Melody Difficulty Classification using Frequent Pattern and Inter-Notes Distance Analysis

Pulung Nurtantio Andono*, Edi Noersasongko, Guruh Fajar Shidik
Khafiizh Hastuti, Sudaryanto Sudaryanto, Arry Maulana Syarif
Faculty of Computer Science, Universitas Dian Nuswantoro, Semarang, Indonesia

***Abstract*—This research proposes a novel method for melody difficulty classification performed using frequent pattern and inter-notes distance analysis. The Apriori algorithm was used to measure the frequency of the notes in the note sequence, in which the melody length is also included in the calculation. In addition, the inter-notes distance analysis was also used to measure the difficulty level of composition based on the distance between successive notes. The classification was performed for traditional Javanese compositions known as Gamelan music. Symbolic representation, in which the Gamelan compositions music sheets were collected as the dataset, was chosen by asking experts to divide the compositions based on their difficulty level into basic, intermediate and advanced classes. Then, the proposed method was implemented to measure the difficulty value of each composition. The difference in the interpretation of the difficulty level between the experts and the difficulty value of the composition is solved by calculating the mean value to obtain the range of difficulty values in each class. Evaluation was performed using confusion matrix to measure the accuracy, precision and recall value, and the results reaching 82%, 82.1% and 82%, respectively.**

***Keywords*—Multi-class classification; frequent analysis; Apriori; Symbolic music; Gamelan**

I. Introduction

This research aims to develop a classifier to discriminate the difficulty level of melody or composition. The topics discussed in this research are closely related to the topic of adaptive learning, including learning to play a musical instrument which becomes the motivation in formulating problems from the difficulty level of composition. Therefore, a brief explanation including the research of adaptive learning literature is deliberately described in this article to emphasize the urgency of the need for a difficulty level classification system in the development of adaptive learning to play musical instruments.

Adaptive learning systems can dynamically adjust learning content based on user abilities and preferences [1]. Adaptive learning systems are developed from simple to complex with a set of rules and self-learning algorithms [2] so that they can provide immediate feedback so that users can stay focused and easily make corrections independently [3]. The difficulty level of the problems that must be solved becomes a challenge in the development of an adaptive learning system. The automatic adjustment of the difficulty level is part of the intelligent tutorial systems that can identify the user's characteristics so that the system can determine the suitable task based on the user's abilities [4]. The automatic adjustment of the difficulty level examples can be found in the works by [5].

Intelligent electronic learning utilizes an adaptive learning approach. However, adaptive learning approaches are rarely found in musical electronic learning, such as in melody learning. Electronic learning to play musical instruments involves melodies, thus classification of melodies (compositions) based on the level of difficulty is still rare as well. Meanwhile, the difficulty level of composition based on the melodic pattern has a positive impact on adaptive learning to play musical instruments [6]. Developing an intelligent musical electronic learning should involve adaptive learning approach. By classifying the difficulty level of composition, the system can determine or provide recommendations containing the composition to be learned based on the user's ability.

Against these facts, a novel approach to develop a classifier that is able to measure the difficulty level of composition was proposed in this research. The classifier was developed using a set of rules defined based on frequent pattern and inter-notes distance analysis. Although melody has a sequence or time series data type, the characteristics of the problems encountered are considered suitable to be solved using frequent mining algorithms rather than sequence mining algorithms. In addition, the inter-notes distance value between successive notes is also used as a parameter in measuring the difficulty level of composition. This makes the analysis of the difficulty level of composition is unique. The forward and backward notes have the same pattern in the inter-notes distance value.

Participation in the preservation of cultural heritage is the motivation in carrying out this research. Thus, the melody difficulty classification system developed in this research is implemented to traditional Javanese compositions known as Gamelan music. However, the results achieved in this research can also be an inspiration to develop a learning system for playing musical instruments of other types of music.

II. Related Work

In electronic learning, an adaptive learning approach is needed to achieve an efficient, effective learning experience and support customization settings for users. Tasks or difficulty classification is part of the adaptive learning that supports the system in determining the suitable task by considering the user’s ability. In general, the system will measure the user's answer by assigning an accuracy weight to be used as a basis for determining the level of task that is match the user's

*Corresponding Author

performance. Therefore, questions or tasks to be solved by the user must be classified based on to the difficulty level. In adaptive learning to play musical instruments, classification can be used as a solution in weighting the difficulty level of composition.

Smart devices and artificial intelligence approach are needed to develop a smart learning environment that supports adaptive learning [7]. However, not all electronic learning systems that uses smart devices to run implementing an adaptive learning approach. Moreover, in learning to play musical instruments, the systems generally are limited to the transformation of musical instruments into digital media as in the work of mobile-based gamelan electronic learning media [8], mobile-based hybrid digital-physical harp instrument [9], 3D virtual traditional Chinese instrument called yangqin [10], virtual traditional Brazilian maracatu [11].

A learning system for playing traditional Javanese musical instruments developed by [12] provides a collection of compositions and tempo preferences that are divided into slow to fast ranges based on the time interval between notes switching. However, the collection is not classified according to the difficulty level of composition, and the choice of tempo is determined by the user. In other words, the system has not yet adopted the ability to automatically and dynamically identify user abilities to recommend the suitable composition and tempo.

A task content analysis team was formed to sort the tasks according to their difficulty level [13], while in the similar case; the difficulty level was designed by developing a curriculum [14]. Both of them used English as their learning content, in which the database of questions of various difficulty levels is easier to find than the composition database with groupings based on the complexity of melodic patterns. Meanwhile, analysis of the difficulty level manually performed has a weakness when dealing with a lot of data or adding data to a task. An adaptive learning system to play a musical instrument can be found in the work by [6], in which the system automatically selects the difficulty level of compositions based on the user's true-false count tracking. Unfortunately, the classification method for the difficulty level of composition is not explained.

The Apriori algorithm is a popular algorithm in association relationship analysis or frequent pattern analysis which is also known as market basket analysis. The Apriori algorithm was modified by counting only the sequential transactions that are frequent so that transaction (a, b) is not the same as transaction (b, a). Furthermore, in sequence data mining, many Apriori-like or Apriori-based algorithms have been proposed by researchers, such as mining Web access sequence [15], or modifying the association rules by adding time constraint [16]. Meanwhile, in the melodic pattern, Apriori-based algorithms have been developed by [17] to generate music, and the Apriori based on Function in a Sequence (AFiS) algorithm proposed by [18] counted only the sequential transactions that are frequent with additional procedures in the form of measuring musical elements based on their position in the sequence as functions to identify frequent sequence patterns.

The Apriori algorithm has been used in various classification problems, such as classification of patient care needs by [19], classification in risk prediction to define disaster rules based on models developed using the Neural Networks method [20], admission planning classification and job prediction in education midwives. This combines the Apriori algorithm with the Decision Tree Algorithm [21]. Besides being used without being combined with other algorithms in finding solutions, the Apriori algorithm is usually used to map and analyze data which is then used as a constraint for other algorithms in making decisions, or to define rules based on data obtained from training results using other algorithms. Selection of Apriori algorithm as the main and single algorithm in classification problems is rarely found, especially in time-series data problems. However, this algorithm has the advantage of managing small datasets to construct good classification rules. In this research, the Apriori algorithm is proven to be an alternative in solving classification problems on time-series data.

III. Methodology

This research aims to measure the difficulty level of composition in order to classified compositions into basic, intermediate and advanced classes. Traditional Javanese compositions were chosen as objects of research with the aim of participating in the preservation of intangible cultural heritage through the implementation of artificial intelligence.

Gamelan music consists of two musical scales, which are pelog and slendro. Each of the musical scale has a different tone frequency. The pelog musical scale consists of seven notes of (1, 2, 3, 4, 5, 6, 7). Meanwhile the slendro musical scale consists of five notes of (1, 2, 3, 5, 6). There are dotted notes on both musical scales which represent moments of silence. Gamelan music uses a mode musical system called pathet. The system determines the characteristics of the composition based on the dominant notes including the arrangement of the order of the notes. The Gamelan composition is divided into various types, such as lancaran, ladrang, ketawang and others. Fig. 1 shows an example of a composition entitled Kembang Pete which is a type of lancaran composition with the lima musical mode and played on the pelog musical scale.

Parameters of the difficulty level of composition were determined by consulting with experts. Three experts were asked to define the parameters to be used in classifying compositions into basic, intermediate and advanced classes. Then, experts were asked to divide 50 compositions in the dataset into these three classes. Experts proposed the melody length and inter-notes distance to be used as parameters to determine the difficulty level of composition. The length of the melody is the number of beats in the composition. The greater number of beats increases the difficulty level in learning to play the instrument. Meanwhile, inter-notes distance is the distance between successive notes is the range between note values. For example, two successive notes of (1, 7) have a higher difficulty to play than (1, 3) because the distance of the first successive notes is 6 units and the distance of the second is 2 units.

Lancaran Kembang Pete, *Laras Pelog Pathet* Lima

6	5	3	2	1	2	3	5
6	5	3	2	1	2	3	5
6	5	6	1	6	5	3	2
5	6	5	4	2	1	6	5

Fig. 1. A Gamelan Music Composition Example.

