Genetic Programming for Medicinal Plant Family Identification System

Indra Laksmana[1], Yeni Herdiyeni[2] & Ervizal A.M. Zuhud[3]

[1]Major of Agricultural Machinery and Equipment, Payakumbuh Agricultural
Polytechnic, Jalan Raya Negara Km 7, Tanjung Pati, Payakumbuh 26271, Indonesia
[2]Department of Computer Science, Faculty of Mathematics and Natural Sciences
Bogor Agricultural University, Darmaga Campus,
Jalan Meranti, Wing 20 Level 5, Bogor 16680, Indonesia
[3]Department of Forest Resources Conservation and Ecotourism, Faculty of Forestry,
Bogor Agricultural University, Darmaga Campus,
Jalan Lingkar Akademik, Bogor 16680, Indonesia
Email: indra@politanipyk.ac.id

Abstract. Information about medicinal plants that is available in text documents is generally quite easy to access, however, one needs some efforts to use it. This research was aimed at utilizing crucial information taken from a text document to identify the family of several species of medicinal plants using a heuristic approach, i.e. genetic programming. Each of the species has its unique features. The genetic program puts the characteristics or special features of each family into a tree form. There are a number of processes involved in the investigated method, i.e. data acquisition, booleanization, grouping of training and test data, evaluation, and analysis. The genetic program uses a training process to select the best individual, initializes a generate-rule process to create several individuals and then executes a fitness evaluation. The next procedure is a genetic operation process, which consists of tournament selection to choose the best individual based on a fitness value, the crossover operation and the mutation operation. These operations have the purpose of complementing the individual. The best individual acquired is the expected solution, which is a rule for classifying medicinal plants. This process produced three rules, one for each plant family, displaying a feature structure that distinguishes each of the families from each other. The genetic program then used these rules to identify the medicinal plants, achieving an average accuracy of 86.47%.

Keywords: *booleanization; fitness evaluation; function set; genetic programming; medicinal plant families; terminal set.*

1 Introduction

As one of the most sophisticated tropical countries, Indonesia has rich and diverse natural resources,among which more than 38,000 species of plants [1]. Groombridge and Jenkins [2] have recorded as many as 22,500 medicinal plants that are dispersed broadly in Indonesia and of which only 4.4 percent are used

Received August 7th, 2013, 1st Revision November 24th, 2013, 2nd Revision December 6th, 2013.

by local people. One of the reasons is lack of information and too little knowledge about the potency of the medicinal plants around them. People can try to classify medicinal plants manually, for example by using a herbarium or textual information in the form of documents, papers, and other literature. All of the information contained in text documents about botany, ecology, distribution, cultivation, benefits, chemical content and many other subjects, is abundant, which naturally hinders manual classification. This process takes very long and needs a certain amount of comprehension, which will surely hamper people to identify medicinal plants, for example determining the family a plant belongs to. Thus, there is a need for a certain set of rules, structure or system to accommodate and accelerate medicinal plant family identification using an heuristic approach.

Recent scholars have implemented heuristic methods for many aspects of life. Stadler [3], for example, used a heuristic method to make databases more structured and then form them into a tree or graph form. Every node represents a document, while one node is linked with another by a labeled edge, thus connecting documents witha similar value. Yuningsih [4] applied a heuristic method using a genetic algorithm (GA) for an image seeking process, which was proven 8.89 times faster than a non-heuristic implementation.

Genetic programming (GP) is the development of a GA that acts as a heuristic search engine based on the biological evolution mechanism. Walker [5] explains that GP is high-accuracy programming, which makes the computer more intelligent and able to solve problems automatically. Yuan, *et al*. [6] have compared the GP method with several other methods *(Rank Boost*, *BM25*, *Rank-SVM*) and suggested the use of GP to handle information retrieval problems. GP automatically makes a ranking that defines the level of relevance to the query in order to make the information more suitable to user need.

This research was aimed at acquiring important taxonomic information that is contained in a text document, implementing GP to determine the best structure or set of rules for identifying medicinal plants based on their characteristics. The objective was to help make this identification process faster, easier and more structured in order for people to recognize medicinal plants around them and make them aware of the special features of the plant families.

2 Methods

The data used in this research were retrieved from an Indonesian language text document. This document contains information regarding types of medicinal plants and is owned by the Plant Diversity Conservation Section, Department of Forest Resources Conservation and Eco-tourism, Faculty of Forestry Bogor

Agricultural Institute. The document is actually a book, entitled *A General Guide to Indonesian Medicinal Plants*, Volume I-X, composed by the Faculty of Forestry, Bogor Agricultural Institute, in cooperation with the Faculty of Forestry, University of Gadjah Mada. The collected data cover 81 species of three families of medicinal plants, i.e. 26 species of the Lamiaceae family, 24 species of the Apiaceae family, and 31 species of the Euphorbiaceae family.

The stages in which the research was conducted are shown in Figure 1.

