

PREDIKSI CURAH HUJAN MENGGUNAKAN ALGORITMA GRAMMATICAL EVOLUTION

¹Sheila Annisa, ²Fhira Nhita, ³Adiwijaya

Ilmu Komputasi Fakultas Informatika Universitas Telkom, Bandung

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

ABSTRACT

Prediction is a method which can estimating future events. Weather information, especially rainfall information, gives lots of advantages in several sector such as fisheries, agriculture, transportation, etc. A significant rainfall fluctuation can caused it's difficult to predict, coupled with the effects of global warming phenomenon, changes in monsoon-season drought shifted continuously.

This research utilized one of Evolutionary Algorithm (EAs) which is Grammatical Evolution which based on biological evolution is used to predict rainfall next month (M+1) for Soreang region, Bandung regency. One of its advantage is its' representation of chromosomes in the form of Backus Naur Form (BNF) which can be define according to the characteristic of problems. This research also analyzed comparison performance between Generational Replacement and Steady State in selection survivor.

Result of this research proved that Grammatical Evolution can predict rainfall M+1 with best performance 71% for Generational Replacement selection and $\pm 75\%$ for Steady State selection. This observation is performed by evaluating as many as 10000 individuals.

Keyword: prediction, rainfall, Grammatical Evolution, BNF, Generational Replacement, Steady State.

ABSTRAK

Prediksi dapat memberikan gambaran mengenai kejadian di masa depan. Prediksi cuaca khususnya curah hujan dapat bermanfaat dalam kelangsungan beberapa sektor seperti perikanan, pertanian, transportasi dan lain-lain. Fluktuasi curah hujan yang cukup signifikan menyebabkan hal tersebut sulit diprediksi, ditambah lagi dengan akibat pemanasan global, perubahan musim hujan-musim kemarau pun turut bergeser terus menerus.

Pada tugas akhir ini digunakan salah satu algoritma *Evolutionary Algorithms* yaitu *Grammatical Evolution* yang berbasis evolusi biologi untuk memprediksi curah hujan satu bulan kedepan (M+1) untuk wilayah Soreang, Kabupaten Bandung. Keunggulan dari *Grammatical Evolution* ialah representasi individu yang berupa *grammar Backus Naur Form* (BNF) yang dapat didefinisikan sesuai dengan karakteristik permasalahan yang dihadapi. Pada tugas akhir ini juga dianalisis perbandingan performansi *Grammatical Evolution* dengan metode seleksi survivor *Generational Replacement* dan *Steady State*.

Algoritma *Grammatical Evolution* menghasilkan prediksi curah hujan M+1 dengan nilai performansi terbaik sebesar $\pm 71\%$ untuk seleksi survivor dengan *Generational Replacement* dan $\pm 75\%$ untuk seleksi survivor dengan *Steady State*. Pengujian ini dilakukan dengan mengevaluasi sebanyak 10000 individu.

Kata kunci: prediksi, curah hujan, *Grammatical Evolution*, BNF, *Generational Replacement*, *Steady State*.

1. PENDAHULUAN

1.1 Latar Belakang

Indonesia dilalui oleh garis khatulistiwa. Hal ini yang menyebabkan Indonesia masuk ke dalam wilayah beriklim tropis. Dewasa ini perubahan musim hujan semakin tidak menentu dan sulit untuk diprediksikan. Prediksi cuaca khususnya curah hujan yang cepat, lengkap dan akurat kini menjadi tuntutan dari berbagai pihak yang membutuhkan informasi tersebut.

Prediksi dapat dilakukan dengan mengamati data-data historis dengan tujuan untuk mendapatkan pola data yang dapat digunakan untuk memprediksi kejadian masa depan. Namun, untuk menemukan pola data yang sesuai dibutuhkan suatu metode pencarian berdasarkan pada data historis.

Evolutionary Algorithms (EAs) merupakan algoritma – algoritma optimasi dan dapat digunakan untuk prediksi. EAs terinspirasi oleh evolusi biologis seperti reproduksi, mutasi, rekombinasi dan seleksi survivor. Pada tugas akhir ini digunakan salah satu algoritma EAs yaitu *Grammatical Evolution* (GE). Algoritma ini mempunyai tingkat adaptif yang baik sehingga dapat menghasilkan suatu model prediksi. *Grammatical Evolution* menggunakan representasi individu yang berupa *Backus Naur Form* (BNF) sehingga bisa digunakan untuk mengevolusi program yang berbeda bahasa.

Pada penelitian sebelumnya, pernah dilakukan prediksi cuaca dengan menggunakan Evolving Neural Network berdasarkan Genetic Algorithm [1]. Pada penelitian tersebut dihasilkan performansi prediksi untuk parameter curah hujan sebesar 61,18%. Selain itu, pernah dilakukan pula penelitian menggunakan *Grammatical Evolution* untuk memprediksi harga dinar di Indonesia [6]. Pada penelitian tersebut dihasilkan performansi terbaik sebesar 99,99% untuk prediksi jumlah pelanggan satu periode berikutnya.