The distance between two successive notes is the same, although the order is reversed. For example, (1, 7) and (7, 1) have the same distance of six units. So, the difficulty level of composition can be classified can be carried out using frequent pattern analysis rather than sequence analysis, including the addition of inter-notes distance analysis that appear in the note sequence. Moreover, the frequent pattern analysis was conducted using the Apriori algorithm, in which the melody length was also included in the calculation. The results of the frequent pattern analysis are then accumulated by the inter-notes distance value to classify compositions into basic, intermediate and advanced clasess.

The methodology used in this research consists of five stages, which are: data preparation, data representation, implementation of the Apriori algorithm, and classification where inter-notes distance analysis is performed.

A. Data Preparation

The dataset which consists of 50 Gamelan compositions in form of symbolic data were collected from a collection of music sheets. The music sheet data were then converted into text format. For example, the composition data shown in Fig. 1 is converted to (6, 5, 3, 2, 1, 2, 3, 5, 6, 5, 3, 2, 1, 2, 3, 5, 6, 5, 6, 1, 6, 5, 3, 2, 5, 6, 5, 4, 2, 1, 6, 5).

Some compositions contain dotted notes in their sequence, and the dotted notes were converted to 0 in order to support computational processing as in [22]. For example, as shown in Fig. 2, the composition entitled Balabak which is a type of ladrang composition with the lima musical mode and played on the pelog musical scale contains dotted notes. The composition is converted into a text format to (3, 2, 3, 1, 3, 2, 3, 5, 3, 2, 3, 1, 3, 2, 3, 5, 0, 0, 7, 6, 5, 4, 2, 1, 3, 2, 3, 1, 3, 2, 3, 5).

Ladrang Balabak, *Laras Pelog Pathet* Lima

3	2	3	1	3	2	3	5
3	2	3	1	3	2	3	5
•	•	7	6	5	4	2	1
3	2	3	1	3	2	3	5

Fig. 2. A Composition Containing Dotted Notes Example.

B. Data Representation

The dataset in this research uses symbolic format data from 50 compositions of ladrang style played in the pelog musical scale and the lima musical mode (the data set is included in the Appendix section, Table VIII). The music sheet data were represented by mapping the note sequence as the basis for determining transactions in frequent pattern analysis using the Apriori algorithm. Further, the results of the analysis were accumulated using inter-notes distance analysis. Frequent pattern analysis was performed per composition. Thus, each composition represents a set of transactions, and the note sequence is mapped as transactions within the composition, while notes of the musical scale are the transaction items. The note sequence mapping for the transactions was performed using the sliding window technique with the implementation on the k-itemset being performed using the following pseudocode:

D = a set of transaction containing note sequence data.

L = the length of D.

K = a K-itemset which is an itemset containing K successive notes.

T = transaction in D which represents a set of notes contained in the musical scale system.

```
for (z = 0; z < L; z++) {
    for (n = 0; n < K; n++) {
      T [z] [n] = D [z + n];
    }
}
```

The pseudocode above results in the last itemset having one less element length than the previous itemsets. In other words, all itemsets will have the same element length except the last itemset. Given a composition containing the note sequence of (3, 2, 3, 1, 3, 2, 3, 5, 3, 2, 3, 1, 3, 2, 3, 5, 0, 0, 7, 6, 5, 4, 2, 1, 3, 2, 3, 1, 3, 2, 3, 5) and the value of K is set to 2, then T will contain ((3, 2), (2, 3), (3, 1), (1, 3), (3, 2), …, (3, 2), (2, 3), (5)).

The melody has a repeating pattern; therefore, the last element in the last itemset will contain the first note by adding the following pseudocode:

```
n = L – K;
p = n + K;
while (n < p) {
    for (z = p – K; z < n; z++) {
      T [n] [K – (n – z)] = D [z - (p – K)]
    }
    n++
}
```

Continuing the previous example, then T will contain ((3, 2), (2, 3), (3, 1), (1, 3), (3, 2), …, (3, 2), (2, 3), (5, 3)), so all itemsets have the same length.

The inter-notes distance analysis was performed by subtracting the value of the higher note from the lower note, or subtracting the value of two same notes. For simplicity, the term unit is used as a measure of the value of the note and distance. There are seven notes and one dotted note denoted by the number 0 as a collection of items, which are (0, 1, 2, 3, 4, 5, 6, 7). Thus, the distance is measured with a scale from 0 to 7 units. Table I shows the measurement of the distance between successive notes.

TABLE I. TWO SUCCESSIVE NOTES DISTANCE MEASUREMENT

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	0	1	2	3	4	5	6	7
1	1	0	1	2	3	4	5	6
2	2	1	0	1	2	3	4	5
3	3	2	1	0	1	2	3	4
4	4	3	2	1	0	1	2	3
5	5	4	3	2	1	0	1	2
6	6	5	4	3	2	1	0	1
7	7	6	5	4	3	2	1	0

Based on the description above, for example, given a K-itemset where K = 2 and let a transaction contains two successive notes of (3, 2), then the distance value will be 1 unit, and a transaction which contains two successive notes of (2, 3) also has the distance value of 1 unit. Meanwhile, a transaction of 2-itemset which contains two successive same notes, such as {0, 0}, or {4, 4}, has the distance value of 0 unit. The distance between successive notes is calculated by subtracting the elements of the first note by the elements of the second note. If any of the results are less than 0 or negative value, the result will be multiplied by -1 to make it a positive value. For example, two successive notes of (2, 5) will result in -3 from subtracting 2 – 5, and -3 will be multiplied by -1 to make a positive value of 3. For K-itemset where K is greater than 2, the distance between three or more successive notes is calculated by adding up all the distances between 2 successive notes on all elements in the itemset. The following is a pseudocode to calculate the inter-notes distance value where R represents the distance value between two successive notes, and S represents the total distance value of the itemset containing more than two successive notes.

```
n=0
while (n < L) {
    for (z = 0; z < K – 1; z++) {
      R [n] [z] = T [n] [z] – T [n] [z + 1];
      if (R [n] [z] < 0) {
            R [n] [z] *= -1;
      }
      S [n] += R [n] [z];
    }
    n++;
}
```

Based on the formulas above, using an example of the note sequences of (3, 2, 3, 1, 3, 2, 3, 5, 3, 2, 3, 1, 3, 2, 3, 5, 0, 0, 7, 6, 5, 4, 2, 1, 3, 2, 3, 1, 3, 2, 3, 5), the data mapping into K-itemset of transactions, for example, where K = 2 and K = 3, and its distance value is as follows:

2-itemset

T = ((3, 2), (2, 3), (3, 1), (1, 3), (3, 2), …, (3, 2), (2, 3), (3, 5), (5, 3)).

S = ((1), (1), (2), (2), (1), …, (1), (1), (2), (2))

3-itemset

T = ((3, 2, 3), (2, 3, 1), (3, 1, 3), (1, 3, 2), (3, 2, 3), …, (3, 2, 3), (2, 3, 5), (3, 5, 3), (5, 3, 2)).

R = ((1, 1), (1, 2), (2, 2), (2, 1), (1, 1), …, (1, 1), (1, 2), (2, 2), (2, 1))

S = ((2), (3), (4), (3), (2), …, (2), (3), (4), (3))

C. Apriori Algorithm Implementation

The Apriori algorithm implementation was performed by mapping each composition as a separate set of transactions, and each transaction contains the same number of beats. In addition, the characteristics of the melody data structure, the classification problems encountered, and the data mapping carried out make the frequent itemset being applied to the 2-itemset pattern. Data mapping was performed based on two successive notes by applying the sliding window technique so as to produce patterns of {beat1, beat2}, {beat2, beat3}, {beat3, beat4}, ..., {last beat, beat1}. Thus, the data mapping can already represent the relationship between successive notes at each beat in the composition.

The following is an example of the Apriori algorithm implementation on a composition that is used as a dummy. Let D is the set of note sequence of the composition entitled Ladrang Balabak-Laras Pelog Pathet Slendro, where each transaction T in D contains items I which are elements of the set of notes in the pelog musical scale, and the transaction is mapped into two successive notes. Thus, with I, M, N, D, and T representing the set of items, the notes sequence, the number of beats in M, the set of transaction T containing note sequence data after data mapping, and the length of D, respectively, then:

I = (0, 1, 2, 3, 4, 5, 6, 7)

M = (3, 2, 3, 1, 3, 2, 3, 5, 3, 2, 3, 1, 3, 2, 3, 5, 0, 0, 7, 6, 5, 4, 2, 1, 3, 2, 3, 1, 3, 2, 3, 5)

N = 32

D = ((3, 2), (3, 1), (3, 2), (3, 5), (3, 2), (3, 1), (3, 2), (3, 5), (0, 0), (7, 6), (5, 4), (2, 1), (3, 2), (3, 1), (3, 2), (3, 5))

L = 16

The next data mapping uses a sliding window technique which doubles the number of beats and the length of the elements in D, as follows:

I = (0, 1, 2, 3, 4, 5, 6, 7)

M = (3, 2, 2, 3, 3, 1, 1, 3, 3, 2, …, 5, 3)

N = 64

D = ((3, 2), (2, 3), (3, 1), (1, 3), (3, 2), …, (3, 2), (2, 3), (3, 5), (5, 3)).

