



## **Proposal Tesis**

# ***Grammatical Evolution* untuk Ekstraksi Fitur dengan Pengukuran *Multi Fitness***

**Go Frendi Gunawan**

**NRP : 5111201033**

DOSEN PEMBIMBING

**Prof. Dr. Ir. Joko Lianto Buliali, Msc.**

PROGRAM MAGISTER

BIDANG KEAHLIAN KOMPUTASI CERDAS VISUAL

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

**2012**

# GRAMMATICAL EVOLUTION UNTUK EKSTRAKSI FITUR DENGAN PENGUKURAN MULTI FITNESS

Nama mahasiswa : Go Frendi Gunawan  
NRP : 5111201033  
Pembimbing : Prof. Dr. Ir. Joko Lianto Buliali, M.Sc

## ABSTRAK

Ekstraksi fitur merupakan salah satu topik yang cukup berpengaruh untuk menyelesaikan masalah klasifikasi. Sampai saat ini, tidak ada cara yang baku untuk menentukan fitur-fitur terbaik dari suatu data. Dalam proposal thesis ini, akan dicoba suatu pendekatan grammatical evolution dengan pengukuran multi fitness guna memperoleh fitur-fitur terbaik dari sebuah data.

Grammatical evolution digunakan untuk mengubah deret angka random menjadi sebuah fungsi berdasarkan *grammar* terdefinisi. Fungsi-fungsi yang terbentuk kemudian akan dijadikan fitur baru dan diukur tingkat kebaikannya untuk memisahkan satu kelas dengan kelas lain. Dalam hal ini, akan ada  $n$  nilai pengukuran untuk setiap fitur, di mana  $n$  adalah jumlah kelas

**Kata Kunci:** ekstraksi fitur, *grammatical evolution*, klasifikasi, multi-fitness.

# GRAMMATICAL EVOLUTION FOR FEATURE EXTRACTION WITH MULTI FITNESS EVALUATION

Name : Go Frendi Gunawan  
Student Identity Number: 5111201033  
Supervisor : Prof. Dr. Ir. Joko Lianto Buliali, M.Sc

## ABSTRACT

Feature Extraction is a significant topic in classification problem solving. Until now, there is no such a standard way to determine the best features of a data. In this thesis proposal, grammatical evolution with multiple fitness evaluation approach will be used in order to extract best features of the data.

Grammatical Evolution is used to change random number set into a mathematic function based on defined grammar. The generated function is then become new features. The goodness of those features to separate a class with another classes is then measured. In this case, there will be  $n$  goodness value, with  $n$  is the count of the classes

**Kata Kunci:** feature extraction, grammatical evolution, classification, multi-fitness.

## DAFTAR ISI

LEMBAR PENGESAHAN PROPOSAL TESIS .....	i
ABSTRAK.....	ii
ABSTRACT.....	iii
DAFTAR ISI.....	iv
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	1
1.3 Batasan Masalah .....	2
1.4 Tujuan Penelitian .....	2
1.5 Manfaat Penelitian .....	2
BAB 2 KAJIAN PUSTAKA DAN DASAR TEORI.....	3
2.1 Ekstraksi Fitur.....	3
2.2 Grammatical Evolution.....	4
2.2.1. Grammar Pada Grammatical Evolution.....	5
2.2.2. Transformasi Genotip ke Fenotip pada Grammatical Evolution.....	7
BAB 3 METODE PENELITIAN.....	8
3.1 Langkah-Langkah Penelitiann.....	8
3.2 Jadwal Kegiatan Penelitian.....	9
3.3 Rancangan Sistem.....	9
3.3.1. Pembuatan Fitur.....	10
3.3.2. Penilaian Fitness.....	10
3.3.3. Pengukuran Performa Fitur.....	11
DAFTAR PUSTAKA.....	12

# **BAB 1**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Ekstraksi fitur merupakan salah satu hal yang paling berpengaruh dalam pemecahan masalah klasifikasi. Pemilihan fitur yang tidak baik akan mengakibatkan kesulitan dalam memisahkan kelas-kelas data. Kegagalan pemisahan kelas-kelas data akan berdampak pada turunnya akurasi dalam proses klasifikasi.

Dalam penelitian sebelumnya Gunawan et al, 2012, telah dicoba suatu pendekatan ekstraksi fitur dengan menggunakan *grammatical evolution*. Dalam penelitian tersebut, terdapat sebuah kelemahan dikarenakan hanya dilakukan 1 tolak ukur global untuk pengukuran *fitness value*. Hal ini mengakibatkan fitur-fitur yang sebenarnya cukup baik secara khusus, justru tersingkirkan karena nilai *fitness* globalnya rendah. Penelitian-penelitian lain seperti Gravilis et al, 2006 dan Gravilis et al, 2008 juga masih menggunakan satu nilai *fitness* saja. Guo et al, 2011 dan Li et al, 2011 menggunakan *classifier* sebagai bagian dalam pengukuran *fitneess*.

