

PARALLEL PROCESS DISCOVERY USING A NEW TIME-BASED ALPHA++ MINER

YUTIKA AMELIA EFFENDI^{1*} AND RIYANARTO SARNO²

¹*Information Systems Study Program, Department of Engineering,
Universitas Airlangga, Surabaya, Indonesia*

²*Department of Informatics, Institut Teknologi Sepuluh Nopember,
Surabaya, Indonesia*

*Corresponding author: yutika.effendi@vokasi.unair.ac.id

(Received: 3rd June 2019; Accepted: 26th October 2019; Published on-line: 20th January 2020)

ABSTRACT: A lot of services in business processes lead information systems to build huge amounts of event logs that are difficult to observe. The event log will be analysed using a process discovery technique to mine the process model by implementing some well-known algorithms such as deterministic algorithms and heuristic algorithms. All of the algorithms have their own benefits and limitations in analysing and discovering the event log into process models. This research proposed a new Time-based Alpha++ Miner with an improvement of the Alpha++ Miner and Modified Time-based Alpha Miner algorithm. The proposed miner is able to consider noise traces, loop, and non-free choice when modelling a process model where both of original algorithms cannot override those issues. A new Time-based Alpha++ Miner utilizing Time Interval Pattern can mine the process model using new rules defined by the time interval pattern using a double-time stamp event log and define sequence and parallel (AND, OR, and XOR) relation. The original miners are only able to discover sequence and parallel (AND and XOR) relation. To know the differences between the original Alpha++ Miner and the new one including the process model and its relations, the evaluation using fitness and precision was done in this research. The results presented that the process model obtained by a new Time-based Alpha++ Miner was better than that of the original Alpha++ Miner algorithm in terms of parallel OR, handling noise, fitness value, and precision value.

ABSTRAK: Banyak sistem perniagaan perkhidmatan menghasilkan sejumlah besar log data maklumat yang payah dipantau. Log data ini akan dianalisis menggunakan teknik proses penemuan bagi memperoleh model proses dengan menerapkan beberapa algoritma terkenal, seperti algoritma deterministik dan algoritma heuristik. Semua algoritma ini memiliki kelebihan dan kekurangannya dalam menganalisis dan mencari log data ke dalam model proses. Kajian ini mencadangkan *Time-based Alpha++ Miner* baru yang merupakan pembaharuan dari algoritma *Alpha++ Miner* dan *Modified Time-based Alpha Miner*. Algoritma baru ini dapat mempertimbangkan kesan bunyi, pusingan, dan pilihan tidak bebas ketika memodelkan model proses di mana kedua algoritma asal tidak dapat menggantikan isu tersebut. *Time-based Alpha++ Miner* baru menggunakan pakai Pola Interval Waktu berjaya memperoleh model proses menggunakan peraturan baru berdasarkan Pola Interval Waktu menggunakan log peristiwa waktu-ganda dan menentukan jujukan dan hubungan selari (AND, OR, dan XOR). Dibandingkan algoritma asal, ia hanya dapat menemukan jujukan dan hubungan selari (AND dan XOR). Bagi membezakan Alpha++ Miner asal dan yang baru termasuk model proses dan kaitannya, penilaian menggunakan nilai padanan dan penelitian telah dijalankan dalam kajian ini. Hasil kajian model proses yang diperoleh oleh *Time-based Alpha++ Miner* baru, adalah lebih baik keputusannya berbanding menggunakan algoritma Alpha++

Miner asal, berdasarkan hubungan selari OR, bunyi kawalan, nilai padanan, dan nilai penelitian.

KEYWORDS: *Alpha++ miner; business process model; parallel process; process discovery; process mining; time interval pattern*

1. INTRODUCTION

Collections of activities that produce services for organizations or companies are called business processes [1]. A lot of services are available in the organization, which requires a means to handle all the services in an efficient way. Information systems were built to facilitate business processes and produce event logs as the documented business processes.

Due to the huge amount of event logs, they will be difficult to observe manually. Well-documented business processes are important because they act as navigator for organizations to operate and analyse their business processes. Obtaining the model of business processes from those event logs is becoming a concern in order to monitor and improve business processes easily and automatically. A process mining technique is a good choice to solve this issue [2].

Process mining consists of several tasks; one of them is process discovery. It is a method to automatically construct a current business process and record variations to the process that occurred within an organization. Because process discovery is a part of the process mining technique, the main goal becomes to discover the process models from an event log that describes the best behaviour of this business process implemented in an organization [3]. To carry out its duties properly, process discovery uses some well-known algorithms to help analyse the event log, such as the deterministic algorithm [4], consisting of Alpha Miner and its renewal versions; Alpha+ Miner, Alpha++ Miner, and Modified Time-based Alpha Miner [5], or the heuristic algorithm [4] i.e. Heuristics Miner and Fuzzy Miner. All of these algorithms have their own limitations and benefits in analysing the event log. In this research, the improvement of Alpha++ Miner and Modified Time-based Alpha Miner, which were the latest improvements on Alpha Miner related to parallel findings, are the main highlights.