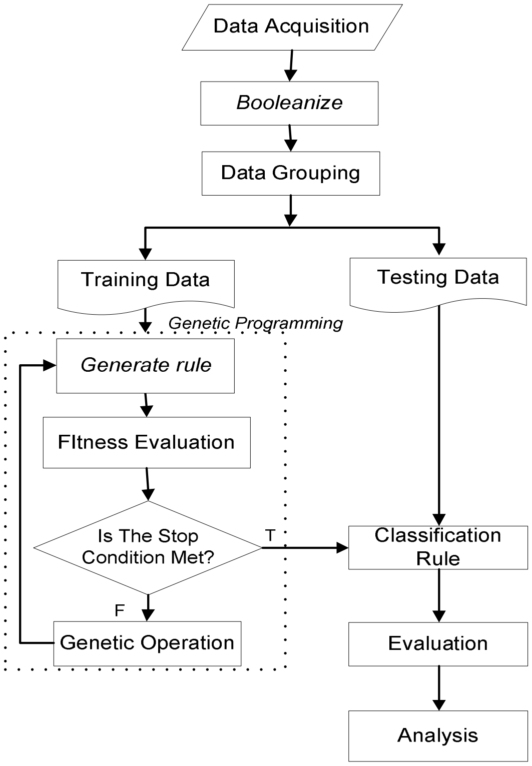


Figure 1 Phases of the research.

The plant families that were used are very similar to each other, share identical characteristics, and have the same morphological traits (i.e. plants with flowers). Furthermore, as can be seen from the number of species found in the document, the three families have many species. The 81 species that were chosen from the three families are listed in Table 1.

Table 1 The species.

Species *Lamiaceae*	**Species *Apiaceae***	**Species *Euphorbiaceae***
Coleus amboinicus	*Centellaasiatica*	*Acalyphaaustralis*
Leonurussibiricus	*Apiumgraveolens*	*Jatrophapodagrica*
Hyptissuaveolens	*Foeniculumvulgare*	*Jatrophagossypifolia*
Coleus scutellarioides	*Eryngiumfoetidum*	*Euphorbia pulcherrima*
Coleus tuberosus	*Hydrocotylesibthorpioides*	*Codiacumvariegatum*
Leucaslavandulifolia	*Daucuscarota*	*Excoecariacochinchinensis*
Rosmarinusofficianalis	*Coriandrumsativum*	*Euphorbia plumerioides*
Salvia coccinea	*Pimpinella alpine*	*Acalyphawilkesiana*
Salvia splendens	*Trachyspermumammi*	*Antidesmabunius*
Orthosiphonaristatus	*Petrosolinumcrispum*	*Croton tiglium*
Ajugareptans	*Carumroxburghianum*	*Bridelia ovate*
Ocimumbasilicum	*Pimpinellaanisum*	*Glochidionrubrum*
Hyptisbrevipes	*Carumcopticum*	*Acalyphahispida*
Ocimum sanctum	*Eryngiumbromeliaefolium*	*Richinuscommunis*
Pogostemon cabin	*Eryngiumfoetidum*	*Baccaurearacemosa*
Menthe arvensis	*Cuminumcyminum*	*Acalyphaindiaca*
Thymus serpylum	*Pimpinellasaxifraga*	*Euphorbia tiraculli*
Thymus vulgaris	*Aegopodiumpodagraria*	*Acalyphamicrophylla*
Mesonapalustris	*Angelica sylvestris*	*Phyllanthusacidus*
Clerodendrumpaniculatum	*Anthriscussylvestris*	*Jatrophacurcas*
Lavandulaofficinalis	*Chaerophyllumtemulentum*	*Aleuritesmoluccana*
Gomphostemmajavanicum	*Heracleumsphondylium*	*Euphorbia milli*
Menthaarvensis	*Meumathamanticum*	*Pedilanthustithymaloides*
Menthapulegium	*Torilis japonica*	*Euphorbia antiquorum*
PogostemonHeyneanus		*Sauropusandrogynus*
Pogostemonhortensis		*Phyllanthusemblica*
		Manihotutilissima
		Phyllanthusreticulatus
		Phyllanthusniruri
		Euphorbia prostata
		Euphorbia hirta

2.1 Booleanization

Booleanization is a process that consists of an attribute coding process. In this research, the attributes were adopted from a selection process of classifiers with regards to the morphological characteristics of the medicinal plants mentioned in the document. Based on 8 physical aspects, such as habitus, leaves, stem, flower, fruit, root, aromatic, and habitat, 63 classifier attributes were chosen that were coded as *X0* to *X62*. The information about each of the species was then transformed into binary values (0 and 1) based on the classifier attributes. The number 0 indicates that there is no such characteristic in the species and the number 1 indicates that the species has the characteristic mentioned in the document. An example of the booleanization process is shown in Table 2.

Table 2 Habitus booleanization.

