International Review on Modelling and Simulations (I.RE.MO.S.), Vol. xx, n. x
June 2008

Rule-Based and Genetic Algorithm for Automatic *Gamelan* Music Composition

Khafiizh Hastuti[1,2], Azhari[1], Aina Musdholifah[1], Rahayu Supanggah[3]

Abstract – *This research aims to develop a system of automatic gamelan music composition. Gamelan is the traditional ensemble music of Java, Indonesia. The authors propose a model of automatic gamelan music composition which consists of knowledge, rules, and random generation. There are three types of knowledge, basic, construction, and melodic knowledge. The basic knowledge contains the general knowledge of gamelan music. The construction knowledge controls the components building a composition. The melodic knowledge controls quality of the sound. The knowledge is transformed into rules of notes arrangement used to produce the characteristic sound of gamelan music. Genetic algorithm is used to generate a composition. Gatra, the smallest unit in a composition that contains four beats (notes), is used as a variable to construct the genes of a chromosome. The fitness value is measured based on the weight of notes distribution, identical gatras and melodic features. The evaluation is conducted to measure the quality of sound of "ladrang laras slendro pathet manyura" composition created by the system. The evaluation is conducted based on Turing test which involves human experts to recognize the composition created by the system. The results show that the model of automatic gamelan music composition proposed in this research is effective. All the gamelan experts state that it is very difficult to find the differences between the composition created by the system when it is randomly arranged and other compositions created by human. 4 out of 6 gamelan experts failed to recognize a composition created by the system.*

Keywords*: Rule-based, Genetic Algorithm, Gamelan Music, Automatic Music Composition*

Nomenclature

A-B-C-D	: Concept of *gatra*.
Balungan	: Composition skeleton
Gamelan	: Traditional music ensemble
Gatra	: The smallest unit of *gamelan* music
Gending	: *Gamelan* song
Laras	: Musical scale in *gamelan* music which consists of *slendro* and *pelog*
Pathet	: A system of categorizing the use of tones
Pin	: Dot notation
Rasa	: Sensation or inner mining or ability to express or perceive feeling
Ricikan	: *Gamelan* instrument
ND	: Number of notes distribution
P	: Data partition
S	: Sequence
TF	: Total number of functions
TN	: Total number of notes
TSI	: Total number of itemsets in a sequence
WC	: Weight chaining
WN	: Weight distribution

I. Introduction

Computer music is created using computer technology for artistic conception. It is one of subject studied in artificial intelligence researches and ongoing experiments, both computer music created entirely using computer and with the help of computer [1] [2].

The use of artificial intelligence in music composition is known as algorithmic composition, where certain algorithms are used to automatically create a music composition. Algorithmic composition is a field of research in computer music which studies the process of automatic music composition by partially or wholly using computer; The algorithmic composition software is programmed to generate music with a certain autonomy [3] [4]. The Algorithmic composition started with the works on Illiac Suite and Xenakis. Illiac Suite, developed by [5], uses the rule-based method, and Xenakis, developed by [6], uses Markov models. The algorithmic composition has grown in many variants of algorithmic approaches, such as generative grammars, Genetic Algorithm, cellular automata, neural networks, machine-learning techniques, expert systems, and others [3] [7] [8].

In this research, the algorithmic composition is used for the traditional ensemble music of Java, Indonesia,

Manuscript received and revised May 2008, accepted June 2008

called *gamelan* music. *Gamelan* music has elements of aesthetic, intellectual, beliefs, customs, order, works of human creativity, nature, rules of life, welfare, and society; hence it cannot be separated from human, humanity, and God [9]. *Gamelan* music uses *gamelan* as an instrument for orchestra or ensemble music, and *gending* as the composition of the song [10]. *Gending* is a general term used to refer to Javanese *gamelan* music composition [11]. *Gamelan* melodies are bound by rules and regulations (Javanese society), which are sacred [12]. The process of creating *gamelan* music should consider the Javanese musical concept, because *gamelan* is not only the mean of performances, but also a part of the Javanese community life, including the concept of cosmology and other concepts of life [13].

In this research, the rule-based and Genetic algorithm approaches are used for automatic *gamelan* music composition. Knowledge of *gamelan* music composition is transformed into rules for automatic composition, and Genetic algorithm is used to arrange notes sequences for composition. Some works on Western ensemble music were *CHORAL* and *Bach in Box*. The rule-based system was used by [14] to develop *CHORAL*, a system for the harmonization of four-part chorales in J.S. Bach' style. The use of rules as constraints in generating composition with the Genetic algorithm was used in *Bach in Box* developed by [15].

Gamelan music is different from Western music. *Gatra* is the smallest unit of *gamelan* music composition which contains four beats, and each beat can be a note or *pin* (dot notation). The notes sequence arrangement in *gatra* is the characteristic of *gamelan* music sound. *Gatra* was analyzed by [16] and [17] to formalize certain types of *gending*. The grammar approach is used by [16] to identify the contour of *gatra* with *srepegan*, a type of *gending*, as the subject of the research. The contour is defined based on pitch scale (higher and lower notes) of notes sequence in a gatra. The structure of *gamelan* music called *gending lampah* is studied using a quasi-linguistic approach [17].

The authors of this paper proposed to use the sequential pattern mining technique to analyze the notes sequences of *gatra*. In a previous work [18], they developed a new sequential pattern mining algorithm called AFiS to formalize the melodic feature of *gamelan* music. The development of AFiS algorithm, which will be discussed later in this paper, was inspired by the philosophical concept at the base of arranging notes sequences of *gatra*.

II. Related Works

Computer music is created using computer technology for artistic conception. It is one of subject studied in artificial intelligence researches and ongoing experiments, both computer music created entirely using computer and with the help of computer [1] [2]. *CHORAL* is an expert system for the harmonization of four-part chorales in J.S. Bach's style. The system uses more than 270 rules to produce multi-view points, such as skeleton, individual melodic lines for each sound, and *schenkerian* voice leading in descant and bass [14]. *Bach in Box* uses rules defined by musical scholars, including a pre-defined melody, to control the search space for four-part baroque harmony [15]. *Bach in Box* uses the Genetic algorithm to generate the composition. Its fitness is measured based on the basis chords, ranges, motion, harmonic interest, beginning and end chords, smoothness and resolution.

