

Human and Computation-based Musical Representation for Gamelan Music

Arry Maulana Syarif¹, Azhari Azhari^{2*}, Suprpto Suprpto³, Khafiizh Hastuti⁴

^{1,2,3}Department of Computer Science and Electronics
Universitas Gadjah Mada, Indonesia

^{1,4}Faculty of Computer Science
Universitas Dian Nuswantoro, Indonesia
e-mail: arry.maulana.s@mail.ugm.ac.id¹, arisn@ugm.ac.id²,
sprpto@ugm.ac.id³, afis@dsn.dinus.ac.id⁴

*Corresponding author: arisn@ugm.ac.id

Published: 20 November 2020

Cite this article (APA): Syarif, A. M., Azhari, A., Suprpto, S., & Hastuti, K. (2020). Human and computation-based musical representation for Gamelan music. *Malaysian Journal of Music*, 9, 82-100. <https://doi.org/10.37134/mjm.vol9.7.2020>

Abstract

A public database containing representative data of *karawitan* traditional music is needed as a resource for researchers who study computer music and *karawitan*. To establish this database, a text-based pitch model for music representation that is both human and computer-based was first investigated. A new model of musical representation that can be read by humans and computers is proposed to support music and computer research on *karawitan* also known as *gamelan* music. The model is expected to serve as the initial effort to establish a public database of *karawitan* music representation data. The proposed model was inspired by Helmholtz Notation and Scientific Pitch Notation and well-established, text-based pitch representation systems. The model was developed not only for pitch number, high or low or middle pitch information (octave information), but for musical elements found in *gamelan* sheet music pieces that include pitch value and legato signs. The model was named *Gending Scientific Pitch Notation* (GSPN). *Gending* is a Javanese word that means “song”. The GSPN model was designed to represent music by formulating musical elements from a sheet music piece. Furthermore, the model can automatically be converted to other music representation formats. In the experiment, data in the GSPN format was implemented to automatically convert sheet music to a binary code with localist representation technique.

Keywords: human and computation based, music representation, computer music, *gamelan*

Introduction

The goal of this research is to provide a public database containing music representation data of *karawitan*. To establish this database, a text-based pitch model for music representation that is both human and computer-based was first investigated. The model is meant to represent musical elements of *karawitan* where pitches and their attributes such as pitch number, pitch register or octaves, pitch values and the legato sign are numerically and alphabetically coded so that they can be read by humans and computers. This model as a representation of the musical parameters of *karawitan* is the first step towards setting up a public database.

Helmholtz Notations and Scientific Pitch Notation are pitch-naming systems known as ABC notations. These systems are used to represent pitches based on their octaves. ABC notations is a text-based pitch representation that can be read by humans and computers. It is different from musical symbols or numbered musical notations that cannot be read by a computer due to the use of symbols or

marks to represent musical elements including pitches. Helmholtz Notations and Scientific Pitch Notation are two formal pitch representation systems that are often used in music research involving a computer program for a part or for the entire process of music creation (Keith, 2010; Trueman, 2007), or in music information retrieval investigations with the objectives to obtain musical inferences resulting from musical features collected from an audio signal, a symbolic representation, or other sources (Downie, 2003).

As music and computer research grows the need for data is not only restricted to a pitch representation but also to other musical elements including note attribute information as well. The availability of data containing those musical representations is very helpful to researchers who research computer music. This kind of public database is commonly found for Western music but not in traditional music such as *karawitan*. *Karawitan* more popularly known as *gamelan* music, is a traditional musical expression from Java, Indonesia.

Karawitan has been an object of research in computer music for several decades (Becker and Becker, 1982; Hastuti et al., 2016 and 2017; Hughes, 1988; Surjodiningrat et al., 1979). Adopting Western notation technology to *karawitan* creates its own, different notation format. The Western notation format pushes the expression of *karawitan* into a different paradigm. In contrast to Western music, musical data representative of *karawitan* is rarely found especially in the form of public data. A database containing musical representation data of *karawitan* is needed as a resource with data that is ready to be used for research that looks at computer music with *karawitan* as its object of investigation.

Related Works

Musical elements consist of tempo, rhythm, pitch, timbre, dynamics, melody and harmony. Pitches are high or low tones that are measured in frequencies of hertz (Hz). In Western music, pitches labelled with the letters A, B, C, D, E, F or G are known as natural notes. The label is repeated from the lowest to the highest pitch known as an octave. The Helmholtz Notation and Scientific Pitch Notation (SPN) are text-based pitch representation models. Helmholtz Notation uses uppercase and lowercase letters followed by a comma or apostrophe to distinguish octaves, for example, C, - C, - C - c - c' - c'' - c''' and represents the lowest C to the highest C. Scientific Pitch Notation uses numbers to represent octaves, for example, C0 - C1 - C2 - C3 - C4 - C5 - C6 - C7. A pitch followed by the number 0 represents the lowest pitch while the higher number that follows a pitch represents the higher pitch. The difference in representation between Helmholtz Notation and Scientific Pitch Notation (SPN) can be seen in this example: The middle C would be C4 in SPN while in Helmholtz Notation it would be c'. In generating four-part harmonies using the Markov Decision Process, Yi and Goldsmith (2007) used commas and apostrophes following the Helmholtz Notation expression to represent high and low octaves. For instance C, is for the lowest C while C' is for highest C. This pitch representation is suitable for computer-based objectives. Compared to the Scientific Pitch Notation format, octaves represented by numbers are more easily readable by humans.

Eigenfeldt and Pasquier (2010) represented chords as pitch class sets that were converted into data arrays. The minor seventh chord with a value of (0 3 7 10) was placed in the first set of arrays. Thus, set 1 has a value of (0 3 7 10) where the value represents the minor seventh chord. Pachet and Roy (2011) generated melody for blues music by representing notes with integers, for example, C4 = 0 or C#4 = Db4 = 1. The representation was implemented in which a bar containing two chords was represented by a mark with a vertical line, a half bar was represented by a slash mark, and a bar containing one chord was represented by writing the chord twice. For instance, "F7 | Eh7/A7 | ..." was represented with "F7 F7 | Eh7/F7 | ...". It would be easier to add a label with a letter than to write the chord twice such as having the lowercase "x" added following a chord to represent a bar with one chord, so that "| F7 F7 |" can be written as "|F7x|".

Binary code was used to represent pitches in the form of vector data format as inputs. Todd (1989) explained that there were two types of pitch representations in binary code which were defined as distributed representation and localist representation. Furthermore, it was explained that in a distributed representation, note A might be coded as 100, note B as 110, and note C as 010. This code can be confusing to artificial neural network programs for the weight values generated from the code. Note A is closer to note B than to note C since note A and note B share two very close values at two

positions (A = **100**, B = **110**), while note A and C share one same value at one position (A = **100**, C = **010**). On the other hand, note C is closer to note B than to note A since note C and B share two same values at two position (B = **110**, C = **010**), while note C and A share one same value at one position (A = **100**, C = **010**). In the examples, bits written in bold illustrate the same values at the same positions. A localist representation treats all notes equally based on one active unit within one duration. Seven notes in a musical scale might be coded with 1000000 for note 1, 0100000 for 2, 0010000 for 3, ... and 0000001 for 7.