This research proposes A New Time-based Alpha++ Miner algorithm to mine a process model along with its relations based on a Time Interval Pattern. The New Time-based Alpha++ Miner is an upgraded version of the Alpha++ Miner [6] and Modified Time-based Alpha Miner [5]. As a process discovery algorithm, the Alpha++ Miner algorithm can handle loop and non-free choice in a business process. This algorithm brings implicit dependencies out of a process model in a Petri Net form for showing non-free choice conditions. As an expansion of Alpha Miner, the steps to discover the model is exactly the same, with the Alpha Miner algorithm using a single-time stamp event log. However, the disadvantages of Alpha++ are that it cannot distinguish the differences between AND and OR in parallel relations and still use sequential relations between activities to obtain a business process. In other words, the Alpha++ algorithm can only determine whether the relations of business processes are sequential and parallel (AND, XOR) in a sequential way. On the other hand, a Modified Time-based Alpha Miner, usually known as a MTBAM algorithm, is the latest update from Alpha Miner which uses time-based information in discovering the process model using a double-time stamp event log. This algorithm can discover sequence and parallel AND, OR, XOR relation. However, this algorithm cannot mine length one loop (L1L), length two loop (L2L), and

non-free choice in the event log to be modelled in a business process. Loop and non-free choice are included in the issues of process discovery. Each issue of process discovery implies that the process discovery cannot present the overall information of an event log in a process model. The similarity between the two algorithms is that they do not consider the noise traces in the event log when obtaining a process model.

To cover the shortcomings of the Alpha++ Miner and MTBAM algorithm, an improvement of the Alpha++ Miner that utilizes Time Interval Pattern is proposed in order to be able to discover process models, including discovering loop, non-free choice, as well as their sequential and parallel relations (AND, OR, XOR) by determining new rules so that the business process cannot only be discovered in a sequential way. In addition, the new one can override noise traces when modelling a process model. To compare our method with the original Alpha++ Miner, fitness and precision will be presented as evaluation of the discovered process models in the experimental result.

The paper is arranged in structured way. Section II is for related work, while in Section III the authors will explain the proposed method; A New Time-based Alpha++ Miner with Time Interval Pattern. Discussion related to experimental results is put in Section IV and Section V will provide the conclusion to this paper.

2. LITERATURE REVIEW

2.1 Process Mining

Process mining is a study where data mining and machine learning are connected, then do analysis in business processes implemented in an organization. As a popular technique, process mining is always used to observe Standard Operating Procedure (SOP) based on event log run in the organization. The overview of process mining and its tasks is shown in Fig. 1. As shown in the figure, process mining consists of several tasks, namely *discovery*, *conformance*, and *enhancement* [4].

First task is called *process discovery*. It is a method to automatically construct a current business process and occurrences of variations to the process that happened in an organization. Because process discovery is a part of the process mining technique, discovering the process models from an event log that describes the best behaviour of the business process implemented in an organization becomes its main goal. A lot of *process discovery* techniques are suggested. Several of them are Alpha algorithm and Heuristics Miner algorithm. The goal of process discovery techniques is not only representing models that show the relations of activities. The other goal is discovering the social network between resources in business processes.

Conformance checking is the second task in process mining. It compares the behaviour captured in the process model and the behaviour captured in the event log. The value of conformance checking is higher if the process model captures more of the behaviour in the event log. It divides into four criteria: fitness, precision, simplicity, and generalization.

The last task of process mining is *enhancement*. An *enhancement* extends or improves the process model which contains the information of the event log. There are two types of *enhancement*. First one is repairing a process model and the second one is extending the process model which refers to the information described in event log, such as durations or allocations. Figure 2 shows the simple explanation of the three tasks in process mining.

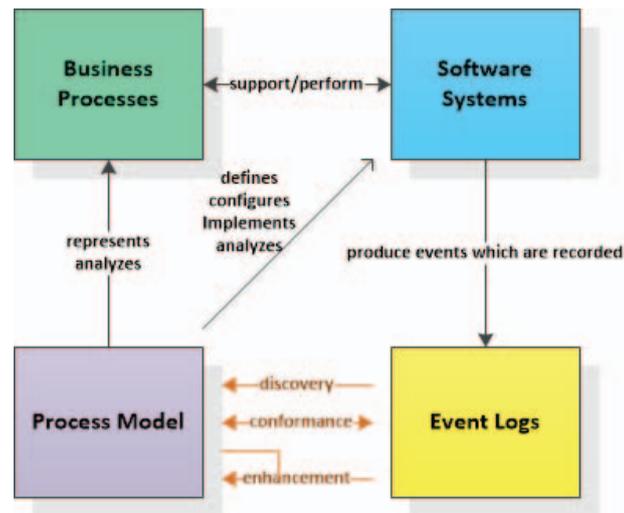


Fig. 1: Mind map of process mining.

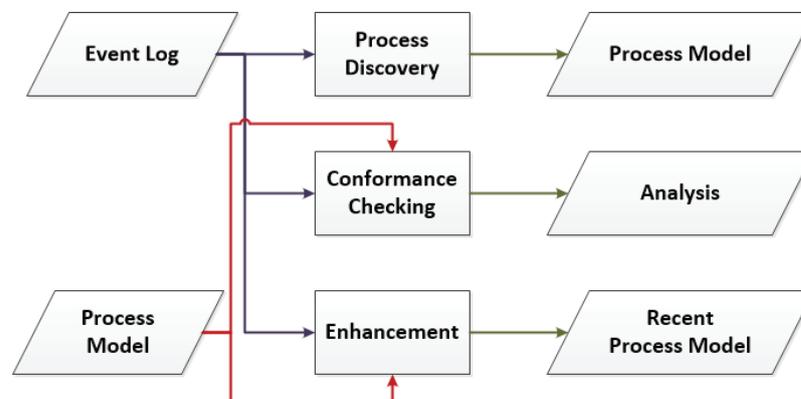


Fig. 2: The explanation of three tasks in process mining.

