

PREDIKSI CURAH HUJAN MENGGUNAKAN ALGORITMA HYBRID NEURAL NETWORK DAN EVOLUTIONARY PROGRAMMING

RAINFALL PREDICTION USING HYBRID NEURAL NETWORK AND EVOLUTIONARY PROGRAMMING ALGORITHM

Irwindi Famesa¹, Fhira Nhita², Adiwijaya³

^{1,2,3}Prodi SI Ilmu Komputasi, Fakultas Informatika, Universitas Telkom

¹mesanda@gmail.com, ²fhiranhita@telkomuniversity.ac.id, ³adiwijaya@telkomuniversity.ac.id

Abstrak

Pertumbuhan dan kualitas hasil pertanian tergantung pada beberapa faktor, salah satunya adalah faktor lingkungan. Tanah merupakan salah satu faktor lingkungan yang berkaitan erat dengan curah hujan karena air sebagai pengangkut unsur hara dari tanah ke akar dan kemudian dilanjutkan pada proses fotosintesis. Hal inilah alasan mengapa prediksi curah hujan patut diketahui.

Pada tugas akhir ini, diterapkan metode peramalan *Simple Moving Average* (SMA) dengan 3-MA dan 5-MA. Untuk membangun sistemnya digunakan *hybrid Artificial Neural Network* (ANN) serta algoritma *Evolutionary Programming* (EP).

Untuk mendapatkan arsitektur dan bobot ANN optimal dibangkitkan 20.000 solusi individu dengan kombinasi ukuran populasi 50 dan generasi 400 dengan akurasi rata-rata pelatihan dan pengujian sebesar 79,24% pada arsitektur 3-1-1 metode 3-MA. Sedangkan untuk 5-MA menghasilkan akurasi rata-rata sebesar 78,45% pada arsitektur 5-2-1.

Kata Kunci : *time series, moving average, ANN, Evolutionary Algorithm, Evolutionary Programming,*

Abstract

Growth and quality of crops depends on several factors, one of them is a environmental factor. Soil is one of the environmental factors closely related to rainfall because the water as a carrier of nutrients from the soil to the roots and then continued in the process of photosynthesis. This is the reason why the prediction of rainfall are noteworthy.

For this final project, applied methods of forecasting of Simple Moving Average (SMA) with 3-MA and 5-MA. To build a hybrid system used Artificial Neural Network (ANN) and algorithms of Evolutionary Programming (EP).

To get optimal ANN architecture and weights are raised 20.000 individual solutions with a combination of 50 population size and 400 generation with an average accuracy of training and testing of 79.24% on the architecture of 3-1-1 3-MA method. As for the 5-MA resulted in an average accuracy of 78.45% in the 5-2-1 architecture.

Keyword : *time series, moving average, ANN, Evolutionary Algorithm, Evolutionary Programming*

1. Pendahuluan

1.1. Latar Belakang

Pertumbuhan dan kualitas hasil pertanian tergantung pada beberapa faktor, salah satunya adalah faktor lingkungan. Tanah merupakan salah satu faktor lingkungan yang berkaitan erat dengan curah hujan karena air sebagai pengangkut unsur hara dari tanah ke akar dan kemudian dilanjutkan pada proses fotosintesis. Hal inilah alasan mengapa prediksi curah hujan patut diketahui.

Peralaman *time series* merupakan salah satu metode statistika yang bertujuan memprediksi suatu kejadian atau keadaan masa depan

berdasarkan data historis sehingga untuk tugas akhir ini, diterapkan metode peramalan *Simple Moving Average* (*Simple MA*) dan untuk membangun sistemnya, digunakan sistem *Evolutionary Artificial Neural Network* (EANN) yang merupakan penggabungan atau *hybrid Artificial Neural Network* (ANN) dan algoritma *Evolutionary Algorithm* (EAs).

Untuk membangun suatu sistem, dibutuhkan proses pembelajaran yang bertujuan mendapatkan kandidat penyelesaian masalah. Dalam hal ini, kemampuan belajar ANN dan EAs

sangat berbeda. ANN sangat baik dalam masalah klasifikasi atau pola permasalahan namun tidak baik untuk permasalahan optimasi. Sebaliknya, EAs sangat baik untuk permasalahan optimasi namun tidak baik untuk permasalahan klasifikasi.

Menurut jurnal yang berjudul “*Neural Network Design Based on Evolutionary Programming*” menyatakan bahwa EP lebih baik daripada algoritma genetika untuk mengembangkan ANN karena algoritma genetika biasanya bergantung pada dua ruang representasional yang berbeda, yaitu rekombinasi dan ruang evaluasi [1]. Hal ini yang menjadikan kelebihan EP untuk mengevolusi ANN. Sebelumnya, pernah dilakukan tugas akhir prediksi curah hujan yang berjudul “*A Rainfall Forecasting using Fuzzy System Based on Genetic Algorithm*” yang menunjukkan bahwa penggabungan algoritma genetika dan fuzzy dapat memodelkan prediksi curah hujan menggunakan data cuaca Kemayoran dengan akurasi lebih dari 75% [5].

Pada tugas akhir ini algoritma *Hybrid Neural Network* dengan *Evolutionary Programming* digunakan untuk prediksi curah hujan wilayah Soreang Kabupaten Bandung. *Hybrid Neural Network* dengan *Evolutionary Programming* akan membentuk model prediksi curah hujan untuk satu bulan kedepan (M+1) dengan M: 1, 2, 3....n sesuai penerapan periode metode *Simple Moving Average* (Simple MA).