The unsupervised learning approach based on a corpus of jazz musical performances is used by implementing a combination of clustering technique and Markov chains. The K-means algorithm is used to statistically collect data in a corpus, and then the data are used as statement in Markov Chain [19]. The Lindenmayer technique is used to generate a composition based on simple inputs from users, and then the input is developed using the probability model, fractal, and chaos [20]. The grammar approach is used to develop *ImprovGenerator*, a system which learns the percussion pattern in live-streaming, and generates accompaniment tracks in real time. The mixed model including a hierarchy structure representing a stochastic context-free grammar is then used to generate patterns of accompaniment music based on history and temporary patterns. The transition probability model is used to improve the generated grammar patterns [21].

Genetic algorithm is used to generate multi-instrumental, guitar-oriented rock music. A unique conversion procedure from numerical values to *abc* language, and from *abc* language to numerical values allows combination of optimization number with variants expression from musical description language. The *abc* language is used to notate music in the ASCII format, and converts a song notated in this language into MIDI format by program [22]. The operators of the Genetic algorithm are modified by [23] to allow pitch schedule and interlude changing significantly. The approach includes a pre-defined rhythm which is set as the initial population. A program developed by [24] is designed to create original music compositions based on rules and musical theory. The parameters of the composition, preferences of genre, tempo, and tone, are controlled by the user, and the Genetic algorithm is used to generate a composition.

III. Proposed Model

In this research, the authors revised their previous proposed model of automatic *gamelan* music composition [25]. During their experiments, they developed a new algorithm called *AFiS* (Apriori based on Functions in sequence) to identify sequential patterns, and *gamelan* notes pattern is one of the subjects which can be used as the subject to identify [18]. The use of AFiS algorithm reduces some procedures of the previous work, and gives more accurate results. The implementation of AFiS algorithm is explained in the section about melodic feature knowledge.

There are three main types of knowledge in our model of automatic *gamelan* music composition, basic, construction and melodic knowledge. The basic knowledge contains the general knowledge of *gamelan* music, such as rules of number and variants of notes for each type of *laras*, rules of number of *gatra* for each type of *gending*, rules of structural ricikan and balungan ricikan, and others. The construction knowledge controls the components building a *gending*, such as number of notes, notes variants and their distribution, number or identical *gatras*. The melodic knowledge is about the quality and characteristics of sound. This knowledge controls notes arrangement to produce a sound with characteristics of *gamelan* music, including the type of *gending* and its *pathet*.

The basic knowledge is based on *gamelan* theory, while construction knowledge and melodic knowledge are built using notes pattern analysis. A combination of basic knowledge, construction knowledge, and melodic knowledge is then transformed into rules of *gamelan* music composition. Furthermore, the genetic algorithm is used to generate a composition. Fig. 1. shows the diagram model of automatic *gamelan* music composition proposed in this research.

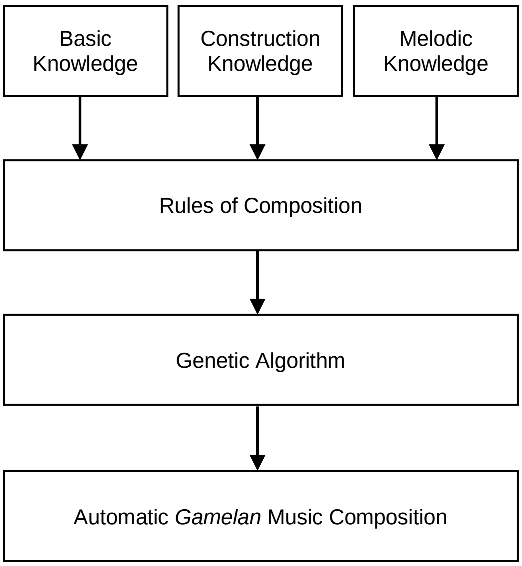

Fig. 1. Diagram model of automatic *gamelan* music composition

IV. *Gamelan* Music Knowledge Base

The basic knowledge contains the *gamelan* theory, such as the *gatra*, the type of *laras*, the type of *pathet*, the type of *gending*, and the rules of structural *ricikan* and *balungan* (skeleton) *ricikan*. The basic knowledge is used by the system to alter notes in terms of *gatra*, to compose a *gending* based on *laras* and *pathet*, and to define the *structural ricikan* as a characteristic of the type of *gending*.

Gatra is the smallest unit of *gamelan* music composition which contains 4 beats, and each beat can be a note or *pin* (dot notation). In *ricikan balungan*, the type of *gatra* is different according to the use, the order of notes and dot notation. For instance, the *gatra* which uses notes for all beats is called *balungan mlaku*. Fig. 2. shows examples of *gatras* including *ricikan balungan*.

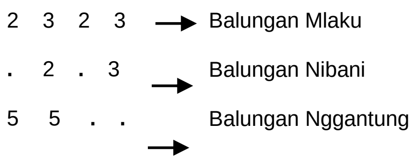

Fig. 2. Examples of *gatras*, including its balungan ricikan

In this research, the use of *gatra* is limited to the type of *balungan mlaku* and *balungan nibani*. The system developed in this research is designed to compose a *gending* in form of *balungan mlaku*, but the composition result can be arranged and modified into *balungan nibani*.

Laras is musical scale in *gamelan* music. There are two types of *laras, laras slendro* and *laras pelog*. *Laras slendro* consists of five notes: 1, 2, 3, 5, 6. *Laras pelog* consists of seven notes: 1, 2, 3, 4, 5, 6, 7. There is a categorization system for the use of tones called *pathet* [26]. *Laras slendro* consists of *pathet nem, pathet sanga*, and *pathet manyura*. *Laras pelog* consists of *pathet lima, pathet nem*, and *pathet barang*. *Pathet* controls the common use of notes for each type of *pathet*. Table I describes the types of *pathet* including their notes.

TABLE I
PATHET

Laras	*Pathet*	Common use of Notes
Slendro	Nem	6, 5, 3, 2
	Sanga	2, 1, 6, 5
	Manyura	3, 2, 1, 6
Pelog	Lima	5, 4, 1, 2
	Nem	2, 1, 6, 5
	Barang	3, 2, 7, 6

There are three types of *ricikan* in *gamelan*: *ricikan balungan*, based on *balungan* (skeleton) or structure of *gending*, *ricikan garap* which extends and completes *ricikan balungan* in arranging the composition, and *ricikan structural* which determines the structure of *gending* [27]. Fig. 3. shows the examples of *ricikan balungan* extending into *ricikan garap*.

