partial a_8} \right| = b_{11} \left| \frac{\partial d_{11}}{\partial a_8} \right| = b_{11}(1),$$

sehingga $b_{10} = b_{11}$ (2.14)

Error pada lapisan 3

Nilai a_9 berasal dari a_7 sehingga nilai b_9 juga

berkorespondensi dengan nilai b_7 .

- Pada tahap maju $a_9 = w_1 f_1 = d_9$, maka

$$\frac{\partial d_9}{\partial w_1} = f_1$$

$$b_7 = \left| \frac{\partial \varepsilon_p}{\partial a_{11}} \right| \left| \frac{\partial d_{11}}{\partial a_9} \right| \left| \frac{\partial d_{11}}{\partial a_7} \right|, \text{ karena } \left(\frac{\partial \varepsilon_p}{\partial a_{11}} \right) \left(\frac{\partial d_{11}}{\partial a_9} \right)$$

$$= \left| \frac{d_9}{\partial a_7} \right|$$

-Pada simpul d_7 dilihat

bahwa: $\frac{\partial d_7}{\partial w_1} = \frac{\partial d_7}{\partial a_5} = \frac{w_2}{(w_1 + w_2)^2}$ dan

$$\frac{\partial d_8}{\partial w_1} = \frac{\partial d_8}{\partial a_5} = -\frac{w_2}{(w_1 + w_2)^2}$$

Sehingga didapat,

$$b_5 = \frac{w_2}{(w_1 + w_2)^2} (b_7 - b_8) \tag{2.16}$$

Dan cara yang sama untuk mencari b_6 .

Error pada lapisan 1

Simpul b_1, b_2, b_3, b_4 adalah nilai keanggotaan dari

input x dan y dan simpul b_1, b_3 berkorespondensi

hanya dengan node b_5 , sehingga,

$$b = b \left| \frac{\partial d_5}{\partial a_1} \right| = b \mu_{a^3}(y) \tag{2.17}$$

$$b_2 = b_6 \left| \frac{\partial d_6}{\partial a_2} \right| = b_6 \mu_{a^4}(y) \tag{2.18}$$

$$b_3 = b_5 \left| \frac{\partial d_5}{\partial a} \right| = b_5 \mu_{a^1}(x) \tag{2.19}$$

$$b_4 = b_6 \left| \frac{\partial d_6}{\partial a_4} \right| = b_6 \mu_{a^2}(x) \tag{2.20}$$

Selanjutnya, nilai tersebut diturunkan untuk mendapatkan nilai parameter a dan c yang baru.

II.3 Pengukuran Performansi

Pengukuran performansi pada penelitian ini

(2.15)

$$|b_7 - b_{11}| = b_9 f_1$$

dan cara yang sama untuk mencari b_8 .

Error pada lapisan 2

Simpul b_7 dan b_8 keduanya berkorespondensi

dengan node b_5 dan b_6 .

$$d_7 = a_7 = \frac{w_1}{w_1 + w_2} = \frac{a_5}{a_5 + a_6} \text{ dan}$$

$$d_8 = a_8 = \frac{w_2}{w_1 + w_2} = \frac{a_6}{a_5 + a_6}$$

dieva
denga
dika
perfo
sian

menggunakan *Weighted Mean Absolute Percentage Error* (WMAPE) [3].

$$WMAPE = \frac{\sum \left| \frac{D_i - F_i}{D_i} \right| * D_i}{\sum D_i} * 100\%$$

(2.21)

Dimana,

D_i : Nilai aktual pada periode ke t

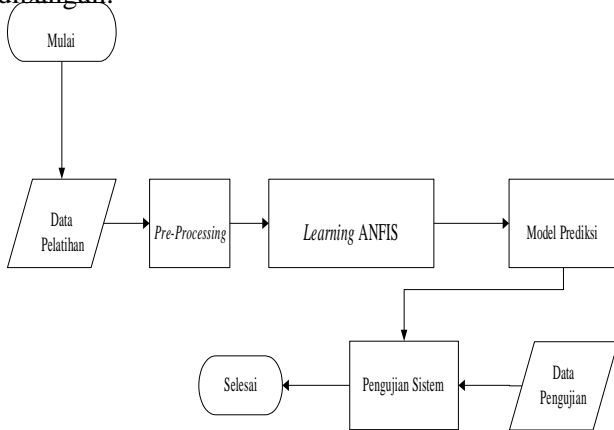
F_i : Nilai Peramalan pada periode ke t

$D_i - F_i$: Nilai Kesalahan (*error*) pada periode ke t

III. GAMBARAN SISTEM

Pada penelitian ini dilakukan pre-processing terlebih dahulu terhadap data yang akan digunakan,

