Identifying Bottlenecks and Fraud of Business Process using Alpha ++ and Heuristic Miner Algorithms (Case study: CV. Wicaksana Artha)

Yurika Caesarita
Information Technology Management
MMT-ITS
yurika16mhs.mmt.its.ac.id

Riyanarto Sarno
Department of Informatics
Institut Teknologi Sepuluh Nopember
riyanarto@if.its.ac.id

Kelly Rossa Sungkono
Department of Informatics
Institut Teknologi Sepuluh Nopember
kelsungkono@gmail.com

Abstract—Every organization or company has a business process to support the structured running tasks. Each of the business processes has a role called resource that is useful for running the processes. However, a few organizations and companies evaluate their business processes. Process mining is a way to evaluate event log data by forming a workflow. Petri net is chosen workflow to analyze the occurrence of bottlenecks. A bottleneck is an event that has a longer waiting time than the other business processes. PROM is used to analyze the business process in mxml formal by using process mining algorithm, such as alpha++ and heuristic miner algorithms. Both algorithms are used to determine the difference of occurring bottlenecks. This research shows that the algorithm affects the location of the occurrence of bottleneck. The location of the bottleneck is seen based on the calculation of the token time in the place (place between 2 transitions or transactions). Heuristic miner can model a business process appropriately and it has a higher performance than alpha++ algorithm because it can calculate the frequency, whereas alpha++ algorithm can not generate the frequency calculation. The heuristic miner algorithm can also check the fraud by utilizing its fitness value. The obtained fitness value will indicate whether there are indications of fraud or no indication in every business process.

Keywords—bottleneck; event log; fraud; heuristic miner; process mining

I. INTRODUCTION

Each organization or company has a business process to support running tasks. Each of the business process has roles called resources that are useful to execute the process. However, only a few of the resources who evaluate their business processes.

The process mining aims to extract the business process from an event log and form the log into a workflow. The event log is a historical record as the evidence of ongoing transaction processing. An event log informs the events in the occurring cases [1]. Those events are processed by process mining algorithm for addressing several issues [2]–[9].

By utilizing the event logs that are generated in an information system, the actual business model will be easier to obtain. Evaluations can run smoothly and more efficiently. The application to perform the process mining is Prom. Prom can also be defined as an expandable framework to support multiple plug-in techniques.

Prom is an open source framework for process mining algorithms. Prom provides a platform to user and process algorithm developers with the goal of actively advancing cutting-edge process mining technology by developing methods that actually work by creating an open community by providing a stable and easily expandable platform that optimally supports mining processes. Some algorithms that are represented in Prom are alpha++ [10] and heuristic miner [11] algorithms.

The alpha++ algorithm [10] is a basic algorithm of the process mining that can be used to analyze and generate the activity patterns of a case. The advantages of alpha++ algorithm is easy to understand for the user. Otherwise, alpha ++ algorithm has an inability to calculate the frequency. Heuristic miner can calculate the existence of frequency and the existence of dependency measure that is an indication strength of causal relation between pair of activities. Heuristic miner algorithm provides a decision restriction.

II. LITERATURE REVIEW

A. Event Logs

An event log are a historical record on an information tool. In an information system, an event log is used as the evidence of an ongoing transaction. The event log record activity sequentially and in parallel, i.e. the tasks that are done together with different people. Each event log will be recorded at a different level. Each abstraction contains the occurrence of some process that is affected by noise.

An event log provides an information about a process. An event log can describe the events of occurring cases [1]. By utilizing the event logs that are generated in an information system, the forming of an actual business model is easier.

978-1-5386-2827-0/17/$31.00 ©2017 IEEE

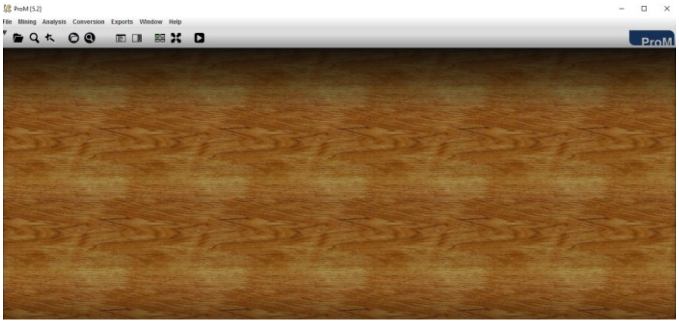


Fig. 1. Interface of Prom

B. Petri Net

Petri net is often used in the representation of common processes that are utilized by process mining algorithms. Petri net allows for explicit concurrency. Using the Petri net, users will be helped in analyzing processes. Petri net is a directed graph that has 2 types of nodes, i.e. place and transition. Places and transitions are connected in a bow. In addition, (WF-net) is a specialty of Petri net. WF-net is proposed to capture the description of the business process control flow. WF-net is considered as a model of formal business process tasks. Petri net is defined as follows:

