Natural Automatic Musical Note Player

by Ahmad Zainul Fanani

Submission date: 08-Apr-2020 08:30PM (UTC+0700)
Submission ID: 1292709761
File name: 6_Jurnal_Telkomnika.pdf (599.98K)
Word count: 5192
Character count: 25125

TELKOMNIKA, Vol.17, No.1, February 2019, pp.235~245
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v17i1.11606

Natural automatic musical note player using time-frequency analysis on human play

Khafiizh Hastuti*[1], Ary Maulana Syarif[2], Ahmad Zainul Fanani[3], Aton Rustandi Mulyana[4]
[1,2,3]Faculty of Computer Science/Universitas Dian Nuswantoro, Semarang, Indonesia
[4]Faculty of Performing Art/Institut Seni Indonesia, Surakarta, Indonesia
*Corresponding author, e-mail: afis@dsn.dinus.ac.id

Abstract

This research aims to develop an automatic gamelan musical note player that can naturally play musical note as human does. A musician estimates time to hit an instrument button in an approximate time which is as close as to the target time. The tolerated time to play a note was identified based on the human play. A gamelan musician was selected to play five note sequences of songs, and the play was recorded to be analyzed. Execution time in hitting instrument buttons in human play was identified using time-frequency analysis and peak detection to define time range which can be tolerated as time value that not too fast or not too late in hitting buttons, and then the result of the analysis was used as parameters to randomize approximate time to play a note. The evaluation shows that the program played all note sequences in the approximate time as human does and the program played more natural and better than another program which played a note as exact as its time target.

Keywords: *automatic musical note player, gamelan music, peak detection, time-frequency analysis*

1. Introduction

Computer music has succeeded transforming conventional music instruments into computer media that gives advantages to access, learn or play a music instrument. Computer music resulted from continuous researches and experiments involves computer which plays part of a composition, performs music, and gives more part of computer in an application related to music [1, 2]. Automatic musical note player application is a part of computer music research, and has been implemented in various types of music applications to give a music demonstration, music games and learning music. For example, *Smule*, a phenomenal virtual music instruments application by Ge Wang, that turns *iPhone* becoming a creative channel to learn, play and perform music [3]. Another application example is *Smart Gamelan Player* designed to support learning of how to play traditional music instruments [4]. This application has a feature that shows *gamelan* music instruments demonstration in playing a note sequence of a song.

The task of automatic musical note player system is to hit an instrument button to play a note sequence based on a defined tempo represented in a time interval value between two consecutive beats. Tempo is measured in beats per minute or a number of beats to play in a minute. Each note in a note sequence has its time target reffered to a time interval value. Time target is used as parameter for an automatic musical note player system to play. Time target parameterization becomes a challenge in developing automatic musical note player for orchestra music. An automatic musical note player system for this type of music will result stiffly sound if each instrument of the orchestra plays every note in a note sequence in an exact time target, and this makes a system lacks of natural touch of human in playing music, such in a work by [4]. Hence, automatically hitting an instrument button to play a note should not be executed as exact as its time target but it should refer to human play that hits an instrument button based on an approximate time.

Approximate time is a range of time based on a time target in which hitting an instrument button can be tolerated as not too fast or not too late. In developing automatic musical player system that refers to human play, approximate time value should be considered as more than a simple random value. It is interesting to be investigated especially for developing

Received July 21, 2017; Revised October 8, 2018; Accepted November 10, 2018

a virtual music orchestra, and it becomes a motivation in this research to develop an automatic musical note player for traditional music orchestra called *gamelan*.

Song in *gamelan* music known as *gendhing* consists of a bar sequence. A bar called *gatra* contains four beats, in which each beat is initially represented with dot note called *pin*. The dot note then can be filled with a pitch or remain in the form of dot note. Rhythm in *gamelan* music is represented with a number of notes in a beat, and the speed in playing the musical note [5]. There are five levels of rhythms in *gamelan* music, which are *lancar* (1/1) where a beat contains one note, *tanggung* (1/2) where a beat contains two notes, *dados* (1/4) where a beat contains four notes, *wiled* (1/8) where a beat contains eight notes and *rangkep* (1/16) where a beat contains sixteen notes. The tempo to play is defined based on a constant interval time value. For example, the rhythm of *lancar*, the time interval value is about 1000 ms between two consecutive beats. Figure 1 shows an illustration of a note sequence consisting of 32 beats in a *gamelan* song entitled *Suwe Ora Jamu*.