L = 32

TABLE II. TRANSACTION T DATA IN D

ID	0	1	2	3	4	5	6	7
1	0	0	1	1	0	0	0	0
2	0	0	1	1	0	0	0	0
3	0	1	0	1	0	0	0	0
4	0	1	0	1	0	0	0	0
5	0	0	1	1	0	0	0	0
6	0	0	1	1	0	0	0	0
7	0	0	0	1	0	1	0	0
…	…	…	…	…	…	…	…	…
31	0	0	0	1	0	1	0	0
32	0	0	0	1	0	1	0	0

Table II shows each transaction record in tabular data format with a sequential two-notes mapping in each transaction.

The next step is to calculate the frequent 2-itemset, including calculating the difficulty weight. The weight of itemset is measured based on the multiplication of the support value in each itemset with the distance between the notes in each itemset. The compositions in the dataset have varied melody lengths, and the longest is the composition with D containing 320 itemset, while the shortest containing 32 itemset.

The weight of itemset W result for each transaction in D is calculated by multiplying the support by the value of the distance between notes S, and dividing by the length of I (the set of items), which is W = (support x S) / the length of I, where I is the set of items containing eight notes. Next, the weight values in each transaction are summed to get the difficulty value. Table III shows the results of calculating the difficulty value of the composition, which is 0.2109.

The procedure performed in the example above was applied to all compositions in the dataset. Table IV shows an example of the results of calculating the difficulty value for the composition in the dataset.

TABLE III. 2-ITEMSET SUPPORT COUNT, TRANSACTION WEIGHT, AND THE DIFFICULTY VALUE OF THE COMPOSITION

Items	Count	Support	Dist.	Weight
{0, 0}	1	0.0312	0	0.0000
{2, 1}	1	0.0312	1	0.0039
{3, 1}	7	0.2187	2	0.0547
{3, 2}	12	0.375	1	0.0469
{4, 2}	1	0.0312	2	0.0078
{5, 0}	1	0.0312	5	0.0195
{5, 3}	5	0.1562	2	0.0391
{5, 4}	1	0.0312	1	0.0039
{6, 5}	1	0.0312	1	0.0039
{7, 0}	1	0.0312	7	0.0273
{7, 6}	1	0.0312	1	0.0039
Difficulty				0.2109

TABLE IV. THE DIFFICULTY VALUE OF COMPOSITIONS RESULTS EXAMPLES

ID	Length	Difficulty
1	80	0.2687
2	128	0.2695
3	32	0.2109
4	32	0.2109
5	96	0.2839
6	64	0.2734
7	64	0.3047
8	192	0.2331
9	128	0.2637
10	96	0.2682

D. Classification

Classification was performed using rules to discriminate compositions into three classes, which are basic, intermediate, and advanced. The difficulty level rules are defined by referring to the lowest value in the middle and advanced classes, the composition included in the basic class must be a composition that has a lower value than the lowest value found in the intermediate class, and the composition included in the intermediate class must be a composition that has a value between the lowest value found in the intermediate class to the lowest value found in the advanced class, while the composition included in the advanced class must be a composition that has a value greater than or equal to the lowest value found in that class. The following is the determination of the difficulty level rules:

IF value < minimum value in the intermediate class

THEN basic class

IF minimum value in the intermediate class ≥ value < minimum value in the advanced class

THEN intermediate class

IF value ≥ minimum value in the advanced class

THENadvanced class

Classification was performed by implementing the rules of the difficulty level based on the 50 compositions which have been divided into three classes by experts. Classification carried out by experts resulted in the division of 19, 17, and 14 compositions into basic, intermediate and advanced classes, respectively. Table V shows the results of the classification of compositions by experts and the value (difficulty level) of each composition. The data are displayed by sorting them based on the composition ID in ascending order.

There are different interpretations in the classification of the level of difficulty between the experts and the value of the level of difficulty. For example, the composition with ID 11 which is based on the assessment of the expert belongs to the basic class, has a difficulty value of 0.2969 that greater than the lowest difficulty value in the middle class, which is 0.2331 for composition ID 8. Moreover, all compositions in the intermediate class have a difficulty value greater than the lowest difficulty value in the advanced class. In this case, the minimum and maximum values in each class need to be rearrangement. However, the rearrangement has an impact on the distribution of compositions by class. In other words, there will be a difference classification results between experts and the classifier.

The results of the classification by experts were used as a basis in determining the rules of the range of values for the level of difficulty for each class. Differences in the interpretation of the classification results by experts with the difficulty level value of each composition are resolved by finding the mean difficulty value in each class and inter-classes. Further, the mean difficulty values inter-classes are used as parameters to determine the value range of the difficulty level in each class. The following is data of the difficulty value of each composition based on its class collected from Table V sorted based on the value in ascending order. It is important to underline that the distribution of compositions into each class is determined by experts.

Basic = 0.1953, 0.1953, 0.1953, 0.2031, 0.2031, 0.2070, …, 0.3516)

Intermediate = (0.2331, 0.2484, 0.2500, 0.2559, 0.2578, …, 0.2734)

Advanced = 0.2406, 0.2578, 0.2695, 0.2760, 0.2773, …, 0.3984)

Next is calculating the mean value in each class. Mean difficulty value of the basic class is 0.2365, and the intermediate is 0.2614, while the advanced class is 0.2930. Finally, the minimum difficulty value in the intermediate class was determined based on the difficulty mean value calculation from the basic class mean value and the intermediate class mean value, which is (0.2365 + 0.2614) / 2 = 0.2489. Meanwhile, the minimum value of the difficulty level for the advanced class was determined based on the mean value calculation from the intermediate class mean value and the advanced class mean value, which is (0.2614 + 0. 2930) / 2 = 0.2772.

IF 0 < difficulty value < 0.2489

THEN basic class

IF 0.2489 ≤ difficulty value < 0.2772

THEN intermediate class

IF 0.2772 ≤ difficulty value ≤ 1

THEN advanced class

TABLE V. CLASSIFICATION RESULTS BY EXPERTS AND THE DIFFICULTY VALUES

Basic			**Intermediate**			**Advanced**		
ID	***Length***	***Value***	***ID***	***Length***	***Value***	***ID***	***Length***	***Value***
3	32	0.2109	1	80	0.2687	2	128	0.2695
4	32	0.2109	6	64	0.2734	5	96	0.2839
11	32	0.2969	8	192	0.2331	7	64	0.3047
14	64	0.1953	9	128	0.2637	13	96	0.2760
19	64	0.1953	10	96	0.2682	15	64	0.2813
20	64	0.2500	12	64	0.2578	24	160	0.3984
21	64	0.2422	16	64	0.2734	27	64	0.3281
22	96	0.2474	17	128	0.2559	30	96	0.2995
23	32	0.3516	18	64	0.2617	36	128	0.2773
26	64	0.2070	25	64	0.2734	38	320	0.2406
29	96	0.2083	28	128	0.2578	40	64	0.2969
32	64	0.2305	31	64	0.2656	41	128	0.2578
33	64	0.2383	35	64	0.2617	45	64	0.2813
34	32	0.3125	37	160	0.2484	46	96	0.3073
39	64	0.2500	42	64	0.2695			
43	64	0.2031	44	96	0.2500			
47	96	0.2031	50	96	0.2604			
48	128	0.2441						
49	96	0.1953						
Mean		**0.2365**	**Mean**		**0.2614**	**Mean**		**0.2930**

IV. Results and Discussion

The difficulty level rules above were evaluated by comparing the content of the three classes set by experts and the classifier. Compositions ID 11, 23 and 34 which are classified by the expert into the basic class are shifted out of the class. The compositions have difficulty value of 0.2969, 0.3516, and 0.3125, respectively, so they are shifted to the advanced class. Still in the basic class, both composition ID 20 and 39 have a difficulty value of 0.25, and they are shifted to the intermediate class. Meanwhile, in the advanced class, compositions ID 13 and 41 that have difficulty value of 0.2760 and 0.2578, respectively, are shifted to the intermediate class, while composition ID 38 that has the difficulty value of 0.2406 is shifted to the basic class.

The intermediate class showed positive results where there was only one classification difference between the experts and the classifier. Composition of ID 8 which is classified by experts into the intermediate class has a difficulty level value of 0.2331 that is in the basic class. Compositions classified by experts into basic classes have a number of beats less than or equal to128, while composition ID 8 is 192 beats long. Although the difficulty value of the composition indicates that it is in the basic class, the number of beats seems to be considered more by experts. This condition also applies to class shifts between basic and advanced classes.

Class shifts were found in several other cases so that the number of compositions in the basic, intermediate and advanced classes originally determined by the expert was 19, 17, and 14, changed to a total of 16, 21, 32 by the classifier, where changes also occur in some of its contents. Table VI shows the comparative results of the classification by experts and the classifier with B, I, A, E, and C, representing basic class, intermediate class, advanced class, experts, and the classifier, respectively.

Evaluation was carried out to measure the performance of the classifier by comparing the results of its classification to the classification by experts. The classifier performance was measured using a confusion matrix, as shown in Table VII. The results of the confusion matrix show that 14 of the 19 compositions classified by experts into basic classes can be identified by the classifier. Meanwhile, in the intermediate and advanced classes, 16 of the 17 compositions and 11 of the 14 compositions, respectively, can be identified by the classifier.