1.2. Perumusan Masalah

Permasalahan yang dijadikan dalam tugas akhir tugas akhir ini terdiri dari :

1. Bagaimana mengimplementasikan algoritma *Hybrid Neural Network* dengan *Evolutionary Programming* dalam memprediksi curah hujan wilayah Soreang Kabupaten Bandung?
2. Bagaimana menentukan bobot yang optimal untuk algoritma *Evolutionary Programming* ?
3. Bagaimana bentuk *Neural Network* terbaik untuk memprediksi curah hujan di wilayah Soreang Kabupaten Bandung?
4. Bagaimana performansi algoritma *Hybrid Neural Network* dengan

Evolutionary Programming dalam memprediksi curah hujan di wilayah Soreang Kabupaten Bandung?

1.3. Tujuan

1. Mengimplementasikan *Hybrid Neural Network* dengan *Evolutionary Programming* sehingga dapat memprediksi curah hujan di wilayah Soreang Kabupaten Bandung.
2. Mengetahui proses optimasi arsitektur dan bobot menggunakan algoritma *Evolutionary Programming* sehingga mendapatkan performansi diatas 75%.
3. Menganalisis hasil performansi yang didapatkan dari algoritma *Hybrid Neural Network* dengan *Evolutionary Programming*.

2. Landasan Teori

2.1. Simple Moving Average

Simple Moving Average (Simple MA) merupakan salah satu metode dalam peramalan yang berfungsi *smoothing* data untuk mengestimasi siklus *trend* (kecenderungan) sehingga memberikan perkiraan yang wajar dan menghilangkan *randomness* [7]. Ciri khas dari *Simple MA* adalah mengestimasi siklus *trend* dengan ketentuan order ganjil. Hal ini dikarenakan prosedur metode ini memerlukan data observasi sebelumnya dan observasi selanjutnya untuk menghasilkan peramalan periode kedepannya.

Perlu diperhatikan dalam menentukan orde, karena jika suatu data terlalu *smooth* atau sangat berbeda dengan data aktual maka data cenderung kehilangan informasi.

Jumlah titik termasuk pada *Simple MA* mempengaruhi *smoothness* hasil dari estimasi. Pada tugas akhir kali ini, diterapkan observasi dengan pola periode 3 bulan dan 5 bulan. Berikut rumusnya:

$$\hat{y}_t = \frac{1}{3} (y_{t-1} + y_t + y_{t+1}) \quad (1)$$

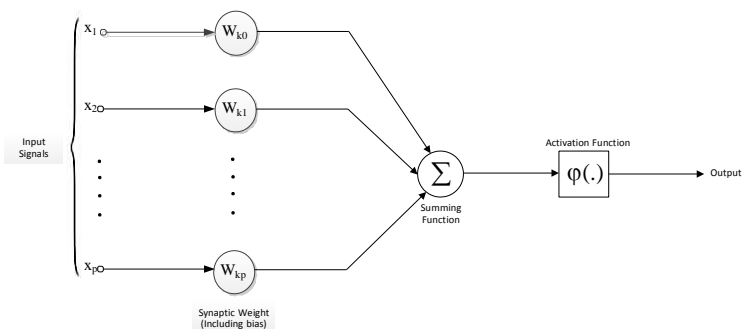
Untuk dua titik pada MA *smoother* maka :

$$\hat{y}_t = \frac{1}{5} (y_{t-2} + y_{t-1} + y_t + y_{t+1} + y_{t+2}) \quad (2)$$

2.2. Artificial Neural Network (ANN)

ANN atau dalam Bahasa Indonesia disebut dengan Jaringan Syaraf Tiruan (JST) merupakan suatu arsitektur jaringan untuk memodelkan cara kerja sistem syaraf manusia. ANN terdiri dari lapisan-lapisan perceptron yaitu terdiri dari *input function*, *activation function*, dan *output*.

Setiap serangkaian elemen *node* akan melakukan operasi matematika yang berfungsi menghasilkan suatu keluaran. Berikut model matematis non-linier dari suatu neuron [8] :



Gambar 1 Model Matematis suatu neuron

Berdasarkan gambar diatas, adapun neuron dinotasikan sebagai k dikalikan dengan bobot yang dinotasikan dengan w. Untuk proses berikutnya adalah penjumlahan dari perhitungan *input* signal dikali bobot sinapsis sehingga menghasilkan persamaan :

$$u_k = \sum_{j=1}^p w_{kj} x_j + b \tag{3}$$

Dimana :

- ⊕ = keluaran dari *linear combiner*
- ⊗, ⊗, ... ⊗ = bobot-bobot neuron k
- ⊗, ⊗, ... ⊗ = sinyal *input*
- b = bias

Fungsi aktivasi bertujuan untuk menterjemahkan hasil keluaran dari *linear combiner*. Fungsi aktivasi sigmoid bersifat non-linier dan baik digunakan untuk menyelesaikan permasalahan dunia nyata yang kompleks [9]. Pada tugas akhir ini, digunakan fungsi sigmoid unipolar [10]. Berikut rumusnya :

$$f(x) = \frac{1}{1 + e^{-x}} \tag{4}$$

Dimana $x = u_k$, sehingga

$$f(x) = \frac{1}{1 + e^{-w_{kj} x_j + b}} \tag{5}$$

Dibutuhkan algoritma pelatihan untuk melatih MLP, salah satu algoritma yang akan diaplikasikan pada tugas akhir ini adalah algoritma umpan maju (*feedforward*). Algoritma ini bertujuan untuk mendapatkan sejumlah *error* sehingga dapat diketahui seberapa baik arsitektur tersebut. Berikut algoritma *feedforward*:

1. Sinyal *input* (x_i) dipropagasi ke *hidden layer*. Sinyal *input* yang terboboti masuk pada tiap-tiap *hidden layer*, ditentukan dengan persamaan (3). Setelah mendapatkan nilai bobot, ditentukan fungsi aktivasi seperti pada persamaan (5). Jika hasil nilainya berada dibawah ambang batas, maka nilai akan diteruskan dengan nilai bias. Untuk langkah selanjutnya yaitu dari *hidden layer* ke sinyal *output* akan sama caranya dengan langkah ini.
2. Hasil *output* dibandingkan dengan target yang harus dicapai. Selisih dari target dan hasil prediksi adalah kesalahan (*error*) yang terjadi.

2.3. Evolutionary Programming (EP)

Untuk terdapat enam hal yang harus diperhatikan [11] :

1. Representasi Individu

Individu pada EP direpresentasikan kedalam vektor bilangan real. Kromosom yang lengkap terdiri dari dua bagian, yaitu :

- Variabel Objek : ⊕, ... ⊕, ⊕
- Ukuran langkah mutasi : ⊗, ... ⊗, ⊗

2. Mutasi

Merupakan satu-satunya proses untuk menghasilkan kromosom baru dengan cara mengubah nilai gen dengan menambahkan bilangan *random* yang dibangkitkan berdasarkan distribusi normal. Berikut rumus mutasi untuk menghasilkan kromosom baru [3] [12] [13]:

$$\sigma'_i = \sigma_i * \exp(\tau'N(0,1) + \tau N(0,1)) \tag{6}$$

$$\oplus = \oplus + \otimes * 0.1 \tag{7}$$

Dimana :

- ⊗ : Hasil mutasi gen *mutation step size*
- x'_i : Hasil gen bobot
- x : Gen bobot

parameter τ' dan τ biasanya diset $(\sqrt{2} \otimes)^{-1}$

dan $(\sqrt{2} \otimes)$ dengan n adalah sejumlah variabel objek [13].

4. Seleksi Survivor

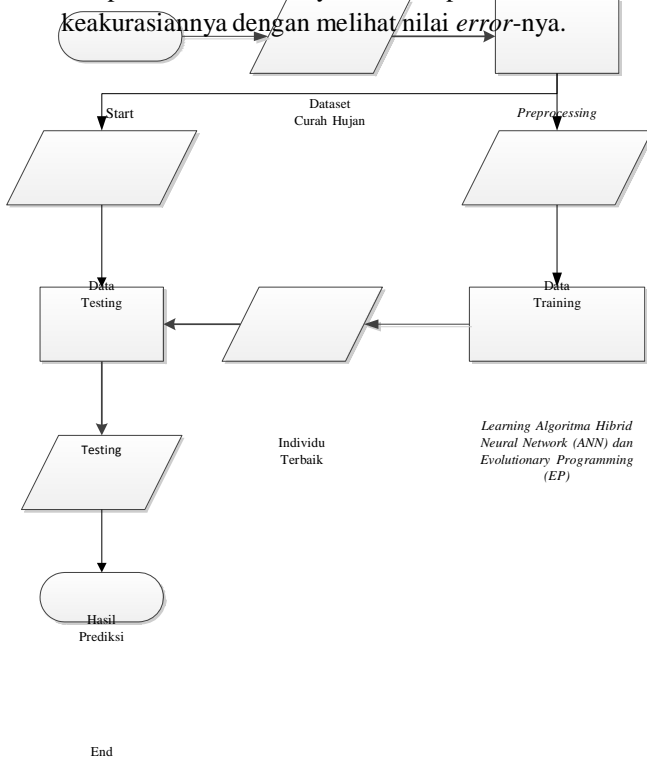
Merupakan penggabungan populasi dari anak dan orang tua untuk di seleksi sebagai populasi orang tua pada generasi berikutnya. Teknik seleksi yang digunakan pada tugas akhir ini adalah *pairwise competition* [14]. Berikut proses dari *pairwise competition*:

- Setiap kromosom orang tua dan anak dibandingkan nilai *fitness* dengan sejumlah p (biasanya di set-10) [11] .
- Lawan dipilih secara acak dari seluruh kromosom populasi gabungan orang tua dan anak.
- Untuk setiap kromosom yang memiliki nilai *fitness* lebih besar dari nilai *fitness* lawan maka kromosom tersebut akan ditandai sebagai pemenang lalu diurutkan

secara ranking dan diambil terbaik dengan sejumlah p.

3. Perancangan Sistem

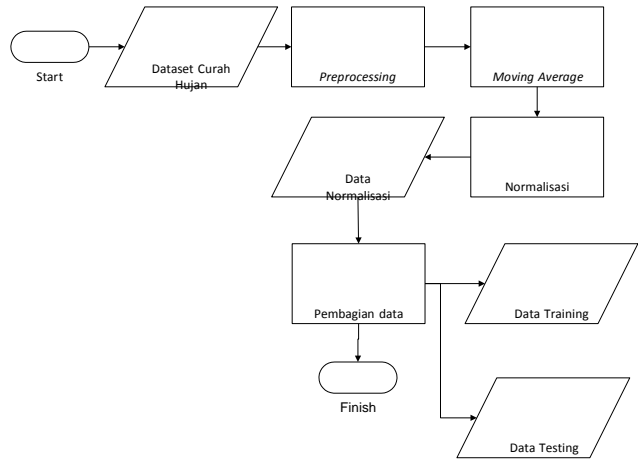
Pada tugas akhir ini, akan dibangun sistem prediksi curah hujan bulan selanjutnya (M+1) dengan menerapkan algoritma *Hybrid Neural Network* (ANN) dan *Evolutionary Programming* (EP). Keluaran yang diharapkan berupa individu terbaik serta arsitektur ANN yang optimal. Setelah didapatkan arsitekturnya, hasil prediksi diukur keakurasiannya dengan melihat nilai *error*-nya.