kemudian data akan di partisi menjadi data pelatihan dan data pengujian dengan proporsi masing-masing 70:30. Data yang digunakan adalah data wilayah Soreang yang diambil dari BMKG Bandung selama 10 tahun. Setelah data dimasukkan kedalam proses pelatihan akan didapatkan hasil prediksi yang akan dihitung performansinya menggunakan Weighted Mean Absolute Percentage Error (WMAPE). Berikut ini merupakan diagram proses dari sistem yang akan dibangun:



Gambar 3.1 Deskripsi umum sistem

III.1 Pre-processing data

Tahap ini merupakan tahap pengolahan data yang siap untuk di proses. Tahap-tahap ini meliputi Moving Average dan partisi data. Berikut ini merupakan penjelasan dari masing-masing:

- Moving Average

Moving Average adalah metode peramalan rata-rata nilai dengan mengambil sekelompok nilai pengamatan yang kemudian dicari rata-ratanya, lalu menggunakan rata-rata tersebut sebagai ramalan untuk periode berikutnya. Pada tugas akhir ini penggunaan *moving average* bertujuan untuk mencari pola data dan juga untuk menghilangkan kemungkinan adanya nilai kosong pada data curah hujan bulanan. Berikut ini merupakan rumus *moving average* [4]:

$$S_t = \frac{1}{n} \sum_{j=i}^{i+n-1} a_j$$

(3.1)

Keterangan :

- s_i = *Moving average* untuk periode ke-i
- a_j = Nilai aktual periode ke-j
- n = Jumlah batas dalam *moving average*

- Normalisasi Data

Normalisasi yaitu mengubah data aktual menjadi nilai *range* dengan interval [0..1] yang bertujuan untuk meminimumkan nilai *error*. Berikut ini merupakan rumus untuk normalisasi[5]:

$$X'_n = \left(\frac{X_n - Min_x}{Max_x - Min_x} \times 0.8 \right) + 0.1 \tag{3.2}$$

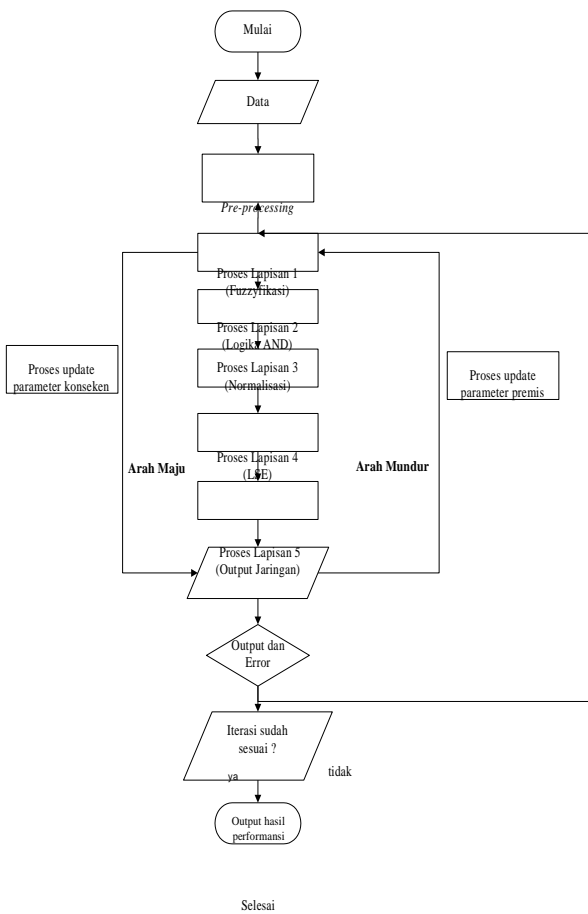
Keterangan :

- X'_n = Data normalisasi ke - n
- X_n = Data aktual ke-n
- Min_x = Data minimum variabel x
- Max_x = Data maksimum variabel x