It is interesting to find out that three compositions of the basic class, which are compositions ID 11, 23 and 34, are shifted to the advanced class by the classifier. The three compositions have a length of 32 beats, and a melody with a length of 32 beats is the shortest composition in the dataset. The length of the melody seems to have a significant role compared to the inter-notes distance value. Another evidence can be seen in composition ID 39 which has a melody length of 320 beats. This composition is classified by experts into the advanced class. Meanwhile, it is shifted by the classifier to the basic class based on the difficulty value. The fact that all compositions classified by experts into the intermediate class have a melody length greater than 32 beats is another evidence.

Overall, the frequent and inter-notes distance analysis proposed in this research show good results in performing multi-class classifications for the difficulty level of composition based on the melodic pattern. The confusion matrix results were then calculated to measure accuracy, precision, and recall of the classifier performance. The performance of the classifier in the basic, intermediate and advanced classes achieved an accuracy of 87.5%, 80%, a precision of 73.7%, 94.1%, and 78.6%, and a recall of 87.5%, 80%, and 78.6%. Meanwhile, in total, the classifier's performance reached an accuracy, precision and recall value of 82%, 82.1%, and 82%.

The rule of difficulty classification is built on the frequency of notes and analysis of the distance between notes that is definitely found in all types of music, the difficulty classification model proposed in this study has a high potential to be applied to various types of music.

TABLE VI. Classification Results by the Classifier

ID	**E**	**C**	**ID**	**E**	**C**	**ID**	**E**	**C**	**ID**	**E**	**C**	**ID**	**E**	**C**
1	I	I	11	B	A	21	B	B	31	I	I	41	A	I
2	A	I	12	I	I	22	B	B	32	B	B	42	I	I
3	B	B	13	A	I	23	B	A	33	B	B	43	B	B
4	B	B	14	B	B	24	A	A	34	B	A	44	I	I
5	A	A	15	A	A	25	I	I	35	I	I	45	A	A
6	I	I	16	I	I	26	B	B	36	A	A	46	A	A
7	A	A	17	I	I	27	A	A	37	I	I	47	B	B
8	I	B	18	I	I	28	I	I	38	A	B	48	B	B
9	I	I	19	B	B	29	B	B	39	B	I	49	B	B
10	I	I	20	B	I	30	A	A	40	A	A	50	I	I

TABLE VII. CONFUSION MATRIX RESULTS

		Classifier		
		B	***I***	***A***
Experts	***B***	15	1	3
	I	1	16	0
	A	2	1	11

V. CONCLUSION AND FUTURE WORK

This research proposes a method for melodic patterns difficulty classification into basic, intermediate and advanced classes. The limited number of datasets is considered unsuitable for computing using a machine learning approach. Hence, instead of using a machine learning approach, classification was performed using a set of rules defined based on frequent pattern and inter-notes distance analysis. In successive notes, the forward notes and backward notes have the same pattern in the value of the distance between the notes. Thus, although the melody has a sequence or time series data type, the characteristics of the problem in the difficulty classification of melodic patterns are more suitable solved using frequent pattern analysis than sequence pattern. As a result, the proposed method is able to build a classifier to discriminate the difficulty level of composition with a good accuracy.

The future projection after completing this research is to implement it into adaptive learning to play musical instruments, and studies related to the classification of difficulty levels in composition are not yet popular even though this topic has relevance in adaptive learning. Fig. 3 shows a workflow diagram of future work where this research position is in the dashed-line box.

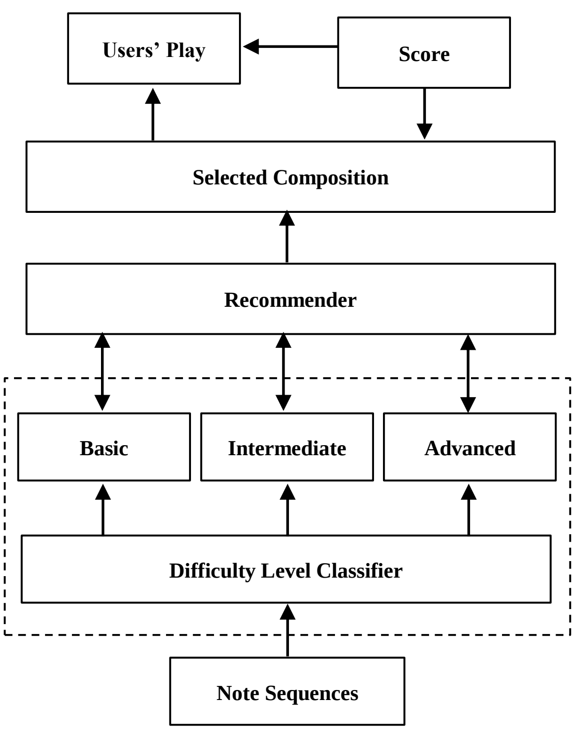

Fig. 3. Research Positions in Future Work.

ACKNOWLEDGMENT

This research is part of the research program of the Gamelan Metaverse lab developed by Universitas Dian Nuswantoro.

REFERENCES

[1] H. Khosravi, S. Sadiq, and D. Gasevi, "Development and Adoption of an Adaptive Learning System: Reflections and Lessons Learned," Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20). Association for Computing Machinery, New York, USA, 2020, pp. 58–64. Doi: 10.1145/3328778.3366900.

[2] V. Mirata, F. Hirt, P. Bergamin, and C. Van der Westhuizen, "Challenges and contexts in establishing adaptive learning in higher education: Findings from a Delphi study," International Journal of Educational Technology in Higher Education, vol. 17, no. 32, 2020, pp. 1-25. Doi: 10.1186/s41239-020-00209-y.

[3] A.V. Vega, O.C. Madrigal, and V. Kugurakova, "Approach of immersive adaptive learning for virtual reality simulator," Proceedings of 3rd Workshop on Advanced Virtual Environments and Education (WAVE2 2021), March 21-24, 2021, pp. 1-8. Doi: 10.5753/wave.2020.211635.

[4] W. Holmes, M. Bialik, and C. Fadel, "Artificial Intelligence in Education Promises and Implications for Teaching and Learning," The Center for Curriculum Redesign, Boston, MA, 2019.

[5] Y. Hao, K.S. Lee, S-T. Chen, and S.C. Sim, "An Evaluative Research of a Mobile Application for Middle School Students Struggling with English Vocabulary Learning," Computers in Human Behavior, 2019, vol. 95, pp. 208-216. Doi: 10.1016/j.chb.2018.10.013.

[6] M. Haug, P. Camps, T. Umland, and J-N. Voigt-Antons, "Assessing Differences in Flow State Induced by an Adaptive Music Learning Software," 2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX), 2020, pp. 1-4. Doi: 10.1109/QoMEX48832.2020.9123132.

[7] H. Peng, S. Ma, and J.M. Spector, "Personalized adaptive learning: an emerging pedagogical approach enabled by a smart learning environment," Smart Learning Environments, 2019, vol. 6, no. 9. Doi: 10.1186/s40561-019-0089-y.

[8] N. Yudana, and S. Wahyono, "The development of Gamelan learning media for android operating system," Indonesian journal of curriculum and educational technology studies, 2019, vol. 7, no. 2, pp. 64-71. Doi: 10.15294/ijcets.v7i2.29443.

[9] G. Presti, D. Adriano, F. Avanzini, A. Baratè, and L.A. Ludovico, "PhonHarp: A Hybrid Digital-Physical Musical Instrument for Mobile Phones Exploiting the Vocal Tract," Audio Mostly 2021 (AM '21). Association for Computing Machinery, New York, NY, USA, 2021, pp. 276-279. Doi: 10.1145/3478384.3478413.

[10] K. Lyu and R. Li, " Development of a Virtual Yangqin App with Unity Based on the Audio Object Pool Pattern," in: Shao X., Qian K., Zhou L., Wang X., Zhao Z. (eds), Proceedings of the 8th Conference on Sound and Music Technology. CSMT 2020. Lecture Notes in Electrical Engineering, vol 761. Springer, Singapore, 2021. Doi: 10.1007/978-981-16-1649-5_3.

[11] D. Lopes, G. Bernardes, L. Aly, and J. Forero, "Tumaracatu: An ubiquitous digital musical experience of maracatu," 11th Workshop on Ubiquitous Music (UbiMus 2021), Matosinhos, Portugal, 2021, October 12. Doi: 10.5281/zenodo.5564751.

[12] A.Z. Fanani, K. Hastuti, A.M. Syarif, and A.R. Mulyana, "Rule-based interactive learning application model on how to play music instruments," International Journal of Emerging Technologies in Learning, 2020, vol. 15, no. 15, pp. 52-63.

[13] S. Ruan, L. Jiang, J. Xu, B.J-K. Tham, Z. Qiu, Y. Zhu, E.L. Murnane, E. Brunskill, and J.A. Landay. "QuizBot: A Dialogue-based Adaptive Learning System for Factual Knowledge. Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, New York, NY, USA, 2019, 1–13. Doi: 10.1145/3290605.3300587.

[14] M. Megahed and A. Mohammed, "Modeling adaptive E-Learning environment using facial expressions and fuzzy logic," Expert Systems

with Applications, 2020, vol. 157, no. 1. Doi: 10.1016/j.eswa.2020.113460.