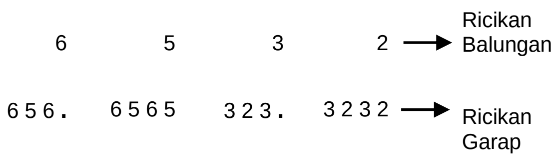

Fig. 3. Ricikan balungan and ricikan garap [27]

Gending is a *gamelan* music composition which can be with or without vocals. There are 7 forms of *gending* included in *gending* alit categories, namely *lancaran, gangsaran, ketawang, ladrang, ayak-ayakan, srepegan and* sampak [27]. The number of *balungan* beats in one *gong*, a *gamelan* instrument, and the setting of the play of *gamelan* instruments of *kethuk*, *kempul*, *kenong*, *kempyang* and *gong* defines the type of *gending* [27]. For example, *gending lancaran* has 16 beats divided into 4 *gatras*; the *kethuk* instrument is played in the first and third beats; the *kempul* instrument is played in the second beat; the *kenong* instrument is played in the fourth beat; the *gong* instrument is played in the fourth beat of the last *gatra*. Another example is *gending ladrang* which has at least 8 *gatras*; each *gatra* consists of 4 beats; the kenong instrument is played in every 2 *gatras*; gong, kenong, and kempul instruments are played together in the last beats.

In this research, the *gending* with *ladrang laras slendro pathet manyura* is used as subject for automatic *gamelan* music composition, and the output of composition is in the form of *ricikan balungan*. Table II shows the basic knowledge.

TABLE II
BASIC KNOWLEDGE

Components	Notes
Gatra	Consists of 4 beats
Balungan Nibani	Every beat in *gatra* is filled with notes.
Minimum number of *gatras* in *ladrang*	8
Minimum number of notes in *ladrang*	32
Laras slendro	Consists of notes 1, 2, 3, 5, 6
Common use of notes for *laras slendro pathet manyura*	3, 2, 1, 6

The construction knowledge controls the components building a *gending*: the number of *gatras* and notes, notes variant and their distribution and the number or identical *gatras*. For example, Fig. 3 shows a composition of *ladrang laras slendro pathet manyura* entitled *Gudhasih*.

Ladrang *"Gudhasih"* Laras Slendro Pathet Manyura

2 1 2 6   2 1 2 6   3 6 3 2   3 2 1 6
5 6 5 6   2 1 3 2   6 1 3 2   3 2 1 6

Fig. 3. *Ladrang laras slendro pathet manyura* entitled "*Gudhasih*"

For the next explanation, the title of *gending* is written in the form of type of *gending* and *tittle*, for example *ladrang Gudhasih*, where ladrang is a type of *gending*, and *Gudhasih* is a title of *gending*. *Ladrang Gudhasih* consists of 8 *gatras* and 32 notes. All variant notes in *laras slendro*, which are 1, 2, 3, 5, 6, are used in this composition. Note 1 is distributed as many of 6, note 2 is distributed as many of 10, note 3 is distributed as many of 6, note 5 is distributed as many of 2, and note 6 is distributed as many of 8. There are 2 identical *gatras* in this composition, which are (2126) and (3216). *Gatras* (2126) are in 1^{st} and 2^{nd} position order, and *gatras* (3216) are in 4^{th} and 8^{th} position order. Table III describes the construction of the *gending*.

TABLE III
GENDING CONSTRUCTION OF GUDHASIH

Components	Notes
Number of *gatras*	8
Number of notes	32
Notes variant	1, 2, 3, 5, 6
Distribution of notes variant (notes: number)	1:6 2:10 3:6 5:2 6:8
Number of identical *gatras*	2 (2126), (3216)
Position order	1^{st}, 2^{th} 4^{rd}, 8^{th}

The construction knowledge is built by analyzing a dataset containing *gendings* in same type of *gending*, *laras*, and *pathet*. The dataset used in this research contains *gending* of *laras slendro pathet manyura*. 15 *ladrang laras slendro pathet manyura* entitled *Bantul*, *Bogaginula*, *Gonjang*, *Gudhasih*, *Kandha Manyura*, *Kembang Pepe*, *Kuwung*, *Lomanis*, *Moncer Alus*, *Sri Katon*, *Suntrut*, *Surengrana*, *Thinik*, *Tropong*, *Wilujeng Alus*, are used as dataset.

The component of number of *gatras* is used by the system to define the number of *gatra* for composition generation. The data collected from *gendings* samples are used as parameter to generate composition. Data of number of notes are collected by multiplying the number of *gatras* by 4 (number of beats in a *gatra*).

The analysis on the number of *gatras* and notes, as well as notes variant used for composition is implemented to all the *gending* samples in dataset. The result shows that 1 *gending* consists of 4 *gatras*, 9 *gendings* consist of 8 *gatras*, and 5 *gendings* consist of 12 *gatras*. This is used as knowledge to define the number of *gatras* in generating composition, including the number of notes. Table IV shows the knowledge of number of *gatras*, and number of notes.

TABLE IV
KNOWLEDGE OF NUMBER OF *GATRAS*, AND NOTES

Components	Value
Number of *Gatras*	4, 8, 12
Number of notes	16, 32, 48

Data of notes variant are collected by identifying the notes variants used in each *gending* sample. All *gending* samples use notes variants 1, 2, 3, 5, 6 in their composition. This is used by the system to define notes variant in composition generation.

Notes variant distribution knowledge is built by analyzing the weight of notes distribution in every *gending* sample. The analysis is implemented based on notes variants used in *gending*. The goal of this analysis is to identify the range of minimum and maximum distribution for each note variant, where the value of minimum and maximum distribution of each note variant is obtained by sorting the weight of each notes variant distribution in all *gending* samples.

In order to accommodate the common use of notes variant as the type of *gending*, *laras*, and *pathet*, the weight of notes distribution is then sequentially calculated in terms of "*previous note distribution defines following note distribution*". Furthermore, the result of the calculation is used to define a range of minimum and maximum distribution for each note variant. Below is the formula to measure the weight of notes distribution, with *WN* denoting weight distribution, *ND* denoting the number of notes distribution, and *TN* denoting the total number of notes in a *gending*:

$$WN = \frac{ND}{TN} \quad (1)$$

As an example, *ladrang Gudhasih* was used to simulate the weight of notes distribution measurement; the results are shown in Table V.

TABLE V
WEIGHT OF NOTES DISTRIBUTION
(SIMULATION IN GUDHASIH)

Notes Variant	ND	TN	WN (ND/TN)
1	6	32	0.188
2	10	32	0.313
3	6	32	0.188
5	2	32	0.063
6	8	32	0.250

The weight of each note distribution is then sequentially calculated in terms of weight of the previous note minus the weight of the following note. This can be called *weight chaining* (*WC*). The formula to measure weight chaining for each notes variant is:

$$WC_k = WN_k - WN_{k+1}$$
$$WC_{end} = WN_{end} - WN_1 \quad (2)$$

Continuing the simulation above, Table VI shows the result of weight chaining.