Habitus	**X0**	**X1**	**X2**	**X3**
Herb/terna	1	0	0	0
Bush/clump	0	1	0	0
Liana	0	0	1	0
Tree	0	0	0	1

Table 3 Attribute booleanization.

Physical Aspect	**Sub Section**	**Attribute Coding**
Habitus		Herb/terna (X0), Bush/clump(X1), Liana (X2), Tree (X3).
	Furry/hairy	Available/Unavailable (X4)
Leaf	Layout	Crossing (X5), Frontal (X6), Round (X7)
	Composition	Singular (X8), Compound (X9)
	Edgeshape	Sharp-pointed (X10), Taper (X11), Circling (X12), Split (X13), Blunt (X14)
	Border shape	Flat (X15), Jaggy (X16), Wavy (X17)
	Shape	Round/Ovoid (X18), Lancet/Ellipse/Loose(X19), Triangle (X20), Needle-shaped (X21), Linear (X22), Finger-shaped(X23)
	Skeleton	Pinnate (X24), Finger-Shaped(X25), Curved (X26), Parallel (X27)
	Supporter	Available/Unavailable (X28)
Stem	Branch	Monopodia (X29), Simpodia (X30)
	Branch shape	Plagiotrophic (X31), Orthotropic (X32)
	Skin inside	White sap (X33), Yellow sap (X34), Red sap(X35), Black sap (X36)
	Leaf footprint	Available/Unavailable (X37)
	Inner cavity	Available/Unavailable (X38)
Flower	Equipment	Complete flower (X39), Incomplete Flower (X40)
	Layout	From the leaf’sbottom edge(X41), Tipof the branch/ stem (X42), Stem/big branch (X43)
	Composition	Limited compound (X44), Unlimited compound (X45)
	Shape	Grain (X46), Panicle (X47), Umbrella (X48)
Fruit	Composition	Single fruit (X49), Compound Fruit (X50)
Habitat	Plant substract	Watery (X51), Muddy (X52), Humid (X53), Dry (X54)
	Living characteristic	Tolerant (X55), Intolerant (X56)
	Living method	Swamp (X57), Scattered(X58)
Root	Shape	Fiber (X59), Taproot (X60)
	Tuber	Available/Unavailable (X61)
Aromatic		Available/Unavailable (X62)

In the booleanization process, species with liana as habitus were transferred to 0 0 1 0. The habitus value is 0 for herb/ternain the X0 position, 0 for bush/clump in the X1 position, and 0 for tree in the X3 position, while the value is 1 in the X2 position. This means that the habitus of the species is liana. 63 attributes were booleanized, whose attributes are shown in Table 3.

2.2 Data Grouping

Initially, the data are grouped into K groups which should have the same size (having equal number of members), and then will be divided into training data and test data. If the total data (N) doesn't exactly divided by the K ($N \ mod \ K \neq 0$), then the last data group (K-1) will have more data than the other groups (K). The process is repeated for K iterations. At the next iteration, part K becomes the test data, while part K-1 is used as the training data [7].

The medicinal plant data taken from the document were divided according to family with an 80-20 proportion. The data grouping was done with K = 5. The data were separated into five equal parts. Then the training data and the test data were divided. Four subsets of training data were used as the training input in the classifying process and one subset of test data was used to examine the model of the training result. The scenario of data grouping is shown in Table 4 and 5.

Table 4 Data grouping scenario.

Fold	**Data**	**Subset**
***Fold* 1**	Training Data Test data	S_1, S_2, S_3, S_4 S_5
***Fold* 2**	Training Data Test data	S_1, S_2, S_3, S_5 S_4
***Fold* 3**	Training Data Test data	S_1, S_2, S_4, S_5 S_3
***Fold* 4**	Training Data Test data	S_1, S_3, S_4, S_5 S_2
***Fold* 5**	Training Data Test data	S_2, S_3, S_4, S_5 S_1

Table 5 Data grouping per family.

Family	**S1**	**S2**	**S3**	**S4**	**S5**	**Total**
Lamiaceae	5	5	5	5	6	26
Apiaceae	5	5	5	5	4	24
Euphorbiaceae	6	6	6	6	7	31
Total	16	16	16	16	17	81

2.3 Genetic Programming

Genetic programming was introduced for the first time by John R. Koza, having been inspired by the ideas of John Holland, who created the genetic algorithm (GA) in 1975, based on Charles Darwin's theory of evolution. In 1992, Koza applied a genetic algorithm to create a system or computer program that was able to do its own programming (*automatic programming)*. This method is named genetic programming [8]. Koza used Genetic programming within a computer program to produce a draft scheme using Lisp computer language as its solution [9].