Gambar 2 Perancangan Sistem

3.1. Preprocessing

Pada tahap ini data akan dipersiapkan sebelum diproses ke sistem. Adapun beberapa tahap yang dilakukan pada data curah hujan kota Soreang adalah :



Gambar 3 Flowchart Preprocessing

1. Moving Average

Dilakukan *Simple MA* dengan 3-MA dan 5-MA. Tujuan dilakukan *Simple MA* selain *smoothing* data adalah menghilangkan *missing value*.

2. Normalisasi

Merupakan langkah mengubah data aktual menjadi nilai *range* dengan interval [0..1]. Berikut rumusnya [17]:

$$X_i = \frac{X_i - \min X}{\max X - \min X} \quad (8)$$

Dimana :

- X_n : Data yang akan dinormalisasikan ke-n
- X_i : Data aktual ke-i
- $\min X$: Nilai minimum dari variabel X
- $\max X$: Nilai maksimum dari variabel X

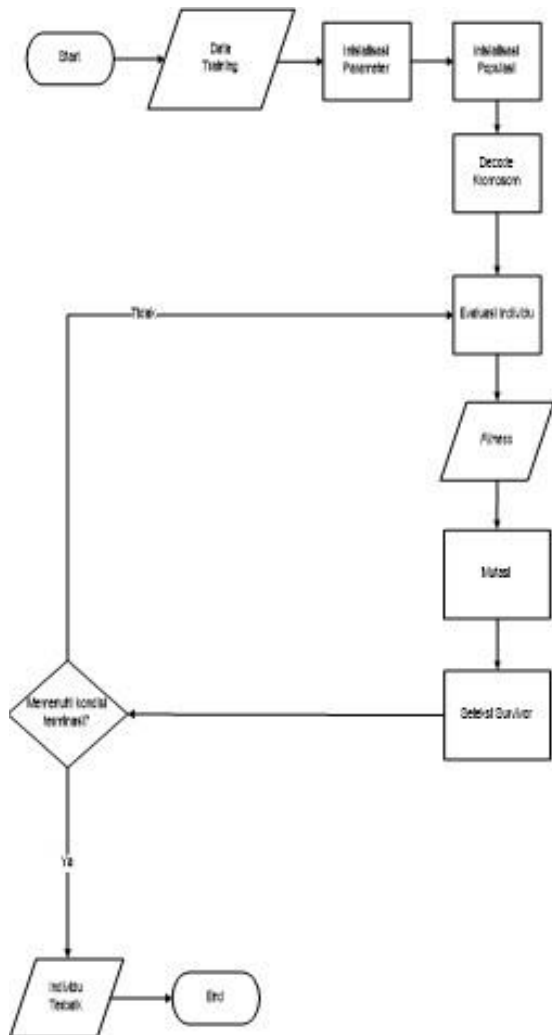
3. Pembagian Data

Data akan dibagi menjadi dua bagian yaitu data *training* dan *testing*.

3.2. Learning Algoritma Hibrid Neural Network (ANN) dan Evolutionary Programming (EP)

Pada proses ini dilakukan tahap *training* yaitu tahap dalam membangun struktur ANN dengan mengadaptasikan algoritma EP dalam menghasilkan individu terbaik kemudian dilakukan juga optimasi struktur ANN sehingga didapatkan arsitektur optimal pada permasalahan prediksi curah

hujan. Adapun tahap yang dilakukan sebagai berikut :



Gambar 4 Flowchart learning hybrid ANN dan EP

1. Inisialisasi Parameter

Pada tahap ini dilakukan set awal parameter-parameter yang digunakan saat proses learning. Berikut rinciannya :

- *Evolutionary Programming (EP)*
 - Kombinasi ukuran populasi dan jumlah generasi
 - Operator mutasi : bobot dan *mutation step size*
- *Feedforward Neural Network*
 Tiap-tiap *layer* berjumlah satu dan menggunakan fungsi aktivasi sigmoid. Berikut jumlah nodenya :
 - *Input node* = 3 dan 5
 - *Hidden node*= 1-12
 - *Output node*=1

2. Inisialisasi Populasi

Suatu kromosom memuat gen-gen bobot, bias dan *mutation step size*. Panjang dari gen *mutation step size* akan selalu sama dengan panjang gen bobot dan bias. Isi awal dari gen-gen ini adalah bilangan *random*. Digunakan nilai *random* distribusi Uniform pada *range* awal bobot, bias dan *mutation step size* untuk mengisi satu kromosom.

Berikut rumus untuk membentuk suatu kromosom serta ilustrasi perincian perhitungan kromosom:

$$\begin{matrix} \text{0.00000} & \text{0.00000000} & = & 2((\text{0.0} + \text{0.0} + (\text{0} + \text{0})) & (9) \end{matrix}$$

Dimana :

- X : *Input node*
- Z : *Hidden node*
- Y : *Output node*

3. Decode Kromosom

Pada tahap ini, tiap kromosom akan di *generate* sebagai gen bobot dan bias untuk struktur ANN.