2.2 Event Log

In the process mining technique, the event log becomes the main input. All three tasks of process mining consider that information systems are capable of recording events. An event log contains several processes. Each process recorded in the event log is called a *case*. Usually, a *case* is symbolized by a number in the event log and consists of a series of ordered *events*.

The event log records several pieces of information about the event, such as *activity*, *time stamp*, and *resource*. *Activity* indicates the name of the ordered event. *Time stamp* indicates the time when the ordered event occurred. *Resource* indicates the person who is doing the event. Process mining might use those pieces of information to provide the best understanding of the real processes.

Table 1 shows a piece of the event log. This event log shows the selection acceptance of a journal. As shown, there are two types of processes. The first process is acceptance of a journal, which is displayed in *case 1*. The second process is rejection of a journal, which is displayed in *case 2*. In fact, not all event logs record only three types of information, like in Table 1. There are several event logs that record other information to describe the process clearly. Based on the time stamp, the event log can be divided into single-time stamp event log and double-time stamp event log.

Table 1: An example of event log

Case ID	Activity	Timestamp	Resource
1	Inviting Reviewers	3/8/2016 10:35	John
1	Getting Reviews from Judges	3/8/2016 10:38	Leo
1	Collecting Reviews	3/8/2016 10:41	Smith
1	Checking Reviews from Judges	3/8/2016 10:44	Carlote
1	Deciding Final Result of Journal based on Reviews	3/8/2016 10:47	Novi, Fiona, Helli
1	Determining The Final Result as Accepted Journal	3/8/2016 10:50	Novi, Fiona, Helli
1	Announcing the Final Result	3/8/2016 10:53	John
2	Inviting Reviewers	3/8/2016 10:56	John
2	Getting Reviews from Judges	3/8/2016 10:59	Dion
2	Collecting Reviews	3/8/2016 11:02	Smith
2	Checking Reviews from Judges	3/8/2016 11:05	Rayna
2	Deciding Final Result of Journal based on Reviews	3/8/2016 11:08	Novi, Fiona, Helli
2	Determining The Final Result as Rejected Journal	3/8/2016 11:11	Novi, Fiona, Helli
2	Announcing the Final Result	3/8/2016 11:14	John

2.2.1 Single-Time Stamp Event Log

Initially, the single-time stamp event log is a form of general event log. The characteristic of the single-time stamp is to only have one time stamp for each activity. The most common time stamp recorded in the event log is a finish time. Both Alpha algorithm [7] and Heuristics Miner algorithm [8] use single-time stamps to analyse the business processes. An example of the single-time stamp event log is shown in Table 1.

2.2.2 Double-Time Stamp Event Log

In some cases, using the single-time stamp event log makes the analysis of business processes difficult. It is because the single-time stamp only records the finish time of the activities, while sometimes the analysis needs to use both of the time stamps of activities to obtain a complete version of the process model. This matter is the reason that double-time stamp event logs appeared. Some algorithms use a double-time stamp event log to determine the model, such as Heuristics Miner with Interval Time [9] and Modified Time-based Alpha Miner [5]. The characteristic of the double-time stamp is to have a start time and a finish time. Because the two time stamps are recorded, they can help solving parallel processes in process mining. This research will use a double-time stamp event log to model the business process.

2.3 Existing Algorithm of Process Discovery

In process discovery, issues appeared during the implementation to discover the model. This becomes the main concern in process discovery. Many algorithms are proposed to handle the issues. The algorithms are divided into two principal categories, namely deterministic algorithms consisting of Alpha, Alpha+, Alpha++, and heuristic algorithms consisting of heuristics miner, heuristics miner with time interval, and fuzzy miner. All of these algorithms have their own limitations and excellences in analysing the event log to model the business processes. Therefore, process discovery in terms of modifying the algorithms becomes the most challenging task for researcher.

Deterministic algorithms define reproducible results [10]. A concept of deterministic algorithms is that behaviours of inputs become behaviours of results. It means that this algorithm delivers all behaviours of event log in the process model. Alpha algorithm is a

deterministic algorithm and it has several modifications, such as Alpha+ and Alpha++ algorithm.

2.3.1 Alpha Algorithm

Alpha algorithm is the first algorithm used in process discovery. It develops causality of activities based on the event log. The event log used in Alpha algorithm is a single time-stamped event log. The Alpha algorithm focuses on the workflow process and is displayed in the form of workflow-nets, which are also part of Petri Nets [7].

To build a process model using Alpha algorithm [4], the first step is to list all activities that are recorded in the event log. Next, a set of beginning activities and a set of last activities are created. A set of beginning activities contains activities that are recorded first for some cases. Meanwhile, a set of last activities consists of activities that occur as end activities of some cases. In the third step, several tuples are created. In the fourth step a set of tuples is obtained from the third step by taking the activities as well as the set inclusion [7]. The fifth step is listing places that connect activities in every tuple constructed, based on the fourth step. The last step is to connect places with their respective activities and a Petri Net is formed.

Alpha algorithm is a basic algorithm and easy to apply. However, Alpha algorithms have some disadvantages. Alpha algorithm cannot deal with short loop, long loop, invisible task, and non-free choice. Because Alpha algorithm cannot handle several issues, other deterministic algorithms appear to improve Alpha algorithm. Those algorithms are Alpha+ algorithm and Alpha++ algorithm.

2.3.2 Alpha+ Algorithm

Alpha+ algorithm is an advanced form of Alpha algorithm. Alpha+ focuses on dealing with short loop conditions [4,5,11]. Both Alpha and Alpha+ algorithms do not consider *noise* in the event log. This algorithm assumes that all information in the event log is correct in order to build an output model, usually presented in a Petri-Net form.