Genetic programming is a development of the genetic algorithm approach that constitutes a heuristic search algorithm. It is based on a natural system and mechanism, i.e. genetics and natural selection. Each solution variable within a genetic program is coded into a string structure representing a gen row, which is a characteristic of the solution. This association is known as the population. All of the individuals within the population are a representation of the solution. A part of the individuals is called a 'chromosome'. Chromosomes evolve in a continuous iteration process called 'generation'. In each of the generations, the individuals are evaluated based on an evaluation function and ultimately the generations inside the genetic program will converge towards the best individual, which is expected to be the optimal solution. According to Poli, *et al*. [10], genetic programming is an evolution of computing in which problems are automatically solved without telling the computer in detail what it must do by deciding the solution shape or structure at the beginning of the program. Genetic programming is more dynamic compared to genetic algorithms.

The individuals in the genetic program used in this research are the representation model or document hierarchy based on the respective attributes of the medicinal plant families. The population is a group of randomly formed rules. Every rule is evaluated with regards to a certain fitness criterion. The primitive form of a genetic program is a compilation of functions (function set). The function set used in this research is a compilation of `AND, OR, NOR,` and some arguments (*terminal set*), which is the booleanization result. The process that follows next is described below.

2.3.1 Generate-rule Process

The generate-rule or initialization phase of each generation is the process of creating a set of individuals. An individual consists of a function set and a terminal set that are randomly generated, followed by a provision of limitation to a specific tree depth and number of nodes. One individual describes one model form or rule that is being sought. An example of a model form or rule is shown in Figure 2.

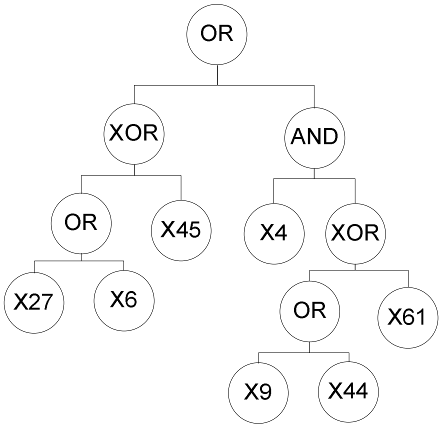


Figure 2 Example of model form or identification rule.

2.3.2 Fitness Evaluation

Fitness evaluation is conducted by counting the total number of individual errors made by the program in predicting one of two classes (*true* or *false*). The expected class is valued as 1 (true) or 0 (false). The lower the fitness value, the lower the amount of falses a single individual has, and the better the created individual is. In this study, the fitness value was acquired from testing the booleanization data in the rule or individual created.

2.3.3 Genetic Operation

Three genetic operators were used in the genetic program, i.e. elitism, crossover, and mutation [11]. The process of genetic operation started with the selection of rules using the tournament method. The winner of the tournament is the individual that occupies the lowest fitness value. Subsequently, the operations of elitism, crossover, and mutation were executed. The elitism operation is the process of copying the winning individual of the tournament into a new population or generation. The crossover operation is an exchange of some parts of the tree structure (gen) from two individuals (parents) with a randomly chosen cross-point. An illustration of crossover is shown in Figure 3. The mutation operation randomly chooses a part of the tree structure in an individual (chromosome) and replaces that part with a function set or terminal set that is adjusted to the selected part. An example of the mutation process is shown in Figure 4.

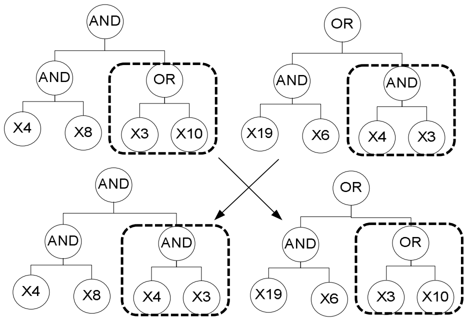


Figure 3 An example of crossover.

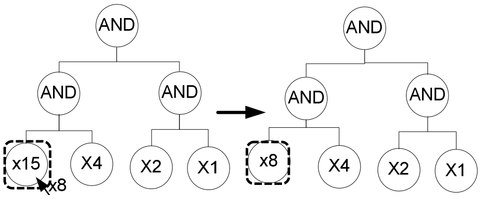


Figure 4 An example of mutation.

2.3.4 The Stop Condition

This process is applied repetitively during the generate-rule phase until it reaches the maximum number of generations. This is set as the stop condition in the genetic program.

3 Results and Discussions

The data of 81 species from three plant families, each having been coded by a binary number (booleanized), were divided into two groups, the training data and the test data, using a 5-fold cross-validation. The training process implemented by the genetic program produces a model form or classification rule for each of the families, giving value 1 to the sought class and 0 to the other classes. The parameters used in the training process are shown in Table 6.

Table 6 Values for *genetic programming* operation.