[15] J. Yang, H. Huang, and S. Jin, "Mining Web Access Sequence with Improved Apriori Algorithm," 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), 2017, pp. 780-784. Doi: 10.1109/CSE-EUC.2017.154.

[16] C. Wang, and X. Zheng, "Application of improved time series Apriori algorithm by frequent itemsets in association rule data mining based on temporal constraint," Evol. Intel., 2020, vol. 13, pp. 39-49. Doi: 10.1007/s12065-019-00234-5.

[17] S.B. Naik, and S. Khan, "Application of Association Rule Mining-Based Attribute Value Generation in Music Composition," in: V. Bhateja, S.C. Satapathy, C.M. Travieso-González, V.N.M. Aradhya (eds) Data Engineering and Intelligent Computing. Advances in Intelligent Systems and Computing, 2021, vol. 1407. Springer, Singapore. Doi: 10.1007/978-981-16-0171-2_36.

[18] K. Hastuti, A. Azhari, A. Musdholifah, and R. Supanggah, "Building Melodic Feature Knowledge of Gamelan Music Using Apriori Based on Functions in Sequence (AFiS) Algorithm," International Review on Computers and Software, 2016, vol. 11, no. 12, pp. 1127-1137. Doi: 10.15866/irecos.v11i12.10841.

[19] K.M. Jhang, M.C. Chang, T.Y. Lo, C.W. Lin, W.F. Wang, and H.H. Wu, "Using The Apriori Algorithm To Classify The Care Needs Of Patients With Different Types Of Dementia," Patient preference and adherence, 2019, vol. 13, pp. 1899–1912. Doi: 10.2147/PPA.S223816.

[20] X. Xie, G. Fu, Y. Xue, Z. Zhao, P. Chen, B. Lu, and S. Jiang, "Risk prediction and factors risk analysis based on IFOA-GRNN and apriori algorithms: Application of artificial intelligence in accident prevention," Process Safety and Environmental Protection, 2018, Doi: 10.1016/j.psep.2018.11.019.

[21] P. Rojanavasu, "Educational Data Analytics using Association Rule Mining and Classification," Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT-NCON), 2019, pp. 142-145, Doi: 10.1109/ECTI-NCON.2019.8692274.

[22] A.M. Syarif, A. Azhari, S. Suprapto, and K. Hastuti, "Human and Computation-based Music Representation for Gamelan Music," Malaysian Journal of Music, 2020, vol. 9, pp. 82-100. Doi: 10.37134/mjm.vol9.7.2020.

Appendix

Dataset

TABLE VIII. Compositions of Ladrang style of the Pelog musical Scale and the Lima musical mode

No	Title	Note Sequence
1	Ladrang Arum Asih	0, 1, 0, 1, 2, 1, 2, 3, 0, 5, 1, 2, 3, 1, 2, 3, 0, 5, 1, 2, 3, 1, 2, 3, 1, 1, 0, 5, 6, 1, 2, 1, 0, 1, 0, 1, 2, 1, 2, 3, 0, 5, 1, 2, 3, 1, 2, 3, 0, 5, 1, 2, 3, 1, 2, 3, 5, 5, 0, 6, 1, 6, 5, 4, 0, 4, 0, 4, 2, 4, 5, 6, 1, 6, 5, 4, 2, 4, 5, 4
2	Ladrang Babar Layar	6, 5, 6, 3, 6, 5, 6, 3, 6, 5, 6, 3, 6, 5, 3, 2, 5, 3, 2, 5, 3, 2, 5, 3, 2, 5, 2, 3, 5, 6, 5, 3, 6, 5, 6, 3, 6, 5, 6, 3, 6, 5, 6, 3, 6, 5, 3, 2, 5, 3, 2, 5, 3, 2, 5, 3, 2, 5, 2, 3, 5, 6, 5, 4, 0, 4, 0, 4, 0, 4, 0, 1, 0, 1, 0, 1, 0, 1, 0, 5, 0, 1, 0, 5, 0, 1, 0, 5, 0, 4, 4, 6, 4, 5, 6, 1, 6, 5, 4, 6, 4, 5, 6, 1, 6, 5, 4, 6, 4, 5, 6, 1, 6, 5, 4, 6, 4, 5, 6, 1, 2, 3, 2, 1, 6, 5, 6, 3
3	Ladrang Balabak	3, 2, 3, 1, 3, 2, 3, 5, 3, 2, 3, 1, 3, 2, 3, 5, 0, 0, 7, 6, 5, 4, 2, 1, 3, 2, 3, 1, 3, 2, 3, 5
4	Ladrang Banten	3, 2, 3, 1, 3, 2, 3, 5, 6, 3, 6, 5, 3, 2, 3, 5, 6, 3, 6, 5, 3, 2, 3, 5, 6, 3, 6, 5, 3, 2, 3, 1
5	Ladrang Banyak Nglangi	0, 6, 1, 2, 1, 6, 4, 5, 0, 6, 1, 2, 1, 6, 4, 5, 0, 6, 1, 2, 1, 6, 4, 5, 3, 3, 0, 0, 2, 1, 2, 3, 0, 1, 2, 3, 5, 1, 2, 3, 0, 1, 2, 3, 5, 1, 2, 3, 0, 1, 2, 3, 5, 1, 2, 3, 1, 1, 0, 0, 6, 5, 4, 5, 0, 2, 2, 0, 6, 5, 4, 5, 0, 2, 2, 0, 6, 5, 4, 5, 0, 2, 5, 4, 0, 2, 5, 4, 0, 2, 5, 4, 2, 1, 6, 5
6	Ladrang Bayemtur	0, 3, 5, 6, 3, 5, 3, 2, 0, 3, 5, 6, 3, 5, 3, 2, 0, 3, 3, 0, 3, 6, 3, 5, 3, 6, 3, 5, 3, 1, 3, 2, 0, 4, 4, 4, 2, 1, 2, 6, 0, 4, 4, 4, 2, 1, 2, 6, 0, 3, 3, 0, 3, 6, 3, 5, 3, 6, 3, 5, 3, 1, 3, 2
7	Ladrang Bedhati	0, 6, 1, 2, 1, 6, 4, 5, 0, 6, 1, 2, 1, 6, 4, 5, 0, 0, 5, 6, 5, 3, 2, 1, 6, 5, 6, 1, 2, 3, 2, 1, 0, 6, 5, 6, 1, 2, 3, 1, 0, 6, 5, 6, 1, 2, 3, 1, 0, 6, 5, 6, 3, 2, 3, 1, 3, 2, 1, 6, 2, 1, 6, 5
8	Ladrang Bima Kurda	2, 1, 2, 1, 2, 5, 6, 1, 2, 1, 2, 1, 2, 5, 6, 1, 2, 1, 2, 1, 2, 5, 6, 1, 5, 6, 1, 2, 5, 3, 2, 1, 3, 3, 0, 0, 1, 2, 3, 2, 3, 1, 2, 0, 3, 2, 1, 2, 3, 1, 2, 0, 3, 2, 1, 2, 0, 1, 6, 0, 5, 6, 2, 1, 5, 5, 0, 0, 1, 6, 5, 3, 2, 3, 5, 0, 1, 6, 5, 1, 2, 3, 5, 0, 1, 6, 5, 3, 0, 2, 1, 0, 5, 6, 2, 1, 2, 1, 2, 1, 2, 5, 6, 1, 2, 1, 2, 1, 2, 5, 6, 1, 2, 1, 2, 1, 2, 5, 6, 1, 5, 6, 1, 2, 5, 3, 2, 1, 3, 3, 5, 3, 1, 2, 3, 2, 3, 1, 2, 3, 1, 2, 3, 2, 3, 1, 2, 3, 1, 2, 3, 2, 0, 1, 6, 0, 5, 6, 2, 1, 5, 5, 6, 5, 7, 6, 5, 3, 2, 3, 5, 6, 7, 6, 5, 3, 2, 3, 5, 6, 7, 6, 5, 3, 0, 2, 1, 0, 5, 6, 2, 1
9	Ladrang Blabak	0, 6, 5, 4, 2, 4, 6, 5, 1, 1, 0, 0, 2, 3, 2, 1, 5, 6, 1, 5, 6, 1, 2, 1, 3, 2, 1, 2, 0, 1, 2, 6, 0, 6, 5, 4, 2, 4, 6, 5, 7, 7, 0, 0, 7, 7, 6, 5, 2, 3, 5, 2, 3, 5, 6, 5, 7, 6, 5, 6, 5, 3, 2, 3, 6, 5, 2, 1, 6, 1, 3, 2, 7, 7, 0, 0, 7, 7, 6, 5, 2, 3, 5, 2, 3, 5, 6, 5, 7, 6, 5, 6, 5, 3, 2, 1, 6, 5, 2, 1, 6, 1, 3, 2, 5, 5, 0, 0, 5, 4, 2, 1, 5, 6, 1, 5, 6, 1, 2, 1, 3, 2, 1, 2, 0, 1, 2, 6
10	Ladrang Dhandanggulo Maskentar	0, 0, 6, 0, 6, 6, 1, 2, 0, 0, 2, 1, 3, 2, 1, 6, 0, 0, 6, 0, 5, 5, 6, 1, 3, 2, 1, 2, 0, 1, 6, 5, 0, 0, 0, 0, 1, 2, 1, 6, 2, 1, 5, 2, 0, 1, 0, 6, 3, 3, 0, 0, 6, 5, 3, 2, 0, 3, 2, 1, 6, 5, 3, 5, 2, 2, 0, 1, 3, 2, 1, 6, 2, 3, 2, 1, 6, 5, 3, 5, 2, 2, 0, 0, 5, 6, 5, 4, 6, 5, 2, 3, 2, 1, 2, 1
11	Ladrang Dhengklung	1, 6, 1, 2, 1, 6, 3, 5, 4, 2, 4, 5, 4, 2, 4, 1, 4, 2, 4, 5, 4, 2, 4, 1, 5, 6, 1, 2, 1, 6, 4, 5
12	Ladrang Dhudha Kondhang	3, 2, 5, 6, 1, 5, 6, 1, 3, 2, 5, 6, 1, 5, 6, 1, 3, 2, 5, 6, 1, 5, 6, 1, 2, 3, 5, 3, 2, 1, 2, 1, 3, 3, 0, 0, 3, 3, 5, 3, 5, 6, 7, 6, 5, 3, 2, 3, 5, 5, 0, 0, 7, 6, 5, 3, 1, 1, 0, 5, 6, 1, 2, 1
13	Ladrang Durma	0, 1, 1, 1, 6, 1, 2, 3, 0, 0, 3, 2, 0, 1, 6, 5, 1, 5, 0, 0, 5, 5, 0, 6, 1, 1, 0, 5, 6, 1, 2, 1, 5, 5, 0, 0, 5, 5, 3, 5, 6, 6, 7, 6, 5, 4, 2, 1, 5, 5, 0, 6, 5, 3, 1, 2, 3, 2, 1, 6, 5, 6, 1, 2, 0, 0, 2, 3, 5, 5, 3, 5, 6, 6, 5, 4, 2, 1, 6, 5, 1, 5, 0, 0, 5, 5, 0, 6, 1, 1, 0, 5, 6, 1, 2, 1
14	Ladrang Eling-Eling	6, 5, 3, 2, 1, 2, 3, 5, 6, 5, 3, 2, 1, 2, 3, 5, 1, 1, 0, 0, 1, 2, 3, 5, 3, 2, 3, 1, 3, 2, 6, 5, 6, 5, 2, 1, 3, 2, 6, 5, 6, 5, 2, 1, 3, 2, 6, 5, 2, 1, 2, 1, 3, 2, 6, 5, 3, 2, 3, 1, 3, 2, 6, 5