4. Evaluasi Individu

Merupakan proses mengevaluasi individu yang mengandung bobot-bobot, bias dan *mutation step size* menggunakan algoritma *feedforward*. Hasil dari algoritma ini adalah nilai fitness yang mencerminkan error pada individu tersebut.

5. Mutasi

Merupakan proses penting karena tahap ini adalah menghasilkan solusi baru atau dikatakan sebagai kromosom anak. Proses ini didapat berdasarkan dari mutasi kromosom orang tua. Berikut ilustrasi proses mutasi.

Kromosom :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----

Langkah 1
 Kromosom dengan panjang 22 gen diisi dengan nilai random sehingga menjadi :

0.2	0.3	-0.8	0.5	0.6	0.7	0.4	-0.5	0.1	0.2	-0.3	0.01	0.03	0.05	0.04	0.08	0.07	0.06	0.02	0.09	0.01	0.042
-----	-----	------	-----	-----	-----	-----	------	-----	-----	------	------	------	------	------	------	------	------	------	------	------	-------

Bila dilakukan mutasi pada gen 1, maka gen 1 memiliki gen *mutation step size* 1 :

0.2	0.01
-----	------

Langkah 2
 Lakukan mutasi *mutation step size* dengan rumus:

$$\sigma'_i = \sigma_i * \exp(\tau * N(0,1) + \tau * N(0,1))$$

Dengan variabel objek= 11, maka untuk operator $\tau=0,2$ dan $\tau=0,4$

Dilakukan juga *me-random* sejumlah nilai dengan bantuan matlab (randn), sehingga didapatkan nilai random distribusi normal dengan masing-masing: 0.5377 dan -0.8947, maka selanjutnya dimasukkan ke rumus

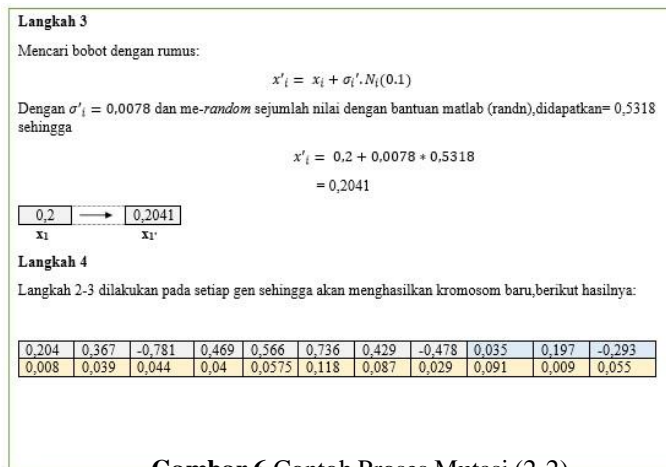
$$\sigma'_i = 0,01 * \exp(0,2 * 0,5377 + 0,4 * -0,8947)$$

$$\sigma'_i = 0,0078$$

Sehingga didapatkan untuk *mutation step* aksen adalah 0.0078

0.01	→	0.0078
σ_1		σ'_1

Gambar 5 Contoh Proses Mutasi (1-2)



Gambar 6 Contoh Proses Mutasi (2-2)

6. Seleksi Survivor

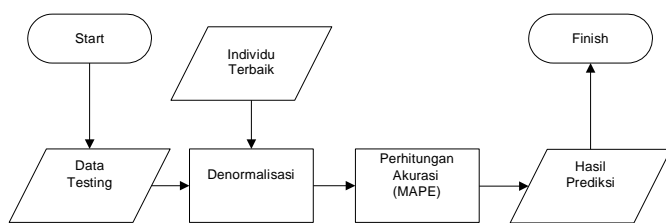
Digunakan metode *pairwise competition*.

7. Kondisi Terminasi

Merupakan suatu kondisi untuk menentukan proses berhenti dilakukan atau tidak. Kondisi pada sistem ini adalah dua kondisi yaitu target MSE tercapai atau mencapai maksimal generasi. Jika belum memenuhi maksimal generasi, maka sistem akan kembali mengevaluasi individu dan melakukan algoritma EP. Jika ya maka sistem diberhentikan karena sudah memenuhi maksimal generasi.

3.3. Testing

Dalam tahap ini dilakukan pengujian hasil dari proses training. Berikut adalah *flowchart* dari proses testing:



Gambar 7 Flowchart Testing

Pada *testing* terdapat proses-proses yang dilakukan. Adapun berikut proses tersebut:

1. Denormalisasi

Merupakan langkah mengembalikan data yang sudah dinormalisasikan menjadi data aktual untuk menghitung nilai MAPE. Hal ini dilakukan untuk mengetahui rata-rata *error* yang dihasilkan dari output perhitungan maju. Berikut rumus untuk denormalisasi:

$$x_i * (\frac{x_{max} - x_{min}}{x'_{max} - x'_{min}}) + x_{min} \quad (10)$$

Dimana :

- x_i = Data normalisasi ke-n
- x'_i = Data aktual ke-n
- x_{max} = Data maksimum variabel X
- x_{min} = Data minimum variabel X

2. Mean Absolute Percentage Error (MAPE)

Merupakan salah satu pengukur kesalahan dalam kasus peramalan. Berikut rumusnya :

$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|PE|}{x_i} \quad (11)$$

Dimana :

P= Nilai aktual

PE= Error

n = Jumlah data

4. Hasil dan Analisis

4.1. Analisis Hasil Pengujian

Berdasarkan uji skenario pengujian, yaitu pencarian jumlah *hidden node* optimal serta pencarian batas nilai random bobot dan *mutation step size* dilakukan dengan arsitektur *input* 3 dan 5 sehingga total jumlah arsitektur adalah 24 arsitektur.