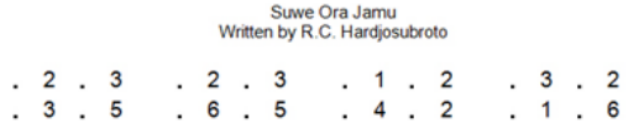

Figure 1. Example of a note sequence of a *gamelan* song entitled *Suwe Ora Jamu*

Instead of using a time interval value for the system to play a note, a random technique parameterized based on an observation on how a *gamelan* musician playing an instruments was proposed in this research. This was to obtain natural touch in playing a note sequence as human does. The parameterization for randomizing time as an approximate time to hit a button was defined by a set of time range value. Given a name *P* for a function to hit a button to play a note in which *P* identifies a set of time range *RT* to define an approximate time *AT* to execute the play, so the notation is P: RT→AT. Time range *RT* is defined based on human play using time-frequency analysis and peak detection.

Time-frequency analysis is commonly used for music signal problems, such as to analyze the sound of musical instruments [6], to analyze the interaction of musical rhythm with melodic structure [7], to analyze various frequency representation of the tone of a traditional instrument from Nusa Tenggara Timur, Indonesia called *gong* [8]. Time-frequency analysis known as spectrograms handles changes in frequency content over time, provides a time-frequency portrait of musical sounds, can describe quick transitions between notes problems [6]. The function in fourier analysis can be represented in both time and frequency domain known as time-frequency analysis [9, 10]. Fourier transform and peak detection were used in time-frequency analysis by [11] for audio segmentation. Peak detection is to identify a time value when human hitting an instrument button. Tempo and beat estimation research was conducted by [12-14]. Spectral analysis, spectral energy flux and peak detection function were used by [12], while k-NN regression was used by [13] and utilization of harmonic separation and periodicity analysis were used by [14] to estimate tempo and beat. These researches aim to define a tempo and to track beats of a music audio source, in which the result can be used to identify a music genre. The methods used in these researches were quite similar with the procedures needed in this work. In this work, tempo and beat estimation were used to observe the way a musicion estimating a tempo in playing a music instrument. Fourier transform and peak detection used in time-frequency analysis by [11, 12] for audio segmentation were used in this research. Peak detection is to identify a time value when human hitting an instrument button, and then the time value was calculated based on a time target to set a tolerated range of time (*RT*) used for randomizing approximate time (*AT*) to play a note.

2. Research Method

Natural automatic musical notes player was developed by analyzing the way a musician estimating execution time referred to the time target of the tempo. Execution time estimation

was represented to approximate time containing a time range value which can be tolerated by time target of the tempo, and the program executes the play of a music instrument in an approximate time. An illustration of a relation between notes sequence NS containing notes S, time target TT containing time value T, and execution time ET containing time value E is shown by Figure 2. The first note (S1) has time target first time target value (T1), and the first execution time (E1) will have time value which is to T1 more or less, and so on. Further, the value of ET was used to define approximate time AT.

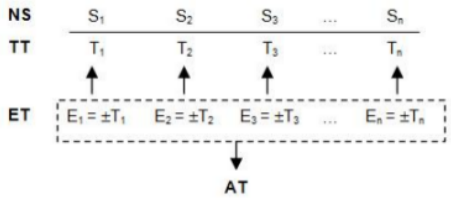

Figure 2. Illustration of relation between notes sequence, time target, execution time, and approximate time in playing music instrument

Execution time was measured by analyzing recording audio of musicians playing a music instrument. The problem in this phase came up with selection of a method to accurately identify execution time of keystroke of hitting an instrument button. The analysis was to obtain data of pitch and time of musical notes played by a musician. Time-frequency analysis was used to accurately find execution time of an interaction of a musician and his instrument music in playing notes sequence. Time-frequency analysis is a method that represents visualization of sound in form of spectrogram. In musical sound analysis domain, spectrogram describes transition between notes including its time. Figure 3 shows the model diagram proposed to develop natural automatic musical notes player.