Hild et al. (1991) used the localist representation of pitch information which contains 12 attributes of harmonic functions. Attributes that were in a pitch were coded by 1 and vice versa. Makris et al. (2017) used the localist representation for drum and bass patterns. The bass was represented using four digits of binary code where the first digit was set to a value of 1 when the bass was played, and set to 0 for the rest event. The next three digits were for bass voice leading activation where 000 represented steady voice leading, 010 for upward voice leading, and 001 for downward voice leading. Colombo et al. (2016) represented melodies based on pitch and duration in the localist representation for automatic melody generation using deep recurrent neural networks. The end of the song and the silent beat features were represented by the value 0. Thus, if there are seven notes the end of the song and the silent beat were represented with 0000000. Liang et al. (2017) deliberately represented music using the pitch and rhythm without other musical elements for automatic stylistic composition of Bach Chorales using recurrent neural networks. It was explained that the purpose of this approach was to train the network to focus learning based on features and instead of being overtaken by music theory. Other systems derived musical representations directly from audio sources. Zhou and Lerch (2015) employed a chord detection system that used constraint Q transform to define frequency bins resulting from a digitised analogue signal and where the frequency bins were equal to a pitch scale. Park and Yoo (2017) extracted and approximated polyphonic melody order to result bins frequency, then to be converted to pitch classes using short-time fourier transform.

Western music technology, including notation systems based on European musical developments, has been accepted and adopted by the Javanese musician with adjustments based on the characteristics of *karawitan* (Becker, 1980). This includes the use of artificial intelligence in *karawitan*. Algorithmic composition, an approach of automatic music generation, has been implemented in *karawitan*. Surjodiningrat et al. (1979) used a statistical approach to set a note sequence of melody skeletal formulation based on bar patterns that frequently occur. Becker and Becker (1982) used a grammar method to set rules for a note sequence of melody skeletal generation while Hughes (1988) used a grammar method based on a quasi-linguistic framework to formulise note sequences of melody skeletal rules. Hastuti et al. (2017) used a rule-based method to define parameters as constraints in a note sequence of melody skeletal automatic generation using genetic algorithm. The dotted note as an additional note in the musical scale system of *karawitan* was converted into value 0 so that it can be processed in a sequential mining algorithm (Hastuti et al., 2016 and 2017). All the researchers focused on the skeletal melody. A skeletal melody is an abstraction of a melody and is similar to a melody chord in Western music. The representation of skeletal melody was commonly restricted on the pitch number without other musical elements or pitch attributes such as pitch value, pitch register and the legato sign.

Proposed Method

A model of music representation was proposed to support this research. The model is a text-based music representation so that the data it generates is readable by humans and computers. It was designed to collect data from music sheets. The model was named *Gendhing Scientific Pitch Notation* (GSPN). The term *gendhing* is based on the word for “song” in the Javanese language. The scientific pitch notation was added as this system was the inspiration in developing the model. Figure 1 shows the GSPN model diagram.

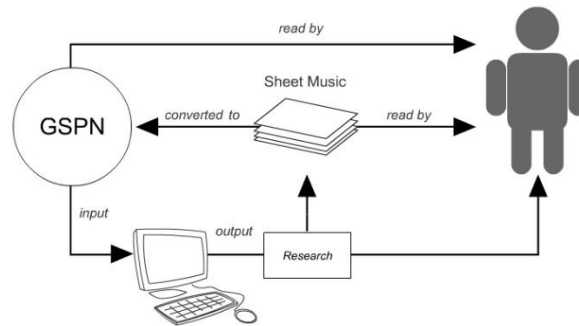


Figure 1. GSPN model

There are four phases in the development of the GSPN model, namely, the musical elements identification, the music representation formulation, the music representation implementation and evaluations.

Musical Elements Identification

In *karawitan* there are compositions that only consist of skeletal melody known as *balungan* (Javanese: skeleton), and compositions that consist of melody and its skeletal melody. Previous research conducted by Surjodiningrat et al. (1979), Becker and Becker (1982), Hughes (1988) and Hastuti et al. (2016 and 2017), focused on the skeletal melody containing pitch number information only. The GSPN model was designed to represent the music of *karawitan* both for the melody and the skeletal melody where the melody contains more pitch information such as pitch register, pitch value and the legato sign. The model development used note sequences of *gerongan* (a part for male chorus sung with the *gamelan*) as the corpus since this part contains melody and skeletal melody tracks. Musical elements identification was conducted to find attributes of pitches in the sheet music pieces. The results were used to define variables in the phase of GSPN formulation. An example of a piece of sheet music containing note sequences of melody and skeletal melody can be seen in Figure 2 with (a) as an abbreviation of the notes sequence of the melody skeletal and (b) as an abbreviation of the notes sequence of the melody.

Ladrang Kawuri, Laras Slendro Pathet Manyura

	(a)	3	3	•	•	3	1	2	3										
	(b)	•	•	•	•	6̣	6̣	•6̣	2	3	•	3	2	•1	1	23	3		
	(a)	6	5	3	2	•	1	2	6̣										
	(b)	•	•	6	5	6	3	1	2	•	1	2	1	3	2	1	6̣		
	(a)	1̇	1̇	•	•	2	3	2	1̇										
	(b)	•	•	•	•	6	5	6	3	1	1	•	1̇	•	1̇	6	2	•3	1̇
	(a)	3̇	2̇	6	5	2	2	3	2										
	(b)	•	•	•	1̇	•	2̇	6	5	6	2	•	2	1	3	•5	2		
	(a)	•	3	2	•	6̣	1	2	3										
	(b)	•	•	•	•	•	•	•	2	•	1	•5	5	6	5				
	(a)	•	5	6	5	1̇	6	5	3										
	(b)	•	3	•	•	6	1̇	2̇	3̇	•	6	3	5	6	6	1̇5	3		
	(a)	•	3	5	6	3	5	3	2										
	(b)	•	•	1̇	1̇	•	2̇	1̇6	1̇	6	•	5	3	6	•	2	2		
	(a)	6̣	1	2	3	2	1	2	6̣										
	(b)	•	•	6̣	2	3	•	56	3	6	56	2	1	•6̣	2	1	6̣		

Figure 2. Example of *karawitan* sheet music

The dataset contained 100 sheets music of the *gerongan* part from two musical scale systems of *slendro* and *pelog*. Data were collected from www.gamelanbvg.com. Each sheet music was analysed based on its structure, extracting the notes sequence as well as musical marks for every pitch. Given that the introductory parts vary and is dependent on the creativity of a performer, this section was excluded from the corpus. In theory, the formulation of GSPN could also be implemented for the introduction part. Other musical symbols excluded from the corpus included the curve symbol and the circle symbol together known as the colotomic that are placed above and around certain notes. The circle symbol signifies the time to hit the *gong* instrument. These musical symbols are guides for playing *gamelan* instruments that appear after a composition has been created. Thus, these symbols are not essential parameters for the process of creating the music.