Pada setiap arsitektur, operator mutasi dengan beberapa percobaan *range* nya (misal untuk satu arsitektur hanya dicoba dengan satu spesifik *range* bobot awal maupun *mutation step size*) tidaklah selalu memberikan akurasi yang baik, sehingga dilakukan semua kombinasi operator pada setiap arsitektur. Berikut contoh percobaan pertama dengan arsitektur 3-1-1.

Range bobot awal: [-0.05, 0.05] Mutation step size: [0.0001, 0.001] MSE: 0.0297 MAPE Training : 37.75% MAPE Testing : 37.53% MAPE rata-rata:37.64%	Range bobot awal: [-0.5, 0.5] Mutation step size: [0.0001, 0.001] MSE : 0.0400 MAPE Training :60.63% MAPE Testing :29.32% MAPE rata-rata:44.973%	Range bobot awal: [-1.5, 1.5] Mutation step size: [0.0001, 0.001] MSE: 0.0414 MAPE Training: 46.35% MAPE Testing: 36.55% MAPE rata-rata:41.45%
Range bobot awal: [-0.05, 0.05] Mutation step size: [0,001,0,01] MSE: 0.0281 MAPE Training :39.31% MAPE Testing :32.46% MAPE rata-rata:35.885%	Range bobot awal: [-0.5, 0.5] Mutation step size: [0,001,0,01] MSE: 0.0295 MAPE Training :45.98% MAPE Testing :26.62% MAPE rata-rata:36.3%	Range bobot awal: [-1.5, 1.5] Mutation step size: [0,001,0,01] MSE: 0.0389 MAPE Training: 58.45% MAPE Testing: 38.60% MAPE rata-rata:48.525%
Range bobot awal: [-0.05, 0.05] Mutation step size: [0,01,0,1] MSE=0.0290 MAPE Training :42.26% MAPE Testing :28.99% MAPE rata-rata:35.625%	Range bobot awal: [-0.5, 0.5] Mutation step size: [0,01,0,1] MSE: 0.0145 MAPE Training : 25.24% MAPE Testing : 17.22% MAPE rata-rata:21.23%	Range bobot awal: [-1.5, 1.5] Mutation step size: [0,01,0,1] MSE: 0.0157 MAPE Training: 30.91% MAPE Testing: 15.95% MAPE rata-rata:23.43%
Range bobot awal: [-0.05, 0.05] Mutation step size: [0.05, 0.1] MSE :0.0298 MAPE Training :51.23% MAPE Testing :30.61% MAPE rata-rata:40.92%	Range bobot awal: [-0.5, 0.5] Mutation step size: [0.05, 0.1] MSE : 0.0290 MAPE Training : 42.26% MAPE Testing : 28.99% MAPE rata-rata:35.625%	Range bobot awal: [-1.5, 1.5] Mutation step size: [0.05, 0.1] 3-1-1= 0.0143 MAPE Training: 25.31% MAPE Testing:16.27% MAPE rata-rata: 20.79%

Gambar 8 Hasil uji skenario pada arsitektur 3-1-1

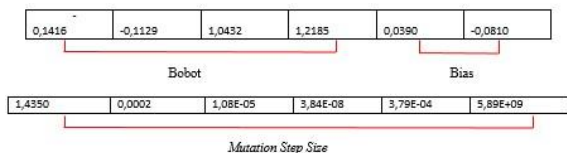
Berdasarkan gambar diatas, terlihat pada arsitektur 3-1-1 memiliki akurasi yang berbeda-beda pada setiap nilai range bobot dan *range mutation step size*. Sehingga dapat dibuktikan

bahwa untuk mendapatkan akurasi yang baik diperlukan observasi pada setiap batas *range* bobot dan *mutation step size*. Berikut ditampilkan hasil terbaik dengan jumlah total 24 arsitektur berdasarkan akurasi rata-rata :