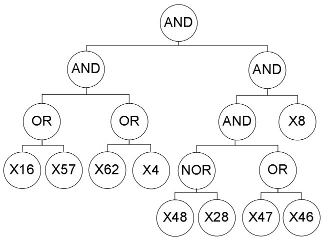
Category	Amount
Number of generations	10, 20, 25
Population size	10 000, 50 000, 100 000
Crossover	0.9
Mutation	0.1
Depth of the tree	5
Maximum node	18, 20, 24

The process was carried out repetitively, based on the parameters above, until the best rule was obtained, which was at the 10th generation with a population of 10,000 and maximum number of nodes 24. Evaluation is conducted by comparing the results of the prediction for each class using a confusion matrix [12]. The genetic program produced three classification rules in a tree form, i.e. the rules for the Lamiaceae, the Apiaceae, and the Euphorbiaceae family. These rules show the characteristic structures that distinguish each of the families.

3.1 Lamiaceae Family

The genetic programming of the 26 species of the Lamiaceae family resulted in nine classifier attributes that were chosen, with three combinations of three operators, i.e. AND, OR and NOR. The produced rule could distinguish the Lamiaceae from the Apiaceae and the Euphorbiaceae family. The rule that was generated is shown in Table 7.

Table 7 Lamiaceae family rule.

Rule	***Boolean***
If (leaf composition is “Single” X8 **and**((flower is “grain-shaped” X46 **or** “panicle” X47)**and** (**doesn’t** “has leaf supporter”X28 **or not formed as flower with** “has umbrella-shape” X48))) **and** ((“Furry”X4**or** “Aromatic”X62) **and** (living method “Swamp” X57**or** leaf border “Jaggy” X16)) **Then**the family is Lamiaceae	 AND AND AND OR OR AND X8 X16 X57 X62 X4 NOR OR X48 X28 X47 X46

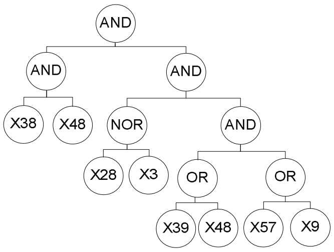
The importance level of each Lamiceace family classifier is based on its operator level. The first and the second level have the sample operator AND, and therefore it will be valued true if two of the inputs below it are true. The X8 classifier (single leaf composition) is a very crucial classifier for the Lamiaceae family, due to the fact that this feature is one of the inputs from the operator AND on the second level. On the third level there are OR and AND operators. The OR operator will be valued *true* if one of the inputs is true, i.e. if one of the classifiers X16 (jaggy leaf border), X57 (swamp living method) is true and one of the classifiers X62 (aromatic), X4 (furry) is also true. On the fourth level, there are NOR and OR operators. The NOR operator is valued *true* if two of its inputs are false, which means that the classifiers X48 (umbrella-shaped flower) and X28 (has leaf supporter) are not possessed by the Lamiacea family. At the OR operator, one of the two classifiers X47 (panicle flower) or X46 (grain flower) must be true, because this operator has input from an AND operator.

The classifiers produced by the genetic program were not completely identical to the classifiers according to [13] and [14]. There were four classifiers representing the nine classifiers produced by the GP process, i.e. furry, aromatic, single leaf, and without leaf supporter. Apart from the number of data and the equipment possessed by the classifier, this difference was inflicted by the function, which was set or was used by the operator, in this case a boolean function (AND, OR, NOR). The boolean function eliminated the excessive classifier attributes, so that the genetic program generated classification rule structure with less classifiers to distinguish the Lamiaceae family from the Apiaceae and the Euphorbiaceae family.

3.2 The Apiaceae Family

The genetic programming of the 24 species of the Apiaceae family resulted in seven classifier attributes that were chosen, with three combinations of the AND, OR, and NOR operators with which this family can be distinguished from the Lamiaceae and the Euphorbiaceae family. The rule that was generated is shown in Table 8.

Table 8 Apiaceae family rule.

Rule	***Boolean***
If (flower shapeis “Umbrella”X48**and** the stem is “Inner cavity”X38) **and** ((habitusis “Tree” X3**or not**“owns a leaf supporter”X28) **and** ((composition of leaf is “Compound”X9**or** life method is “Swamp”X57)**and** (flower shape is “Umbrella”X48**or** “Complete flower”X39))) **Then** the family is Apiaceae	 AND AND AND X38 X48 NOR AND X28 X3 OR OR X39 X48 X57 X9

The importance level of each Apiaceae family classifier is based on its operator level. The first and the second level have the same operator (AND), which means it will be true if the two inputs below it are both true. The X38 classifier (inner cavity) and the X48 classifier (umbrella-shaped flower) are the most important classifiers for the Apiaceae family, because they are among the inputs from the AND operator on the second level. On the third level, there are the operators NOR and AND. The NOR operator is true if two of its inputs are false, which means that the X28 (has leaf supporter) and the X3 (tree habitus) do not characterize the Apiaceae family. The AND operator has input from the two OR operators on the fourth level, which means that one of the classifiers X48 (umbrella-shaped flower) and X39 (perfect flower) must be true and one of the classifiers X57 (swamp living method) and X9 (compound flower composition) must also be true.