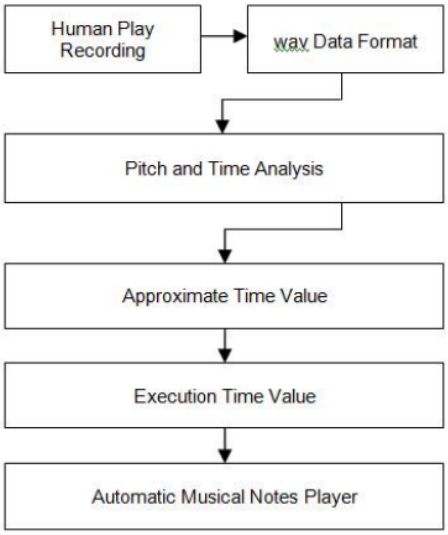

Figure 3. Model diagram of the development of automatic musical notes player

The experiment in this research was conducted by developing an automatic *gamelan* musical note player limited into the rhythm of *lancar* and melodic abstraction note played using melodic abstraction instruments such as *saron, demung*, *peking* and *slenthem*. The proposed method was divided into data acquisition, pitch and time analysis and implementation.

2.1. Data Acquisition

The experiment used instrument of *gamelan* music for developing automatic musical notes player. There are two types of musical scale in *gamelan* music, which are *slendro* which consists of five pitches of (1, 2, 3, 5, 6) and *pelog* which consist of seven pitches of (1, 2, 3, 4, 5, 6, 7). The frequency of pitches of these types of musical scale is different, and the frequency of pitches of *gamelan* music is different with that in western music. A set of gamelan orchestra can contain both *gamelan slendro* and *gamelan pelog*, or only one of these types. Both types of *gamelan* musical scale consist of instruments classified based on their function, which are a group of instruments that plays notes sequence of melody skeleton of songs, such as *demung, saron, peking*, and *slenthem*; a group of instruments that plays notes of songs, such as *gender* and *rebab*; a group of instruments that plays notes which have function to define the structure of a song, such as *kenong* and *gong*.

The development of automatic musical notes player was limited to *demung,* an instrument of *gamelan pelog* that plays notes sequence of melody skeleton of songs. *Demung* is a type of xylophone instrument that has a register containing range of notes which are (1, 2, 3, , 5, 6, 7). Figure 4 shows an illustration of *demung*.

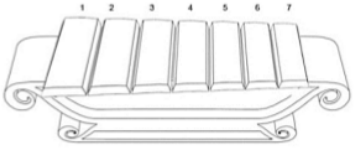

Figure 4. *Demung*, a gamelan music instrument used to play notes sequences of melody skeleton

Notes sequence of a melody skeleton are played in a constant interval time, such as a beat per second for slow tempo, or a beat per half a second for tempo that is faster than slow. For an example, a song entitled "*Suwe Ora Jamu*" (Figure 1) consists of 32 beats filled with dot notes and number notes. The use of slow tempo with one beat per second defines the first beat filled with a dot note has time target at 1st second, the second beat filled with a note of 2 has time target at 2nd seconds, the third beat filled with a dote note has time target at 3rd seconds, the fourth beat filled with a note of 3 has time target at 4th seconds, and so on. This makes time target to play all beats in the song is 32 seconds.

Data acquisition was to obtain audio signal data of *gamelan* song played by a musician in the rhythm of *lancar* where the time interval value is about 1000ms between two consecutive beats. The acquisition was conducted by recording a *gamelan* musician playing *saron,* one of melodic abstraction instruments. The play was recorded into .wav file format with sample rate of 48 khz and 16 bit mono. The musician played a note sequence of five *gamelan* songs collected from www.gamelanbvg.com. The *gamelan* songs used as dataset were described in Table 1. The musical note structure of *gamelan song* can be seen in Figure 5 that depicts the musical note of the first song in the dataset.

Table 1. Dataset of *Gamelan* Songs

No	Song Title	Number of Beats
1	Suwe Ora Jamu	32
2	Kebo Giro	80
3	Gugur Gunung	64
4	Lasem	64
5	Manyarmilar	72

2.2. Pitch and Time Analysis

Time interval value was used as target time for a *gamelan* musician knowing when a button should be hit to play a note. Human plays it based on intuition or musical sense. This behavior show the way a musician estimating time to play in an approximate time which is as close as to the target time. Time-frequency analysis was used to identify the approximate time from a *gamelan* musician while playing a note sequence. The analysis was conducted by performing fast fourier transform (FFT) technique to remove noise, and then followed by performing peak detection technique to find the exactly time value of an approximate time from the *gamelan* musician play. The distance of execution time resulted from peak detection to the time target was measured to set range time used to parameterize a random value for approximate time value. Figure 5 shows an example of data input and data after noise removal using FFT which are then used to identify peaks values.