In *karawitan* the musical scale system is called *laras* with two types of musical scale systems namely *slendro* and *pelog*. The *slendro* musical scale system consists of five pitches represented by notes numbered as 1, 2, 3, 5 and 6, while *pelog* consists of seven pitches represented by notes numbered as 1, 2, 3, 4, 5, 6 and 7. The tuning of pitches in *slendro* and *pelog* are different from what is used in Western music and also have different tunings. For example, the frequency of note 1 in *slendro* is different from note 1 in *pelog*, as well as the others. Apart from that, *gamelan* sets that have the same music scale do not share the same frequency range. In addition, there is also a dotted note (punctuation mark) to represent a silent moment. There is also a musical mode known as *pathet* which is determined by the dominant pitch or the most often played pitch in a certain beat. A typical *slendro* musical scale consists of three musical mode systems, namely *manyura*, *sanga* and *nem*. Additionally, the *pelog* musical scale consists of three musical mode systems namely *barang*, *lima* and *nem*. The *nem* musical mode system in *slendro* is different from *nem* in the *pelog*.

The pitch region in *karawitan* functions like an octave in Western music and is divided into low notes, middle notes and high notes. A dotted mark above a note indicates a high note, a dotted mark below a note indicates a low note and a note without a dotted mark is a middle note. The legato sign as used in Western music is also used in the dance notation system and means that two or more notes are played continuously and uninterrupted by subsequent notes.

A typical line contains two bars or *gatra* and a bar contains four beats. A beat is where the notes are placed. The number of notes in a beat is determined by the rhythm. For an example, a composition rhythmic value of 1/2 (a half note or minim) means that each beat in the composition has a value of 2. The total number of note value is the sum from two notes or more. The horizontal line above the note shows the value of the note. The bar divisions in the line is not shown explicitly. Calculation of the value of the rhythm and note can be used to define a bar.

Musical Representation Formulation

The identification of musical elements resulted with nine variables used for the GSPN formulation: the musical scale system, musical mode system, rhythm, line (phrasing), the composition duration, musical notes, pitch region, horizontal line (note values) and legato sign.

The *slendro* musical scale system contains three mode systems which are *manyura*, *nem* and *sanga* while the *pelog* musical scale contains three musical mode systems namely *barang*, *lima*, and *nem*. The *slendro* musical scale is abbreviated as S and the *pelog* scale is abbreviated as P. The musical mode systems in each musical scale system are coded with numbers and in alphabetical order: Code 1 for *manyura* in *slendro* and *barang* in *pelog*; code 2 for *nem* in *slendro* and *lima* in *pelog*; and code 3 for *sanga* in *slendro* and *nem* in *pelog*. The GSPN formulation for the musical scale and mode system with L for *laras* (musical scale system), S for *slendro*, and P for *pelog* is:

$$\begin{aligned} L &= \{P, S\} \\ S &= \{S_1, S_2, S_3\} \\ P &= \{P_1, P_2, P_3\} \end{aligned}$$

Rhythm as it is known, refers to the duration in a bar and it is associated with the notation value used. In this context, rhythm is divided into *lancar* or 1/1 the value of one beat (semibreve), *tanggung* or 1/2 for the value of two notes (minim), *wiled* or 1/4 for the value of four notes (crotchet), *dados* or

1/8 for the value of eight notes (quavers) and *rangkep* or 1/16 for sixteen notes (semiquavers). The GSPN formulation for rhythm is designated as R and is:

$$R = \{R_1, R_2, R_3, R_4, R_5\}$$

where:

$$\begin{aligned} R_1 &= 1/1 \\ R_2 &= 1/2 \\ R_3 &= 1/4 \\ R_4 &= 1/8 \\ R_5 &= 1/16 \end{aligned}$$

The objective of the development of the GSPN is to represent musical elements of *gamelan* and therefore includes how to write a composition title that is also formulated. The format for writing the composition title are sorted by musical scale, mode, and the rhythm connected with a hyphen between these three musical elements. For example, “*Ladrang Kawuri: S₁-R₂*” are read as a composition entitled “*Ladrang Kawuri*” played in *slendro* (S) scale with *manyura* (S₁) musical mode system, and the rhythm of *tanggung* (R₂ or 1/2). Another example is “*Lancaran Suwe Ora Jamu: P₃-R₁*” read as a composition entitled “*Lancaran Suwe Ora Jamu*” played in *pelog* (P) scale, with musical mode *nem* (P₃) and rhythm of *lancar* (R₁ or 1/1).

The *slendro* musical scale system is comprised of five notes, 1, 2, 3, 5, 6 while the *pelog* is seven notes typically, 1, 2, 3, 4, 5, 6, 7. Punctuation marks in both of the musical scale systems are converted into numeric ‘0’ for computation purposes. The note was coded using the union of two sets containing notes that are in *slendro* and *pelog*. Below is the GSPN formulation for the note number, where T stands for the note, T₁ for notes in *slendro* musical scale system, and T₂ stands for notes in the *pelog* scale system.

$$\begin{aligned} T_1 &= \{0, 1, 2, 3, 5, 6\} \\ T_2 &= \{0, 1, 2, 3, 4, 5, 6, 7\} \\ T &= T_1 \cup T_2 = \{0, 1, 2, 3, 4, 5, 6, 7\} \end{aligned}$$

Notes coded for both musical scale systems are written in as a set even though they are in different frequencies. The ‘if-then’ rules can process data from the musical scale system to differentiate the note and its frequency. For example, a given frequency dataset of *F slendro* = {fs₀, fs₁, fs₂, fs₃, fs₅, fs₆}, and *F pelog* = {fp₀, fp₁, fp₂, fp₃, fp₄, fp₅, fp₆, fp₇}. Thus, the ‘if-then’ rules were used and can be defined as:

$$\begin{aligned} \text{IF } (L = S) \\ \text{THEN } T &= \{0, 1, 2, 3, 5, 6\} \\ \text{AND } F &= \{fs_0, fs_1, fs_2, fs_3, fs_5, fs_6\} \end{aligned}$$

$$\begin{aligned} \text{IF } (L = P) \\ \text{THEN } T &= \{0, 1, 2, 3, 4, 5, 6, 7\} \\ \text{AND } F &= \{fp_0, fp_1, fp_2, fp_3, fp_4, fp_5, fp_6, fp_7\} \end{aligned}$$