TABLE VI
WEIGHT CHAINING OF NOTES DISTRIBUTION
(SIMULATION IN LADRANG GUDHASIH)

Notes	WN	WC
1	0.188	-0.125
2	0.313	0.125
3	0.188	0.125
5	0.063	-0.188
6	0.250	0.063

The above procedure is implemented to all *gending* samples. Furthermore, the value of weight chaining of notes distribution of all *gending* samples is concatenated based on note variant, and the range of minimum and maximum distribution of each note is defined based on the lowest and highest values of weight sorted from concatenation.

The minimum distribution of note variant can be defined using the following formula, with *T* denoting the total number of *gending*s in a dataset:

$$\sum_{k=0}^{k=T} \text{Min (WC)} \quad (3)$$

While the maximum distribution of note variant can be defined using the below formula:

$$\sum_{k=0}^{k=T} \text{Max (WC)} \quad (4)$$

Table VII shows the result of the process of measuring weight chaining (WC) of notes distribution for each *gending* sample, and then concatenating and sorting the results based on notes variant to obtain the minimum and maximum values of notes distribution weight.

TABLE VII
MINIMUM AND MAXIMUM OF WEIGHT CHAINING

Notes	Distribution	
	Min	Max
1	-0.188	0
2	-0.084	0.25
3	-0.084	0.219
5	-0.219	0.188
6	-0.125	0.125

Identical *gatra*s are common in *gending* composition. All *gending* samples have identical *gatra*s in their composition. The identical *gatra*s knowledge contains the number of *gatra*s used more than one time in a composition, and their position order. Table VIII shows the identification result of identical *gatra*s in each sample *gending*, including their position order in the composition.

TABLE VIII
IDENTICAL *GATRAS* (IG): NUMBER AND POSITION ORDER

Gending ID	Number of IG	Position Order
01	2	(3, 7), (4, 8)
02	2	(1, 3), (2, 4, 8)
03	2	(1, 4, 5), (3, 11)
04	2	(1, 2), (4, 8)
05	2	(4, 8),(9, 10)
06	4	(2, 3, 10, 11), (4, 8, 12) , (5, 6), (7, 9)
07	2	(1, 10), (2, 11)
08	1	(3, 4, 8)
09	1	(1, 4, 8)
10	1	(1, 2, 3, 7, 8)
11	2	(4, 6), (1, 2, 8)
12	1	(1, 2)
13	2	(1, 2), (4, 8), (5, 6)
14	3	(1, 2, 5), (3, 7), (9, 10)
15	2	(1, 4), (5, 7)

The number of identical *gatra*s identified in *gending* samples is 1, 2, 3, 4.The data are used by the system to define the number of identical *gatra*s in composing a *gending*.

Identical *gatra*s knowledge is built by identifying the *gatra* pattern based on position order. $A \Rightarrow B$ is used to construct the pattern of identical *gatra*s, where *A* and *B* represent the position order. If one of the identical *gatra*s is positioned in *A* order, then the other *gatra*(s) can be positioned in *B* order, where *B* can be more than one position order.

$A \Rightarrow B$ is used to identify the identical *gatras*, puts the first order is *A*, and the following order(s) is (are) *B*. For example, based on the identical *gatras* analysis conducted on 20 *gending* samples, 3 gendings have identical *gatras* with fifth *gatra* as the first order (A), and the patterns in each *gending* are (5, 6), (5, 12), (5, 8, 10). The inference is that the fifth *gatra* can be reused on 6, 8, 10 and 12th *gatras*. Table IX shows the identical *gatras* pattern knowledge.

TABLE IX
IDENTICAL *GATRAS* PATTERN KNOWLEDGE

First Order	Following Order(s)
1	{2},{ 3}, {4}, {5}, {7}, {8}, {10}
2	{3}, {4}, {8}, {10}, {11}
3	{4}, {7}, {8}, {11}
4	{6}, {8}, {12}
5	{6}, {8}, {10}, {12}
7	{9}
9	{10}

The melodic knowledge is used by the system to arrange notes sequences that fit the sound characteristics of *gamelan* music. The notes arrangement must contain the concept of *A-B-C-D* of *gatra*, where *A* denotes *maju* (forward), *B* denotes *mundur* (back), *C* denotes *maju* (forward), and *D* denotes *seleh* (end point of a journey). The value (notes) of four beats in *gatra* must contain the concept of *A-B-C-D*. This concept controls the hierarchy of function of every beat in a gatra. *D* is the strongest part, since it is the musical point reference, *B* is the second part, *A* is the third part, and *C* is the weakest part. The strong or weak level of parts of a *gatra* is defined by the notes filled in each part. The chosen notes in the previous part and the following part define the strength or weakness of the parts. The correct arrangement of notes that fits the concept of *A-B-C-D* produces sound that fits the characteristic sound of gamelan music.

In the previous work [25], the *AFiS* algorithm was proposed for sequential pattern mining, and the algorithm was implemented to identify the melodic feature of *gamelan* music. The AFiS algorithm uses functions in a sequence, where each function contains an item based on its order. Furthermore, the functions are chained in terms of sequential pattern.

In this research, the AFiS algorithm was used to build the melodic knowledge of ladrang *laras slendro pathet manyura* in form of *gatra*. First is functions definition, where the concept of *A-B-C-D* is used as function. Function *A* contains the first note of *gatra*, function *B* contains the second note of *gatra*, function *C* contains the third note of *gatra*, and function *D* contains the fourth note of *gatra*. Next is the data partition phase, where each *gatra* represents a partition. Data partition can be formulated as the below pseudocode [16]:

S : sequence
TSI : total number of itemsets in a sequence
TF : total number of functions
P : data partition

```
n = 0
While ( n < (TSI / TF ) ) {
        P [n] = [ ]
        n++
}

For ( n = 0; n < TSI; n++ ){
      For ( k = 0; k < TF; k++){
            P [n] [k] = S [ (k*TF) + n ]
      }
}
```

Each note in *gatra* is then altered in each function based on its order. Table X shows the simulation of function definition, and data partition for *ladrang Gudhasih*.