Dalam penelitian ini, akan dibuat suatu standar baru dalam penilaian *fitness*. Penilaian *fitness* tersebut akan dilakukan dengan skenario 1 vs all untuk semua kelas. Pada akhir proses, diharapkan akan ditemukan sejumlah  $n$  fitur terbaik, di mana  $n$  sama dengan jumlah kelas dalam proses klasifikasi.

### **1.2 Perumusan Masalah**

Dalam penelitian ini, masalah-masalah yang akan diselesaikan dirumuskan sebagai berikut:

1. Bagaimana menentukan formula untuk mengukur nilai *fitness* dari sebuah fitur
2. Bagaimana membuat skenario 1 vs all untuk semua kelas

3. Bagaimana melakukan pengujian atas fitur-fitur yang sudah di *generate*

### **1.3 Batasan Masalah**

Batasan masalah dalam penelitian ini adalah:

1. Data yang diproses adalah data numerik
2. Data yang diproses harus lengkap dan bisa dipisahkan per kelas.

### **1.4 Tujuan Penelitian**

Menciptakan dan menguji suatu metode baru dengan menggunakan prinsip *grammatical evolution* untuk mengekstraksi fitur pada data numerik.

### **1.5 Manfaat Penelitian**

Hasil penelitian dapat digunakan sebagai salah satu langkah pre-processing sebelum melakukan proses klasifikasi.

Dengan metode ekstraksi fitur dalam tahapan pre-processing, diharapkan proses klasifikasi data yang tidak memiliki korelasi langsung terhadap kelas dapat dilakukan dengan lebih baik.

## **BAB 2**

### **KAJIAN PUSTAKA DAN DASAR TEORI**

Pada bab ini akan dibahas beberapa teori dasar yang menunjang dalam pembuatan Tugas Akhir.

#### **2.1 Ekstraksi Fitur**

Ekstraksi Fitur merupakan sebuah teknik untuk memilih fitur-fitur terbaik yang bisa digunakan untuk mengklasifikasikan data secara paling sederhana.

Di dalam proses ekstraksi fitur, terkadang dimunculkan fitur-fitur baru berdasarkan fitur-fitur original. Proses tersebut sering pula dikenal dengan sebutan feature construction. Namun ada kalanya, hanya digunakan sebagian dari fitur-fitur original, atau biasa disebut sebagai feature selection. Dalam berbagai kasus, seringkali keduanya dilakukan bersamaan, sehingga muncul fitur-fitur baru yang berdampingan dengan beberapa fitur original.

Adapun fitur-fitur hasil ekstraksi bisa dikatakan baik, jika berhasil memisahkan data berdasarkan kelas yang diharapkan dengan tingkat kesalahan sekecil mungkin. Walau demikian, tingkat kompleksitas masing-masing fitur dan tingkat kompleksitas operasi untuk melakukan pemisahan data juga perlu dipertimbangkan. Kerumitan pada saat pembuatan fitur baru maupun pada saat operasi klasifikasi akan berpengaruh buruk pada performa proses klasifikasi itu sendiri.

Untuk menggambarkan tujuan ekstraksi fitur secara lebih jelas, maka disediakan contoh data numerik pada tabel 2.1. Jika digambarkan dalam bentuk cartesian dengan fitur x dan fitur y sebagai aksis dan ordinat, maka kelas A, B dan C tidak bisa dipisahkan secara linear. Pengklasifikasian data seperti ini dengan menggunakan neural network misalnya, akan membutuhkan banyak hidden neuron yang berujung pada peningkatan kompleksitas proses klasifikasi.

**Tabel 2.1. Contoh Data numerik**

Fitur original		kelas
x	y	
0	0	A
1	1	A
-1	-1	A
1	-1	A
-1	1	A
2	2	B
-2	-2	B
-2	2	B
2	-2	B
3	3	C
-3	-3	C
3	-3	C
-3	3	C

Proses ekstraksi fitur diharapkan dapat menyederhanakan proses klasifikasi. Proses ekstraksi fitur dapat saja menghasilkan sebuah fitur tunggal  $x^2 + y^2$  yang sanggup memisahkan data secara linear. Fitur tersebut cukup baik, namun bukan yang terbaik, karena memiliki kompleksitas yang cukup tinggi jika dibandingkan dengan fitur-fitur original.

Salah satu alternatif yang lebih baik adalah dengan menggunakan  $\text{abs}(x) + \text{abs}(y)$ . Fitur ini memiliki kompleksitas yang lebih rendah daripada fitur sebelumnya, namun tetap mampu memisahkan data secara linear.