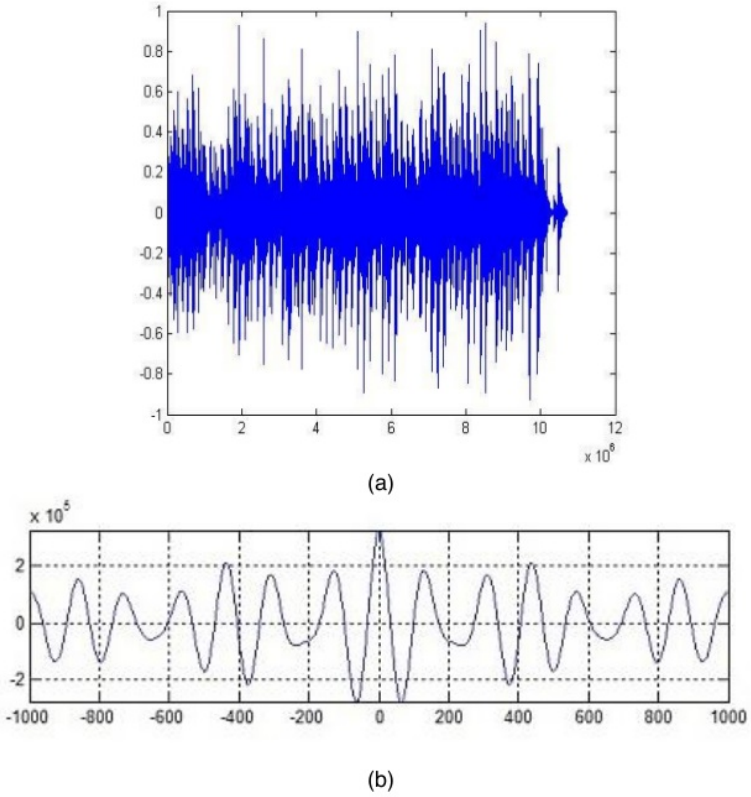

Figure 5. Signal data input (a), after noise removal using FFT (b)

The distance of execution time resulted from peak detection to the time target was measured to set range time used to parameterize a random value for approximate time value. For example, given a time interval value at 1000ms (rhythm of *lancar*) and the play was started from time value at 0, so the target time for each note can be formulated as:

$$TT = \sum_{k=0}^{k=L-1} [TI \; x \; K]$$

where:
TT=Time target
TI=Time interval
L=Number of beats

Execution time was identified based on the peak value resulted from peak detection. The formula to find execution time value is shown as follow:

$$ET = \sum_{k=0}^{k=L-1} [TT_k - PT_k]$$

where:
ET=Execution time
PT=Peak time
L=Number of beats

Time range setting was used as parameters to random a time value to play a note. The data of execution time are sorted to find the lowest and the highest approximate time value. The lowest value will be negative value when the *gamelan* musician playing a note before target time. The highest value will be positive value when a *gamelan* musician playing a note after target time. The lowest and highest execution time value were used to parameterize a random value for the system playing a note in every time target, in which each value of time target was summed with the lowest value and the highest value. Below is the formula of time range setting.

$$DS = Sort\ (ET)$$
$$RT = \sum_{k=0}^{k=L-1} [(TT_k + DS[0]), (TT_k + DS[L-1])]$$
$$AT = Random\ (RT)$$

where:
DS=Data sorted from ET
RT=Random value parameter.
AT=Approximate time

For example, given a time interval value is at 1000 ms, and the approximate time to play a note in the second segment is:
TI=1000ms.
ET=[0, 11,-31, 150, 87].
DS=[-31, 0, 11, 87, 150].
TT_3=2000ms.
RT_3=((2000+ (-31)), (2000 + 150))
AT_3=Random (1969, 2150)

Based on the example above, the approximate time *AT* to play the third note in a note sequence with time target of 2000 ms is a time in a time range *RT* of 1969 to 2150 ms, thus the system will randomize the approximate time value based on this time range.