Tabel 1 Hasil kumulatif akurasi seluruh arsitektur

No Observasi	Arsitektur	Operator Mutasi		MSE	MAPE Training	MAPE Testing	MAPE rata-rata	Akurasi Rata-rata
		Range bobot awal	Mutation Step Size					
1	3-1-1	[-1.5;1.5]	[0.05; 0.1]	0.0143	25.31%	16.27%	20.79%	79,21%
2	3-2-1	[-0.05;0.05]	[0.05; 0.1]	0.0113	24.58%	20%	22.29%	77,71%
3	3-3-1	[-1.5;1.5]	[0.01; 0.1]	0.0098	24.93%	19.43%	22.18%	77,82%
4	3-4-1	[-0.5;0.5]	[0.001; 0.01]	0.0096	24.06%	20.21%	22,135%	77,87%
5	3-5-1	[-1.5;1.5]	[0.0001;0.001]	0.0105	27.23%	21.83%	24,34%	75,46%
6	3-6-1	[-1.5;1.5]	[0.001; 0.01]	0.0100	28.42%	18.94%	23,675%	76,33%
7	3-7-1	[-0.5;0.5]	[0.0001; 0.001]	0.0099	22.87%	22.08%	22,475%	77,53%
8	3-8-1	[-0.5;0.5]	[0.001; 0.01]	0.0098	28.53%	20.14%	24,335%	75,67%
9	3-9-1	[-0.5;0.5]	[0.001; 0.01]	0.0097	28.96%	23.37%	26.16%	73,84%
10	3-10-1	[-0.5;0.5]	[0.001;0.01]	0.0096	24.14%	20.93%	22,335%	77,47%
11	3-11-1	[-0.5;0.5]	[0.0001; 0.001]	0.0106	28.58%	20.94%	24.76%	75,24%
12	3-12-1	[-0.5;0.5]	[0.001; 0.01]	0.0119	25.97%	22.45%	24,21%	75,79%
13	5-1-1	[-1.5;1.5]	[0.0001; 0.001]	0.0390	26.98%	19.35%	23,165%	76,84%
14	5-2-1	[-0.5;0.5]	[0.0001;0.001]	0.0100	26.13%	16.98%	21,555%	78,45%
15	5-3-1	[-0.5;0.5]	[0.05; 0.1]	0.0097	26.56%	16.84%	21,9%	78,1%
16	5-4-1	[-0.5;0.5]	[0.05; 0.1]	0.0103	28.72%	17.49%	23,105%	76,9%
17	5-5-1	[-0.5;0.5]	[0.01; 0.1]	0.0108	31.08%	15.81%	23,445%	76,55%
18	5-6-1	[-0.05;0.05]	[0.001; 0.01]	0.0099	26.38%	17.89%	22,135%	77,87%
19	5-7-1	[-0.05;0.05]	[0.0001; 0.001]	0.0152	31.32%	22.74%	27,03%	72,97%
20	5-8-1	[-0.5;0.5]	[0.001; 0.01]	0.0128	31.09%	15.74%	24,415%	75,59%
21	5-9-1	[-0.5;0.5]	[0.001; 0.01]	0.0099	29.95%	17.61%	25,78%	76,22%
22	5-10-1	[-0.05;0.05]	[0.001;0.01]	0.0157	32.53%	15.95%	24,24%	75,76%
23	5-11-1	[-0.5;0.5]	[0.001; 0.01]	0.0139	28.69%	21.24%	24,965%	75,04%
24	5-12-1	[-0.05;0.05]	[0.001; 0.01]	0.0135	28.24%	18.55%	23,395%	76,61%

Berdasarkan hasil tabel diatas, didapatkan bahwa arsitektur 3-1-1 merupakan arsitektur terbaik menggunakan metode *Simple Moving Average* 3-MA, sehingga didapatkan hasil peramalan curah hujan serta individu terbaik.



Gambar 9 Individu terbaik arsitektur 3-1-1

Setelah mendapatkan arsitektur serta operator murasi terbaik, maka langkah selanjutnya adalah dibangkitkan solusi dengan beberapa kombinasi untuk mengetahui kombinasi pembangkit solusi terbaik. Berikut hasilnya :

Tabel 2 Hasil pengujian dari kombinasi pembangkit solusi

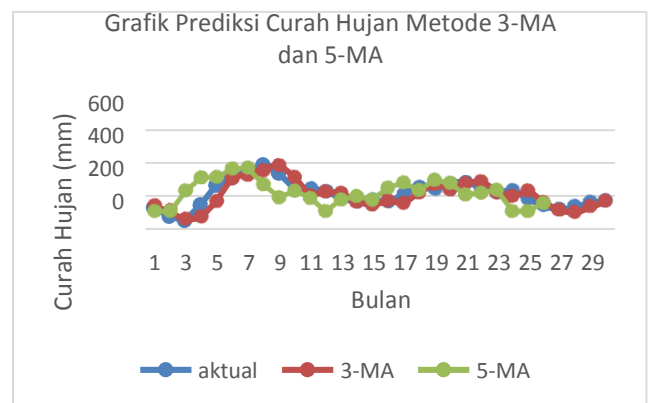
No Observasi	Pembangkit Solusi		Arsitektur	Operator Mutasi		MSE	MAPE Training	MAPE Testing	MAPE rata-rata	Akurasi Rata-rata
	Generasi	Populasi		Range bobot awal	Mutation Step Size					
1	200	100	3-1-1	[-1.5;1.5]	[0.05; 0.1]	0.0184	26.62%	20.09	23,325%	76,675%
2	100	200	3-1-1	[-1.5;1.5]	[0.05; 0.1]	0.0130	27.1%	17,84%	22,97%	77,03%
3	300	30	3-1-1	[-1.5;1.5]	[0.05; 0.1]	0.0143	25.31%	16.27%	20.79%	79,21%
4	200	200	3-2-1	[-1.5;1.5]	[0.0001;0.001]	0.0110	25.79%	25,23%	27,22%	72,78%
5	100	200	3-2-1	[-1.5;1.5]	[0.0001;0.001]	0.0189	48.03%	22.89%	34,38%	65,62%
6	300	30	3-2-1	[-1.5;1.5]	[0.0001;0.001]	0.0100	26.13%	16.98%	21,525%	78,45%

Berdasarkan hasil tabel diatas, maka dalam memprediksi curah hujan dengan metode *Simple Moving Average* dengan 3-MA dan 5-MA menggunakan *hybrid ANN* dan *Evolutionary Programming* didapatkan bahwa pada kedua metode tersebut memiliki perbedaan dari segi jumlah *hidden node* dan operator mutasi pada *range mutation step size*.

Untuk metode 3-MA, digunakan *hidden node* dengan jumlah 1 saja sedangkan metode 5-MA dibutuhkan 2 *hidden node*. Kemudian, pada *range mutation step size* metode 5-MA, menggunakan *range* yang lebih kecil daripada metode 3-MA yaitu [0.0001, 0.001].