## 2.2 Grammatical Evolution

Dikarenakan setiap data akan memiliki karakteristik yang berbeda-beda. Oleh sebab itu, tidak ada rumus baku untuk menghasilkan fitur baru secara umum. Cara yang cukup masuk akal untuk menghasilkan fitur-fitur semacam itu adalah dengan *trial and error*.

Algoritma genetika dan turunan-turunannya terbukti cukup baik dalam men-simulasikan trial and error yang dilakukan oleh manusia. *Grammatical evolution* merupakan salah satu turunan algoritma genetika yang memiliki context



free *grammar*, sehingga sangat cocok digunakan untuk mengekstraksi fitur dari data-data numerik.

Pada *grammatical evolution*, sebuah individu memiliki dua buah representasi. Representasi yang pertama adalah representasi genotip, sedangkan representasi yang kedua adalah representasi fenotip.

Representasi genotip di sini berupa sekumpulan angka sebagaimana layaknya pada algoritma genetika. Di sini representasi genotip akan digunakan untuk mengevolusikan sebuah start symbol pada sebuah *grammar* untuk menjadi sebuah kalimat. Kalimat tersebutlah yang menjadi representasi fenotip individu.

Representasi genotip pada *grammatical evolution* tidaklah berbeda dengan representasi individu pada algoritma genetika, yakni berupa sekumpulan angka. Angka yang dimaksud bisa berupa angka biner maupun desimal. Contoh representasi genotip pada *grammatical evolution* dalam bentuk biner adalah 11001001.

Representasi genotip perlu diubah ke dalam bentuk fenotip dengan menggunakan *grammar*. Representasi fenotip pada *grammatical evolution* dapat berupa fungsi matematika ataupun sebuah program komputer, tergantung pada *grammar* yang didefinisikan. Contoh representasi fenotip yang berupa fungsi matematika adalah  $X+1$ ,  $(X+3)*2$  dan seterusnya.

Untuk proses penentuan *fitness* value, yang digunakan adalah representasi fenotip, sedangkan untuk operasi genetika, yang digunakan adalah representasi genotip.

### **2.2.1. Grammar Pada Grammatical Evolution**

Seperti yang telah disebutkan, bahwa untuk mentransformasikan representasi genotip menjadi representasi fenotip dibutuhkan sebuah *grammar*. *Grammar* di sini sebenarnya mirip dengan *grammar* dalam bahasa natural. Hanya saja, direpresentasikan dalam bentuk backus naur form (BNF). Dalam sebuah *grammar* terdapat beberapa bagian penting, antara lain:

**Tabel 2.2. Contoh Grammar**

Node Notation	Node	Aturan Produksi	Notasi Aturan
(A)	<expr>	<expr><op><expr>	(A1)
		<num>	(A2)
		<var>	(A3)
(B)	<op>	+	(B1)
		-	(B2)
		*	(B3)
		/	(B4)
(C)	<var>	x	(C1)
		y	(C2)
(D)	<num>	1	(D1)

- T : Terminal set. Merupakan node-node yang sudah tidak mungkin dievolusikan
- N : Non-terminal set. Merupakan node-node yang masih mungkin dievolusikan
- P : Production rules. Merupakan keseluruhan *grammar*
- S :Start symbol. Merupakan salah satu anggota N yang digunakan sebagai node

Semisal, didefinisikan production rules (P) seperti pada tabel 2.2, maka  $T=\{+, -, *, /, x, y, 1\}$ . Node-node yang menjadi anggota T, sudah tidak mungkin dapat dievolusikan. Sedangkan yang dimaksud N adalah  $\{<expr>, <op>, <var>, <num>\}$ . Node-node tersebut masih mungkin berevolusi. Semisal node <op>, dapat berevolusi menjadi +, -, \*, ataupun /. Sementara itu, yang menjadi start symbol (S) adalah <expr>. Jadi jika ada sebuah genotip yang akan dicari representasi fenotipnya, maka akan digunakan node <expr> sebagai node awal.

### 2.2.2. Transformasi Genotip ke Fenotip pada Grammatical Evolution

Seandainya kita memiliki sederetan angka representasi genotip dalam bentuk biner 11.01.00.10.01, maka proses untuk mendapatkan fenotipnya dapat digambarkan secara lengkap pada tabel 2.3.