TABLE X
FUNCTION DEFINITION AND DATA PARTITION
(SIMULATION IN LADRANG GUDHASIH)

Partition Number	Data Partition (*Gatras*)	A	B	C	D
1	<2, 1, 2, 6>	2	1	2	6
2	<2, 1, 2, 6>	2	1	2	6
3	<3, 6, 3, 2>	3	6	3	2
4	<3, 2, 1, 6>	3	2	1	6
5	<5, 6, 5, 6>	5	6	5	6
6	<2, 1, 3, 2>	2	1	3	2
7	<6, 1, 3, 2>	6	1	3	2
8	<3, 2, 1, 6>	3	2	1	6

The sequential patterns are built by chaining the functions. In this experiment, the sequential pattern of *gatra* consists of <A, B, C, D>, <B, C, D, A>, <C, D, A*, B*>, <D, A*, B*, C*>, where the asterisk denotes the next partition. Table XI shows the simulation of a sequential pattern for *ladrang Gudhasih*.

TABLE XI
SEQUENTIAL PATTERNS CREATION
(SIMULATION IN LADRANG GUDHASIH)

<A, B, C, D>	**<B, C, D, A*>**	**<C, D, A*, B*>**	**<D, A*, B*, C*>**
<2, 1, 2, 6>	<1, 2, 6, 2>	<2, 6, 2, 1>	<6, 2, 1, 2>
<2, 1, 2, 6>	<1, 2, 6, 3>	<2, 6, 3, 6>	<6, 3, 6, 3>
<3, 6, 3, 2>	<6, 3, 2, 3>	<3, 2, 3, 2>	<2, 3, 2, 1>
<3, 2, 1, 6>	<2, 1, 6, 5>	<1, 6, 5, 6>	<6, 5, 6, 5>
<5, 6, 5, 6>	<6, 5, 6, 2>	<5, 6, 2, 1>	<6, 2, 1, 3>
<2, 1, 3, 2>	<1, 3, 2, 6>	<3, 2, 6, 1>	<2, 6, 1, 3>
<6, 1, 3, 2>	<1, 3, 2, 3>	<3, 2, 3, 2>	<2, 3, 2, 1>
<3, 2, 1, 6>	<2, 1, 6, ->	<1, 6, -, ->	<6, -, -, ->

Next is candidate selection, where the itemsets with a length that is not equal to the length of functions is eliminated. The pattern of <A, B, C, D> of *ladrang*

Gudhasih contains 8 itemsets, while the other patterns contain 7 itemsets. A candidate defined as frequent is measured using the minimum support value. The given minimum support is 1, which means that an itemset must have at least 1 transaction to be defined as frequent. By setting 1 as the minimum support value, all candidates are frequent.

The above process of function definition, data partition, sequential pattern creation and support counting are implemented to all *gending* samples. Furthermore, all itemsets in all *gending* samples are concatenated based on each pattern. Since the given minimum support value is 1, the weight of an itemset after concatenation is not counted. The duplicate itemsets in each chain are removed. Table XII shows the result of pattern concatenation of <A, B, C, D>, <B, C, D, A>, <C, D, A*, B*>, <D, A*, B*, C*> for all *gending* samples.

TABLE XII
RESULTS OF SEQUENTIAL PATTERN CONCATENATION OF ALL *GENDING* SAMPLES

<A, B, C, D>	<1, 2, 1, 6>, <1, 2, 5, 3>, <1, 6, 1, 6>, <1, 6, 2, 3>, <1, 6, 3, 2>, <1, 6, 5, 3>, <2, 1, 2, 6>, <2, 1, 3, 2>, <2, 1, 5, 3>, <2, 1, 6, 5>, <2, 3, 1, 6>, <2, 3, 2, 1>, …, <6, 3, 2, 1>
<B, C, D, A*>	<1, 2, 6, 1>, <1, 2, 6, 2>, <1, 2, 6, 3>, <1, 2, 6, 5>, <1, 3, 2, 3>, <1, 3, 2, 5>, <1, 3, 2, 6>, <1, 5, 3, 1>, <1, 5, 3, 2>, <1, 5, 3, 5>, <1, 6, 5, 6>, <2, 1, 6, 2>, …, <6, 5, 6, 3>
<C, D, A*, B*>	<1, 6, 2, 1>, <1, 6, 2, 3>, <1, 6, 3, 1>, <1, 6, 3, 2>, <1, 6, 3, 6>, <1, 6, 5, 3>, <1, 6, 5, 6>, <2, 1, 2, 1>, <2, 1, 2, 3>, <2, 1, 2, 6>, <2, 1, 3, 2>, <2, 1, 5, 6>, …, <6, 5, 6, 3>
<D, A*, B*, C*>	<1, 2, 1, 2>, <1, 2, 3, 2>, <1, 2, 6, 2>, <1, 2, 6, 5>, <1, 3, 2, 1>, <1, 5, 6, 5>, <2, 1, 6, 1>, <2, 1, 6, 3>, <2, 3, 1, 3>, <2, 3, 1, 5>, <2, 3, 2, 1>, <2, 3, 2, 3>, …, <6, 5, 6, 5>

Next is the prune phase to set the following *gatra* based on the previous. Pruning of *gatra* is set by chaining the functions (sequential patterns) as seen in Fig. 4.

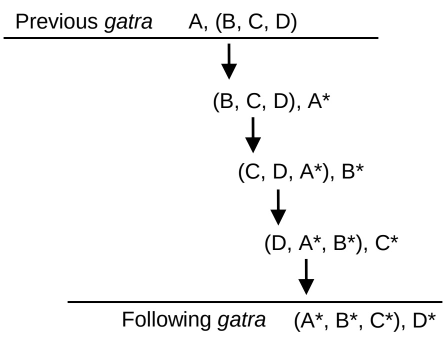

Fig. 4. Pruning by chaining the functions

Table XIII shows an example of *gatra* pruning by chaining the functions for gatra (1216) and (1253). Function <A, B, C, D> is for the previous *gatra*, and function <A*, B*, C*, D*> is for the following *gatra*.