4.2. Analisis dengan 3-MA dan 5-MA

Setelah dilakukan uji skenario, diperoleh bahwa metode 3-MA memberikan akurasi lebih besar daripada 5-MA. Hal ini membuktikan bahwa tidak selamanya semakin besar periode dari suatu *Simple Moving Average* lebih baik. Jika data terlalu *smooth* maka cenderung menghilangkan informasi. Berikut hasil grafik prediksi 3-MA dan 5-MA.



Gambar 10 Hasil prediksi 3-MA dan 5-MA

5. Kesimpulan dan Saran

5.1. Kesimpulan

1. *Hybrid Algoritma Artificial Neural Network* pada algoritma *Evolutionary Programming* menghasilkan arsitektur terbaik 3-1-1 pada prediksi curah hujan menggunakan metode *Simple Moving Average* 3-MA serta arsitektur 5-2-1 pada *Simple Moving Average* 5-MA.
2. Hasil prediksi 3-MA dengan arsitektur 3-1-1 menghasilkan akurasi sebesar 79,24% dan 5-MA dengan arsitektur 5-2-1 menghasilkan akurasi sebesar 78,45%.
3. Untuk mencari arsitektur terbaik harus dicoba dengan berbagai kombinasi pada operator mutasi seperti *range*

bobot dan *range mutation step size* karena hal tersebut berpengaruh besar pada akurasi suatu arsitektur.

4. Semakin batas *range* pada operator semakin besar maka waktu *compile* program semakin lama.
5. Metode 3-MA memberikan akurasi lebih tinggi daripada 5-MA. Hal ini membuktikan tidak selamanya semakin tinggi periode pada *Simple Moving Average* lebih baik karena, jika data terlalu *smooth* maka cenderung menghilangkan informasi.
6. Hasil prediksi yang dioptimasi memberikan performansi diatas 75%.

5.2. Saran

1. Pada penelitian selanjutnya untuk prediksi curah hujan dapat menggunakan algoritma EAs yang lain selain *Evolutionary Programming* sebagai perbandingan dari hasil performansi *Evolutionary Programming*.
2. Mencoba metode lain pada proses mutasi seperti Cauchy atau Levy.

6. Daftar Pustaka

- [1] J. Fang dan Y. Yi, "Neural Network Design Based on Evolutionary Programming," dalam *Artificial Intelligence in Engineering 11*, Elsevier Science, 1997, pp. 155-161.
- [2] J. A. Portilla-Figueras, S. Salcedo-Sanz, A. Oropesa-García dan C. Bousoño-Calzón, "Cell size determination in WCDMA systems using an evolutionary," *Computers & Operations Research*, vol. 35, pp. 3758-3768, 2008.
- [3] A. Hoofar, "Evolutionary Programming in Electromagnetic Optimization: A Review," *IEEE Transactions On Antennas And Propagation*, vol. 55, pp. 523-537, Maret 2007.
- [4] V. Jamnejad dan A. Hoorfar, "Design of Corrugated Horn Antennas by Evolutionary Optimization Techniques," *IEEE Antennas And Wireless Propagation Letter*, vol. 3, pp. 276-279, 2004.
- [5] F. Nhita, Adiwijaya dan U. Wisesty, dalam *Forecasting of Weather Data in Indonesia Using Evolving Neural Network on Genetic Algorithm*, The Second International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE), 2014, pp. 78-82.
- [6] R. S. Crowder, "Predicting the Mackey–Glass time series with cascaded correlation learning," dalam *Proc. Connectionist Models Summer School*, 1990, p. 117–123.
- [7] S. Makridakis, S. C. Wheelright dan R. J. Hyndman, *Forecasting*.
- [8] S. Haykin, "Neural Network," dalam *A Comprehensive Foundation*, New York, Macmillan Publishing Company, 1994, pp. 8;138-149.
- [9] Suyanto, "Membangun Mesin Ber-IQ Tinggi," dalam *Soft Computing*, Bandung, Informatika, 2008, pp. 65-70.
- [10] Z. Tang dan P. A. Fishwick, "Feed Forward Neural Nets as Models for Time Series Forecasting," *Computer and Information Sciences, University of Florida*, 1993.
- [11] Suyanto, *Evolutionary Computation, Komputasi Berbasis "Evolusi" dan "Genetika"*, Bandung: INFORMATIKA, 2008.
- [12] X. Yao, Y. Liu dan G. Lin, "Evolutionary Programming Made Faster," *IEEE Transaction on Evolutionary*, July 1999.
- [13] N. Sinha, R. Chakrabarti dan P. K. Chattopadhyay, "Evolutionary Programming Techniques for Economic Load Dispatch," *IEEE Transactions On Evolutionary Computation*, vol. 7, pp. 83-94, Februari 2003.
- [14] X. Yao, "Evolving Artificial Neural Networks," *Proceedings of the IEEE*, vol. 87, pp. 1423-1447, 1999.
- [15] X. Yao dan L. Yong, "A New Evolutionary System for Evolving Artificial Neural Networks," *IEEE Transaction Neural Network*, vol. 8, pp. 694-713, May 1997.
- [16] D. Puspitaningrum, "Pengantar Jaringan Saraf Tiruan," Yogyakarta, Andi, 2006.
- [17] D. Y., *The Oxford Dictionary of Statistical Terms*, Great Britain: Oxford University Press, 2003.