**Tabel 2.3. Proses Transformasi Genotip ke Fenotip**

Before	Gene	Rule	After Transformation
<expr>	11 -> 3	<expr><op><expr>	<expr><op><expr>
<expr>	01 -> 1	<num>	<num><op><expr>
<num>	-	1	1<op><expr>
<op>	00 -> 0	+	1+<expr>
<expr>	10 -> 2	<var>	1+<var>
<var>	01 -> 1	y	1+y

Proses transformasi diawali dengan start symbol (dalam hal ini <expr>). Selanjutnya diambil sebuah segmen dari genotip (dalam hal ini 11). Segmen tersebut dapat pula dinyatakan dalam bilangan decimal (dalam hal ini 3). Node <expr> memiliki 3 kemungkinan perubahan (A0 : <expr><op><expr>, A1:<num>, dan A2:<var>). Untuk menentukan aturan mana yang akan digunakan, maka dilakukan operasi modulo (sisanya), di mana segmen genotip terpilih akan dibagi dengan jumlah kemungkinan evolusi. Karena  $3 \bmod 3 = 0$ , maka dipilihlah aturan A0, yakni <expr><op><expr>. Proses ini dilanjutkan terus sampai seluruh node telah bertransformasi menjadi anggota terminal set (T).

Dalam contoh transformasi di tabel 2.3, diperoleh representasi fenotip dari 11.01.00.10.01 adalah 1+Y

## **BAB 3**

### **METODE PENELITIAN**

#### **3.1 Langkah-Langkah Penelitiann**

Pada penelitian ini, terdapat beberapa tahapan penyelesaian yang akan dilakukan, yang masing-masing tahapan menggunakan suatu metode tertentu. Adapun tahapan dan metode yang digunakan adalah sebagai berikut (gambar 3.1):

1. Studi literatur dan pencarian dataset

Proses ini terdiri atas pencarian referensi-referensi pendukung yang sesuai, baik dari buku, jurnal, maupun artikel. Proses tersebut dilanjutkan dengan pencarian data-data numerik yang tersedia di internet sesuai dengan batasan permasalahan.

2. Menyusun *grammar* dan rancang bangun sistem

Proses ini terdiri atas perancangan formula *grammar* dan algoritma umum dalam proses ekstraksi fitur

3. Menyusun rancangan pengujian sistem

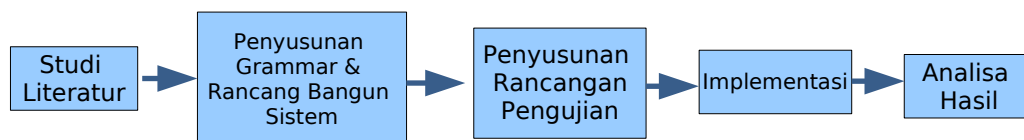
Dalam proses ini ditentukan skenario pengujian. Pengujian yang dimaksud dapat berupa perbandingan hasil klasifikasi dengan ekstraksi fitur dalam penelitian ini, ekstraksi fitur dalam penelitian sebelumnya, dan tanpa ekstraksi fitur. Performa klasifikasi (seperti kompleksitas) akan turut dihitung. Untuk proses klasifikasi sendiri akan digunakan soft SVM

4. Mengimplementasikan sistem

Sistem akan dibuat dalam bahasa pemrograman python yang umum digunakan dalam kepentingan penelitian.

5. Menganalisis hasil yang diperoleh untuk menghasilkan kesimpulan

Hasil ekstraksi fitur pada langkah nomor 4 akan diuji sesuai dengan rancangan pada langkah no 3. Selanjutnya akan disimpulkan apakah hasil penelitian lebih baik dari penelitian sebelumnya. Seandainya tidak lebih baik, maka akan dianalisa penyebab kegagalannya.



**Gambar 3.1 Skema Metode Penelitian**

### 3.2 Jadwal Kegiatan Penelitian

Pada Tabel 3.1 diperlihatkan jadwal kegiatan penelitian selama 4 bulan. Jadwal akan disajikan perminggu selama 4 bulan mulai dari November 2012 sampai dengan Februari 2013.

**Tabel 3.1 Jadwal Rencana Kegiatan Penelitian**

Kegiatan	Bulan															
	November 2012				Desember 2012				Januari 2013				Februari 2013			
Studi Literatur	■	■	■	■	■	■	■	■								
Analisa dan Desain Algoritma					■	■	■	■								
Pengembangan Perangkat Lunak							■	■	■	■	■	■	■	■	■	
Analisa Hasil Pengujian											■	■	■	■	■	
Penulisan Laporan									■	■	■	■	■	■	■	■

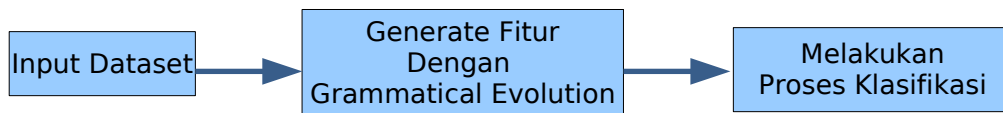
### 3.3 Rancangan Sistem

Secara umum, algoritma yang akan digunakan adalah sebagai berikut:

Input dataset

1. Melakukan *grammatical evolution* untuk menciptakan fitur-fitur
2. Generate genotip
  1. Transform genotip menjadi fenotip (fitur-fitur baru), sesuai dengan aturan *grammar* yang disediakan
  2. Hitung nilai *fitness* dari setiap fenotip (fitur-fitur baru) terhadap setiap kelas.
  3. Pilih fitur-fitur terbaik

3. Membangun *feature space* berdasarkan fitur-fitur terbaik, dan melakukan proses klasifikasi.



**Gambar 3.2. Skema algoritma**

### 3.3.1. Pembuatan Fitur

Proses pembuatan fitur dilakukan dengan menggunakan *grammatical evolution*. Proses ini ditujukan untuk membuat sebanyak mungkin calon fitur yang akan dinilai tingkat *fitness* nya.

Proses ini dimulai dengan pendefinisian *grammar*. *Grammar* yang telah didefinisikan, kemudian akan digunakan untuk mentransformasi sejumlah genotip yang dihasilkan secara random menjadi sejumlah fenotip. Setiap fenotip akan dihitung nilai *fitness* nya. Selanjutnya semua fenotip akan diurutkan berdasarkan nilai *fitness*.

### 3.3.2. Penilaian Fitness

Secara umum, sebuah fitur akan dikatakan baik apabila:

- Proyeksi data terhadap fitur tidak tumpang tindih antara kelas-kelas yang berbeda
- Dalam proyeksi data terhadap fitur tidak ada kelas lain di antara kelas yang diharapkan
- Jarak intra-class dari kelas yang diharapkan cukup kecil, sedangkan jarak inter-class terhadap kelas-kelas yang tidak diharapkan cukup besar.
- Fitur relatif tidak kompleks

Ada dua cara yang akan digunakan untuk menilai *goodness* dari sebuah fitur. Cara pertama adalah dengan menggunakan akurasi classifier. Cara kedua adalah dengan mengukur keterpisahan data secara empiris. Dalam penelitian ini, kedua cara

tersebut digunakan. Adapun pengukuran keterpisahan data secara empiris dilakukan dengan menggunakan rumus sebagai berikut:

$$\text{fitness} = (\text{SI} + 2 * (1 - \text{IP}) + 3 * (1 - \text{CP})) / 6$$

SI = Separability Index

IP = Intrusion Proportion

CP = Collision Proportion

Collision Proportion merupakan perbandingan antara jumlah data dalam satu kelas yang tidak terpisahkan dengan data dari kelas lain dengan keseluruhan data

Intrusion Proportion merupakan perbandingan antara jumlah data dalam satu kelas yang berada di antara nilai minimum dan maksimum dari kelas lain dengan keseluruhan data. Separability index dihitung dengan

#### **3.3.2.1. Skenario Feature Selection**

Dalam skenario ini, akan dipilih subset dari fitur original yang paling mampu memisahkan kelas dalam data secara optimum. Penilaian fitness dilakukan dengan memanfaatkan akurasi separator.

#### **3.3.2.2. Skenario Global Separator**

Dalam skenario global, akan dipilih sebuah fitur yang mampu memisahkan semua kelas secara cukup baik. Penilaian fitness dilakukan dengan cara mengukur keterpisahan data secara empiris.

#### **3.3.2.3. Skenario Local Separator**

Dalam skenario lokal, akan dipilih  $n$  fitur yang mampu memisahkan  $n$  kelas secara cukup baik. Semisal  $C$  adalah himpunan kelas yang ada, maka kelas pertama disimbolkan sebagai  $C_1$ , kelas ke dua disimbolkan sebagai  $C_2$ , dan kelas terakhir disimbolkan sebagai  $C_n$ . Dalam kasus ini, akan dibuat  $n$  buah fitur yang akan memisahkan setiap kelas dengan komplemen nya.

Maka untuk  $n$  buah kelas, akan dibuat  $n$  buah fitur.  $F_1$  akan bertugas memisahkan  $C_1$  dengan kelas-kelas lain.  $F_2$  akan bertugas memisahkan  $C_2$  dengan kelas-kelas lain, demikian seterusnya.

Penilaian fitness dalam skenario ini dilakukan dengan cara mengukur keterpisahan data secara empiris.

#### **3.3.2.4. Skenario Multi Accuration**

Skenario ini merupakan variasi dari skenario local separator. Yang berbeda dari skenario ini adalah cara menilai fitness. Pada skenario, digunakan akurasi classifier sebagai penentu fitness.

#### **3.3.2.5. Skenario Tatami Local Separator**

Skenario ini merupakan pengembangan dari skenario local separator. Pengembangan tersebut berasal dari hipotesa bahwa jika sebuah kelas telah terpisah dari semua kelas lainnya, maka pada proses selanjutnya, kelas yang sudah terpisah tersebut bisa diabaikan. Dalam skenario ini, akan tercipta  $n-1$  buah fitur.