Pada tugas akhir ini algoritma *Grammatical Evolution* digunakan untuk prediksi curah hujan wilayah Soreang. *Grammatical Evolution* akan membentuk model prediksi curah hujan yang optimal untuk satu bulan kedepan (M+1) dengan M: 1, 2, 3....n.

1.2 Perumusan Masalah

Permasalahan yang diangkat dalam penelitian tugas akhir ini terdiri dari:

- Bagaimana mengimplementasikan *Grammatical Evolution* dalam memprediksi curah hujan wilayah Soreang?

- Bagaimana perbandingan performansi *Grammatical Evolution* dengan menggunakan seleksi survivor *Generational Replacement* dan *Steady State* dalam memprediksi curah hujan di wilayah Soreang?

- Bagaimana kompleksitas waktu asimptotik (Big O) algoritma *Grammatical Evolution* yang dibangun?

1.3 Tujuan

Untuk menjawab beberapa permasalahan yang ada, maka tujuan dari penelitian tugas akhir ini adalah:

- Menganalisis dan mengimplementasikan algoritma *Grammatical Evolution* sehingga mampu memprediksi parameter curah hujan di wilayah Soreang
- Menganalisis perbandingan hasil performansi yang didapat dari algoritma *Grammatical Evolution* dengan menggunakan seleksi survivor *Generational Replacement* dan *Steady State*
- Menganalisis kompleksitas waktu asimptotik (Big O) dari algoritma *Grammatical Evolution*

2. LANDASAN TEORI

2.1 Pengertian *Grammatical Evolution*

Grammatical Evolution merupakan salah satu algoritma EAs yang menggunakan representasi individu berupa *Backus Naur Form* (BNF). *Production rules* BNF pada *Grammatical Evolution* menggunakan pemetaan *genotype* ke *phenotype* proses biologi di alam dan pendefinisian *production rules* BNF yang dapat dibangun fleksibel sesuai dengan permasalahan. Pada *Grammatical Evolution* juga terjadi proses seleksi individu, sehingga pada akhir proses evolusi akan terpilih individu-individu terbaik sebagai solusi yang paling optimal. dengan kelebihan ini, GE dapat melakukan pencarian untuk sangat banyak kemungkinan model prediksi, baik linier maupun non-linier.

2.2 *Backus Naur Form* (BNF)

BNF adalah notasi yang digunakan untuk mengekspresikan *grammar* yang dapat berupa terminal dan non-terminal sebagai berikut:

- Terminal-terminal, berupa item atau operator yang dapat muncul dalam kamus tersebut, antara lain; +, -, dan sebagainya.
- Non-terminal, dapat dikembangkan ke dalam satu atau lebih terminal dan non-terminal [3].

Suatu *grammar* dapat merepresentasikan {N,T,P,S} dimana N adalah himpunan non-terminal, T adalah himpunan terminal, P adalah himpunan *production rules* yang memetakan elemen-elemen dalam N menjadi T,

dan S adalah start symbol yang berupa satu simbol non-terminal (anggota N). Berikut ini adalah contoh BNF untuk sebuah ekspresi sederhana:

| | |
|---|---|
| $N = \{ \text{expr, op, pre_op} \}$ $T = \{ \text{sin, cos, tan, log, +, -, /, *, X, ()} \}$ $S = \langle \text{expr} \rangle$ P = dapat direpresentasikan sebagai: | |
| (1) $\langle \text{expr} \rangle$ | $:: = \langle \text{expr} \rangle \langle \text{op} \rangle \langle \text{expr} \rangle$ (A) $ (\langle \text{expr} \rangle \langle \text{op} \rangle \langle \text{expr} \rangle)$ (B) $ \langle \text{pre_op} \rangle (\langle \text{expr} \rangle)$ (C) $ \langle \text{var} \rangle$ (D) |
| (2) $\langle \text{op} \rangle$ | $:: = +$ (A) $ -$ (B) $ /$ (C) $ *$ (D) |
| (3) $\langle \text{pre_op} \rangle$ | $:: = \text{Sin}$ (A) $ \text{Cos}$ (B) $ \text{Tan}$ (C) $ \text{Log}$ (D) |
| (4) $\langle \text{var} \rangle$ | $:: = X$ |

Gambar 1 Contoh Grammar dari suatu BNF [3]

Production rules diatas memiliki empat aturan besar. Pada aturan nomor (1), suatu ekspresi non-terminal $\langle \text{expr} \rangle$ dapat dikembangkan ke dalam salah satu dari empat pilihan yaitu A, B, C atau D. Kemudian dari aturan nomor (1) dapat dikembangkan menjadi empat aturan lain yaitu 1A, 1B, 1C, 1D. Sehingga dari keempat aturan besar tersebut, production rules tersebut memiliki $4 + 4 + 4 + 1 = 13$ aturan.