15	Ladrang Eling-Eling Subasiti	6, 5, 3, 2, 1, 2, 3, 5, 6, 5, 6, 1, 3, 2, 6, 5, 6, 5, 2, 1, 3, 2, 3, 1, 2, 1, 5, 6, 1, 1, 2, 1, 6, 1, 6, 1, 6, 1, 2, 1, 4, 5, 4, 5, 6, 1, 6, 5, 0, 3, 2, 1, 6, 5, 6, 1, 6, 6, 2, 1, 2, 6, 3, 5
16	Ladrang Glendheh	5, 6, 1, 0, 1, 3, 1, 2, 5, 6, 1, 0, 1, 3, 1, 2, 5, 6, 1, 0, 1, 3, 1, 2, 5, 6, 5, 4, 2, 1, 2, 1, 5, 5, 0, 0, 6, 4, 6, 5, 0, 5, 5, 5, 6, 4, 6, 5, 0, 0, 4, 0, 2, 0, 4, 0, 2, 4, 6, 5, 0, 4, 2, 1
17	Ladrang Golong	0, 6, 1, 2, 1, 6, 3, 5, 3, 2, 3, 0, 3, 6, 3, 5, 3, 2, 3, 0, 3, 6, 3, 5, 4, 2, 1, 2, 1, 6, 3, 5, 0, 0, 5, 2, 3, 5, 6, 5, 0, 0, 5, 6, 7, 7, 5, 6, 0, 6, 3, 5, 6, 7, 5, 6, 7, 5, 3, 2, 5, 6, 5, 3, 6, 5, 6, 3, 6, 5, 6, 3, 6, 5, 6, 3, 6, 5, 3, 2, 3, 1, 6, 1, 2, 3, 5, 3, 6, 5, 3, 5, 3, 2, 3, 2, 3, 5, 6, 3, 5, 6, 7, 6, 7, 5, 6, 7, 6, 5, 3, 2, 5, 6, 5, 4, 2, 1, 2, 1, 3, 5, 3, 2, 1, 6, 3, 5
18	Ladrang Gudhasih	0, 6, 1, 2, 1, 6, 4, 5, 3, 1, 3, 2, 1, 6, 4, 5, 2, 2, 0, 0, 2, 3, 2, 1, 5, 6, 1, 2, 1, 6, 4, 5, 0, 5, 5, 5, 6, 4, 6, 5, 2, 4, 5, 6, 5, 4, 2, 1, 0, 2, 4, 5, 4, 2, 4, 1, 5, 6, 1, 2, 1, 6, 4, 5
19	Ladrang Gunung Kembar	0, 3, 2, 3, 5, 6, 5, 3, 0, 3, 2, 3, 5, 6, 5, 3, 0, 0, 3, 2, 3, 5, 6, 5, 0, 0, 3, 5, 3, 2, 3, 1, 0, 1, 2, 0, 2, 3, 2, 1, 0, 1, 2, 0, 2, 1, 3, 2, 0, 1, 6, 5, 6, 6, 5, 6, 3, 3, 2, 3, 5, 6, 5, 3
20	Ladrang Hastama	2, 1, 2, 4, 5, 4, 2, 1, 2, 1, 2, 4, 5, 4, 2, 1, 3, 2, 1, 2, 0, 1, 6, 5, 1, 5, 0, 6, 1, 0, 2, 1, 5, 5, 0, 0, 5, 5, 4, 5, 6, 6, 5, 6, 4, 5, 6, 5, 6, 5, 4, 2, 1, 6, 4, 5, 1, 5, 0, 6, 1, 0, 2, 1
21	Ladrang Jong Layar	0, 0, 0, 0, 2, 2, 3, 2, 0, 0, 3, 5, 0, 0, 3, 2, 0, 0, 3, 5, 0, 0, 3, 2, 5, 3, 2, 5, 2, 3, 5, 6, 0, 0, 0, 0, 6, 6, 5, 6, 0, 0, 3, 5, 0, 0, 3, 2, 0, 0, 3, 5, 0, 0, 3, 2, 5, 3, 2, 5, 2, 3, 5, 6
22	Ladrang Kagok	0, 1, 1, 1, 5, 6, 2, 1, 0, 1, 1, 1, 5, 6, 1, 2, 0, 0, 2, 4, 5, 0, 6, 5, 6, 6, 5, 4, 2, 1, 2, 1, 5, 5, 0, 0, 5, 5, 3, 5, 0, 0, 5, 6, 7, 6, 5, 6, 0, 6, 5, 3, 2, 2, 3, 2, 0, 0, 2, 4, 5, 0, 6, 5, 7, 6, 5, 6, 5, 4, 2, 1, 3, 2, 1, 2, 0, 1, 6, 5, 0, 6, 1, 2, 0, 1, 6, 5, 1, 1, 0, 5, 6, 1, 2, 1
23	Ladrang Kapirekta	0, 6, 1, 2, 1, 6, 4, 5, 0, 6, 1, 2, 1, 6, 4, 5, 3, 5, 3, 5, 6, 1, 6, 5, 7, 6, 2, 4, 2, 1, 6, 5
24	Ladrang Kodhokan	0, 6, 0, 3, 0, 5, 0, 2, 0, 6, 0, 3, 0, 5, 0, 2, 0, 6, 0, 3, 0, 5, 0, 2, 0, 6, 0, 5, 0, 3, 0, 2, 0, 1, 3, 2, 5, 6, 1, 2, 0, 1, 3, 2, 5, 6, 1, 2, 0, 1, 3, 2, 5, 6, 1, 2, 0, 6, 0, 5, 0, 3, 0, 2, 0, 3, 3, 0, 3, 5, 2, 3, 5, 2, 3, 0, 3, 5, 2, 3, 5, 2, 3, 0, 3, 5, 2, 3, 0, 6, 0, 5, 0, 3, 0, 2, 0, 7, 0, 7, 0, 6, 0, 5, 0, 4, 0, 2, 0, 4, 0, 1, 0, 4, 0, 2, 0, 4, 0, 1, 0, 4, 0, 6, 0, 4, 0, 5, 0, 0, 5, 6, 7, 7, 6, 7, 0, 6, 5, 6, 7, 7, 6, 7, 0, 6, 5, 6, 7, 7, 6, 7, 0, 6, 0, 5, 0, 3, 0, 2
25	Ladrang Kombang Mara	5, 3, 2, 1, 5, 3, 2, 1, 5, 3, 2, 1, 2, 1, 6, 5, 6, 1, 6, 5, 6, 1, 6, 5, 6, 1, 2, 3, 5, 3, 2, 1, 5, 5, 0, 0, 4, 4, 2, 5, 0, 0, 1, 6, 5, 4, 2, 5, 0, 0, 1, 6, 5, 4, 2, 1, 6, 1, 2, 3, 5, 3, 2, 1