Secara konsep, jika terdapat  $n$  buah kelas, yang masing-masing disimbolkan dengan  $C_1$  sampai dengan  $C_n$ , maka akan dipilih satu kelas yang paling terpisah dari kelas-kelas lain. Kelas ini selanjutnya disimbolkan sebagai  $C^*1$ .

Kemudian diekstrak fitur  $F_1$  yang bertugas untuk memisahkan  $C^*$  dan  $C-C^*1$ . Proses akan diulang dengan  $C-C^*1$  sebagai himpunan kelas yang baru. Di sini  $C^*1$  diabaikan, karena sudah terpisah dari kelas-kelas lain. Selanjutnya akan dipilih  $C^*2$  yang baru, dari  $C-C^*1$ .  $C^*2$  dan  $C-C^*1$  akan dipisahkan oleh fitur  $F_2$ . Demikian seterusnya sampai  $C^*n-1$  dan fitur  $F_{n-1}$ .

Inspirasi skenario ini berasal dari bentuk rantai tradisional Jepang yang dikenal dengan sebutan tatami. Adapaun penentuan fitness dalam skenario ini menggunakan pengukuran keterpisahan data secara empiris.

### **3.3.3. Pengukuran Performa Fitur**

Untuk klasifikasi, akan digunakan classifier *Naive-Bayes* yang sudah umum digunakan. Dalam proses ini, akan dilakukan berbagai skenario perbandingan sesuai dengan yang telah dibahas pada subbab sebelumnya.



Diharapkan skenario tatami akan memperoleh hasil yang lebih baik dibandingkan dengan skenario-skenario lain.

Pengujian dilakukan terhadap berbagai macam data. Selain data sintesis yang sengaja dibuat untuk menguji hipotesa, percobaan juga akan dilakukan pada data iris dan data e.coli yang telah umum dipakai dalam penelitian-penelitian sejenis.

## **BAB 4**

### **HASIL PENELITIAN DAN PEMBAHASAN**

Percobaan skenario diimplementasikan dengan menggunakan bahasa pemrograman python 2.7 dan beberapa library eksternal. Adapun library eksternal yang digunakan adalah scipy, numpy, matplotlib dan scikit-learn.

Source code program dan hasil lengkap penelitian diletakkan di repository github. Repository tersebut berlisensi open-source dan bisa diakses secara publik di alamat <https://github.com/goFrendiAsgard/feature-extractor>.

Dalam percobaan yang dilakukan terdapat beberapa skenario yang diujikan pada berbagai macam data.

#### **4.1 Pengujian Terhadap Data Sintesis 01**

Untuk kepentingan uji coba penelitian, maka dibuat sebuah data sintesis. Data ini dibuat dengan menggunakan aplikasi libre office dengan penentuan kelas sesuai dengan formula percabangan yang dibuat secara khusus.

Dalam data sintesis ini ada 4 atribut antara lain attack, defense, agility dan stamina. Setiap atribut memiliki pola sebaran random uniform. Berdasarkan keempat atribut tersebut, data dibagi menjadi 3 buah kelas dengan aturan sebagai berikut:

- Kelas defender jika  $\text{attack/defense} \geq 1.2$
- Kelas demon\_hunter jika  $\text{attack/defense} < 1.2$  dan  $\text{agility} \geq \text{stamina}$
- Kelas wizard jika  $\text{attack/defense} < 1.2$  dan  $\text{agility} < \text{stamina}$

Pengujian terhadap data ini dilakukan dengan menggunakan 5 skenario pengestrasian fitur yang telah dibahas pada subbab 3.3.2.

Percobaan dilakukan dua kali. Percobaan pertama dilakukan dengan melibatkan keseluruhan dataset sebagai training sekaligus testing. Percobaan kedua melibatkan skenario folding. Dalam skenario folding digunakan 5 fold

dengan masing-masing melibatkan 80% dataset sebagai training set dan 20% dataset sebagai test set.

Pada skenario pertama, Grammatical Evolution dengan skenario tatami memperoleh akurasi tertinggi baik pada training maupun testing. Hal ini sesuai dengan hipotesis yang diajukan.

Pada skenario kedua, grammatical evolution dengan skenario tatami dan grammatical evolution dengan skenario local separability masing-masing menunjukkan hasil yang cukup baik.