2.3 Representasi Individu

Pada Grammatical Evolution, suatu kromosom yang merepresentasikan individu bisa dipandang sebagai

untaian biner. Kromosom ini di transkripsi ke dalam untaian integer dengan aturan bahwa setiap 8 bit mengodekan satu angka integer. Sehingga setiap angka integer bisa bernilai antara 0 sampai 255. Production rules berfungsi untuk mentranslasikan untaian integer tersebut sehingga menjadi sebuah fungsi. Berikut ini merupakan tabel ilustrasi dari representasi individu [3].

Tabel 1 Contoh Representasi Individu pada GE [3]

| | | | | | | | |
|----|----|---|----|-----|-----|-----|-----|
| 40 | 83 | 4 | 62 | 237 | 107 | 247 | 179 |
|----|----|---|----|-----|-----|-----|-----|

Kromosom yang berupa untaian integer tersebut akan ditranslasikan menggunakan production rules. Translasi dimulai dengan mengeksekusi bagian non-terminal yang dijadikan start symbol, dalam hal ini $\langle \text{expr} \rangle$. Pada production rules tersebut, terdapat empat pengembangan

untuk $\langle \text{expr} \rangle$, untuk memilih satu dari empat pilihan, maka dilakukan operasi modulo. Hasil proses tranlasi kromosom diatas dapat ditampilkan sebagai berikut:

Tabel 2 Proses Translasi Kromosom [3]

| Gen | Mod | Aturan Terpilih | Ekspresi |
|-----|-----|-----------------|---|
| | | - | $\langle \text{expr} \rangle$ |
| 40 | 0 | 1A | $\langle \text{expr} \rangle \langle \text{op} \rangle \langle \text{expr} \rangle$ |
| 83 | 3 | 1D | $\langle \text{var} \rangle \langle \text{op} \rangle \langle \text{expr} \rangle$ |
| - | - | - | $X \langle \text{op} \rangle \langle \text{expr} \rangle$ |
| 4 | 0 | 2A | $X - \langle \text{expr} \rangle$ |
| 62 | 2 | 1C | $X - \langle \text{pre_op} \rangle \langle \text{expr} \rangle$ |
| 237 | 1 | 3B | $X - \text{cos} (\langle \text{expr} \rangle)$ |
| 107 | 3 | 1D | $X - \text{cos} (\langle \text{var} \rangle)$ |
| - | - | - | $X - \text{cos} (X)$ |

2.4 Nilai Fitness

Nilai Fitness digunakan untuk mengevaluasi performansi suatu individu. Dalam kenyataannya, individu dengan nilai fitness rendah tidak dapat bertahan, sedangkan individu yang memiliki nilai fitness tinggi akan tetap hidup [4]. Karena prediksi curah hujan ini masuk ke dalam permasalahan minimasi, maka rumus nilai fitness yang digunakan adalah:

$$f = \frac{1}{(k + b)} \tag{1}$$

Keterangan:

f = nilai fitness

b= konstanta / bilangan yang sangat kecil

K= rata-rata kesalahan absolut prediksi untuk semua data aktual

2.5 Proses Grammatical Evolution

Secara garis besar, proses evolusi yang digunakan pada Grammatical Evolution menggunakan semua operator yang seperti proses evolusi pada GA, yaitu seleksi orangtua, CrossOver, mutasi, dan seleksi survivor. Akan tetapi, Grammatical Evolution menggunakan dua operator tambahan, yaitu duplicate dan prune pada proses dekode kromosom. Berikut ini adalah ilustrasi proses pengerjaan algoritma Grammatical Evolution:

1. Inisialisasi Populasi
Proses ini merupakan proses awal yaitu membangkitkan nilai-nilai gen dalam kromosom secara acak.
2. Dekode Kromosom
yaitu mengubah individu/kromosom atau solusi atau fungsi menggunakan *production rule* yang didefinisikan dalam *grammar* BNF. Jika kromosom dinyatakan tidak *valid*, maka dilakukan *duplicate*. Jika fungsi kromosom sudah dapat dihasilkan sebelum semua gen digunakan, maka dilakukan *prune* pada gen yang tidak digunakan.
3. Evaluasi Individu
yaitu menghitung kesalahan prediksi dan nilai *fitness* berdasarkan solusi yang telah di hasilkan dari proses dekode kromosom.
4. Elitisme
yaitu menggandakan kromosom yang memiliki nilai *fitness* tertinggi dari setiap populasi. Hal ini dilakukan agar solusi terbaik yang pernah dicapai tidak hilang.
5. *Linear Fitness Ranking*
yaitu melakukan penskalaan untuk semua kromosom dalam populasi. Hal ini dilakukan untuk menghindari kesalahan pada seleksi orang tua yang menggunakan proporsi nilai *fitness*.
6. Seleksi orangtua
yaitu proses pemilihan kromosom sebagai orangtua dengan menggunakan *roulette wheel*.
7. *CrossOver*
yaitu salah satu metode untuk menghasilkan kromosom baru yang didapatkan dari proses menyilangkan kedua gen orangtua. Peluang keberhasilan proses ini ditentukan dengan probabilitas *CrossOver* (P_c).
8. Mutasi
yaitu mengubah isi suatu gen. peluang terjadinya mutasi dinyatakan dengan (P_m).
9. Seleksi survivor
yaitu proses penggantian kromosom menggunakan salah satu dari kedua metode seleksi survivor, yaitu *Generational Replacement* atau *Steady State*.