The musical scales were distributed in three pitch regions as low, middle and high notes. A dotted mark below the note indicates low notes and are coded with a lowercase ‘a’. A dotted mark above the note indicates a high note and is coded with lowercase ‘b’. A note without any dotted mark indicates a middle note and is written without a dotted mark. Punctuation marks that are converted into ‘0’ do not belong to any pitch region since it represents a rest or silent moment notes are not to be played. A punctuation mark which represents a silent moment is written as in the middle note which is in the original form. Here is the GSPN formula for pitch region where W stands for the note region:

$$W = \{\emptyset, a, b\}$$

The note values are indicated by a horizontal line mark above the notes. Any notes without any horizontal line mark has value 1, any note with a horizontal line mark has value 1/2, and any note with double horizontal lines has value 1/4. The note values are represented using uppercase 'A' and 'B', where 'A' is for note value 1/2 and 'B' is for note value 1/4. Any notes with value 1 are written in its original form without 'A' or 'B'. The use of uppercase letters differentiates the note value from the pitch region codes that use lowercase letters. This is the GSPN formula for the note value where V stands for the note value:

$$V = \{\emptyset, A, B\}$$

The legato sign is coded based on the number of notes in the sign. The first note in a legato sign is coded with lowercase 'x' while the last note in a legato sign is coded with lowercase 'y'. Any notes that are not placed in any legato sign and any notes in between the first and the last note in a legato sign are written in its original form without 'x' or 'y'. Here is the GSPN formula for the legato sign with G stands for the note legato:

$$G = \{\emptyset, x, y\}$$

The format for the GSPN model in formulating the note sequence including the pitch attributes is:

$$T + W + V + G$$

Table 1 shows an example of music representation in the GSPN model for a composition titled, "Ladrang Kawuri", played in the scale of *slendro* and in *manyura* mode with rhythm of 1/2 (*tanggung*).

Table 1

GSPN music representation of "Ladrang Kawuri"

Ladrang Kawuri: S1-R2			
Lines	Bars	Skeleton	Melody
1	1-2	33003123	00006a6a0A6aA2x3y032x0A1Ay1x2A3Ay3
2	3-4	65320126	0065x6y3x1y201213216a
3	5-6	1b1b002321	000065Ax6Ay3Ax1bAy1b01b01bx6y2bx0A3bAy1b
4	7-8	3b2b652232	0001b02bx6y5x6y202x1y3x0A5Ay2
5	9-10	03206a123	00000000201x0A5Ay5x6y5x
6	11-12	05651b653	03y0061b2b3b06x3y5x6y6x1bA5Ay3
7	13-14	03563532	001b1b02bx1bA6Ay1bx6y053x6y022
8	15-16	6a1232126	006a2x3y05Ax6Ay3x65A6Ay21x0A6aAy2x1y6a

Musical Representation Implementation

The implementation of the GSPN model was conducted in two phases. The first phase represents the sheet music data using the GSPN format. The second phase converts the GSPN into a binary code pitch representation using localist representation techniques.

The dataset used sheet music from *gerongan*, a type of composition that contains melody and its skeletal melody. All the sheet music pieces were converted to the GSPN format. The process was manually conducted through manually writing the codes for all the musical elements based on the rules of GSPN. The melody and its skeletal melody were coded separately. The codes for musical elements in a melody that consist of lines and bars were concatenated into a single line code. This process was also used for the skeletal melody. Table 2 below shows an example of the concatenation process of coding for the melody and its skeletal melody of “*Ladrang Kawuri*”, a composition that was coded in Table 1 above.

Table 2

Concatenation process of coding for melody and melody skeletal

Ladrang Kawuri: S₁-R₂	
Melody Skeletal	Melody
33003123653201261b1b002321	00006a6a0A6aA2x3y032x0A1Ay1x2A3Ay30065x6y
3b2b65223203206a12305651b6	3x1y201213216a000065Ax6Ay3Ax1bAy1b01b01bx6
53035635326a1232126	y2bx0A3bAy1b0001b02bx6y5x6y202x1y3x0A5Ay2
	00000000201x0A5Ay5x6y5x03y0061b2b3b06x3y5
	x6y6x1bA5Ay3001b1b02bx1bA6Ay1bx6y053x6y02
	2006a2x3y05Ax6Ay3x65A6Ay21x0A6aAy2x1y6a

The GSPN model allows for all musical elements from music sheets to be used for computational processes. The method of extracting data information from the GSPN format is conducted by splitting the input by characters. Later the coding approach is used to define musical elements and note attributes based on the formula of T + W + V + G. The extraction was conducted by pairing the notes with their attributes in the form and order of (T + W), (T + V) and later (T + G).

The experiment on the GSPN model implementation was conducted by developing a formula to be implemented in a computer program for processing data in the GSPN format. The program was expected to accurately extract data information from the GSPN format and to accurately convert the data that was produced via the extraction and back into the music sheet format in its original structure. The program was designed to encompass note attributes in this way:

- MT = {note number}
- MW = {note region}
- MV = {note value}
- MG = {note legato}

The length of MT is equal to MW, MV and MG, where MT consists of data of the original note number of each pitch, MW data of the note region value of each pitch, MV the note value of each pitch, and MG data of legato notes value of each pitch.

The melody part (see Figure 2) is an example of GSPN model implementation used to simulate the data extraction method. The first step was to declare variables representing pitch attributes in numerical format as in the following:

- T in GSPN = {0, 1, 2, 3, 4, 5, 6, 7}
- T in number format = {0, 1, 2, 3, 4, 5, 6, 7}
- W in GSPN = {∅, a, b}
- W in number format = {0, 1, 2}
- V in GSPN = {∅, A, B}
- V in number format = {1, 0.5, 0.25}
- G in GSPN = {∅, x, y}
- G in number format = {0, 1, 2}

The next step involved splitting the GSPN data by character as in the following example:

```
Input GSPN data =
"0000660A6Bx1bBy5x06y1b2bx0A3bAy1bAx2bA1by600003b3b3bBx5bB2bAy1x02by1bBx2bB6Ay
3x0A5Ay2Ax5A3y20000660A6Bx1bBy5x06y1b2bx0A3bAy1bAx2bA1by6001b2bx1bA6Ay3Ax5A3
y2003Ax5Ay301Ax2A1y6a"
melody data = "input".split("");
melody data = {0, 0, 0, 0, 6, 6, 0, A, 6, B, x, 1, b, B, y, 5, x, 0, 6, y, 1, b, 2, b, x, 0, A, 3, b, A, y, 1, b, A,
x, 2, b, A, 1, b, y, 6, 0, 0, 0, 3, b, 3, b, 3, b, B, x, 5, b, B, 2, b, A, y, 1, x, 0, 2, b, y, 1, b, B, x, 2, b, B, 6,
A, y, 3, x, 0, A, 5, A, y, 2, A, x, 5, A, 3, y, 2, 0, 0, 0, 6, 6, 0, A, 6, B, x, 1, b, B, y, 5, x, 0, 6, y, 1, b, 2,
b, x, 0, A, 3, b, A, y, 1, b, A, x, 2, b, A, 1, b, y, 6, 0, 0, 1, b, 2, b, x, 1, b, A, 6, A, y, 3, A, x, 5, A, 3, y, 2,
0, 0, 3, A, x, 5, A, y, 3, 0, 1, A, x, 2, A, 1, y, 6, a}
```