TABLE XIII
EXAMPLE OF *GATRAS* PRUNING

<A, B, C, D> (Previous)	<B, C, D, A*>	<C, D, A*, B*>	<D, A*, B*, C*>	<A*, B*, C*, D*> (Following)
<1, 2, 1, 6>	<2, 1, 6, 2>	<1, 6, 2, 1>	<6, 2, 1, 2>	<2, 1, 2, 6>
			<6, 2, 1, 3>	<2, 1, 3, 2>
			<6, 2, 1, 5>	<2, 1, 5, 3>
		<1, 6, 2, 3>	<6, 2, 3, 1>	<2, 3, 1, 6>
	<2, 1, 6, 3>	<1, 6, 3, 1>	<6, 3, 1, 2>	<3, 1, 2, 6>
		<1, 6, 3, 2>	<6, 3, 2, 1>	<3, 2, 1, 6>
			<6, 3, 2, 3>	<3, 2, 3, 1>
				<3, 2, 3, 2>
		<1, 6, 3, 6>	<6, 3, 6, 3>	<3, 6, 3, 2>
	<2, 1, 6, 5>	<1, 6, 5, 3>	<6, 5, 3, 1>	<5, 3, 1, 6>
			<6, 5, 3, 5>	<5, 3, 5, 3>
				<5, 3, 5, 6>
		<1, 6, 5, 6>	<6, 5, 6, 2>	<5, 6, 2, 1>
			<6, 5, 6, 3>	<5, 6, 3, 2>
			<6, 5, 6, 5>	<5, 6, 5, 3>
				<5, 6, 5, 6>
<1, 2, 5, 3>	<2, 5, 3, 1>	<5, 3, 1, 2>	<3, 1, 2, 1>	<1, 2, 1, 6>
			<3, 1, 2, 5>	<1, 2, 5, 3>
		<5, 3, 1, 6>	<3, 1, 6, 5>	<1, 6, 5, 3>
	<2, 5, 3, 2>	<5, 3, 2, 1>	<3, 2, 1, 2>	<2, 1, 2, 6>
			<3, 2, 1, 3>	<2, 1, 3, 2>
	<2, 5, 3, 5>	<5, 3, 5, 2>	<3, 5, 2, 5>	<5, 2, 5, 3>
		<5, 3, 5, 3>	<3, 5, 3, 2>	<5, 3, 2, 1>
			<3, 5, 3, 5>	<5, 3, 5, 3>
				<5, 3, 5, 6>
		<5, 3, 5, 6>	<3, 5, 6, 5>	<5, 6, 5, 3>
				<5, 6, 5, 6>

V. Composition Rules

Rules of composition are defined based on basic knowledge, construction knowledge, and melodic knowledge. Therefore, there are basic rules, construction rules, and melodic rules.

In this research, the type of *gending* used for automatic *gamelan* music composition is the ladrang *laras slendro pathet manyura*. The basic knowledge is used to construct a *gending*. The system built in this research sets the number of *gatra*s to generate a composition using the knowledge of number of *gatra*s and notes (Table IV), and sets the beats in *gatra* based on the type of *balungan* using the basic knowledge (Table II). The generation of composition is controlled by construction and melodic rules. The system randomizes the collection of *gatras* in a sequential pattern of <A, B, C, D> (Table XII) to create notation sequence of composition. The result of the composition is measured using the weight chaining of notes distribution (Table VII), the number and distribution of identical *gatra*s (Table VIII), and *gatra*s pruning (Table XIII).

VI. Genetic Algorithm for Composition Generation

The genetic algorithm is used to generate composition of ladrang *laras slendro pathet manyura*. There are three main constraints in composition generation: weight distribution of notes, identical *gatras*, and melodic features. The objective function is formulated as ($x1 + x2 + x3$). Variable $x1$ denotes the weight of notes distribution. The later (Table 5) is used to measure each note variant. There are 5 notes variants to distribute, and value 1 is given to a note which fits the knowledge, otherwise the value given is 0. The weight of notes distribution for every note in the composition must fit the knowledge, so the constraint of weight of notes distribution is satisfied with value 5. Variable $x2$ denotes the minimum number of identical *gatras* in a composition. There must be at least 1 *gatra* used as identical *gatra*, and the distribution fits the knowledge of idenctical *gatras*. The value 1 is given if the condition is fulfilled. Variable $x3$ denotes the number of *gatras* to compose, where each *gatra* will have value 1 if its pruning is correct as the melodic feature represented in knowledge and in rules of *gatra* pruning. For example, if there are 8 *gatras* composition to generate, and each *gatra* has correct pruning, then the value of each *gatra* is 1, and the total value is 8. So, the objective function value of a composition which consists of 8 *gatras* is (5 + 1 + 8).

The chromosome consists of as many genes as the number of *gatras* to generate. If there are 8 *gatras* to generate, then each chromosome consists of 8 genes. The value of genes is set by randomizing the itemsets of a sequential pattern *<A, B, C, D>*. A number of chromosomes are created for the initial population, and then the procedures of selection and crossover are implemented to chromosomes. Rank selection was used to limit the elimination of chromosomes which do not fit. One-point crossover is used to create children (Fig. 5).

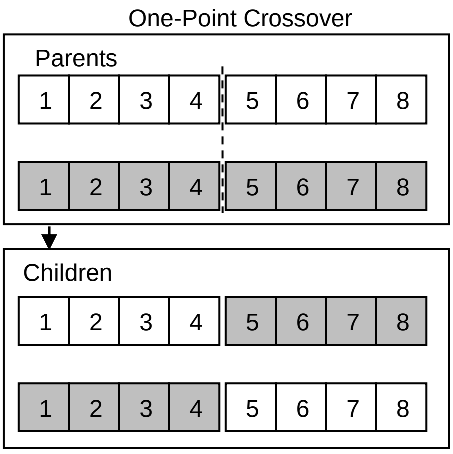

Fig. 5. One-point crossover to create children

The fitness of chromosomes is measured by validating the weight of notes distribution for each note, identical *gatras*, and the melodic feature. The mutation procedure is implemented if there are no chromosomes fitting. The process is restarted from the selection phase if there are still no chromosomes fitting after mutation.

VII. Implementation

The model proposed in this research is implemented by developing a system for automatic *gamelan* music composition. In this experiment, an 8 *gatras* composition of ladrang *laras slendro pathet manyura* was created. 100 chromosomes were set. Each chromosome consists of 8 genes which represent *gatras*. The value of genes is set by randomizing the collection of *gatras* in a sequential pattern *<A, B, C, D>*. Furthermore, the procedures of selection, crossover, and mutation at a rate of 0.1, constrained to weight chaining of notes distribution, identical *gatras*, and melodic features, are used to search the best composition. The generation is limited to 1000 times. The results show that system can generate a composition in 24 times generation. The composition of ladrang *laras slendro pathet manura* generated by the system is shown in Fig. 6.

Composition of *ladrang laras slendro pathet manyura* generated by system

5 6 5 6   2 1 2 6
3 2 5 3   1 2 5 3
5 3 5 6   5 3 5 6

Fig. 6. Composition generated by system

The composition of *ladrang laras slendro pathet manyura* generated by the system can satisfy the constraints of weight chaining of notes distribution, identical *gatras*, and melodic features. Each note in the composition has a weight chaining value that fits the rules, so each note has value 1, and the total value of weight chaining ($x1$) is 5. Table XIV shows the weight of notes distribution of the composition created by the system, where the weight of each note distribution fits the range of minimum and maximum weight of notes distribution (WC).