2.6 Pengukuran Performansi

Weighted Mean Absolute Percentage Error (WMAPE) merupakan salah satu metode

pengukuran performansi. Untuk prediksi dengan

ukuran variabel kecil (terdapat data aktual yang bernilai 0 atau mendekati 0) WMAPE memiliki kelebihan yang membuat nilai yang tidak terdefinisi (infinite) menjadi terdefinisi [8]. Berikut ini

penjelasan rumus yang digunakan untuk menghitung WMAPE [9]:

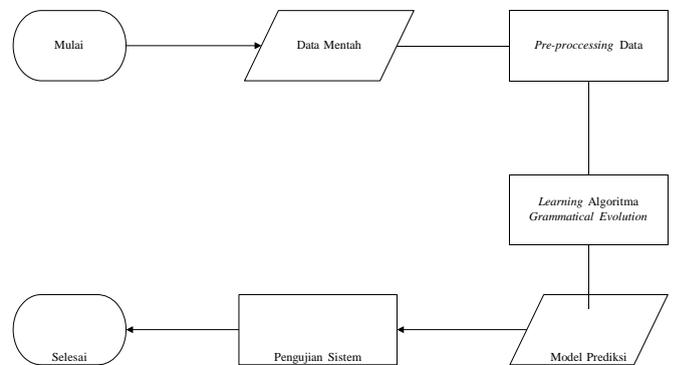
$$WMAPE = \frac{\sum \left| \frac{D_i - F_i}{D_i} \right| * D_i}{\sum D_i} \quad (2)$$

Keterangan:

- D_i = Data aktual ke- i
 F_i = Data prediksi ke- i
 $D_i - F_i$ = Nilai kesalahan (error) pada periode ke i

3. GAMBARAN SISTEM

Pada penelitian ini akan dibangun sistem prediksi curah hujan dengan menggunakan algoritma *Grammatical Evolution* dengan data *time series* parameter curah hujan bulanan menggunakan data BMKG Bandung wilayah Soreang. Fungsi prediksi dikatakan baik jika dapat menghasilkan nilai WMAPE yang kecil.



Gambar 2 flowchart deskripsi sistem prediksi curah hujan

3.1 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:

a. Moving Average

Penggunaan *Moving Average* bertujuan untuk menemukan pola data dan menghilangkan adanya kemungkinan nilai kosong pada data:

$$S_t = \frac{1}{n} \sum_{j=i}^{i+n-1} a_j \quad (3)$$

Keterangan:

S_t = *Moving average* untuk periode ke- i

a_j = Nilai aktual periode ke- j

n = Jumlah batas dalam *moving average*

b. Normalisasi

Normalisasi data yaitu mengubah data aktual menjadi *range* dengan interval [0...1] dengan tujuan meminimumkan *error* dan terdapat pencilaan data. [5]:

$$X_{ni} = ((\frac{Xi - \min(X)}{\max(X) - \min(X)}) * 0,8) + 0,1 \quad (3)$$

Keterangan:

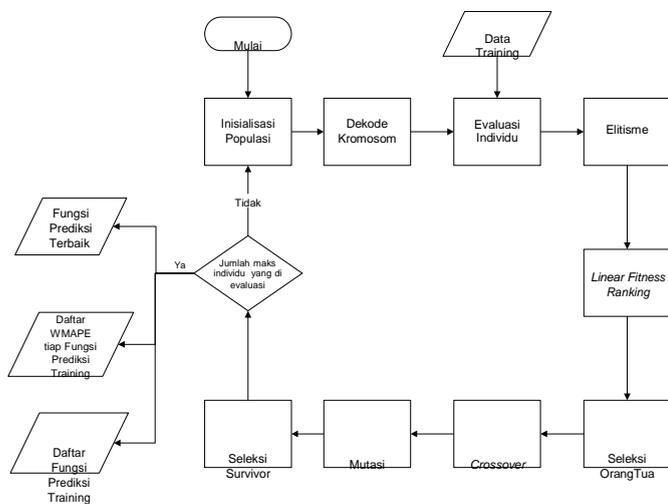
- X_{ni} = Data aktual normalisasi ke -i
- X_i = Data aktual dengan range data asli ke-i
- X = Data aktual dengan range data asli

c. Partisi Data

- Data training: 84 data (Januari 2003 – Desember 2009)
- Data testing: 36 data (Januari 2010 – Desember 2012)