The formulas above were implemented in the experiment. A *gamelan* musician who has 27 years experience in playing *gamelan* instruments was selected to play note sequences of five *gamelan* songs in the form of melodic abstraction using a melodic abstraction instrument called *saron*. The play of each song was separately recorded, and then the data were analyzed using Matlab to conduct time-frequency analysis and peak detection. Table 2 shows the result of time-frequency analysis for five songs used in the dataset in which each song was played in the rhythm of *lancar* with time interval for two consecutive beats at 1000 ms. The columns in the table are labeled with *GS* which stands for index of *gamelan* songs in the dataset, *IN* for index of notes, *NS* for note sequence, *TT* for time target, *PT* for peak time and *ET* for execution time.

There were 312 peaks values of execution time obtained from five songs. The execution time values from all samples were concatenated and sorted to find lowest and highest values of execution time as seen below with *DA* stands for result of data of time-frequency analysis, *DC* stands for result of data concatenation, and *DS* stands for result of data sorting.

Table 2. Result of Time-frequency Analysis

GS	IN	NS	TT	PT	ET (TT-PT)
			(in milliseconds)		
1	1	.	0	0	0
	2	2	1000	985	15
	3	.	2000	2236	-236
	…	…	…	…	…
	32	6	32000	32378	-378
2	1	.	0	0	0
	2	6	1000	1074	-74
	3	.	2000	2133	-133
	…	…	…	…	…
	80	5	80000	79987	13
3	1	.	0	0	0
	2	6	1000	1142	-142
	3	.	2000	1947	53
	…	…	…	…	…
	64	2	64000	63892	108
4	1	3	0	0	0
	2	5	1000	891	109
	3	6	2000	2212	-212
	…	…	…	…	…
	64	6	64000	64231	-231
5	1	.	0	0	0
	2	2	1000	1081	-81
	3	.	2000	2143	-212
	…	…	…	…	…
	72	7	72000	71967	33

DA = [[0, 15, -236, …, -378] [0, -74, -133, …, 13] [0, -174, -142, …, 108] [0, 109, -212, …, -231] [0, -81, -212, …, 33]]

DC = [0, 15, -236, …, -378, 0, -74, -133, …, 13, 0, -174, -142, …, 108, 0, 109, -212, …, -231, 0, -81, -212, …, 33]

DS = [-328, …, 0, …, 246]

The result of the lowest value was -328, and the highest value was 246. The result was used for the range time as parameters to randomize approximate time for playing a note sequence.

2.3. System Design

An automatic musical notes player system was developed to play notes sequence of melody skeleton using *demung* instrument of *gamelan* music. Notes sequences data were inputted into the system. The system works by playing a note sequence data with output of sound and animation of a note selected by order. Figure 6 shows workflow diagram of automatic musical notes player system.

Automatic musical notes player system was developed on mobile application platform. System development was divided into four phases, which were note sequence audio-visual, notes sequences data input, programming and testing, and interface design. Each phase was described in following discussions. Outputs of automatic musical notes player were sound of notes and its visual displayed using its instrument button picture. Sounds of seven *demung* instrument buttons were separately recorded and saved into .wav format file, which were: *sound1.wav, sound2.wav, sound3.wav, sound4.wav, sound5.wav, sound6.wav* and *sound7.wav*.

The instrument images were manipulated by adding its mallet picture to visualize the play of notes sequence. There were eight pictures to visualize execution of silent/dot note, and note 1 until note 7. The mallet was positioned in the bottom-center of the instrument to visualize silent and execution of dot notes, while for other notes visualization, the mallet was positioned above the each instrument button. Each picture was saved into .png format file and labeled with *pic0.png, pic1.png, pic2.png, pic3.png, pic4.png, pic5.png, pic6.png,* and *pic7.png*. Figure 7 shows some examples of pictures of instrument buttons visualization.

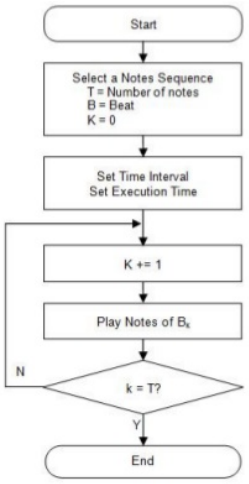

Figure 6. Workflow diagram of automatic musical notes player systems

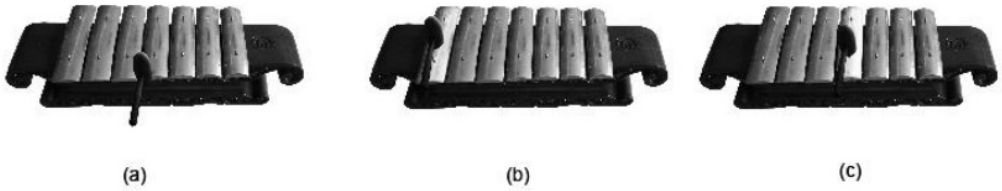

Figure 7. Instrument buttons visualization with
(a) for silent and dot note visualization, (b) for note 1 execution, (c) for note 4 execution