To obtain a Petri Net using Alpha+ algorithm, there are several steps that must be followed [11]. The first step is to list all of activities based on the event log. The second step is to list activities that have dependency relations with itself (have *one loop* condition). The third step is to create a set of activities that do not consist of *one loop* activities. The fourth step is to create a set of arcs that connect *one loop* activities. The fifth step is to discover a workflow net from a set of activities from the third step. The sixth step is combine the fifth step and the fourth step to build a Petri Net. It can be concluded that Alpha+ algorithm cannot deal with skip conditions. The other disadvantage of Alpha+ algorithm is that it cannot override noise traces when modelling a process.

2.3.3 Alpha++ Algorithm

Alpha++ algorithm is also an expansion of the Alpha algorithm. It offers a method for handling non-free choice. This algorithm also brings implicit dependencies out of the process model in a Petri Net form for showing non-free choice conditions. The Alpha++ algorithm states that this algorithm is able to discover explicit and implicit dependencies between tasks.

Alpha++ algorithm has several steps to mine non-free choice conditions [12]. An outline of the steps is determining possible implicit dependencies between activities and reducing the possible implicit dependencies by several steps. The obtained implicit dependencies will be added in a process model as the output of the Alpha++ algorithm. The Alpha++ algorithm has the same disadvantages as the Alpha+ algorithm. However,

the Alpha++ algorithm has an extra advantage. That is handling non-free choice conditions.

2.3.4 Modified Time-based Alpha Algorithm

Modified Time-based Alpha algorithm (MTBAM) is the latest update on the deterministic algorithm [5]. This algorithm uses time-based information from a double-time stamp event log to discover the model. Not only is it able to take advantage of the double-time stamp event log, but it is also able to define OR relationships in a process model. However, although this algorithm can overcome some of Alpha Miner's weaknesses, it still cannot discover loops, non-free choices, and it also cannot override noise traces when modelling a process.

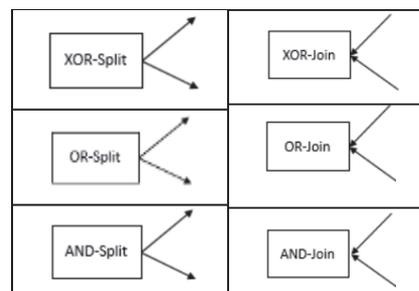
3. RESEARCH METHOD

This section presents the event log, temporal pattern and integrated discovery approach for discovering business processes in this research.

3.1 Differentiation of Parallel Relations of Activities

A process model has relations to connect all activities. Relations in the process model consist of a sequence and parallel relations. Sequence relation is a relation to connect one activity to another activity directly. Parallel relations are divided into AND, XOR, and OR. Each parallel relation will be presented in Petri Nets form as in Table 2.

Table 2: Parallel relations presented in Petri Nets



Almost all algorithms in process discovery can distinguish AND and XOR relations correctly, but not OR relations. OR relations have high possibilities of appearing in the process model, but a lot of algorithms have difficulty in classifying OR relations. Many algorithms will construe OR relations into XOR or AND whereas these two relations are very different from the function of the OR relation.

3.2 A New Time-based Alpha++ Miner with Time Interval Pattern

A new Time-based Alpha++ Miner algorithm will be introduced in this section. This algorithm is divided into three steps: defining the double-time stamp event log, determining the sequence relation, classifying the parallel relation, defining the loop condition and non-free choice, and the last, establishing the process model.

3.2.1 Defining the Double-Time Stamp Event Log

The Double-time stamp event log will be used as an input in this research. However, if there is only a single-time stamp provided in an organization, using the technique explained in [5], a single-time stamp event log can be converted into a double-time stamp event log. After it is obtained, the next step is to determine the relations. In this step, we will pay attention to noise traces in the event log. The event log will be processed without

noise, with 10% noise, with 30% noise, and with 50% noise. We separate the discovery process based on noise traces because they will affect the obtained process model as well as the fitness and precision values.

3.2.2 Determining the Relations

As explained in Section 3.1, relations in process model consist of sequence and parallel AND, OR, XOR relations. However, besides those relations, there are conditions wherein the activities proceed to themselves again or return to the previous activity rather than advance to the next activity. This is known as a loop condition [13].

Other than that, non-free choice is also an issue to be solved. The inability to identify a non-free choice process is one of the issues in discovering process models. This is because several processes implement non-free choice. For example, consider the selection of provinces and cities in software systems. Usually, the choice of cities appears appropriate to the selected province. The city cannot be chosen if it is not a part of the selected province. It means that the selection of provinces and cities is a non-free choice process.

This research proposes definitions to determine the sequence and parallel relations from an event log considering noise traces by utilizing a time interval pattern. The definitions in the proposed algorithm are:

Definition 3.1 There are *event log* (EL) and *trace* (σ) wherein $\sigma \in EL$. The types of time interval pattern between activity X and activity Y ; $X(X_s, X_f)$ and $Y(Y_s, Y_f)$, according to which $X, Y \in EL$ can be classified into *before*, *meet*, *is-started-by*, *is-completed-with*, *overlap*, *contain*, and *equal*.