The note number data (MT) extraction is used to only identify the notes in melody data. This data contains information about notes, note value, note region and legato. MT extraction was conducted by removing all attributes that follow the note number using this formula:

$$MT = \bigcup_{k=1}^{k \leq P} \{x_k \in X \mid x_k \in T\}$$

MT = {note number}

T = a set of musical scale = {0, 1, 2, 3, 4, 5, 6, 7}

X = {melody data}

P = Number of set X

The results of the original note data extraction were:

```
MT = {0, 0, 0, 0, 6, 6, 0, 6, 1, 5, 0, 6, 1, 2, 0, 3, 1, 2, 1, 6, 0, 0, 0, 0, 3, 3, 3, 5, 2, 1, 0, 2, 1, 2, 6, 3, 0, 5, 2,
5, 3, 2, 0, 0, 0, 0, 6, 6, 0, 6, 1, 5, 0, 6, 1, 2, 0, 3, 1, b, 2, b, 1, b, 6, 0, 0, 1, b, 2, b, 1, b, 6, 3, 5, 3, 2, 0, 0, 3,
5, 3, 0, 1, 2, 1, 6}
```

The total number of notes in a melody can be used to calculate the length of MT elements. Humans can read the note number data by interpreting 0 as a punctuation mark while the GSPN format and the computer program defined a punctuation mark as 0. All notes other than the punctuation mark are read in their original notes:

GSPN: {0, 0, 0, 0, 6, 6, 0, 6, 1, ..., 6}

Humans: {., ., ., ., 6, 6, ., 6, 1, ..., 6}

Computers: {0, 0, 0, 0, 6, 6, 0, 6, 1, ..., 6}

The note region data (MW) extraction was executed based on the structure of the note number (T) and note region (W). Later, the data from MW was converted to numerical format. Any subsets of MW that only consist of the element of T are converted to value 0. Any subsets that consist of a pair of elements of T and 'a' are converted to value 1 and any subsets that consist of a pair of elements of T and 'b' are converted to value 2. The values of 0, 1 and 2 represent middle note, low note and high note respectively. The following formulas are used to extract note region data and to convert information into numerical format:

$$WX = \bigcup_{k=1}^{k \leq P} \{x_k \in X \mid x_k \in T \cup W\}$$

$$WX = WX \cup \{\emptyset\}$$

$$WY = \bigcup_{k=1}^{k < N} \{(y_k, y_{k+1}), y_k, y_{k+1} \in Y \mid y_k \in T, y_{k+1} \in T \cup W\}$$

$$MW = \bigcup_{k=1}^{k \leq M} \{Z_x \mid Z_x = f((x_k, y_k)), (x_k, y_k) \in WY\}$$

$$f((x_k, y_k)) = \begin{cases} 1, & y_k = a \\ 2, & y_k = b \\ 0, & \text{else} \end{cases}$$

T = a set of musical scale = {0, 1, 2, 3, 4, 5, 6, 7}
 W = {∅, a, b}
 X = {melody data}
 P = Number of set X
 WX = {melody data extraction containing elements of T or W}
 Y = WX
 N = Number of set Y
 WY = {subsets resulted from WX}
 Z = {WY}
 M = Number of set Z
 MW = {note region data in number format based on WY}

The formula above resulted in producing WX with note number (T) and note region (W) data only. WY encompasses subsets that contain (elements of T, elements of T) or (elements of T, elements of W) and (Elements of T, ∅) for the last note. Furthermore, MW is determined based on the second element of subsets in WY with value 0 if the second element of a subset in WY is an element of T or ∅, 1 if the second element is of a subset in WY as ‘a’, and 2 if the second element of a subset in WY is ‘b’. The results of each process of note region data extraction in GSPN format were:

melody data = {0, 0, 0, 0, 6, 6, 0, A, 6, B, x, 1, b, B, y, ..., 6, a}
 WX = {0, 0, 0, 0, 6, 6, 0, 6, 1, b, ..., 6, a, ∅}
 WY = {(0,0) (0,0) (0,0) (0,6) (6,6) (6,0) (0,6), (6,1), (1, b), ..., (6, a)}
 MW = {0, 0, 0, 0, 0, 0, 0, 0, 2, ..., 1}

Humans can interpret the code ‘a’ as low note, ‘b’ as high, and the note that is not followed by any coding as the middle note. In the computer program, the codes are converted into numerical format. The notes without any code are converted into value of numerical 0, code “a” the value of 1, and ‘b’ the value of 2. Below are the examples of note region data extraction in GSPN format, read by human and the computer. The conversions are written in a form of *note number or punctuation mark-note region* with ‘mn’ represent middle note, ‘ln’ for low and ‘hn’ stands for high notes.

GSPN: {0, 0, 0, 0, 6, 6, 0, 6, 1b, ..., 6a}
 Humans: {.-mn, .-mn, .-mn, .-mn, .-mn, 6-mn,.-mn, 6-mn, 1-hn, ..., 6-ln}
 Computer: {0-0, 0-0, 0, 0-0, 0-0, 6-0, 6-0, 0-0, 6-0, 1-2, ..., 6-1}

The process of note region data extraction was implemented to extract the note value data (MV). The note value data (MV) extraction was employed based on the structure of note number data (T) and note value data (V). The following equations were used to extract note value data and to convert the data into numerical format:

$$VX = \bigcup_{k=1}^{k \leq P} \{x_k \in X \mid x_k \in T \cup V\}$$

$$VX = VX \cup \{\emptyset\}$$

$$VY = \bigcup_{k=1}^{k < N} \{(y_k, y_{k+1}), y_k, y_{k+1} \in Y \mid y_k \in T, y_{k+1} \in T \cup V\}$$

$$MV = \bigcup_{k=1}^{k \leq M} \{Z_x | Z_x = f((x_k, y_k)), (x_k, y_k) \in VY\}$$

$$f((x_k, y_k)) = \begin{cases} 0.5, & y_k = A \\ 0.25, & y_k = B \\ 1, & \text{else} \end{cases}$$

where:

T = a set of musical scale = {0, 1, 2, 3, 4, 5, 6, 7}

V = {∅, A, B}

X = {melody data}

P = Number of set X

VX = {melody data extraction containing elements of T or V}

Y = VX

N = Number of set Y

VY = {subsets resulted from VX}

Z = {VY}

M = Number of set Z

MV = {note value data in number format based on VY}

The formula above resulted in VX comprising note number (T) and note region (V) data only. The VY encompasses subsets that contain (elements of T, elements of T) or (elements of T, elements of V), and (Elements of T, ∅) for the last note. Further, MV is determined based on the second element of subsets in VY with value 0 if the second element of a subset in VY is element of T or ∅, 1 if the second element of a subset in VY is 'A', and 2 if the second element of a subset in VY is 'B'. The results of each process of note value data extraction in the GSPN format were derived via:

melody data = {0, 0, 0, 0, 6, 6, 0, A, 6, B, x, 1, b, B, y, ..., 6, a}

VX = {0, 0, 0, 0, 6, 6, 0, A, 6, B, 1, B, ..., 6, ∅}

VY = {(0,0) (0,0) (0,0) (0,6) (6,6) (6,0) (0, A), (6, B), (1, B), ..., (6,∅)}

MV = {1, 1, 1, 1, 1, 1, 0.5, 0.25, 0.25, ..., 1}

Humans can read data by interpreting code 'A' as the note with a value of 1/4, 'B' as note value of 1/8, and notes not associated with any code value of 1. In the computer program the codes are converted into numerical format. Notes without any code might be converted into numerical value of 0, 'A', or converted into value 1 and 'B' into value 2. Examples of the note value data extraction in GSPN format to be read by humans and converted by computers and presented in the form of *note number or dot note-note value* are:

GSPN: {0, 0, 0, 0, 6, 6, 0A, 6B, 1B, ..., 6}

Humans: {.-1, .-1, .-1, .-1, 6-1, 6-1, .-0.5, 6-0.25, 1-0.25, ..., 6-1}

Computers: {0-1, 0-1, 0-1, 0-1, 0-1, 6-1, 6-1, 0-0.5, 6-0.25, 1-0.25, ..., 6-1}

In the following example, the rhythm value R is R₂ or 1/2 or a beat with note values of 2, and if a bar consists of four beats the number of the note values for each bar is 8. Thus, the note value (V) can be used to detect beat, bar, line and composition duration as is illustrated in the following Figure 3 illustration.

0	0	0	0	6	6	0A	6Bx	1bBy	5x
1	1	1	1	1	1	1/2	1/4	1/4	1
Beat 1		Beat 2		Beat 3		Beat 4			
Bar 1									
0	6y	1b	2bx	0A	3bAy	1bAx	2bA	1by	6
1	1	1	1	1/2	1/2	1/2	1/2	1	1
Beat 5		Beat 6		Beat 7				Beat 8	
Bar 2									

The first line of
a composition entitled
'Ladrang Wilujeng'

Figure 3. Illustration of beat, bar, line and melody length detection

Based on *karawitan* theory the number of beats in a bar is a constant value of 4 while the number of bars in a line is a constant value of 2. The number of beats in a line are the number of beats in a bar multiplied by the number of bars in the line which is $4 \times 2 = 8$.

The rhythm value was used to calculate the number of lines since it defines the number of note values in a beat. The denominator in the rhythm value was used as a constant value to calculate the number of lines. For instance, the value of 2 is the denominator for a rhythm of 1/2. Thus, the value of 2 is used as the constant value to calculate the number of lines. The following formulas were used to calculate the number of lines, bars and number of beats in the composition.

$$kr = 4$$

$$rb = 2$$

$$kb = (kr \cdot rb)$$

$$vm = \sum_{k=1}^{k \leq p} MV_k$$

$$jb = vm / (Rt \cdot kb)$$

$$jr = (jb \cdot rb)$$

$$jk = (jr \cdot kr)$$

$$vb = (vm / jb)$$

$$vr = (vb / rb)$$

$$vk = (vr / kr) = Rt$$

where:

- Rt = denominator value of the rhythm value
- MV = note region value data
- kr = the number of beats in a bar = 4
- rb = the number of bars in a line = 2
- kb = the number of beats in a line = $(kr \cdot kb) = 8$
- jb = the number of lines in a melody
- jr = the number of bars in a melody.
- jk = the number of beats in a melody.
- vm = Total number of notes value in a melody
- vb = Total number of notes value in a line
- vr = Total number of notes value in a bar.
- vk = Total number of notes value in a beat = Rt

The process of note region and note value data extractions was used to extract data on legato (MG). The legato (MG) extraction was conducted based on the structure of note number (T) as well as legato (G) data. The formulas below were used to extract legato data and to convert the extracted data into numerical format:

$$GX = \bigcup_{k=1}^{k \leq P} \{x_k \in X \mid x_k \in T \cup G\}$$

$$GX = GX \cup \{\emptyset\}$$

$$GY = \bigcup_{k=1}^{k < N} \{(y_k, y_{k+1}), y_k, y_{k+1} \in Y \mid y_k \in T, y_{k+1} \in T \cup G\}$$

$$MG = \bigcup_{k=1}^{k \leq M} \{Z_x \mid Z_x = f((x_k, y_k)), (x_k, y_k) \in GY\}$$

$$f((x_k, y_k)) = \begin{cases} 1, & y_k = x \\ 2, & y_k = y \\ 0, & \text{else} \end{cases}$$

where:

T = a set of musical scale = {0, 1, 2, 3, 4, 5, 6, 7}

G = {∅, x, y}

X = {melody data}

P = Number of set X

GX = {melody data extraction containing elements of T or G}

Y = GX

N = Number of set Y

GY = {subsets resulted from GX}

Z = {GY}

M = Number of set Z

GW = {legato data in number format based on GY}

The formulas above produced the result of GX comprising note number (T) and legato (G) data only. The GY encompasses subsets that contain (elements of T, elements of T) or (elements of T, elements of G), and (Elements of T, ∅) for the last note. Furthermore, MG is determined based on the second element of subsets in GY with value 0 if the second element of a subset in GY is 'x', 1 if the second element of a subset in GY is 'x', and 2 if the second element of a subset in GY is 'y'. The results of each process of note value data extraction in the GSPN format were:

melody data = {0, 0, 0, 0, 6, 6, 0, A, 6, B, x, 1, b, B, y, ..., 6, a}

GX = {0, 0, 0, 0, 6, 6, 0, 6, x, 1, y, ..., 6, ∅}

GY = {(0,0) (0,0) (0,0) (0,6) (6,6) (6,0) (0,6), (6,1), (1, 2), ..., (6, ∅)}

MG = {0, 0, 0, 0, 0, 0, 1, 2, ..., 0}

Humans might interpret code 'x' as signifying where the note legato started, code 'y' as where it ends, and a note not followed by any code as non-legato or notes between x and y. By using a computer program the codes are converted into numerical format. A note without any codes is converted into the numerical value of 0, 'x' into the value of 1 and 'b' into numerical value 2. Table 3 immediately below shows an example of legato data extraction in GSPN format to be read by humans and computers. The conversions are written in term of note number or punctuation mark-legato where 'nl' stands for non-legato, 'sl' for start of the legato and 'el' for the end of legato.