3.2 *Learning algoritma Grammatical Evolution*

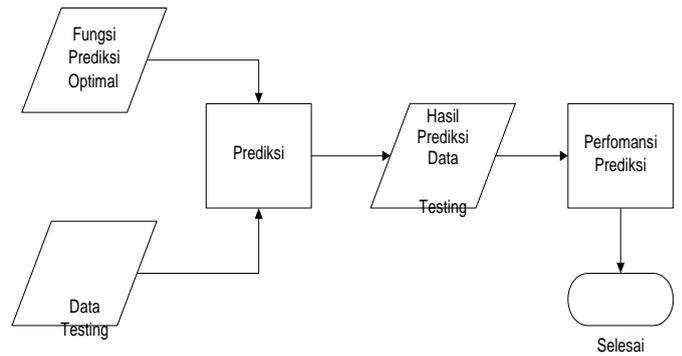
Tahap ini bertujuan untuk membangun model prediksi yang berupa fungsi yang akan di evaluasi dengan menggunakan data *training*. Berikut merupakan *flowchart learning Grammatical Evolution*:



Gambar 3 *flowchart* proses *learning Grammatical Evolution*

3.3 *Pengujian Sistem*

Pada tahap ini dilakukan prediksi dengan menggunakan fungsi prediksi terbaik yang didapatkan dari tahap sebelumnya dengan acuan data *testing* sejumlah M periode sebagai inputan proses. Hasil dari proses ini yaitu fungsi prediksi optimal dan performansi algoritma Grammatical Evolution dalam prediksi curah hujan M+1.



Gambar 4 proses pengujian (*testing*) *Grammatical Evolution*

3.3 *Pendefinisian BNF*

Pada tahap ini dilakukan pengujian terhadap penggunaan BNF yang digunakan. BNF ini didefinisikan sesuai dengan karakteristik prediksi data *time series*. Pola yang mungkin terbentuk menyatakan penjumlahan antara korelasi data curah hujan (dinyatakan dengan variabel x) atau kesalahan prediksi curah hujan (dinyatakan dengan variabel y). Berikut ini adalah definisi *grammar* yang digunakan:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------------|--|------------------|-----|--------------------|-----|----------------|-----|-------|-----|---------|-----|----|-----|------|-----|------|-----|--|--|------|-----|------|-----|--|--|------|-----|---|-----|--|--|
| N= | { <i>expr</i> , <i>op</i> , <i>expr</i> } | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| T= | { <i>sin</i> , <i>cos</i> , +, -, *, /, 0,1, 0,2, 0,3, 0,4, 0,5, 0,6, 0,7, 0,8, 0,9, 2, 3, x1, x2, x3, x4, y1, y2, y3, y4} | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| S= | { <i>expr</i> } | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1) <expr>::= | <table border="0"> <tr><td><expr><op><expr></td><td>(A)</td></tr> <tr><td>(<expr><op><expr>)</td><td>(B)</td></tr> <tr><td><pre_op><expr></td><td>(C)</td></tr> <tr><td><var></td><td>(D)</td></tr> <tr><td><const></td><td>(E)</td></tr> </table> | <expr><op><expr> | (A) | (<expr><op><expr>) | (B) | <pre_op><expr> | (C) | <var> | (D) | <const> | (E) | | | | | | | | | | | | | | | | | | | | |
| <expr><op><expr> | (A) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (<expr><op><expr>) | (B) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <pre_op><expr> | (C) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <var> | (D) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <const> | (E) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2) <pre_op>::= | <table border="0"> <tr><td>Sin</td><td>(A)</td></tr> <tr><td>Cos</td><td>(B)</td></tr> </table> | Sin | (A) | Cos | (B) | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Sin | (A) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Cos | (B) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3) <op>::= | <table border="0"> <tr><td>+</td><td>(A)</td></tr> <tr><td>-</td><td>(B)</td></tr> <tr><td>*</td><td>(C)</td></tr> <tr><td>/</td><td>(D)</td></tr> </table> | + | (A) | - | (B) | * | (C) | / | (D) | | | | | | | | | | | | | | | | | | | | | | |
| + | (A) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - | (B) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| * | (C) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| / | (D) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4) <const>::= | <table border="0"> <tr><td>0, 1</td><td>(A)</td><td>0, 6</td><td>(F)</td><td>3</td><td>(K)</td></tr> <tr><td>0, 2</td><td>(B)</td><td>0, 7</td><td>(G)</td><td></td><td></td></tr> <tr><td>0, 3</td><td>(C)</td><td>0, 8</td><td>(H)</td><td></td><td></td></tr> <tr><td>0, 4</td><td>(D)</td><td>0, 9</td><td>(I)</td><td></td><td></td></tr> <tr><td>0, 5</td><td>(E)</td><td>2</td><td>(J)</td><td></td><td></td></tr> </table> | 0, 1 | (A) | 0, 6 | (F) | 3 | (K) | 0, 2 | (B) | 0, 7 | (G) | | | 0, 3 | (C) | 0, 8 | (H) | | | 0, 4 | (D) | 0, 9 | (I) | | | 0, 5 | (E) | 2 | (J) | | |
| 0, 1 | (A) | 0, 6 | (F) | 3 | (K) | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0, 2 | (B) | 0, 7 | (G) | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0, 3 | (C) | 0, 8 | (H) | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0, 4 | (D) | 0, 9 | (I) | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0, 5 | (E) | 2 | (J) | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5) <var>::= | <table border="0"> <tr><td>x 1</td><td>(A)</td><td>y1</td><td>(E)</td></tr> <tr><td>x 2</td><td>(B)</td><td>y2</td><td>(F)</td></tr> <tr><td>x 3</td><td>(C)</td><td>y3</td><td>(G)</td></tr> <tr><td>x 4</td><td>(D)</td><td>y4</td><td>(H)</td></tr> </table> | x 1 | (A) | y1 | (E) | x 2 | (B) | y2 | (F) | x 3 | (C) | y3 | (G) | x 4 | (D) | y4 | (H) | | | | | | | | | | | | | | |
| x 1 | (A) | y1 | (E) | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| x 2 | (B) | y2 | (F) | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| x 3 | (C) | y3 | (G) | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| x 4 | (D) | y4 | (H) | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Gambar 5 definisi *grammar* BNF yang digunakan