Definition 3.2 Based on *event log* (EL), *trace* (σ) and activities $X, Y \in EL$, the relations of process models both sequential and parallel can be distinguished as follows:

- NotRelated*, $X \# Y$ iff $XOY \wedge YOX$
- Sequence*, $X \rightarrow Y$ iff $X > Y \wedge YOX$
- Parallel*, $X \parallel Y$ iff $X > Y \wedge Y > X \vee \{XOY \vee X @ Y \vee X_f Y_s \vee X \approx Y \vee X_p Y\}$
- AND*, $X \bullet Y$ iff $X \parallel Y$ and there is no $X \otimes Y$ in σ in EL
- OR*, $X \oplus Y$ iff $X \parallel Y$ and there is $X \otimes Y$ in σ in EL
- XOR*, $X \otimes Y$ iff $X \parallel Y$ if there is only X or Y in σ in EL

3.2.3 Establishing a Process Model

This research proposed how to discover process models using a new Time-based $\alpha++$ Miner algorithm. There are 10 steps in this proposed algorithm to obtain the process model.

- Step I.** A double-time stamp event log is an input
- Step II.** Define Place (before and after), Start Activities, End Activities, Loop, Non-free Choice, Relations, and Rule Numbers in Time Interval Pattern

II.1 Steps to determine the loop:

$$LL = \{t \in T_L \mid \exists \sigma = t1t2 \dots tn \in W; i \in \{1, 2, \dots, n\} [t = ti - 1 \wedge t = ti]\} \quad (1)$$

where:

LL : loop

t : activity

T_L : transition between activities

W : workflow of process model

II.2 Steps to set the rules in the time interval pattern

The model of time interval pattern is based on a double-time stamp to mine the process model. [14] introduced the use of time-based process mining algorithms. In her research, the definition of a time-based process mining algorithm consists of *before and meets, overlaps, contains, is-finished-by, equals, and starts*. In this research, we specify and categorize an improvement of time interval pattern between activities in business processes. We divide the sequence relations into *before* and *meet*, whereas parallel relations become *is-started-by, is-completed-with, overlap, contain* and *equal*. Table 3 presents the rules and the relations proposed in this research.

Table 3: Rules of Determining Activity Relations by Algorithm with HMM

Number	Rules	Relations
1	<i>before</i> : wherein $X > Y$ iff $X_f \leq Y_s$	Sequence
2	<i>meet</i> : wherein $X > Y$ iff $X_f \leq Y_s$	Sequence
3	<i>is-started-by</i> : wherein $X_f Y$ iff $X_f = Y_f \wedge X_s < Y_s \wedge Y_s < X_f$	Parallel
4	<i>is-completed-with</i> : wherein $X \rho Y$ iff $X_s = Y_s \wedge X_f = Y_f$	Parallel
5	<i>overlap</i> : wherein $X O Y$ iff $X_f > Y_s \wedge X_f < Y_f$	Parallel
6	<i>contain</i> : wherein $X @ Y$ iff $X_s < Y_s \wedge Y_f > X_s \wedge X_f > Y_f$	Parallel
7	<i>equal</i> : wherein $X \approx Y$ iff $X_s = Y_s \wedge X_f = Y_f$	Parallel

II.3 Steps to specify the parallel relations:

XOR relation

$$\text{If Avg PPM} \leq \text{Min ASR in EL, then XOR} \quad (2)$$

OR relation

$$\text{If Min ASR} \leq \text{Avg PPM} \leq \text{Avg ASR in EL, then OR} \quad (3)$$

AND relation

$$\text{If Avg ASR} \leq \text{Avg PPM in EL, then AND} \quad (4)$$

where:

Min ASR : minimum value of all sequential relations in EL

Avg ASR : average of all sequential relations in EL

Avg PPM : average of parallel relations with the same parent activity in process model (the frequency of each activity is both directly and indirectly followed by another activity)

Step III. Generate process models by following these steps:

III.1 Generate set of transition $T_L = \{t \in T \mid \exists_{\sigma \in L} t \in \sigma\}$ (5)

III.2 Generate set of input $T_I = \{t \in T \mid \exists_{\sigma \in L} t = first(\sigma)\}$ (6)

III.3 Generate set of output $T_o = \{t \in T \mid \exists_{\sigma \in L} t = last(\sigma)\}$ (7)

III.4 Generate the places $P_L = \{p_{(A,B)} \mid (A,B) \in Y_L\} \cup \{I_L, O_L\}$ (8)

Step IV. Display the final result of business process in Petri Net [15]

$$\alpha(L) = (P_L, T_L, F_L) \quad (9)$$

where:

α : A business process in Petri Net form

P : place

In the business process, it is symbolized by circle shape and named as p1, p2, p3, etc.

T : transition

In the business process, it shows the activities with square shape

F : function

In the business process, function means an arc.

$F = \{P \times T\}$ means the arc connects the place (P) and transition (T)

Step V. Evaluation of the process model using Fitness and Precision [16,17].

4. RESULTS AND DISCUSSION

The experimental results will present the process model to prove that our new approach can mine business processes using a new Time-based Alpha++ Miner algorithm utilizing a time interval pattern.

4.1 Case Study and Event Log

a) Case Study

The experiment data are business processes in a textile industry where the event log is generated from PT. XYZ, a yarn production company in Jakarta, Indonesia. The business processes of PT. XYZ consist of 11 activities as described in Table 4. The pieces of the event log from PT. XYZ are shown in Table 5. The process model will be presented in a Petri Net form [15,18].

b) Data Set

The case study uses a double-time stamp event log that has information including case id, activities, start time, and finish time.

This event log contains 50 cases in four conditions. These conditions are event log without noise, event log with 10% noise, event log with 30% noise, and event log with 50% noise. The event log after being processed using the proposed method explained in Section 3 in this experiment is presented in Table 6.