TABLE XIV
WEIGHT CHAINING OF NOTES DISTRIBUTION OF COMPOSITION CREATED BY THE SYSTEM

Notes	ND	TN	WN (ND/TN)	WC	Value (x1)
1	4	32	0.125	-0.094	1
2	7	32	0.219	0.000	1
3	7	32	0.219	-0.031	1
5	8	32	0.250	0.063	1
6	6	32	0.188	0.063	1

The number of *gatra* used as identical *gatras* and its position order fulfil the condition of rules of identical *gatras*. There is one *gatra* used as identical *gatra*, which is *gatra* (5 3 5 6), and is positioned at 5^{th} and 6^{th} order in the composition. The value for identical gatras (*x2*) of this composition is 1 (Table XV).

TABLE XV
IDENTICAL GATRAS OF COMPOSITION CREATED BY THE SYSTEM

Components	Notes	Value (x2)
Number of identical *gatras*	1 (5 3 5 6)	1
Position order	5^{st}, 6^{th}	

The *gatra*s pruning in this composition fit the rules of melodic features. The value of *gatras pruning* of each gatra is 1, if the following *gatra* is matches the prune of the previous *gatra*. The evaluation shows that all *gatras* in the composition have matching following *gatras* in their pruning. The composition has 8 *gatras*, and each *gatra* has a matching following *gatra*, therefore the total value (x3) achieved is 8 (Table XVI).

TABLE XVI
GATRAS PRUNING OF COMPOSITION CREATED BY THE SYSTEM

NO	Previous Gatra	Following Gatra	Gatra Prunning	Value
1	5 6 5 6	2 1 2 6	True	1
2	2 1 2 6	3 2 5 3	True	1
3	3 2 5 3	1 2 5 3	True	1
4	1 2 5 3	5 3 5 6	True	1
5	5 3 5 6	5 3 5 6	True	1
6	5 3 5 6	2 1 3 2	True	1
7	2 1 3 2	3 2 1 6	True	1
8	3 2 1 6	5 6 5 6	True	1

The objective function which must be fulfilled is (*x1* + *x2* + *x3*), where *x1* = 5, *x2* = 1, and *x3* = 8. The implementation of the genetic algorithm can satisfy the constraints to generate an 8 *gatras* composition of *ladrang laras slendro pathet manyura.*

VIII. Evaluation

The evaluation is conducted to measure the quality of sound of the composition of *ladrang laras slendro pathet manyura* created by the system. A framework of evaluation of algorithmic composition proposed by [28] was used. The evaluation is conducted based on Turing test which involves human experts to recognize the composition created by the system.

In this evaluation, a collection of *ladrang laras slendro pathet manyura* was used, which consists of 5 *gending*s composed by humans: *Ghudasih, Lomanis, Sri Katon, Thinik,* and *Wilujeng Alus*, and add a composition generated by the system to the collection. The *gendings* in collection are arranged randomly as seen in Table XVII. Further, six *gamelan* experts with a background of leaders of *gamelan* studio, *gamelan* practitioners, and lecturers, are asked to recognize a composition created by the system in the collection, to assess the difficulty level in recognizing a composition created by system, and to define the *pathet* of each composition.

TABLE XVII
COLLECTION OF GENDINGS EVALUATED BY GAMELAN EXPERTS

ID	Gending Tittle	Notes
G1	Gudhasih	2 1 2 6 2 1 2 6 3 6 3 2 3 2 1 6 5 6 5 6 2 1 3 2 6 1 3 2 3 2 1 6
G2	Lomanis	1 6 3 2 3 1 3 2 3 2 1 6 3 2 1 6 5 6 5 6 2 1 5 3 2 1 2 6 3 2 1 6
G3	Sri Katon	2 1 2 6 2 1 2 6 2 1 2 6 3 6 3 2 5 6 5 3 1 6 5 3 2 1 2 6 2 1 2 6
G4	System	5 6 5 6 2 1 2 6 3 2 5 3 1 2 5 3 5 3 5 6 5 3 5 6 2 1 3 2 3 2 1 6
G5	Thinik	2 1 2 6 2 1 2 6 1 6 3 2 3 2 1 6 5 3 5 6 5 3 5 6 2 1 3 2 3 2 1 6
G6	Wilujeng Alus	2 3 1 6 3 6 3 2 5 3 1 6 2 3 1 6 5 6 5 6 2 1 3 2 5 6 5 6 3 2 1 6

The evaluation results show that two out of six *gamelan* experts can correctly recognize a composition created by the system. The evaluation continues by asking the experts to evaluate the difficulty level in recognizing a composition created by a human or the system. The value in scale of 1-5 is given to evaluate the difficulty level, where value 1 indicates very easy and value 5 indicates very difficult. All the experts give value 5 in recognizing a composition created by humans or generated by the system. To define the *pathet* of each gending, all the experts state that all the *gendings* in the collection have *manyura* as type of *pathet*.

Table XVIII shows the evaluation result, with item **G4* indicating the composition generated by system. Column I in the table is about recognizing a composition created by the system, with symbol ✓ denoting a correct answer, and *x* denoting a wrong answer; Column II is about the difficulty level with value in scale of 1-5, where value 5 indicates very difficult; Column III is about the type of *pathet* of each gending in collection, where 5 composition created by humans have *manyura* as their type of *pathet*, and a composition generated by systems is aimed at *pathet manyura*.

TABLE XVIII
RESULT OF THE EVALUATION

Experts	I		II	III
1	*G4	✓	5	All Manyura
2	G3	x	5	All Manyura
3	*G4	✓	5	All Manyura
4	G6	x	5	All Manyura
5	G6	x	5	All Manyura
6	G2	x	5	All Manyura

IX. Conclusion and Future Works

The evaluation by *gamelan* experts results show that the composition generated by the system is very difficult to be differentiated from the composition created by humans. 4 out of 6 *gamelan* experts failed to recognize a composition created by the system. Expert 1 and 3 could recognize the composition generated by system, but they agreed that it was very difficult to recognize it, as well as other experts. All experts stated that all the evaluated *gending* have *pathet manyura*. This proves that the system can generate a composition with a specific *pathet*. The model of automatic *gamelan* music composition proposed in this research is effective, but the element of *rasa* which can be translated as sensation or inner mining or ability to express or perceive feelings [29], is still not accommodated in this model. The AFiS algorithm can only be used for *balungan mlaku* and *balungan nibani*. On the other hand, it needs more types of *balungan* to explore elements of *rasa*. In the future works, the AFiS algorithm can be revised to accommodate more types of *balungan* in analyzing the notes sequence pattern of *gamelan* music, or the model proposed in this research can be improved by adding elements of *ricikan garap*, or conducting a classification which includes element of *rasa* for data training.