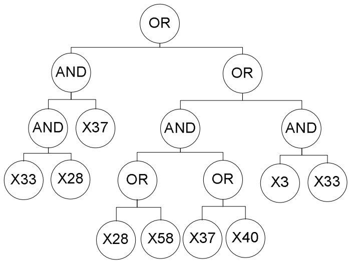
The generated classifiers were not completely identical to the classifiers according to [13] and [15]. There were five classifiers representing the seven classifiers produced by the GP process, i.e. inner cavity, compound leaf composition, no leaf supporter, umbrella-shaped flower and perfect flower (hermaphrodite flower). Apart from the number of data and the equipment possessed by the classifier, this difference was due to fact that the function set or operator used was a boolean function. The boolean function eliminated the excessive classifier attributes, so that the GP couldgenerate a classification rule structure with fewer classifiers to distinguish this family from the Lamiaceae and the Euphorbiaceae family.

3.3 The Euphorbiaceae Family

The genetic programming of the 31 species of the Euphorbiaceae family resulted in 9 (nine) classifiers with a combination of two operators, AND and OR, which can distinguish this family from the Lamiaceae and the Apiaceae family. The rule that was generated is shown in Table 9.

The importance level of each Euphorbiaceae family classifier can be seen from its operator level. On the first level, there is an OR operator, which means that it is valued *true* if one of the inputs below it (OR and AND) are true. The initial observation is implemented into the AND operator on the second level. If the classifiers X37 (leaf footprint), X33 (white sap) and X28 (leaf supporter) are true, there is no need to trace back to the OR operator inputs on level two. In contrast, if one of the three classifiers is false, then there should be a tracing back to the other inputs of the OR operator on level two.

Table 9 Euphorbiaceae family rule.

<table>
<tr><th>Rule</th><th><b><i>Boolean</i></b></th></tr>
<tr><td><b>If</b>
((“leaf print observed”X37) <b>and</b> (“has leaf supporter”X28 <b>and</b> “has white sap”X33))
<b>or</b>
((“has white sap”X33 <b>and</b> habitus “tree”X3) <b>or</b> ((“incomplete flower”X40<b>or</b> “leaf print observed”X37) <b>and</b> (life method “Scattered”X58<b>or</b>“has leaf supporter”X28)))
<b>Then</b> thefamily is<i>Euphorbiaceae</i></td><td>
</td></tr>
</table>

The OR operator on level two gets input from the AND operator, which means that there should be no tracing back to both of the inputs if one of the operators is true. In contrast, if one of the classifiers X33 (has white sap) and X3 (habitus tree) is false, then there should be a tracing back to the other inputs of the AND operator on level three. This AND operator has input from two OR operators, meaning that one of the classifiers X28 (has leaf supporter) and X58 (scattered life method) must be true and one of the classifiers X37 (print leaf) and X40 (incomplete flower) must also be true.

The classifiers produced were not completely identical with the classifiers according to [13] and [15]. There were four classifiers that likely represented the six classifiers produced by the GP process, i.e. tree habitus, having leaf supporter, incomplete flower (most likely single sex) and white sap (contains sap). Apart from the number of data and the equipment possessed by the classifier, this difference is due to fact that the function set or operator used was a boolean function. The boolean function eliminated the excessive classifier attributes, so that the GP could generate a classification rule structure with fewer classifiers to distinguish this family from the Lamiaceae and the Apiaceae family.

3.4 Evaluation

A total of 81 species of medicinal plants belonging to three plant families were divided into training data and test data. Fold 1 contained training data of 64 species and test data of 17 species. The results of Fold 1 are shown in Table 10.

Table 10 Confusion matrix of Fold 1.

Fold 1		***Actual Class***			
		Lamiaceae	**Apiaceae**	**Euphorbiaceae**	**Other Families**
Predicted Class	Lamiaceae	6	0	0	0
	Apiaceae	0	3	0	0
	Euphorbiaceae	0	0	7	0
	Other families	0	1	0	0

The data of the 17 species that were taken as test data originated from the Lamiaceae family (6), the Apiaceae family (4), and the Euphorbiacea family (7). Table 10 shows that there was one failure in the classification activity. The failure occurred due to the fact that one of the important classifiers referred to in the rule gained is not possessed by this species. This classifier was equipped with the operator AND, which implies that both of its inputs must be true. Inner cavity and (AND) umbrella-shaped flower states that these two classifiers must exist in the species, while the *Eryngiumfoetidum* has a grain-shaped flower.

Referring to [13] and [15], the Apiaceae family also has the characteristics of a crossing leaf layout and seldom in face-to-face position. Both of these classifiers were not equipped with AND, OR, and NOR operators, which made it unclear how important these classifiers are for the Apiaceae family. They were not found in any classification rule generated by the GP, so that the accuracy of the classification result in Fold 1 was 94.11%. The calculation was executed as follows:

$$\text{Accuracy of fold 1} = \frac{6+3+7}{17} \text{x } 100\% = 94.11\%$$

Table 11 Confusion matrix of Fold 2.