Sounds of notes of instrument buttons and pictures of instrument buttons visualization were embedded into the program. The program will call each sound and each picture according to a note which was played on the execution time. The program was designed to automatically play notes sequences inputted into the system. Total 30 notes sequences of melody skeleton including notes sequences used as dataset were used for the songs library in the program. All notes sequences were formatted as array data, and the dot notation was translated into value 0. Figure 8 shows the illustration of a notes sequence input formatted as array data.

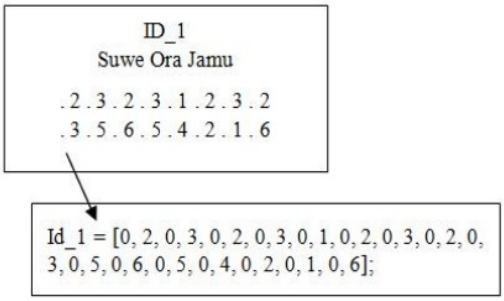

Figure 8. Note sequence formatted as array data

Array data of notes sequences used in songs library of the program were described in Table 3.

Table 3. Notes Sequences Data Set

ID	Notes Sequences
ID_1	[0, 2, 0, 3, 0, 2, 0, 3, 0, 1, 0, 2, 0, 3, 0, 2, 0, 3, 0, 5, 0, 6, 0, 5, 0, 4, 0, 2, 0, 1, 0, 6]
ID_2	[0, 6, 0, 5, 0, 3, 0, 2, 0, 3, 0, 2, 0, 6, 0, 5, 0, 6, 0, 5, 0, 3, 0, 2, 0, 3, 0, 2, 0, 6, 0, 5, 0, 6, 0, 5, 0, 6, 0, 7, 0, 6, 0, 7, 0, 6, 0, 5, 0, 6, 0, 5, 0, 6, 0, 7, 0, 6, 0, 7, 0, 6, 0, 5, 0, 7, 0, 6, 0, 3, 0, 2, 0, 3, 0, 2, 0, 6, 0, 5]
ID_3	[0, 6, 0, 7, 0, 6, 0, 7, 0, 3, 0, 5, 0, 7, 0, 6, 0, 2, 0, 7, 0, 2, 0, 7, 0, 6, 0, 5, 0, 2, 0, 3, 0, 5, 0, 6, 0, 5, 0, 6, 0, 2, 0, 3, 0, 6, 0, 5, 0, 2, 0, 3, 0, 2, 0, 3, 0, 6, 0, 5, 0, 3, 0, 2]
ID_4	[3, 5, 6, 5, 3, 5, 3, 2, 5, 3, 2, 3, 5, 6, 7, 6, 3, 5, 6, 5, 3, 5, 3, 2, 5, 3, 2, 3, 5, 6, 7, 6, 0, 2, 6, 2, 2, 6, 0, 2, 6, 0, 2, 6, 3, 2, 7, 6, 0, 2, 6, 0, 2, 6, 0, 2, 6, 0, 2, 6, 3, 2, 7, 6]
ID_5	[0, 2, 0, 7, 0, 2, 0, 6, 0, 2, 0, 6, 0, 2, 0, 7, 0, 2, 0, 7, 0, 2, 0, 6, 0, 3, 0, 5, 0, 3, 0, 2, 0, 3, 0, 2, 0, 3, 0, 5, 0, 7, 0, 6, 0, 3, 0, 2, 0, 6, 0, 6, 0, 6, 0, 6, 0, 0, 0, 0, 3, 5, 6, 7]

Phase of programming and testing was conducted by writing codes for automatic musical notes player program application, and testing to make sure that there were no errors found in the program, and all functions in the program can properly works. Below is the algorithm to automatically play a notes sequence:

// Songs library = [id_1, id_2, …, id_30];
// Sounds library = [sound1.wav, sound2.wav, …, sound7.wav]
// Pictures library = [pic0.png, pic1.png, …, pic7.png];
// Tempo = 1000 ms;
// Timer = 0 millisecond
(1) Select a notes sequence from songs library.
(2) Set duration by multiplying number of notes in selected notes sequence with tempo.
(4) Set execution time for all notes by randomizing approximate time value.
(5) Set timer to 0 milliseconds.
(6) Play sound and display picture for every note by order in selected notes sequence when timer reach time value of execution time data.
(7) Stop the play when timer value equals to duration value.