GSPN: {0, 0, 0, 0, 6, 6, 0, 6x, 1y, ..., 6}

Humans: [-nl, .nl, .nl, .nl, 6-nl, 6-nl, 0-nl, 6-sl, 1-el, ..., 6-nl]

Computers: [0-0, 0-0, 0-0, 0-0, 6-0, 6-0, 0-0, 6-1, 1-2, ..., 6-0]

Table 3

GSPN format to localist representation

GSPN Format	Computation Values		Localist Representation		
	T	W	T	W	(T+W)
0	0	0	10000000	100	10000000100
0	0	0	10000000	100	10000000100
0	0	0	10000000	100	10000000100
0	0	0	10000000	100	10000000100
6a	6	1	00000010	010	00000010010
6a	6	1	00000010	010	00000010010
0A	0	0	10000000	100	10000000100
6aA	6	1	00000010	010	00000010010
...
3	3	0	00010000	100	00010000100

GSPN to Binary Code Conversion

The GSPN model was designed to support a computation process. Thus, it should be able to support an automatic conversion to binary code format. The model allows all data from musical elements to be converted by using a computational process.

An experiment of converting GSPN to binary code was conducted using localist representation technique. The following explanation is an example of GSPN conversion to localist representation with two note attributes, namely the musical scale and pitch region. There were eight notes comprised of 0, 1, 2, 3, 4, 5, 6, and 7. Thus there were eight inputs of the musical scale in the localist representation. Note 0 is represented with 10000000, note 1 with 01000000, note 2 with 00100000, ..., note 7 with 00000001. The pitch region coded as “a” was denoted by 010, the middle note written in its original number was represented by 100, and high note coded with “b” and represented by 001. The localist representation for the musical scale and the pitch region used 11 inputs, consisted of 8 inputs for the musical scale, and 3 inputs for the pitch region. For example, the middle note of 2 was represented by 01000000100 and the high note of 2 was represented by 01000000001.

The experiment shows that all data in the GSPN format were successfully converted into binary code format by means of localist representation technique as shown in Table 2 with T standing for the musical scale, and W for the pitch region.

Evaluation

The GSPN model was evaluated using two techniques. The first technique involved calculating the note value to detect beat, bar, line and melody length where the resulting values were expected to fulfill the numerical-based structure of a *gamelan* melody. The second approach involved developing a computer program to generate sheet music based on the GSPN data format. Sheet music generated by a computer program was expected to follow the original sheet music structure and any error caused by manually converting the sheet music into the GSPN format could be detected using these techniques. A single

mistake made in manually typing in the code of the GSPN would produce wrong data. For example, 6B is for a middle note of 6 which has a note value of 1/4. If it is accidentally written as 6b, the note 6 will have a note value of 1. This can then lead to a wrong calculation of total note values in a composition along with a wrong distribution of notes, beats, bars and lines.

The pseudocode used at an earlier stage for the evaluation of note sequence per beat, bars and lines distribution including checking the notes value of every beat in a melody is:

```

z = 0;
while (a < jb) {
  for (b = 0; b < jt - a; b++) {
    c += MV [b + a];
    if (c <= vk) {
      MTK [d].push(MT [b + a]);
    }
  }
  for (b = 0; b < MTK [d].length; b++) {
    MVk += MV [d] [b];
  }

  MVr.push (MVk);
  a += MTK [d].length;
  d += 1;
  d = 0;
  vk = 0;
  z++
}

```

where:

jb = the number of lines in a melody
 jt = the number of notes in a melody
 MV = note value data
 MT = note number data
 vk = Total number of notes value in a beat = Rt
 MTK = origin notes per beat
 MVK = Total number of notes value per beat
 MVr = Total notes value per bar

Below is an example of implementation of the pseudocode above for the first line of a composition played in the rhythm of 1/2. The line is showed in the Figure 4.

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

Figure 4. A line used as an example of evaluation using the first technique

The first evaluation calculates note values of each note to detect beat, bar, line and composition length. The results of the implementation of the first evaluation technique for the first line of the composition above are shown immediately below. In this example, each beat will have total a note value of 2 since the rhythm value is 1/2, each bar will have total note value of 8 since a bar consists of 4 beats and each line will have total note value of 16 since a line consists of 2 bars.

```

MTk in r1 = [[0, 0], [0, 0], [6, 6], [0, 6, 1, 5]];
MVk in r1 = [[1, 1], [1, 1], [1, 1], [0,5, 0,25, 0,25, 1]];
MVr in r1 = [[2], [2], [2], [2]];
vr in r1 = [8];
MTk in r2 = [[0, 6], [1, 2], [0, 3, 1, 2], [1, 6]]
MVk in r2 = [[1, 1], [1, 1], [0,25, 0,25, 0,25, 0,25], [1, 1]];

```

$MVr \text{ in } r2 = [[2], [2], [2], [2]];$
 $vr \text{ in } r2 = 8;$
 $MTb \text{ in } b1 = MTr \text{ } r1 * MTr \text{ } r2;$
 $MVb \text{ in } b1 = MVb \text{ } r1 * MVb \text{ } r2;$
 $vr \text{ } b1 = vr \text{ } r1 + vr \text{ } r2;$

where:

$r = \text{bar}$
 $b = \text{line}$
 $MTk = \text{note sequence per beat data}$
 $MVk = \text{notes value per beat data}$
 $MTr = \text{note sequence in a bar data}$
 $MVr = \text{notes value per bar data}$
 $Vr = \text{total number of notes value per bar}$
 $MTb = \text{note sequence in lines}$

The results from the first evaluation were compared to the original sheet music. The result of the comparisons exhibited that 100 sheet music pieces in GSPN format are similar to the original sheet music. Table 4 below presents an example of a result from the first evaluation derived from a composition labelled with ID number 001 in the dataset, where Y in the status column represents “yes,” which means that the data is similar to the original sheet music.

Table 4

An example of comparison evaluation results

Data	Status
ID: 001	Y
Number of lines: 8	Y
Total note value per lines: 16, 16, 16, 16, ..., 16	Y
Total note value of lines: 128	Y
Number of bars: 16	Y
Total note value per bars: 8, 8, 8, 8, 8, 8, 8, 8, ..., 8	Y
Total note value of bars: 128	Y
Number of beats: 64	Y
Total note value per beats: 2, 2, 2, 2, 2, 2, 2, ..., 2	Y
Total note value of beats: 128	Y

The second evaluation was completed by adding a function to the computer program to display data from the GSPN into sheet music format. The comparison was conducted based on a structure of the note sequence and its attributes generated from the computer program and into the original music sheet. The results of the evaluation show that all 100 sheet music pieces generated from the computer program have a degree of similarity in their structure to the original sheet music.

Ladrang Kawuri, Laras Slendro Pathet Manyura

```

      - - - - -
. . . . 6 6 . 6 2 3 . 3 2 . 1 1 2 3 3
      x y x y x y x y x y x y
. . 6 5 6 3 1 2 . 1 2 1 3 2 1 6
      x y x y x y x y x y x y
. . . . 6 5 6 3 1 1 . 1 . 1 6 2 . 3 1
      x y x y x y x y x y x y x y
. . . . 1 . 2 6 5 6 2 . 2 1 3 . 5 2
      x y x y x y x y x y x y
. . . . . . . . . 2 . 1 . 5 5 6 5
      x y x y x y x y x y x y
. 3 . . 6 1 2 3 . 6 3 5 6 6 1 5 3
      y x y x y x y x y x y x y
. . 1 1 . 2 1 6 1 6 . 5 3 6 . 2 2
      x y x y x y x y x y x y
. . 6 2 3 . 5 6 3 6 5 6 2 1 . 6 2 1 6
      x y x y x y x y x y x y x y

```

Figure 5. An example of a sheet music piece generated from a computer program

Figure 5 above shows an example of a sheet music piece generated from the program based on its data from the GSPN format. In the example, the legato sign was displayed with letter x for beginning of the legato and y for the end of the legato. The sheet music was generated based on data for a melody displayed in Figure 2.