Fold 2		***Actual Class***			
		Lamiaceae	**Apiaceae**	**Euphorbiaceae**	**Other Families**
Predicted Class	Lamiaceae	3	0	1	0
	Apiaceae	0	5	0	0
	Euphorbiaceae	1	0	5	0
	Other families	1	0	0	0

The test data from Fold 2 covered16 species, coming from the Lamiaceae family (5), the Apiaceae family (5), and the Euphorbiaceae family (6). Table 11 shows that there were three failures in the classifying process. They occurred in the Lamiaceae family (the species *Menthe ardencies* and *Thymus vulgaris*) and the Euphorbiaceae family (species *Aleuritesmoluccana*). The species *Menthe arvensis* was not identified because it did not obtain an important classifier from the three rules produced. The *Thymus vulgaris* was classified as belonging to the Euphorbiaceae family, while the species *Aleuritesmoluccana* from the Euphorbiaceae family was classified as belonging to the *Lamiaceae* family. The cause of the failure was that the aromatic classifier is owned by the *Aleuritesmoluccana* species and not by the *Thymus vulgaris* species. This failure was influenced by the AND operator, which states that two of the input classifiers (classifier attributes) have to be owned by its species. The accuracy of the identification results from Fold 2 was 81.25%. The calculation was executed as follows:

$$\text{Accuracy of fold 2} = \frac{3+5+5}{16} \times 100\% = 81.25\%$$

Table 12 Confusion matrix of Fold 3.

Fold 3		***Actual Class***			
		Lamiaceae	**Apiaceae**	**Euphorbiaceae**	**Other Families**
Predicted Class	Lamiaceae	4	0	0	0
	Apiaceae	0	5	0	0
	Euphorbiaceae	1	0	5	0
	Other families	0	0	1	0

The test data in Fold 3 covered 16 species, originating from the Lamiaceae family (5), the Apiaceae family (5), and the Euphorbiaceae family (6). The classification results from Fold 3 are shown in Table 12. There were two classification failures, which occurred in the Lamiaceae family (*Ocinum sanctum* species) and the Euphorbiaceae family (*Euphorbia tiraculli* species). The failures in the classifying process happened because the *Euphorbiatiraculli* species does not have a leaf supporter, while the OR operator states that one of two classifiers must be owned by its species. Apart from a leaf supporter, this species also has a complete flower, not a tree habitus, and does not have a leaf footprint, which made it not identified as belonging to the Euphorbiaceae family. This species did not have an important classifier in any of the three rules produced. The *Ocimum sanctum* species is considered a false identification, because it has important characteristics of the rules generated, so that it was identified as belonging to two families, i.e. the Lamiaceae and the Euphorbiaceae family. The accuracy of the identification resultsof Fold 3 was 87.50%. The calculation was executed as follows:

$$\text{Accuracy of fold 3} = \frac{5+5+4}{16} \times 100\% = 87.50\%$$

Table 13 Confusion matrix Fold 4.

Fold 4		***Actual Class***			
		Lamiaceae	**Apiaceae**	**Euphorbiaceae**	**Other Families**
Predicted Class	Lamiaceae	4	0	0	0
	Apiaceae	0	5	0	0
	Euphorbiaceae	1	0	5	0
	Other families	0	0	1	0

The test data in Fold 4 covered 16 species, originating from the Lamiaceae family (5), the Apiaceae family (5), and the Euphorbiaceae family (6). The classification results from Fold 4 are shown in Table 13. There was a failure in identifying the Rosmarinusofficianalis and the Croton tiglium species. This failure occurred because the Croton tiglium species does not have white sap equipped with the OR operator. Like the Euphorbiaceae family, this species does not have a tree habitus or white sap, so that it didnot have an important classifier in any of the three rules created. The Rosmarinusofficianalis species is part of the Lamiaceae family, but was identified as belonging to the Euphorbiaceae family, because the Rosmarinusofficianalis species has white sap like the Euphorbiaceae family rule states. The accuracy of the identification result from Fold 4 was87.50%. The calculation was executed as follows:

$$\text{Accuracy of fold 4} = \frac{4+5+5}{16} \times 100\% = 87,50\%$$

Table 14 Confusion matrix Fold 5.

Fold 5		*Actual Class*			
		Lamiaceae	**Apiaceae**	**Euphorbiaceae**	**Non-Three Families**
Predicted Class	Lamiaceae	3	0	0	0
	Apiaceae	0	5	0	0
	Euphorbiaceae	1	0	5	0
	Other families	1	0	1	0

The test data in Fold 5 covered 16 species, originating from the Lamiaceae family (5), the Apiaceae family (5), and the Euphorbiaceae family (6). The classification results from Fold 5 are shown in Table 14. A failure occurred in identifying the *Coleus amboinicus,* the *Leonurussibiricus* and the *Jatrophagossypifolia* species. This happened because the *Coleus amboinicus* species was classified as possessing a leaf foot print by the rule produced with the NOR operator, where this operator is true if there is no input (no classifier attributes). The *Jatrophagossypifolias* species does not have a leaf supporter, like the rule for the Euphorbiaceae family states.