Table 4: Lists of Activities

Activities	Activity code used in this experiment
<i>Getting-good-receive</i>	A
<i>Opening-blending-bale</i>	B
<i>Opposing-spike</i>	C
<i>Blowing-air</i>	D
<i>Striking-cotton</i>	E
<i>Carding</i>	F
<i>Framing-drawing</i>	G
<i>Framing-roving</i>	H
<i>Combing</i>	I
<i>Framing-ring</i>	J
<i>Winding-cone</i>	K

Table 5: Part of Event Log

PP1	A	6/20/2014 10:32	6/20/2014 13:42	PP48	A	9/1/2014 0:15	9/1/2014 16:30
PP1	B	6/20/2014 13:42	6/20/2014 16:52	PP48	B	9/1/2014 16:30	9/2/2014 8:45
PP1	C	6/20/2014 16:52	6/20/2014 20:02	PP48	D	9/2/2014 8:45	9/3/2014 1:00
PP1	D	6/20/2014 17:49	6/20/2014 20:59	PP48	C	9/2/2014 16:52	9/4/2014 1:22
PP1	E	6/20/2014 20:59	6/21/2014 0:09	PP48	E	9/4/2014 1:22	9/4/2014 17:37
PP1	F	6/21/2014 0:09	6/21/2014 3:19	PP48	F	9/4/2014 17:37	9/5/2014 9:52
PP1	G	6/21/2014 3:19	6/21/2014 6:29	PP48	H	9/5/2014 9:52	9/6/2014 2:07
PP1	H	6/21/2014 5:13	6/21/2014 8:23	PP48	G	9/5/2014 18:00	9/7/2014 2:30
PP1	I	6/21/2014 8:23	6/21/2014 11:33	PP48	I	9/7/2014 2:30	9/8/2014 11:00
PP1	J	6/21/2014 11:33	6/21/2014 14:43	PP48	J	9/8/2014 11:00	9/9/2014 19:30
PP1	K	6/21/2014 14:43	6/21/2014 17:53	PP48	K	9/9/2014 19:30	9/11/2014 4:00

Table 6: Event log after being processed using proposed method

Place (Before)	Place (After)	Start Activities	Finish Activities	Relation	Rule Number
PP1	PP2	A	B	Sequence	2
PP2	PP3	B	C,D	Parallel	5
PP3	PP4	C,D	E	Parallel	5
PP4	PP5	E	F	Sequence	1
PP5	PP6	F	G,H	Parallel	6
PP6	PP7	G,H	I	Parallel	7
PP7	PP8	I	J	Sequence	1
PP8	PP8	J	J	Sequence	2
PP8	PP9	J	K	Sequence	1
PP9	PP10	K	<i>e</i>	-	1

4.2 Experimental Results

Based on the event log in Table 5, there are sequence and parallel activities in the business processes of PT. XYZ. All activities are in sequence relations except for: *opposing-spike*, *blowing-air*, *framing-roving*, and *framing-drawing*. Activities of *Opposing-spike* and *Blowing-air* are parallel following rule number 5 according to section

3, meanwhile activities *Framing-drawing* and *Framing-roving* follow rule number 6 (contain) and 7 (equal) respectively. Therefore, four of them are parallel relations.

By implementing a new Time-based Alpha++ algorithm for parallel processes utilizing a Time Interval Pattern as explained in Section 3 and by following all steps, the final process model is described in Fig. 3. The type of parallel relations in the process model after (2), (3), and (4) are applied, are also shown in Table 7.

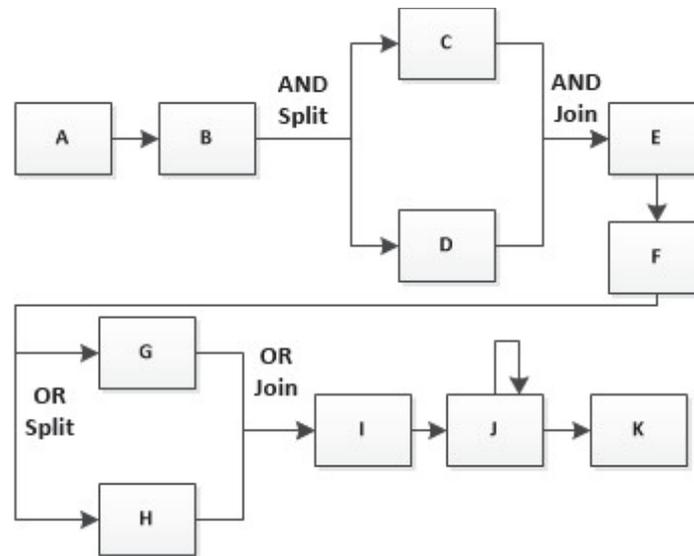


Fig. 3: Process model obtained by using proposed algorithm.

Table 7: Relations of process model generated by proposed algorithm

Place (Before)	Place (After)	Start Activities	Finish Activities	Relation
PP1	PP2	A	B	Sequence
PP2	PP3	B	C,D	Parallel AND
PP3	PP4	C,D	E	Parallel AND
PP4	PP5	E	F	Sequence
PP5	PP6	F	G,H	Parallel OR
PP6	PP7	G,H	I	Parallel OR
PP7	PP8	I	J	Sequence
PP8	PP8	J	J	Sequence
PP8	PP9	J	K	Sequence
PP9	PP10	K	e	-

The evaluation by the proposed algorithm was also tried out in event logs that contain 10% noise, 30% noise, and 50% noise. The percentage of noise shows the total of cases that are noise in the event log. For example, 10% noise means that 10% of all cases restored in the event log are noise. Based on the experiments, the process models from those event logs are the same as the process model displayed in Fig. 3.