4. HASIL DAN ANALISIS

Berdasarkan skenario yang telah dijelaskan sebelumnya, telah dilakukan 10 kali observasi pengujian untuk setiap kombinasi parameter dengan seleksi survivor *Generational Replacement* dan *Steady State*.

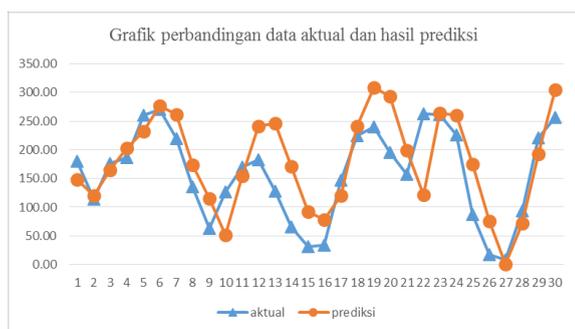
4.1 Generational Replacement

Tabel 3 Hasil Pengujian dengan *Generational Replacement*

| Uk Pop | Pc | Pm | Best WMAPE training | Best WMAPE testing | Average WMAPE training | Average WMAPE testing | |
|--------|-----|-----|---------------------|--------------------|------------------------|-----------------------|--------|
| 50 | 0.9 | 0.3 | 21.54% | 29.84% | 24.77% | 32.41% | |
| | | 0.1 | 20.72% | 29.50% | 25.76% | 32.15% | |
| | 0.7 | 0.3 | 21.10% | 29.33% | 25.49% | 31.68% | |
| | | 0.1 | 21.39% | 29.42% | 26.69% | 33.64% | |
| | 100 | 0.9 | 0.3 | 25.32% | 31.82% | 26.64% | 33.16% |
| | | | 0.1 | 22.42% | 29.69% | 25.86% | 31.99% |
| 0.7 | | 0.3 | 22.86% | 29.65% | 27.44% | 33.33% | |
| | | 0.1 | 24.24% | 28.84% | 26.77% | 32.01% | |

Berdasarkan observasi pencarian fungsi prediksi terbaik, dapat disimpulkan dari tabel diatas bahwa hasil skenario dengan penggunaan seleksi survivor *Generational Replacement* menunjukkan performansi yang paling baik pada ukuran populasi 50 dengan penggunaan Pc 0,7 dan Pm 0,3 yaitu dengan nilai best WMAPE testing sebesar 29,33% sehingga memiliki nilai performansi sebesar 70,67%.

Jika dilihat dari nilai *Average WMAPE testing*, kombinasi parameter tersebut juga menghasilkan nilai kesalahan (*error*) yang paling kecil, yaitu 31,68% sehingga memiliki nilai performansi sebesar 68,32%.. Jika digambarkan dalam bentuk grafik, hasil pengujian observasi dengan *best WMAPE testing* terbaik dapat digambarkan sebagai berikut:



Gambar 6 Hasil prediksi M+1 dengan *Generational Replacement*

Dari gambar diatas, hasil prediksi terlihat mengikuti fluktuasi dari data aktualnya, namun hasil tersebut belum mendekati nilai sebenarnya. Selain itu, seleksi survivor menggunakan *Generational Replacement* dengan menyimpan dua individu terbaik setiap generasinya juga mempengaruhi banyaknya individu baru yang terbentuk sehingga memungkinkan terjadinya perubahan fungsi terbaik setiap generasinya.