III.2 Pelatihan *Adaptive Neuro Fuzzy Inference System*

Proses pelatihan pada *Adaptive Neuro Fuzzy Inference System* (ANFIS) menggunakan dua langkah yaitu langkah maju dan langkah mundur. Langkah maju akan membangkitkan nilai parameter

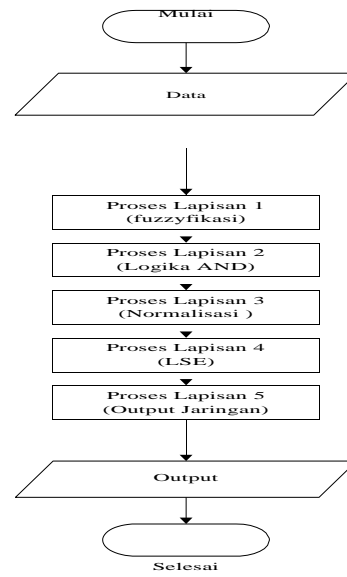
premis di iterasi pertama pada langkah maju dan memperbaiki parameter konsekuen di lapisan empat dengan menggunakan *Least Square Estimator* (LSE). Sedangkan, pada langkah mundur akan mempropagasi balik nilai maju melalui *Backpropagation* dengan menggunakan *Gradient Descent* untuk memperbaiki parameter premis yang berada di lapisan satu. Berikut ini merupakan ilustrasi proses pelatihan ANFIS :



Gambar 3.2 Flowchart Pelatihan ANFIS

III.3 Pengujian Sistem

Proses pengujian dilakukan setelah mendapatkan parameter premis dan konsekuen yang telah di dapatkan dari proses pelatihan. Proses pengujian hanya menerapkan tahap maju pada pada ANFIS. Tujuan dilakukannya proses pengujian adalah untuk mengetahui seberapa baik sistem yang telah dibangun dalam proses pelatihan memberikan hasil prediksi yang mendekati nilai sebenarnya. Berikut ini merupakan ilustrasi pengujian pada ANFIS :



Gambar 3.3 Flowchart Pengujian ANFIS

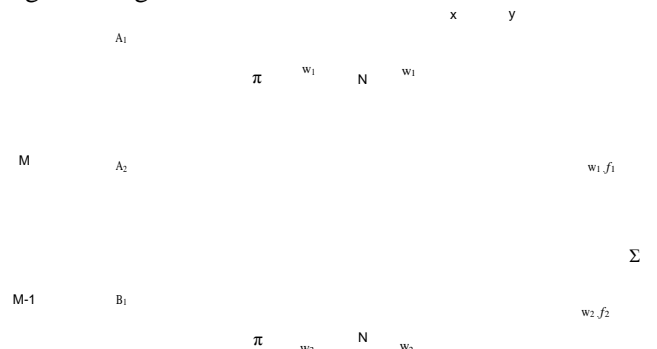
IV. HASIL DAN ANALISIS

Pada proses pelatihan ini digunakan beberapa kombinasi untuk mendapatkan arsitektur terbaik dengan *error* terkecil. Beberapa kombinasi yang digunakan akan dijelaskan pada tabel berikut ini :

Tabel I Tabel Skenario

Skenario	Input	Himpunan Fuzzy	Rule	Iterasi	Learning Rate
1 - 36	2	[2, 3, 4]	[min	[50,	[0.1,
37 - 72	3		,	70,	0.01]
73 - 108	4		max]	100]	

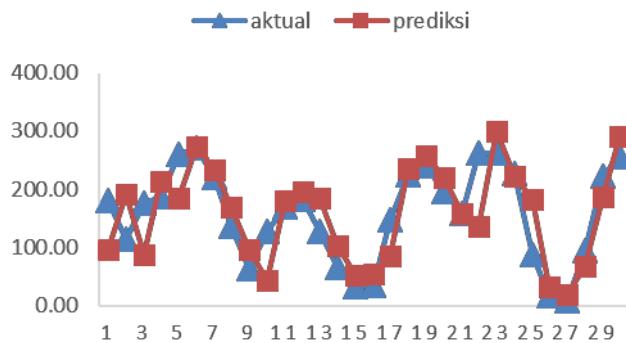
Setelah dilakukan 108 skenario berdasarkan tabel diatas, didapatkan hasil terbaik dengan hasil performansi 91,17% pada proses pelatihan dan 80% pada proses pengujian. Berikut ini adalah arsitektur terbaik yang didapatkan dari beberapa kombinasi yang telah digunakan :



Gambar 4.1 Arsitektur Terbaik

Arsitektur ini menggunakan dua *input* yaitu M dan M-1 pada simpul *input*, lalu pada lapisan pertama menggunakan dua simpul pada setiap input yang berarti menggunakan 2 nilai linguistik, pada lapisan kedua menggunakan *rule* minimum (*half-connected*) yang berarti *rule* = jumlah nilai linguistik. Pada proses pelatihan ini menggunakan *learning rate* 0,1 dan iterasi sebanyak 100.