3. Results and Analysis

An automatic *gamelan* musical note player mobile application was developed using Adobe Flash program. The interface design of automatic musical notes player consisted of two main parts, which were songs library which contained title songs and its notes sequence, and play section which automatically presented the play of selected musical notes sequence including outputs of sound and instrument visualization. Figure 9 shows the interface design of automatic musical notes player for *gamelan* instrument named *demung*.

(a) (b)

Figure 9. Interface design, (a) songs library (b) automatic play

For an addition, the program was designed to display a note sequence data and play it instrument animation and sounds as outputs, and three melodic abstraction instruments were used in the program which were *saron, demung* and *slentem*. Figure 10 shows the screenshot of the automatic *gamelan* musical note player application.

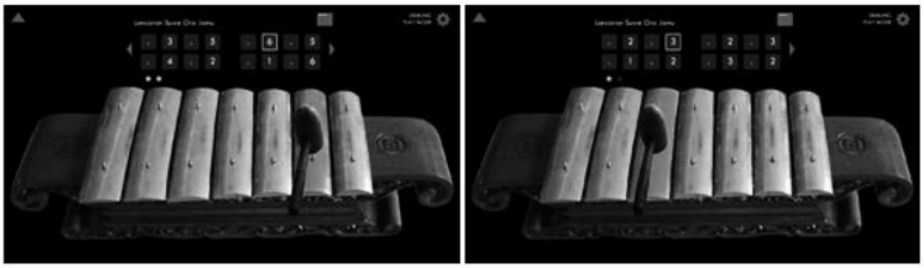

Figure 10. Screenshot of automatic gamelan musical note player program

The evaluation was simply conducted by recording the automatic play by the program using note sequences of five song samples. The recording audio files were analyzed using procedures implemented in the phase of data analysis which were time-frequency analysis and peak detection. The expectation was that all notes were played by the program in approximate time as the range time obtained from observing a *gamelan* musician play which were random value between (-328, 246) as addition for time target value. The result shows that all notes were played in approximate time as expected. Table 4 shows the execution times in playing the instrument by the program developed in this research denoted with Program I, and by the program developed by [4] denoted as Program II as the comparation. The columns in the table are labeled with *IN* which stands for index of notes, *NS* for note sequence, *TT* for time target and *ET* for execution time.

Table 4. Example of Execution Time Results for *Suwe Ora Jamu*

IN	NS	TT	RT	ET	
			Random (-328, 246)	Program I (TT + RT)	Program II (TT)
1	.	0	151	151	0
2	2	1000	-302	698	1000
3	.	2000	-153	1847	2000
4	3	3000	-114	2886	3000
5	.	4000	-155	3845	4000
6	2	5000	-25	4975	5000
7	.	6000	51	6051	6000
8	3	7000	-111	6889	7000
...	...	...	...	...	...
32	6	31000	124	31124	31000

Another evaluation was conducted by asking three experts to compare the 5 plays recorded by program I (the program developed in this research) with the plays recorded by Program II (*Smart Gamelan Player* developed by [4]). The task for the experts was to choose the plays that sound natural. All experts stated that the plays recorded by Program II were not natural and all plays were straight and very clean since all the instruments exactly play a note in the same time, while all plays recorded by Program I sound more natural.

4. Conclusion

This research aims to develop an automatic *gamelan* musical note player that can naturally play a musical note as human does. A function *P* to hit a button to play a note in a

tolerated time was formulated into P: RT→AT. The value of time range (RT) was defined based on human play using time-frequency analysis. The time range was used as parameters to random approximate time for playing a note. After observing plays by a musician and conducting pitch and time analysis, the approximate time AT value was a random value of time range RT which was a value between-328 until 246 milliseconds from the time target value. An automatic *gamelan* musical note player program was developed based on this function. The evaluation shows that the program played all note sequences in the approximate time as human does. Based on the expert opinions, the program developed in this research played sounds more natural and better than *Smart Gamelan Player* developed by [4].

For the next works, the results of this research will be used to develop a model of musical notes reader, a program that can read play a musical sheet, and to develop an interactive music instrument program which can detect and measure the tempo of the play by the user.