This research also presents the results of the same case study and event log processed by the original Alpha++ algorithm in order to compare and prove that our proposed algorithm can discover the process model better than that of the original Alpha++ Miner. The process model of the Alpha++ Miner will be obtained using ProM [19,20], a process mining tool to accelerate the process discovery. Fig.4 shows the resulting process model and Table 8 presents the result of its relations.

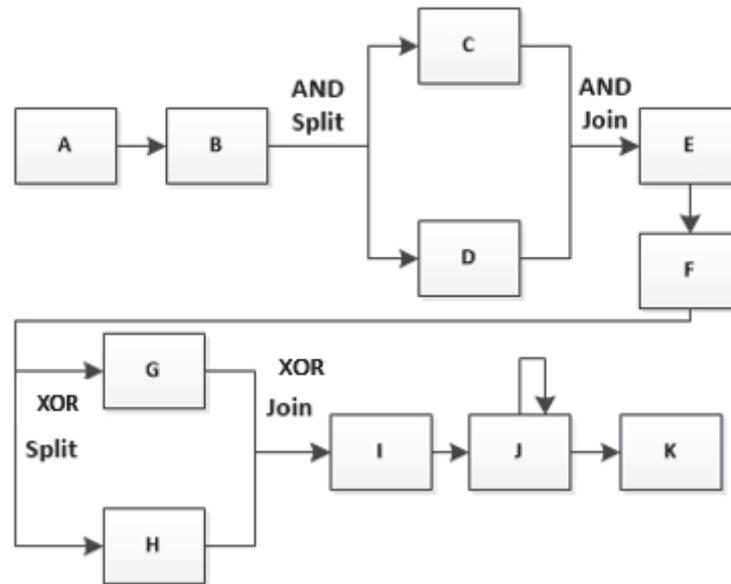


Fig. 4: Process model obtained by applying Alpha++ Miner.

Table 8: Relations of process model generated by original algorithm

Start Activities	Finish Activities	Relation
A	B	Sequence
B	C,D	Parallel AND
C,D	E	Parallel AND
E	F	Sequence
F	G,H	Parallel XOR
G,H	I	Parallel XOR
I	J	Sequence
J	J	Sequence
J	K	Sequence
K	<i>e</i>	-

The results presented in Fig. 4 and Table 8 show that Alpha++ Miner cannot define an OR relation in a process model and considers it as an XOR relation. The reason for this is that the original Alpha++ Miner has difficulties in interpreting the OR relation and is only able to define either AND or XOR relations, so it automatically construes OR relations into XOR or AND if there is an OR relation in the process model, whereas these two relations are very different from the function of OR relation.

4.3 Evaluation of the Business Processes

The quality of an algorithm can be measured from the results of that algorithm. In the data mining context, especially clustering and classification, the proposed algorithm is measured based on the obtained classes of respective data. The accuracy, recall, and precision are mostly used criteria to measure the quality of data mining algorithms.

Process mining also has several criteria to measure the quality of algorithms [4,16]. There are fitness, precision, generalization, and simplicity. Our proposed algorithm for parallel processes utilizing time interval pattern is evaluated by measuring fitness and precision value.

The definition of fitness is the total number of cases in the event log displayed in the process model. Fitness is measured using the formula explained in [15]. [15] reviewed that there are two ways to calculate the fitness of an obtained process model. The first way measures the fitness by calculating the percentage of activities that can be generated in the process model. The second way is by calculating the percentage of traces that can be represented in the process model. This evaluation uses the second way to calculate fitness. The equation of the second way is shown in (10).

$$\text{Fitness} = \frac{n_{ct}}{n_t} \quad (10)$$

Table 9: The details of fitness and precision of the proposed algorithm

Process Model of Event Log	Fitness			Precision		
	n_{ct}	n_t	Result	n_{ctm}	n_{tm}	Result
Event Log without noise	50	50	1	16	16	1
Event Log with 10% noise	45	50	0,9	16	16	1
Event Log with 30% noise	35	50	0,7	16	16	1
Event Log with 50% noise	25	50	0,5	16	16	1

The precision measures how many traces in process model occur in the event log. The precision can be calculated from the total of correct traces divided by the total of all traces that are generated from process model. The correct traces are combination traces that are generated from the process model and occur in the event log. The equation of precision is shown in (11).

$$\text{Precision} = \frac{n_{ctm}}{n_{tm}} \quad (11)$$

This evaluation measures the quality of the proposed algorithm based on the process models, as shown in Fig. 3. The calculation details of fitness using (10) and precision by applied (11) of the proposed algorithm are presented in Table 9. The final fitness and precision of the proposed algorithm as well as the original algorithm for parallel processes are shown in Table 10.

Based on Table 10, the proposed algorithm for parallel processes has relatively high fitness and high precision to produce a process model from the event log without noise or with noise of 10%, 30%, and 50% compared with the original Alpha++ Miner itself.