26	Ladrang Kudhawa	3, 2, 3, 1, 3, 2, 3, 1, 3, 2, 3, 1, 0, 2, 3, 5, 0, 0, 5, 6, 7, 7, 6, 5, 3, 2, 3, 5, 3, 2, 3, 1, 5, 5, 0, 0, 5, 5, 3, 5, 6, 5, 3, 2, 1, 2, 3, 5, 6, 5, 3, 2, 1, 6, 3, 5, 3, 2, 3, 5, 3, 2, 3, 1
27	Ladrang Kumara Maya	6, 1, 6, 2, 6, 1, 6, 5, 6, 1, 6, 2, 6, 1, 6, 5, 0, 5, 5, 5, 6, 4, 6, 5, 1, 2, 1, 6, 5, 4, 2, 1, 5, 6, 1, 6, 5, 4, 2, 1, 5, 6, 1, 6, 5, 4, 2, 1, 6, 6, 0, 0, 6, 5, 4, 2, 4, 5, 6, 5, 2, 1, 6, 5
28	Ladrang Langen Branta	0, 1, 0, 1, 6, 1, 2, 3, 5, 6, 5, 3, 2, 1, 2, 1, 0, 5, 5, 0, 5, 6, 1, 2, 3, 3, 5, 3, 2, 1, 2, 1, 0, 1, 0, 1, 6, 1, 2, 3, 5, 6, 5, 3, 2, 1, 2, 1, 0, 5, 5, 0, 5, 6, 1, 2, 3, 5, 3, 2, 1, 6, 3, 5, 6, 5, 6, 0, 6, 5, 2, 1, 3, 5, 3, 2, 1, 6, 3, 5, 0, 4, 4, 2, 4, 5, 2, 1, 3, 5, 3, 2, 1, 6, 3, 5, 6, 5, 6, 0, 6, 5, 2, 1, 3, 5, 3, 2, 1, 6, 3, 5, 0, 4, 4, 2, 4, 5, 2, 1, 3, 3, 5, 3, 2, 1, 2, 1
29	Ladrang Larastangis	0, 1, 1, 1, 2, 3, 2, 1, 0, 1, 1, 1, 2, 3, 2, 1, 0, 2, 1, 0, 2, 1, 6, 5, 0, 0, 5, 6, 1, 2, 3, 2, 0, 0, 0, 0, 2, 2, 3, 2, 0, 0, 2, 3, 2, 1, 2, 1, 0, 2, 1, 0, 2, 1, 6, 5, 0, 0, 5, 6, 1, 2, 3, 2, 5, 5, 0, 0, 5, 5, 3, 5, 0, 0, 5, 6, 7, 7, 6, 7, 0, 0, 0, 0, 7, 6, 5, 3, 0, 0, 2, 5, 0, 3, 2, 1
30	Ladrang Lebdajiwa	0, 6, 1, 2, 1, 6, 4, 5, 0, 6, 1, 2, 1, 6, 4, 5, 1, 1, 0, 0, 5, 6, 1, 2, 1, 3, 1, 2, 0, 1, 6, 5, 0, 6, 1, 2, 1, 6, 4, 5, 0, 6, 1, 2, 1, 6, 4, 5, 0, 0, 0, 0, 6, 4, 6, 5, 2, 4, 5, 6, 5, 4, 1, 2, 6, 6, 0, 0, 2, 1, 6, 5, 1, 2, 1, 6, 5, 4, 1, 2, 1, 1, 0, 0, 5, 6, 1, 2, 1, 3, 1, 2, 0, 1, 6, 5
31	Ladrang Manik Maninten	0, 1, 1, 1, 2, 3, 2, 1, 5, 5, 0, 0, 6, 1, 6, 5, 0, 5, 3, 5, 6, 1, 6, 5, 6, 1, 2, 1, 2, 3, 2, 1, 5, 5, 0, 0, 6, 1, 6, 5, 6, 1, 2, 1, 2, 3, 2, 1, 0, 1, 6, 1, 2, 3, 2, 1, 6, 6, 5, 4, 2, 1, 2, 1
32	Ladrang Maya	0, 1, 1, 1, 5, 6, 2, 1, 2, 1, 2, 3, 5, 3, 2, 1, 3, 2, 1, 2, 0, 1, 6, 5, 1, 5, 0, 6, 1, 0, 2, 1, 5, 5, 0, 0, 5, 5, 6, 5, 7, 6, 5, 6, 5, 4, 2, 1, 5, 5, 0, 6, 4, 5, 6, 5, 6, 6, 5, 4, 2, 1, 2, 1
33	Ladrang Menggak Layar	3, 2, 5, 6, 1, 5, 6, 1, 3, 2, 5, 6, 1, 5, 6, 1, 3, 2, 5, 6, 1, 5, 6, 1, 2, 3, 5, 3, 2, 1, 2, 1, 3, 3, 0, 0, 3, 3, 5, 3, 5, 6, 7, 6, 5, 3, 2, 3, 0, 3, 5, 6, 7, 6, 5, 3, 5, 6, 5, 3, 2, 1, 2, 1
34	Ladrang Nusantara	6, 5, 1, 6, 2, 1, 6, 5, 6, 5, 1, 6, 2, 1, 6, 5, 7, 6, 5, 6, 3, 5, 3, 2, 5, 3, 1, 6, 2, 1, 6, 5
35	Ladrang Obah	0, 6, 1, 2, 3, 1, 6, 5, 0, 1, 5, 6, 1, 2, 3, 2, 3, 2, 3, 5, 6, 5, 3, 2, 1, 3, 1, 2, 0, 1, 6, 5, 0, 5, 5, 5, 6, 4, 6, 5, 6, 5, 6, 1, 2, 1, 6, 5, 1, 2, 1, 6, 5, 4, 1, 2, 1, 3, 1, 2, 0, 1, 6, 5
36	Ladrang Pacarcina	0, 3, 2, 1, 6, 1, 3, 2, 0, 3, 2, 1, 6, 1, 2, 3, 0, 2, 5, 3, 0, 2, 5, 3, 5, 5, 6, 1, 2, 3, 1, 2, 5, 5, 0, 0, 5, 5, 3, 5, 0, 0, 5, 6, 7, 6, 5, 6, 0, 6, 5, 3, 2, 3, 6, 5, 7, 6, 5, 4, 2, 1, 2, 1, 0, 0, 0, 0, 1, 1, 2, 1, 3, 5, 3, 2, 0, 1, 6, 5, 0, 0, 2, 3, 5, 6, 7, 6, 3, 5, 6, 5, 3, 2, 1, 2, 6, 1, 6, 2, 6, 1, 6, 5, 6, 1, 6, 2, 6, 1, 6, 5, 6, 1, 6, 2, 6, 1, 6, 5, 3, 3, 6, 5, 3, 2, 1, 2