The rule produced was using the AND operator, so the two species do not have an important classifier in any of the three rules produced. The *Leonurussibiricus* species is part of the Lamiaceae family, but was identified as belonging to the Euphorbiaceae family, because this species has a leaf footprint, which is an important classifier for the Euphorbiaceae family. The accuracy of the identification results from Fold 5 was 81.25%. The calculation was made as follows:

$$\text{Accuracy of fold 5} = \frac{3+5+5}{16} \times 100\% = 81.25\%$$

The evaluation of the system's performance was determined by the average value of the accuracy numbers from all folds, which was 86.32%. The calculation was made as follows:

$$\text{Accuracy} = \frac{94.11+81.25+87.50+87.50+81.25}{5} \times 100\% = 86.32\%$$

4 Conclusion

A genetic program using booleanization was used to describe the Lamiaceae, the Apiaceae, and the Euphorbiaceae family characteristics in order to identify these respective medicinal plant families. The program was implemented repetitively until a regulation or rule in the best tree form was obtained. The experiment began from the change of the generation until reaching the numbers of maximum node and the depth tree. Node selection, consisting of operators (*function set*) and attributes (*terminal set*), was carried out randomly for each generate-rule, crossover and mutation iteration. The operators used were AND, OR, and NOR (booleanization), which may eliminate excessive classifier attributes.

The rules or regulations generated by the genetic program identified plants from the Lamiaceae, the Apiaceae and the Euphorbiaceae family with an average accuracy level of 86.32%. People can utilize the hierarchy generated by the genetic program to recognize important classifiers of each family in order to identify the family of these medicinal plants.

References

[1] Bappenas, *Indonesia Biodiversity and Action Plan 2003-2020*, Jakarta, http://www.bappenas.go.id/node/82/406/strategi-dan-rencana-aksi-keane karagaman-hayati-indonesia--indonesian-biodiversity-strategy-and-action -plan---ibsap-nglish-version/ (7 December 2012).

[2] Groombridge, B., & Jenkins, M., *World Atlas of Biodiversity, Earth's Living Resources in the 21st Century*, Berkeley University of California Press, https://archive.org/details/worldatlasofbiod02groo (10 September 2012).

[3] Stadler, P.F., *Fitness Landscape*, https://www.bioinf.uni-leipzig.de/~studla/Publications/PREPRINTS/01-pfs-004.pdf (15 September 2012).

[4] Yuningsih, F., *Image Searching Using Heuristic Method for Image Retrieval System* (Text in Indonesian), Master Thesis, Department of Computer Science, Faculty of Mathematics and Natural Sciences, Bogor Agricultural University, Bogor, 2009.

[5] Walker, M., *Introduction to Genetic Programming*, http://www.cs.montana.edu/~bwall/cs580/introduction_to_gp.pdf (9 September 2012).

[6] Yuan, J.Y., Lin, J.Y., RenKe, H. & Yang, W.P., *Learning to Rank for Information Retrieval Using Genetic Programming*, http://www2.sdufe.edu.cn/wangsq/teaching/2012-09_CI/readings6-RankGP.pdf (13 February 2012).

[7] Bramer, M., *Principles of Data Mining*, London, Springer, 2007.

[8] Lukas, I.A., *The Ending of A King-Rook-King (KRK) Chess Gameby Using Genetic Programming* (Text in Indonesian), National Conference of System and Informatics, November 15th, 2008, Bali, Indonesia KNS, pp. 328-334, 2008.

[9] Koza, J.R., *Genetic Programming On the Programming of Computers by Means of Natural Selection*, London, MIT Press, 1992.

[10] Poli, R., Langdon W.B. & McPhee, N.F., *A Field Guide to Genetic Programming*, Creative Commons, 2008.

[11] Carvalho, M.G., Laender, A.H., Goncalves, M.A. & Silvia, A.S., *A Genetic Programming Approach to Record Deduplication*, IEEE Transactions on Knowledge and Data Engineering, **24**(3), pp. 399-412, 2012.

[12] Davis, J. & Goadrich, M., *The Relationship Between Precision-Recall and ROC Curves, International Conference on Machine Learning*. Appearing in Proceedings of the 23rd International Conference on Machine Learning, pp. 233-240, 2006.

[13] Keng, H., *Orders and Family of Malayan Seed Plants*, Singapore University Press, 1978.

[14] Balgooy, M.M.J.V., *Malesian Seed Plants*, *Portraits of Non-Tree Families, 3nd ed*, National Herbarium Nederland-University Leiden Branch, 2001.

[15] Tjitrosoepomo, G., *Morphology of Plants* (Text in Indonesian), Yogyakarta, Indonesia: UGM University Press, 2009.