References

[1] S Keith, *Bridging The Gap: Thoughts On Computer Music And Contemporary (Popular) Electronic Music*. Proceedings of the 2010 Australasian Computer Music Conference, Canberra. 2010: 37-42.
[2] D Keislar. A historical view of computer music technology. In: R.T. Dean. The Oxford Handbook of Computer Music. New York: Oxford University Press. 2009: 11-43.
[3] C Dannen, iPhone Design Award Winning Project, *Apress*, New York. 2009
[4] K Hastuti, AZ Fanani, AM Syarif. *Virtual Player of Melodic Abstraction Instruments for Automatic Gamelan Orchestra*, 2nd International Conference on Science in Information Technology (ICSITech), Balikpapan. 2016: 30-34.
[5] Martopangrawit, Pengetahuan Karawitan, ASKI, Surakarta. 1969.
[6] JF Alm, JS Walker, Time-Frequency Analysis of Musical Instruments, *Society for Industrial and Applied Mathematics* (SIAM). 2002; 44 (3): 457-476.
[7] X Cheng, JV Hart, and JS Walker. Timefrequency analysis of musical rhythm, *Notices of the American Mathematical Society*. 2009; 56:344-60, available at http://www.ams.org/notices/200903/
[8] YCH Siki, NMR Mamulak. Time-frequency analysis on gong timor music using short-time fourier transform and continuous wavelet transform. *International Journal of Advances in Intelligent Informatics*. 2017; 3 (3): 146-153.
[9] N Lenssen, D Needell. An Introduction to Fourier Analysis with Applications to Music, *Journal of Humanistic Mathematics*. 2014; 4 (1): 72-89.
[10] A Degani, M Dalai, R Leonardi, P Migliorati, *Time-Frequency Analysis of Musical Signals Using the Phase Coherence*, Proc. of the 16th Int. Conference on Digital Audio Effects (DAFx-13), Maynooth. 2013: 2-5.
[11] C Meyer, M Spiertz. *Audio Segmentation Using Different Time-Frequency Representations*, Poster 2008: 12th International Student Conference on Electrical Engineering, Prague. 2008: 1-5.
[12] Miquel Alonso, Bertrand David, Gael Richard, *Tempo and Beat Estimation of Musical Signals*, ISMIR 2004 5th International Conference on Music Information Retrieval, Barcelona. 2004: 32-37.
[13] Antti Eronen and Anssi Klapuri. Music Tempo Estimation with K-NN Regression, *IEEE Transactions on Audio, Speech, and Language Processing*. 2010; 18 (1): 50-57.
[14] Aggelos Gkiokas, Vassilis Katsouros, George Carayannis and Themos Stafylakis. *Music Tempo Estimation and Beat Tracking by Applying Source Separation and Metrical Relations*. 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto. 2012: 421-424.

Natural Automatic Musical Note Player

ORIGINALITY REPORT

6%	3%	2%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	sinta3.ristekdikti.go.id Internet Source	1%
2	Submitted to University of Malaya Student Paper	1%
3	dedyrw.staff.telkomuniversity.ac.id Internet Source	1%
4	Submitted to Study Group Australia Student Paper	1%
5	Submitted to Pennsylvania College of Technology Student Paper	1%
6	eprints.dinus.ac.id Internet Source	1%
7	"13th International Conference on Biomedical Engineering", Springer Science and Business Media LLC, 2009 Publication	<1%
8	en.pudn.com Internet Source	<1%

9	www.freepatentsonline.com Internet Source	<1%
10	Jeremy F. Alm, James S. Walker. "Time-Frequency Analysis of Musical Instruments", SIAM Review, 2002 Publication	<1%
11	Khafiizh Hastuti, Azhari Azhari, Aina Musdholifah, Rahayu Supanggah. "Rule-Based and Genetic Algorithm for Automatic Gamelan Music Composition", International Review on Modelling and Simulations (IREMOS), 2017 Publication	<1%
12	Khafiizh Hastuti, Azhari Azhari, Aina Musdholifah, Rahayu Supanggah. "Building Melodic Feature Knowledge of Gamelan Music Using Apriori Based on Functions in Sequence (AFiS) Algorithm", International Review on Computers and Software (IRECOS), 2016 Publication	<1%
13	link.springer.com Internet Source	<1%
14	Submitted to Universitas Dian Nuswantoro Student Paper	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		