37	Ladrang Pasang Wetan	0, 0, 0, 0, 2, 2, 3, 2, 0, 0, 2, 3, 5, 6, 5, 3, 0, 0, 5, 3, 2, 1, 2, 6, 1, 2, 0, 6, 1, 2, 3, 2, 0, 0, 0, 0, 2, 2, 3, 2, 0, 0, 2, 3, 5, 6, 5, 3, 0, 0, 5, 3, 2, 1, 2, 6, 3, 5, 0, 2, 3, 5, 6, 5, 0, 0, 0, 0, 5, 5, 3, 5, 6, 6, 0, 3, 6, 5, 3, 5, 3, 2, 0, 3, 5, 6, 0, 3, 5, 6, 0, 3, 6, 5, 3, 2, 0, 0, 3, 0, 1, 2, 3, 2, 0, 2, 1, 6, 5, 6, 1, 6, 0, 0, 1, 6, 0, 0, 1, 6, 7, 7, 0, 0, 5, 6, 7, 6, 0, 2, 2, 0, 2, 3, 1, 2, 0, 0, 2, 3, 5, 6, 5, 3, 0, 0, 5, 3, 2, 1, 2, 6, 1, 2, 0, 6, 1, 2, 3, 2
38	Ladrang Playon	0, 6, 1, 2, 1, 6, 4, 5, 3, 3, 6, 5, 3, 2, 1, 6, 5, 6, 1, 2, 3, 2, 1, 2, 1, 6, 5, 4, 2, 4, 6, 5, 0, 5, 4, 2, 1, 2, 4, 5, 6, 5, 4, 2, 1, 2, 4, 5, 6, 5, 4, 2, 1, 2, 3, 2, 6, 6, 0, 7, 5, 6, 7, 6, 0, 6, 5, 4, 2, 2, 3, 2, 0, 0, 2, 4, 5, 0, 6, 5, 6, 5, 4, 2, 1, 0, 2, 1, 0, 2, 4, 5, 4, 2, 4, 1, 0, 2, 4, 5, 4, 2, 4, 1, 0, 2, 4, 5, 4, 2, 4, 1, 0, 2, 4, 5, 4, 2, 4, 1, 5, 5, 0, 0, 4, 5, 6, 5, 6, 5, 4, 2, 1, 2, 3, 2, 6, 6, 0, 7, 6, 5, 4, 5, 6, 5, 4, 2, 1, 6, 4, 5, 0, 6, 1, 2, 1, 6, 4, 5, 0, 6, 1, 2, 1, 6, 4, 5, 3, 3, 6, 5, 3, 2, 1, 6, 5, 6, 1, 2, 3, 2, 1, 2, 1, 6, 5, 4, 2, 4, 6, 5, 3, 3, 0, 0, 3, 3, 5, 3, 6, 5, 2, 1, 6, 1, 2, 3, 6, 5, 2, 1, 0, 5, 6, 1, 0, 4, 1, 2, 4, 5, 6, 5, 6, 5, 4, 2, 1, 6, 5, 6, 3, 5, 3, 2, 3, 2, 1, 6, 5, 6, 1, 2, 3, 2, 1, 2, 1, 6, 5, 4, 2, 4, 6, 5, 6, 5, 4, 2, 1, 1, 2, 1, 1, 6, 5, 4, 2, 4, 6, 5, 0, 2, 4, 5, 4, 2, 4, 1, 0, 2, 4, 5, 4, 2, 4, 1, 0, 2, 4, 5, 4, 2, 4, 1, 3, 3, 6, 5, 3, 2, 1, 6, 5, 6, 1, 2, 3, 2, 1, 2, 1, 6, 5, 4, 2, 4, 6, 5
39	Ladrang Prasaja	1, 5, 0, 6, 1, 1, 2, 1, 2, 3, 5, 3, 2, 1, 2, 1, 3, 3, 0, 0, 6, 5, 3, 2, 1, 3, 1, 2, 1, 6, 4, 5, 0, 5, 5, 5, 6, 4, 6, 5, 7, 6, 5, 6, 5, 4, 2, 1, 5, 5, 0, 6, 5, 3, 1, 2, 1, 3, 1, 2, 1, 6, 4, 5
40	Ladrang Pujiwidada	0, 6, 2, 1, 3, 2, 6, 5, 1, 6, 1, 5, 1, 6, 1, 2, 3, 6, 3, 5, 3, 1, 3, 2, 3, 1, 3, 2, 1, 6, 4, 5, 0, 5, 5, 5, 6, 4, 6, 5, 1, 2, 1, 6, 5, 4, 1, 2, 3, 6, 3, 5, 3, 1, 3, 2, 3, 1, 3, 2, 1, 6, 4, 5
41	Ladrang Randha Ngangsu	1, 2, 1, 6, 5, 6, 5, 0, 5, 6, 1, 2, 3, 2, 1, 0, 1, 2, 1, 6, 5, 6, 5, 0, 2, 4, 2, 4, 5, 6, 4, 5, 0, 6, 5, 4, 2, 4, 2, 0, 2, 4, 2, 4, 5, 6, 4, 5, 0, 4, 4, 5, 0, 4, 4, 5, 0, 1, 1, 2, 3, 2, 1, 0, 0, 0, 3, 2, 0, 1, 6, 5, 1, 5, 0, 6, 1, 0, 2, 1, 0, 1, 1, 1, 5, 6, 2, 1, 3, 2, 6, 5, 2, 4, 6, 5, 0, 6, 5, 6, 1, 1, 2, 1, 4, 4, 6, 5, 2, 4, 6, 5, 6, 6, 0, 0, 5, 6, 1, 6, 5, 2, 4, 5, 1, 1, 2, 1
42	Ladrang Raraskaton	2, 1, 2, 6, 2, 1, 6, 5, 2, 1, 2, 6, 2, 1, 6, 5, 2, 1, 2, 6, 2, 1, 6, 5, 2, 3, 5, 3, 3, 2, 1, 2, 2, 3, 0, 0, 6, 5, 3, 2, 6, 5, 3, 5, 3, 2, 1, 2, 3, 2, 1, 6, 2, 1, 6, 5, 2, 1, 2, 6, 2, 1, 6, 5
43	Ladrang Rasamulya	0, 1, 1, 1, 2, 3, 2, 1, 0, 1, 1, 1, 2, 3, 2, 1, 5, 6, 1, 0, 2, 1, 6, 5, 0, 0, 5, 6, 1, 1, 2, 1, 5, 5, 0, 0, 5, 5, 6, 5, 7, 6, 5, 6, 5, 4, 1, 2, 0, 0, 2, 4, 5, 0, 6, 5, 6, 6, 5, 4, 2, 1, 2, 1
44	Ladrang Retna Kedhiri	0, 6, 1, 2, 1, 6, 4, 5, 0, 6, 1, 2, 1, 6, 4, 5, 3, 1, 3, 2, 3, 1, 3, 2, 5, 6, 1, 2, 1, 6, 4, 5, 6, 5, 4, 2, 1, 2, 4, 5, 6, 5, 4, 2, 1, 2, 4, 5, 6, 5, 4, 2, 1, 2, 3, 2, 6, 6, 0, 7, 6, 5, 4, 5, 7, 6, 5, 6, 5, 4, 2, 1, 0, 2, 4, 5, 4, 2, 4, 1, 0, 2, 4, 5, 4, 2, 4, 1, 5, 6, 1, 2, 1, 6, 4, 5
45	Ladrang Retnaningsih	0, 6, 1, 2, 1, 6, 4, 5, 0, 6, 1, 2, 1, 6, 4, 5, 1, 1, 0, 0, 5, 6, 1, 2, 1, 3, 1, 2, 0, 1, 6, 5, 0, 5, 5, 5, 6, 4, 6, 5, 0, 5, 5, 5, 6, 4, 6, 5, 6, 5, 4, 2, 1, 1, 2, 1, 5, 6, 1, 2, 0, 1, 6, 5
46	Ladrang Santi Mulya	6, 1, 6, 5, 6, 1, 6, 5, 2, 4, 5, 6, 5, 4, 2, 1, 6, 5, 6, 1, 6, 5, 6, 1, 2, 3, 2, 1, 2, 1, 6, 5, 2, 1, 6, 5, 2, 1, 6, 5, 0, 6, 3, 2, 1, 6, 3, 5, 0, 0, 5, 0, 5, 3, 2, 1, 2, 6, 2, 1, 3, 2, 6, 5, 6, 6, 0, 0, 4, 5, 6, 1, 2, 1, 6, 5, 4, 5, 6, 1, 3, 2, 1, 2, 1, 6, 4, 5, 0, 6, 1, 2, 1, 6, 3, 5
47	Ladrang Sarilaya	2, 2, 0, 0, 2, 2, 3, 2, 0, 0, 2, 3, 5, 6, 5, 3, 0, 3, 2, 1, 3, 2, 1, 2, 3, 3, 2, 1, 6, 5, 3, 5, 1, 1, 0, 0, 1, 1, 2, 1, 3, 2, 1, 2, 0, 1, 6, 5, 0, 0, 2, 3, 5, 6, 5, 6, 3, 5, 6, 5, 3, 2, 1, 2, 6, 6, 0, 0, 6, 6, 5, 6, 0, 0, 7, 6, 5, 3, 2, 3, 0, 3, 2, 1, 3, 2, 1, 2, 3, 3, 2, 1, 6, 5, 3, 5
48	Ladrang Sayang Gunung	0, 6, 1, 2, 0, 1, 6, 5, 0, 6, 1, 2, 0, 1, 6, 5, 1, 1, 0, 0, 5, 6, 1, 2, 3, 3, 1, 2, 0, 1, 6, 5, 3, 3, 0, 0, 2, 1, 2, 3, 6, 5, 3, 2, 3, 1, 2, 3, 6, 5, 3, 2, 3, 1, 2, 3, 1, 1, 3, 2, 0, 1, 6, 5, 3, 3, 0, 0, 2, 1, 2, 3, 6, 5, 3, 2, 3, 1, 2, 3, 1, 2, 3, 5, 0, 3, 2, 1, 3, 2, 1, 2, 0, 1, 6, 5, 1, 2, 1, 6, 5, 4, 2, 1, 3, 2, 1, 2, 0, 1, 2, 6, 0, 0, 6, 5, 6, 1, 2, 1, 0, 3, 0, 2, 0, 1, 6, 5
49	Ladrang Sembawa	0, 1, 1, 1, 2, 3, 2, 1, 0, 1, 1, 1, 2, 3, 5, 3, 0, 3, 5, 6, 7, 6, 5, 3, 5, 3, 2, 3, 2, 1, 2, 1, 5, 5, 0, 0, 5, 5, 3, 5, 0, 0, 5, 6, 7, 6, 5, 6, 0, 6, 5, 3, 6, 5, 3, 5, 7, 6, 2, 1, 2, 3, 5, 3, 0, 3, 3, 3, 2, 1, 2, 1, 0, 1, 1, 1, 2, 3, 5, 3, 0, 3, 5, 6, 7, 6, 5, 3, 5, 3, 2, 3, 2, 1, 2, 1
50	Ladrang Singa-Singa	0, 6, 5, 6, 1, 2, 1, 6, 0, 6, 5, 6, 1, 2, 1, 6, 0, 6, 5, 6, 1, 2, 1, 6, 5, 6, 5, 4, 2, 4, 5, 6, 0, 6, 6, 6, 5, 4, 2, 1, 0, 1, 2, 4, 5, 4, 2, 1, 0, 1, 2, 4, 5, 4, 2, 1, 3, 2, 1, 6, 2, 4, 2, 1, 0, 0, 1, 2, 3, 2, 1, 2, 0, 2, 1, 0, 1, 2, 1, 6, 0, 6, 5, 6, 1, 2, 1, 6, 5, 6, 5, 4, 2, 4, 5, 6