References

[1] D. Trueman, Why a Laptop Orchestra, *Cambridge Journal Volume 12*, 2007.

[2] S. Keith, Bridging The Gap: Thoughts On Computer Music And Contemporary (Popular) Electronic Music, *Proceedings of the 2010 Australasian Computer Music Conference*, 2010.

[3] J.D. Fernandez, F. Vico, AI Methods in Algorithmic Composition: A Comprehensive Survey, *Journal of Artificial Intelligence Research 48*, 2013.

[4] E.R. Miranda, *Composing Music with Computer*, (Focal Press, 2004).

[5] L.A. Hiller, Jr., L.M. Isaacson, *Experimental Music: Composition with an Electronic Computer*, (McGraw-Hill Book Company, Inc, 1959).

[6] I. Xenakis, *Formalized Music: Thought and Mathematics in Composition*, (Pendragon Press, 1992).

[7] G. Nierhaus, *Algorithmic Composition: Paradigms of Automated Music Generation*, (Springer Wien, 2009).

[8] K. Essl, *Algorithmic Composition, Cambridge Companion to Electronic Music*, ed. by N. Collins and J. d'Escrivan, (Cambridge University Press, 2007).

[9] S. Hastanto, *Konsep Pathet Dalam Karawitan Jawa*, (ISI Press, 2009)

[10] H. Kridalaksana, F.X. Rahyono, D. Puspitorini, S. Widodo, Darmoko, *Wiwara: Pengantar Bahasa dan Kebudayaan Jawa*, (Gramedia Pustaka Umum, 2001).

[11] R. Supanggah, *Bothekan Karawitan I*, (Masyarakat Seni Pertunjukan Indonesia, 2002).

[12] M.C. Ricklefs, *Sejarah Indonesia Modern 1200-2008*, (PT Serambi Ilmu Semesta, 2008).

[13] S. Nakagawa, *Musik dan Kosmos: Sebuah Pengantar Etnomusikologi*, (Yayasan Obor Indonesia, 2000).

[14] K. Ebcioglu, An Expert System for Chorale Harmonization, *AAAI-86 Procedings*, 1986.

[15] R.A. McIntyre, Bach in a Box: The Evolution Of Four Part Baroque Harmony Using The Genetic Algorithm, *IEEE World Congress on Computational Intelligence*, 1994.

[16] J. Becker, A. Becker, *A Grammar of the Musical Genre Srepegan*, (University of Texas Press, Texas, 1982)

[17] D. W. Hughes, *Deep Structure and Surface Structure in Javanese Music: A Grammar of Gendhing Lampah* (University of Illinoi Press, Illinois, 1988).

[18] K. Hastuti, A. Azhari, A. Musdholifah, R. Supanggah, Building Melodic Feature Knowledge Of *Gamelan* Music Using Apriori Based On Functions In Sequence (AFiS) Algorithm, *International Review on Computers and Software (I.RE.CO.S.), Vol. 11, no. 12*

[19] J. Gillick, K. Tang, R.M. Keller, Learning Jazz Grammars, *Proceedings of the Sound and Music Computing Conference*, 2009.

[20] P. Pestana, Lindenmayer systems and the harmony of fractals, *Chaotic Modeling and Simulation (CMSIM) 1*, 2012.

[21] K. M. Kitani, H. Koike, ImprovGenerator: Online Grammatical Induction For On-The-Fly Improvisation Accompaniment, *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2010

[22] T.M. Oliwa, Genetic Algorithm and the abc Music Notation Language for Rock Music Composition, *Proceedings of the 10th annual conference on Genetic and evolutionary computation, ACM*, 2008

[23] D. Matic, A Genetic Algorithm for Composing Music, *Yugoslav Journal of Operations Research, Volume 20*, 2010.

[24] N. Fortier, M.V. Dyne, A Genetic Algorithm Approach to Improve Automated Music Composition, *International Journal of Computers, Issue 4, Volume* 5, 2011.

[25] K. Hastuti, A Method for Automatic Gamelan Music Composition, *International Journal of Advances in Intelligent Informatics Vol 2, No 1, 2016*.

[26] Sumarsam, *Introduction To Javanese Gamelan*, (Spiral-bound, 1988)

[27] R. Supanggah, *Bothekan Karawitan II: Garap*, (ISI Press, 2007).

[28] M. Pearce, G. Wiggins, Towards a framework for the evaluation of machine compositions, *Proceedings of the AISB 2001*, 2001.

[29] M. Benamou, Rasa: Affect and Intuition in Javanese Musical Aesthetics, Oxford University Press, 2010.

Authors' information

[1]Faculty of Mathematics and Natural Science, Universitas Gadjah Mada, Yogyakarta, Indonesia

[2]Faculty of Computer Science, Universitas Dian Nuswantoro, Semarang, Indonesia.

[3]Faculty of Performing Arts, Institut Seni Indonesa, Surakarta, Indonesia.

Khafiizh Hastuti received her M.Kom degree in 2012 from informatics Engineering Department of Universitas Dian Nuswantoro. Her research interests are artificial intelligent, data mining, and software engineering.
Email: afis@dsn.dinus.ac.id

Azhari Department of Computer Science and Electronics Universitas Gadjah Mada. Undergraduate Statistics Universitas Gadjah Mada, Master of Software Engineering Institut Teknologi Bandung, Doctor of Computer Science Universitas Gadjah Mada. Research interests are Intelligent agent, software engineering, project management.
Email: arisn@ugm.ac.id

Aina Musdholifah Department of Computer Science and Electronics Universitas Gadjah Mada.Undergraduate and master of computer science Universitas Gadjah Mada, Ph.D Computer Science Universiti Teknologi Malaysia (UTM). Research interests are genetic algorithm, fuzzy system.
Email: aina_m@ugm.ac.id

Rahayu Supanggah Professor of Ethnomusicology and Music Composition, at ISI Surakarta. Doctorat de 3ème cycle Etnomusicologie, diplomé 1985. Maestro of karawitan composer.
Email: supanggah@yahoo.com