<b>Percobaan</b>		<b>GA Select</b>	<b>GE Global</b>	<b>GE Local</b>	<b>GE Multi</b>	<b>GE Tatami</b>
<b>Whole</b>	<b>Training</b>	90.00%	69.78%	90.65%	92.17%	<b>95.87%</b>
	<b>Testing</b>	90.00%	69.78%	90.65%	92.17%	<b>95.87%</b>
	<b>Total</b>	90.00%	69.78%	90.65%	92.17%	<b>95.87%</b>
<b>Fold 1 dari 5 fold</b>	<b>Training</b>	88.89%	71.82%	91.06%	93.49%	<b>95.39%</b>
	<b>Testing</b>	86.81%	67.03%	92.30%	<b>95.60%</b>	<b>95.60%</b>
	<b>Total</b>	88.47%	70.87%	91.30%	93.91%	<b>95.43%</b>
<b>Fold 2 dari 5 fold</b>	<b>Training</b>	88.89%	73.44%	<b>97.28%</b>	91.05%	95.93%
	<b>Testing</b>	92.30%	73.62%	<b>97.80%</b>	91.20%	<b>97.80%</b>
	<b>Total</b>	89.56%	73.48%	<b>97.39%</b>	91.08%	96.30%
<b>Fold 3 dari 5 fold</b>	<b>Training</b>	87.80%	73.98%	<b>94.85%</b>	93.49%	<b>94.85%</b>
	<b>Testing</b>	90.10%	71.42%	<b>95.60%</b>	93.40%	<b>95.60%</b>
	<b>Total</b>	88.26%	73.47%	<b>95.00%</b>	93.48%	<b>95.00%</b>
<b>Fold 4 dari 5 fold</b>	<b>Training</b>	89.43%	73.44%	94.03%	92.95%	<b>95.39%</b>
	<b>Testing</b>	85.71%	73.62%	94.50%	89.01%	<b>96.70%</b>
	<b>Total</b>	88.69%	73.48%	94.13%	92.17%	<b>95.65%</b>
<b>Fold 5 dari 5 fold</b>	<b>Training</b>	92.68%	74.25%	<b>98.64%</b>	96.20%	97.56%
	<b>Testing</b>	86.81%	71.43%	<b>97.80%</b>	86.81%	92.30%
	<b>Total</b>	91.52%	73.70%	<b>98.47%</b>	94.34%	96.52%

## 4.2 Data Sintesis 02

Data sintesis 02 memiliki struktur yang hampir sama dengan data sintesis 01. Namun di sini, diberikan lebih banyak kelas.

Dalam percobaan dilakukan dua jenis skenario ujicoba. Untuk ujicoba pertama, di mana data training sama dengan data testing, GE Tatami menunjukkan hasil yang sangat baik:

Sementara pada skenario kedua, dimana data dibagi menjadi 5 fold, GE tatami kembali menunjukkan hasil yang baik.

<b>Percobaan</b>		<b>GA Select</b>	<b>GE Global</b>	<b>GE Local</b>	<b>GE Multi</b>	<b>GE Tatami</b>
<b>Whole</b>	<b>Training</b>	85.80%	54.65%	83.19%	87.60%	<b>91.02%</b>
	<b>Testing</b>	85.80%	54.65%	83.19%	87.60%	<b>91.02%</b>
	<b>Total</b>	85.80%	54.65%	83.19%	87.60%	<b>91.02%</b>
<b>Fold 1 dari 5 fold</b>	<b>Training</b>	85.77%	55.08%	88.01%	87.19%	<b>93.29%</b>
	<b>Testing</b>	86.78%	57.02%	92.56%	85.95%	<b>92.56%</b>
	<b>Total</b>	85.97%	55.46%	88.91%	86.95%	<b>92.66%</b>
<b>Fold 2 dari 5 fold</b>	<b>Training</b>	85.97%	55.89%	88.21%	84.76%	<b>92.68%</b>
	<b>Testing</b>	81.82%	54.55%	88.43%	80.17%	<b>92.56%</b>
	<b>Total</b>	85.15%	55.62%	88.25%	83.84%	<b>92.66%</b>
<b>Fold 3 dari 5 fold</b>	<b>Training</b>	86.38%	55.28%	89.63%	83.13%	<b>93.29%</b>
	<b>Testing</b>	77.68%	50.41%	86.77%	82.64%	<b>95.86%</b>
	<b>Total</b>	84.67%	54.32%	89.07%	83.03%	<b>93.86%</b>
<b>Fold 4 dari 5 fold</b>	<b>Training</b>	84.96%	51.82%	87.80%	85.36%	<b>93.90%</b>
	<b>Testing</b>	85.95%	57.85%	88.43%	90.98%	<b>91.73%</b>
	<b>Total</b>	85.15%	53.61%	87.92%	86.46%	<b>93.47%</b>
<b>Fold 5 dari 5 fold</b>	<b>Training</b>	84.35%	54.67%	85.57%	86.18%	<b>93.29%</b>
	<b>Testing</b>	87.60%	56.19%	85.12%	87.60%	<b>94.21%</b>
	<b>Total</b>	84.99%	54.97%	85.48%	86.46%	<b>93.47%</b>

#### 4.3 Data Iris

Data iris merupakan dataset yang cukup banyak dipakai dalam penelitian. Data ini terdiri dari 3 kelas dan 4 atribut. Data iris bersifat multi-variate, terdiri dari 150 baris.