Table 10: The Comparison of fitness and precision from both proposed and original algorithm

Algorithm	Fitness	Precision
A New Time-based Alpha++ Miner	0.7	1
Original Alpha++ Miner	0.6	0.83

5. CONCLUSIONS

The conclusions for this paper are as follows:

- This research paper focused on introducing a new approach to mine parallel business processes using a time interval pattern that was implemented in Alpha++ Miner algorithm. The proposed method defined seven types of time interval patterns which included both sequence and parallel relations and grouped them into rules.
- A New Time-based Alpha++ algorithm was an upgraded version of Alpha++ Miner algorithm. Compared to original Alpha++ Miner algorithm, the improved algorithm could detect sequence and parallel time stamp using time interval pattern and identifies them as sequence and parallel processes, discovered loop, determined non-free choice, and also distinguished AND, OR, XOR relations in a correct way. Meanwhile, Alpha++ Miner algorithm was not able to define the differences between OR and XOR in parallel relations according to the results of the experiment.
- The experimental results proved that our new process discovery approach could mine the business processes with parallel AND and OR relations, which could not be obtained by the original Alpha++ Miner algorithm. In addition, the results of fitness and precision also clearly stated that a new Time-based Alpha++ also gave better results rather than that of the original Alpha++ Miner algorithm.

REFERENCES

- [1] Sienou A, Karduck AP, Lamine E, Pingaud H. (2018) Business Process and Risk Models Enrichment: Considerations for Business Intelligence. IEEE International Conference on e-Business Engineering. DOI: 10.1109/ICEBE.2008.123
- [2] Peña MR, Bayona-Oré S. (2018) Process Mining and Automatic Process Discovery. 7th International Conference On Software Process Improvement (CIMPS). DOI: 10.1109/CIMPS.2018.8625621
- [3] Saylam R, Sahingoz OK. (2013) Process mining in business process management: Concepts and challenges. International Conference on Electronics, Computer and Computation (ICECCO). DOI: 10.1109/ICECCO.2013.6718246
- [4] Van der Aalst WMP. (2011) Process mining: discovery, conformance and enhancement of business processes. Springer Science and Business Media. <http://dx.doi.org/10.1007/978-3-642-19345-3>
- [5] Effendi YA, Sarno R. (2018) Modeling Parallel Business Process Using Modified Time-based Alpha Miner. International Journal of Innovative Computing, Information and Control, 14(5). DOI: 10.24507/ijicic.14.05.1565
- [6] Wen L, Van der Aalst WMP, Wang J, Sun J. (2007) Mining Process Models with Non-Free-Choice Constructs. Data Mining and Knowledge Discovery, 15(2):145-180.
- [7] Van der Aalst WMP, Weijters T, Maruster L. (2004) Workflow mining: Discovering process models from event logs. IEEE Transactions on Knowledge and Data Engineering, 16(9):1128-1142.
- [8] Weijters AJMM, Van der Aalst WMP, Alves de Medeiros AK. (2006) Process Mining with the Heuristic-miner algorithm. Technische Universiteit Eindhoven, Tech. Rep. WP, 166:1-34.
- [9] Burattin A, Sperduti A. (2010) Heuristics Miner for Time Intervals. European Symposium on Artificial Neural Networks - Computational Intelligence and Machine Learning, d-side publi., ISBN 2-930307-10-2.
- [10] Gehrke N, Werner M. (2013) Process mining. Das Wirtschaftswachstum, 42(7):934-943.

-
- [11] Alves de Medeiros AK. (2004) Process mining: Extending the α -algorithm to mine short loops. Eindhoven, Netherland: BETA working paper series, WP 113, Eindhoven University of Technology, Eindhoven.
- [12] Wen L, Van der Aalst WMP, Wang J, Sun J. (2007) Mining Process Models with Non-Free-Choice Constructs. *Data Mining and Knowledge Discovery*, 15(2):145-180.
- [13] Sarno R, Effendi YA, Haryadita F. (2016) Modified Time-Based Heuristics Miner for Parallel Business Processes. *International Review on Computers and Software (IRECOS)*, 11 (3):249-260. <http://doi.org/10.15866/irecos.v11i3.8717>
- [14] Sutrisnowati RA, Bae H, Dongha L, Minsoo K. (2014) Process Model Discovery based on Activity Lifespan. *International Conference on Technology Innovation and Industrial Management*, pp.137-156. <http://dx.doi.org/10.1016/j.eswa.2014.05.055>
- [15] Effendi YA, Sarno R. (2018) Implementation of the Semantic Web in Business Process Modeling Using Petri Nets,” *International Conference on Information and Communications Technology (ICOIACT)*. DOI: 10.1109/ICOIACT.2018.8350724
- [16] Effendi YA, Sarno R. (2018) Conformance Checking Evaluation of Process Discovery Using Modified Alpha++ Miner Algorithm. *International Seminar on Application for Technology of Information and Communication*, pp. 435 – 440. DOI: 10.1109/ISEMANTIC.2018.8549770.
- [17] Burattin A, Maggi FM, Sperduti A. (2016) Conformance checking based on multi-perspective declarative process models. *Expert Systems with Applications*, 65:194-211. DOI: 10.1016/j.eswa.2016.08.040
- [18] Ishihara K, Hiraishi K. (2001) The completeness of linear logic for Petri net models. *Logic Journal of the IGPL*, 9(4):549 - 567. DOI: 10.1093/jigpal/9.4.549
- [19] De Cnudde S, Claes J, Poels G. (2014) Improving the Quality of the Heuristics Miner in Prom 6.2. *Expert System with Application*, 41:7678-7690.
- [20] Van Dongen BF, Verbeek HMW, Van der Aalst WMP, De Medeiros AKA, Weijters AJMM. (2005) The ProM Framework: A New Era in Process Mining Tool Support. *26th International Conference Applications and Theory of Petri Nets (ICATPN)*. DOI: 10.1007/11494744_25