Conclusion and Future Works

A music representation model for the *karawitan* known as *Ghending Scientific Pitch Notation* (GSPN) was developed to represent musical elements from sheet music to a format that is readable by humans as well as computers. The GSPN model was successfully implemented through a simple computer program that can read data from the GSPN format and later display the data in *karawitan* sheet music format. The GSPN data also successfully converted the sheet music and was represented by binary numbers.

In the future the GSPN data will be used to build XML-based representation for *karawitan*, where the work will encompass images of music sheet, or audio, as sources for an automatic music representation. This type of work can be seen in a work of MusicXML by Michael Good, a standard open format for exchanging digital sheet music between applications (Good, 2013). The manual technique to convert sheets music to the GSPN format may result any mistyping. Thus, an image processing approach and pattern recognition method can be implemented to capture *karawitan* sheet music to be automatically converted to the GSPN format.

In the future, the research will also focus on the development of a *karawitan* music representation database in the form of a GSPN format. The database will be published online in order to make it available and ready for computational processes. This resource will support research projects involving computer music and *karawitan*.

Examples of music representation of *karawitan* in the form of GSPN and binary representation can be accessed at <https://drive.google.com/drive/folders/1O16OXD7N5HoPdQ8Gj2WLAYr5BzHZo36V>

References

- Becker, J. (1980). *Traditional Music in Modern Java: Gamelan in a Changing Society*. The University Press of Hawaii. <https://doi.org/10.2307/j.ctv9zcyj8.9>
- Becker, J., & Becker, A. (1982). *A Grammar of the Musical Genre Srepegan*. University of Texas Press.
- Colombo, F., Muscinelli, S.P., Seeholzer, A., Brea, J., & Gerstner, W. (2016). Algorithmic Composition of Melodies with Deep Recurrent Neural Networks. *Proceedings of 1st Conference on Computer Simulation of Musical Creativity, UK, 1, 2-12*.
- Downie, J. S. (2003). Music Information Retrieval. *Annual Review of Information Science and Technology, 37(1)*, 295-340.
- Eigenfeldt, A., & Pasquier, P. (2010). Realtime Generation of Harmonic Progressions Using Controlled Markov Selection. *Proceedings of the First International Conference on Computational Creativity, Portugal, 1, 16-25*.
- Good, M. (2013). Exchange and Publish Scores with MusicXML. <https://www.musicxml.com/publications/>
- Hastuti, K., Azhari, A., Musdholifah, A., & Supanggih, R. (2016). Building Melodic Feature Knowledge of Gamelan Music Using Apriori Based on Functions in Sequence (AFiS) Algorithm. *International Review on Computers and Software, 11(12)*, 1127-1137. <https://doi.org/10.15866/irecos.v11i12.10841>
- Hastuti, K., Azhari, A., Musdholifah, A., & Supanggih, R. (2017). Rule-Based and Genetic Algorithm for Automatic Gamelan Music Composition. *International Review on Modelling and Simulations, 10(3)*, 2010-212. <https://doi.org/10.15866/iremos.v10i3.11479>
- Hild, H., Feulner, J., & Menzel, W. (1991). HARMONET: A Neural Net for Harmonizing Chorales in the Style of J.S.Bach. *Proceedings of Advances in Neural Information Processing Systems, 4*, 267-274.
- Hughes, D. W. (1988). *Deep Structure and Surface Structure in Javanese Music: A Grammar of Gending Lampah*. University of Illinois Press.
- Keith, S. (2010). Bridging the Gap: Thoughts on Computer Music and Contemporary (Popular) Electronic Music. *Proceedings of the 2010 Australasian Computer Music Conference, Australia, 2010*, 37-42.
- Liang, F., Gotham, M., Johnson, M., & Shotton, J. (2017). Automatic Stylistic Composition of Bach Chorales with Deep LSTM. *18th International Society for Music Information Retrieval Conference, Suzhou, China, 18*, 449-456.
- Makris D., Kaliakatsos-Papakostas M., Karydis I., & Kermanidis K.L. (2017). Combining LSTM and Feed Forward Neural Networks for Conditional Rhythm Composition. In Boracchi G., Iliadis L., Jayne C., & Likas A. (Eds.), *Engineering Applications of Neural Networks. EANN 2017: vol. 744. Communications in Computer and Information Science* (pp. 570-582). Springer. https://doi.org/10.1007/978-3-319-65172-9_48
- Pachet, F., & Roy, P. (2011). Markov constraints: Steerable generation of Markov sequences. *Constraints, 16*, 148-172.
- Park, H., Yoo, C. (2017). Melody Extraction and Detection through LSTM-RNN with Harmonic Sum Loss. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, 2017*, 2766-2770, <https://doi.org/10.1109/ICASSP.2017.7952660>
- Surjodiningrat, W., Khandelwal V.K., & Soesianto, F. (1977). *Gamelan dan Komputer: Analisa Patet dan Komposisi Gending Jawa Laras Slendro*. Universitas Gadjah Mada.
- Todd, P. M. (1989). A Connectionist Approach to Algorithmic Composition. *Computer Music Journal, 13(4)*, 27-43.
- Trueman, D. (2007). Why a Laptop Orchestra. *Cambridge Journal, 12*, 171 – 179.
- Yi, L., & Goldsmith, J. (2007). Automatic Generation of Four-part Harmony. *Proceedings of the Fifth UAI Conference on Bayesian Modeling Applications Workshop, Canada, 268*, 81–86.
- Zhou, X., & Lerch, A. (2015). Chord Detection using Deep Learning. *16th International Society for Music Information Retrieval Conference, USA, 16*, 52-58.

Biography

Arry Maulana Syarif is a doctoral candidate in Computer Science Department of Universitas Gadjah Mada, and a lecturer at Department of Computer Science Universitas Dian Nuswantoro. His research interests are data mining, artificial intelligence and machine learning.

Azhari Azhari is currently teaching at the Department of Computer Science and Electronics Universitas Gadjah Mada. His research interests are Intelligent agent, software engineering and project management.

Suprpto Suprpto is currently teaching at the Department of Computer Science and Electronics Universitas Gadjah Mada. His research interests are machine learning, artificial intelligence, and graph theory.

Khafiizh Hastuti is a lecturer at the Faculty of Computer Science, Universitas Dian Nuswantoro and also the founder of Gamelan Research Institute. Her research interests are artificial intelligence, software engineering, project management and algorithmic composition.