Dengan nilai parameter yang telah didapatkan dari proses pelatihan yang menghasilkan hasil performansi 91,78%, maka dilakukan pengujian dengan data uji dan didapatkan hasil nilai WMAPE 20% yang berarti performansinya adalah 80 %. Hasil perbandingan hasil prediksi dengan data aktual digambarkan dalam grafik berikut ini :



Gambar 4.2 Grafik perbandingan hasil

IV.1 Analisis Waktu Asimtotik

Pada bagian ini akan membahas tentang kompleksitas waktu asimtotik pada algoritma *Adaptive Neuro Fuzzy Inference System* (ANFIS) yang telah dibuat berdasarkan kompleksitas asimtotik (big O). Arsitektur ANFIS terdiri dari 5 lapisan dan berikut ini adalah perhitungan kompleksitas waktu pada setiap lapisan :

Lapisan 1
 $T_n = n^2 = O(n^2)$ (4.1)

Lapisan 2
 $T_n = 2 + n^2 = O(n^2)$ (4.2)

Lapisan 3
 $T_n = 1 + n = O(n)$ (4.3)

Lapisan 4
 $T_n = 3 + n + n^2 + n^3 = O(n^3)$ (4.4)

Lapisan 5
 $T_n = 1 = O(n)$ (4.5)

V. KESIMPULAN DAN SARAN

V.1 Kesimpulan

Dari hasil pengujian dapat diambil beberapa kesimpulan, antara lain :

1. Karakteristik data yang fluktuatif menghasilkan hasil performansi yang lebih rendah jika dibandingkan dengan penggunaan data yang tidak terlalu fluktuatif.
2. Beberapa kombinasi yang digunakan seperti jumlah *input*, jumlah himpunan *fuzzy*, jumlah *rule*, nilai *learning rate* dan iterasi dapat mempengaruhi hasil performansi, dapat dilihat dari hasil performansi yang didapatkan ketika melakukan beberapa percobaan.
3. Setelah dilakukan proses pembelajaran, arsitektur dengan 2 jumlah *input*, 2 himpunan *fuzzy*, *rule* (*half-connected*) adalah arsitektur terbaik yang menghasilkan performansi pengujian sebesar 80%.

V.2 Saran

Beberapa saran untuk penelitian ini, antara lain :

1. Penggunaan parameter premis dan konsekuen yang berbeda.
2. Penggunaan jumlah *input*, jumlah himpunan *fuzzy*, jumlah *rule* yang berbeda.
3. Penggunaan *learning rate* yang berbeda.

DAFTAR PUSTAKA

[1] Kusumadewi, Sri, Hartati, Sri, 2010. *Neuro-Fuzzy: Integrasi Sistem Fuzzy & Jaringan Syaraf Edisi 2*. Yogyakarta: Graha Ilmu.

[2] Suyanto. 2008. *Soft Computing: Membangun Mesin Ber-IQ Tinggi*. Bandung: Informatika.

- [3] Wilson, Tom, 2012."Forecast Accuracy and Uncertainty of Australian Bureau of Statistics State and Territory Population Projections". International Journal of Population Research.
- [4] E.W. Weisstein, "Moving Average," Wolfram MathWorld, [Online]. Available: <http://mathworld.wolfram.com/MovingAverage.html> [Diakses 18 Desember 2014].
- [5] Rismala Rita, Liong The Houw, Ardiyanti Arie (2013). "Prediction of Malaria Incidence in Banggai Regency Using Evolving Neural Network", international Conference on Technology, Informatics, Management, Engineering & Environment (TIME-E 2013).
- [6] Jang, Jyh-Shing Roger; 1993; *ANFIS: Adaptive-network based fuzzy inference system*; Departement of Electrical Engineering and Computer Science, University of California.