4.2 Steady State

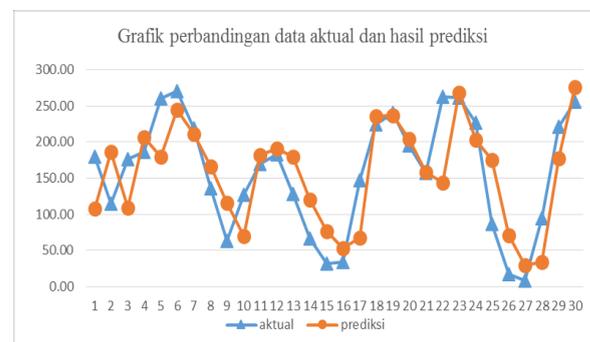
Tabel 4 Tabel Hasil Skenario Pengujian dengan *Steady State*

| Uk Pop | Pc | Pm | Best WMAPE training | Best WMAPE testing | Average WMAPE training | Average WMAPE testing |
|--------|-----|-----|---------------------|--------------------|------------------------|-----------------------|
| 50 | 0.9 | 0.3 | 24.17% | 27.66% | 27.50% | 32.76% |
| | | 0.1 | 22.01% | 29.91% | 27.73% | 33.93% |
| | 0.7 | 0.3 | 23.18% | 29.44% | 26.11% | 32.10% |
| | | 0.1 | 22.02% | 26.58% | 25.93% | 30.91% |
| 100 | 0.9 | 0.3 | 21.12% | 25.95% | 23.91% | 30.26% |
| | | 0.1 | 21.10% | 25.65% | 26.18% | 30.67% |
| | 0.7 | 0.3 | 22.07% | 28.12% | 25.99% | 32.50% |
| | | 0.1 | 21.92% | 27.88% | 24.81% | 31.29% |

Berdasarkan Tabel 4, terlihat bahwa hasil skenario dengan penggunaan seleksi survivor *Steady State* menunjukkan nilai *best WMAPE testing* ukuran populasi 100 dengan penggunaan Pc 0,9 dan Pm 0,1 yaitu dengan nilai sebesar senilai 25,65% sehingga memiliki nilai performansi sebesar 74,35%.

Namun, jika dilihat dari nilai *Average WMAPE testing*, kombinasi parameter dengan ukuran populasi 100, Pc 0,9 dan Pm 0,3 menghasilkan nilai kesalahan (*error*) yang paling kecil, yaitu 30,26% sehingga memiliki nilai performansi sebesar 69,74%

Penentuan ukuran populasi, Pc, dan Pm dapat mempengaruhi performansi yang dihasilkan. Penggunaan ukuran populasi yang besar dapat memperbanyak kemungkinan variasi dari individu, sedangkan penggunaan generasi yang tidak terlalu besar dapat mengurangi kemungkinan dihasilkan nilai *fitness* yang konvergen. Jika digambarkan dalam bentuk grafik, hasil pengujian observasi dengan *best WMAPE testing* terbaik untuk curah hujan M+1 dapat digambarkan sebagai berikut:



Gambar 7 Hasil prediksi M+1 dengan *Steady State*

Berdasarkan Gambar 7, terlihat fluktuasi hasil prediksi pada awalnya berbanding terbalik dengan data

aktualnya, namun semakin lama semakin mengikuti pola data aktualnya.

Seleksi survivor dengan *Steady State* dengan menyimpan sejumlah M individu terbaik setiap generasi dapat memperbesar peluang terbentuknya individu baru dari gen-gen orangtua terbaik pada generasi tersebut.

Proses *learning* mengakibatkan individu dalam populasi mengalami perbaikan nilai *fitness* sehingga diharapkan setiap individu menjadi konvergen menjadi individu terbaik. Besarnya populasi pada setiap generasi mempengaruhi kecepatan konvergensi. Semakin besar ukuran populasi akan mengakibatkan konvergensi yang lambat, akan tetapi apabila jumlah populasi semakin kecil maka adanya kemungkinan terjadinya konvergensi prematur. Penggunaan dua tipe penggantian solusi *Generational Replacement* dan *Steady State* menghasilkan performansi yang berbeda dan waktu komputasi yang berbeda. Dari seluruh observasi yang dilakukan, didapatkan satu fungsi prediksi terbaik untuk *Generational Replacement* (GR) maupun *Steady State* (SS):

Tabel 5 Parameter Observasi dari Skenario Terbaik

| Seleksi Survivor | Uk Pop | Pc | Pm | Fungsi Prediksi optimal |
|------------------|--------|-----|-----|---|
| GR | 50 | 0,7 | 0,3 | $\sin((x_4 + 0,7 - \cos(\cos(\sin(2 * \sin(x_2) - \sin(y_4)) - y_4)))) / \cos(0,1)$ |
| SS | 100 | 0,9 | 0,1 | $x_4 / (0,5 + x_3)$ |

4.3 Analisis Kompleksitas Waktu Asimptotik

Kompleksitas waktu asimptotik yang telah dibangun oleh algoritma GE adalah $O(G(P^2 + ni))$, kompleksitas ini dipengaruhi oleh G yang merupakan generasi, P yang merupakan populasi, n yang merupakan jumlah data dan i yang merupakan jumlah gen.