Pada percobaan pertama, dimana semua data digunakan untuk training sekaligus testing, GE dengan pengukuran fitness berdasarkan akurasi klasifier menunjukkan hasil terbaik (98%), disusul GE Tatami (97.3%).

Pada percobaan kedua, GE Tatami dan GE Global Fitness menunjukkan hasil terbaik.

<b>Percobaan</b>		<b>GA Select</b>	<b>GE Global</b>	<b>GE Local</b>	<b>GE Multi</b>	<b>GE Tatami</b>
<b>Whole</b>	<b>Training</b>	96.67%	<b>98.00%</b>	97.33%	97.33%	<b>98.00%</b>
	<b>Testing</b>	96.67%	<b>98.00%</b>	97.33%	97.33%	<b>98.00%</b>
	<b>Total</b>	96.67%	<b>98.00%</b>	97.33%	97.33%	<b>98.00%</b>
<b>Fold 1 dari 5 fold</b>	<b>Training</b>	96.67%	97.50%	<b>98.33%</b>	<b>98.33%</b>	96.67%
	<b>Testing</b>	90.00%	<b>100%</b>	96.67%	93.33%	<b>100%</b>
	<b>Total</b>	95.53%	<b>98.00%</b>	<b>98.00%</b>	97.33%	97.33%
<b>Fold 2 dari 5 fold</b>	<b>Training</b>	96.67%	98.33%	98.33%	<b>99.16%</b>	<b>99.16%</b>
	<b>Testing</b>	93.33%	96.67%	90.00%	<b>98.33%</b>	96.67%
	<b>Total</b>	96.00%	98.00%	96.67%	98.00%	<b>98.67%</b>
<b>Fold 3 dari 5 fold</b>	<b>Training</b>	96.67%	<b>98.33%</b>	96.67%	<b>98.33%</b>	97.50%
	<b>Testing</b>	96.67%	96.67%	<b>100%</b>	93.33%	93.33%
	<b>Total</b>	96.67%	<b>98.00%</b>	97.33%	97.33%	96.67%
<b>Fold 4 dari 5 fold</b>	<b>Training</b>	96.67%	98.33%	97.50%	97.50%	<b>98.33%</b>
	<b>Testing</b>	96.67%	96.67%	96.67%	96.67%	<b>100%</b>
	<b>Total</b>	96.67%	98.00%	97.33%	97.33%	<b>98.67%</b>
<b>Fold 5 dari 5 fold</b>	<b>Training</b>	96.67%	<b>98.33%</b>	96.67%	97.50%	97.50%
	<b>Testing</b>	93.33%	96.67%	96.67%	96.67%	<b>100%</b>
	<b>Total</b>	96.00%	<b>98.00%</b>	96.67%	97.33%	<b>98.00%</b>

#### **4.4 Data E-Coli**

Data E-Coli merupakan dataset yang cukup banyak dipakai dalam penelitian. Data ini terdiri dari

**BAB 5**  
**KESIMPULAN DAN SARAN**

## DAFTAR PUSTAKA

- [1] Gunawan G. F., Gosaria S, Arifin A. Z. (2012). “*Grammatical Evolution For Feature Extraction In Local Thresholding Problem*”, Jurnal Ilmu Komputer dan Informasi, Vol 5, No 2 (2012)
- [2] Harper R., Blair A. (2006). “*Dynamically Define Functions in Grammatical Evolution*”, IEEE Congress of Evolutionary Computation, July 16-21, 2006
- [3] Gavrilis D., Tsoulous I. G., Georgoulas G., Glavas E. (2005). “*Classification of Fetal Heart Rate Using Grammatical Evolution*”, IEEE Workshop on Signal Processing Systems Design and Implementation, 2005.
- [4] Gavrilis D., Tsoulous I. G., Dermatas E. (2008). “*Selecting and Constructing Features Using Grammatical Evolution*”, Journal Pattern Recognition Letters Volume 29 Issue 9, July, 2008 Pages 1358-1365 .
- [5] Guo L., Rivero D., Dorado J., Munteanu C. R., Pazos A. (2011). “*Automatic feature extraction using genetic programming: An application to epileptic EEG classification* ”, Expert Systems with Applications 38 Pages 10425-10436
- [6] Li B., Zhang P.Y., Tian H., Mi S.S., Liu D.S., Ruo G.Q. (2011). “*A new feature extraction and selection scheme for hybrid fault diagnosis of gearbox*”, Expert Systems with Applications 38 Pages 10000-10009