Jika pada tugas akhir ini menggunakan skenario dengan maksimum generasi adalah 200, jumlah populasi maksimum 100, jumlah gen rata-rata 100 dan jumlah data 84 maka akan diperoleh nilai proses yang dilakukan GE untuk sekali proses *training* adalah 3.680.000 proses.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan observasi yang telah dilakukan, tugas akhir ini dapat disimpulkan sebagai berikut:

1. Pendefinisian *grammar* BNF merupakan komponen penting pada *Grammatical Evolution*. Hal ini dikarenakan *grammar* yang digunakan pada BNF sangat menentukan fungsi prediksi yang dihasilkan.
2. *Grammatical Evolution* dapat memberikan performansi yang cukup baik dan layak dalam memprediksi curah hujan $M+1$ yaitu berkisar 60% hingga 75%. Hasil observasi terbaik untuk seleksi survivor *generational replacement* yaitu 70,67% dengan ukuran populasi 50, Pc 0,7 dan Pm 0,3. Sedangkan pada seleksi survivor *steady state* yaitu 74,35% dengan ukuran populasi 100, Pc 0,9 dan Pm 0,1. Nilai hasil prediksi belum mendekati nilai sebenarnya, namun pola yang dihasilkan dapat mengikuti pola data aktual. .
3. Performansi algoritma dengan menggunakan seleksi survivor *steady state* memungkinkan didapatkan hasil yang lebih baik dibandingkan dengan *generational replacement*. Hal ini disebabkan karena pada *steady state* separuh individu terbaik pada generasi sebelumnya masih dimasukkan kedalam generasi berikutnya. Sehingga memungkinkan terjadinya *crossover* antar dua individu terbaik pada generasi-generasi berikutnya.
4. Kompleksitas waktu asimptotik yang dihasilkan dari fungsi prediksi yang dihasilkan oleh algoritma *Grammatical Evolution* adalah $O(G(P^2 + ni))$. Hal ini tergantung pada jumlah maksimum generasi, ukuran populasi, jumlah gen, dan banyaknya jumlah data yang digunakan.

5.2 Saran

Saran dari tugas akhir ini yaitu:

1. Penggunaan metode seleksi orangtua selain *roulette wheel* dan penggunaan > 1 titik potong untuk proses *CrossOver* dan mutasi sehingga memungkinkan didapatkan kromosom bervariasi yang akan di evaluasi.
2. Mengombinasikan *Grammatical Evolution* dengan algoritma lainnya yang diharapkan dapat menghasilkan performansi yang lebih baik
3. Penggunaan parameter cuaca yang lain yang mempengaruhi curah hujan sebagai acuan untuk prediksi ke depannya.

6 DAFTAR PUSTAKA

- [1] F.Nhita dan Adiwijaya, 2013."A Rainfall Forecasting using *Fuzzy* System Based on Genetic Algorithm".International Conference of Information and Communication Technology (ICoICT)
- [2] Ryan, Conor. Gecco 2006 Grammatical Evolution Tutorial.(<http://www.grammaticalevolution.org/tutorial.pdf>, diakses 21 April 2014 pukul 20.10 WIB)
- [3] Suyanto. (2008). Evolutionary Computation Komputasi Berbasis Evolusi dan Genetika, Informatika, Bandung
- [4] Suyanto. (2008). Soft Computing Membangun Mesin ber-IQ Tinggi, Informatika, Bandung.
- [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] Wibowo Agung Toto, Widhiantika Bunga Ayu (2010). "Peramalan Harga Dinar di Indonesia Menggunakan Grammatical Evolution", Konferensi Nasional Sistem dan Informatika. (Bali 2010)
- [7] Nurcahyo Septian, F. Nhita & Adiwijaya, "Rainfall Prediction in Kemayoran Jakarta Using Hybrid Genetic Algorithm (GA) and Partially Connected Feedforward Neural Network (PCFNN)", The Second International Conference on Information and Communication Technology (ICOICT 2014).
- [8] Hyndman, rob, dkk, 2005. "Another Look at Measures of Forecast Accuracy" of Monash University
- [9] Wilson, Tom, 2012." Forecast Accuracy and Uncertainty of Australian Bureau of Statistics State and Territory Population Projections". International Journal of Population Research