Petri net is a pair of 4 tuples (P, T, A, W) where:

- P is a set consisting of places, P= {p1,p2,…pn}
- T is a set consisting of transitions, T={tl,t2,…tn}
- A is a consisting of arcs, A (P x T) u (T x P)
- Wis a function of weight, W: A {1,2,3}

C. ProM

Prom is a tool to perform a process mining. Prom can be defined as an expandable framework to support multiple plug-in techniques. This tool is an open source framework for process mining algorithms. It also provides a platform to users and process algorithm developers who have the goal of actively advancing advanced process mining technology. These technological advances can be made by developing methods that actually work in a stable and easy-to-develop platform that supports optimal process mining. Prom also offers a variety of plugins that allow to apply developments in process mining . Fig. 1 shows the interface of ProM.

D. Business Process

Business Process is a set of activities that is designed to accomplish an the strategic goals of organizations [12]. The business process is a work sequence that relates one task to another for solving a particular problem or achieving its goal. A business process can be broken down into several sub processes.

A business process can be broken down into sub processes that have their own attributes. Business processes can be defined as a sequential work, and the processes use company resources to produce a product [13]. The illustration of the actual business process can be derived from the form of event log history [14].

E. Process mining

The process mining is the result of business process extraction from the log files of each company's business processes with the aim to better understand and optimize each activity. The process mining is well suited for organizations or companies that use the development process and is useful for evaluating new developments. From the obtained data, it will find a pattern to provide insight into the behavior of the process. The business process can be considered from various perspectives.

The process mining combines the technical knowledge of computational intelligence, data mining and process analysis. In a process mining, events will be recorded whatever is done and when or who that executes the events [15]. Process mining is used for several kinds of systems, such as system generates event log (recording of various actual behavior). The purpose of the process mining is to extract the information from the business process by utilizing the information recorded in the event log.

The process mining is as a tool that helps the owner of a system in analyzing the things that really happen in the business process. The process mining seeks to provide and help build a realistic vision of the operational process. This process mining can help, monitor and improve the real process by extracting knowledge of the event log [16]. It also helps users to know the business process quickly and objectively in analyzing data that includes event log.

F. Alpha++ Algorithms

The alpha++ algorithm is the basic algorithm of the process mining which can be used to analyze and generate the activity patterns of a case. The advantages of alpha++ algorithm is easy to understand for the user, but alpha++ algorithm has an inability to calculate the frequency. Alpha++ algorithm is suitable for the business process that is not too much selection procedure. Because of that, alpha++ is suggested to overcome business process that has "short loops". However, alpha++ algorithm has a disadvantage, i.e. this algorithm can not know any noise happened in the processes.

The alpha++ algorithm is a deficiency improvement in the alpha algorithm. The alpha++ algorithm is able to overcome several issues, icluding length-one-loop, length-two-loop, invisible task, duplicate task, implicit places, and non-free choice [9].

G. Heuristic miner

Heuristic miner is designed to handle the noise in event logs that are not owned by alpha++ algorithm. Heuristic miner can calculate the existence of frequency to existence of dependency measure that is an indication strength of causal relation between pair of activities [10]. This algorithm provides a decision restriction.

Heuristic Miner is the second algorithm of a process mining that closely follows the alpha algorithm. This algorithm was developed by Dr. Ton Weitjers, who uses the heuristic approach to address many problems and making this algorithm becomes more suitable in practice.

TABLE I. CASE ACTIVITY

CaseID	ActivityName	StartTime	Resource
1	Receive order	28/04/2017 07:05:00 AM	Erick
1	Check completeness	28/04/2017 07:15:00 AM	Erick
1	Check fix status	28/04/2017 07:20:00 AM	erick
1	Matching prices	28/04/2017 07:30:00 AM	Erick
1	Choose supplier	28/04/2017 07:35:00 AM	Yurika
1	Call factory	28/04/2017 07:40:00 AM	Yurika
1	Check availability	28/04/2017 07:45:00 AM	Yurika
1	Large order	28/04/2017 07:50:00 AM	Yurika
1	Make decision	28/04/2017 08:00:00 AM	Yurika
1	Fast process	28/04/2017 09:05:00 AM	Erick
1	Admin process	28/04/2017 09:10:00 AM	Yurika
1	Payment	28/04/2017 09:20:00 AM	Bigane
1	Delivery	28/04/2017 09:25:00 AM	Ceto
1	Send customer	28/04/2017 09:35:00 AM	Ceto
2	Receive order	29/04/2017 07:05:00 AM	Erick
2	Check completeness	29/04/2017 07:10:00 AM	Erick
2	Check fix status	29/04/2017 07:20:00 AM	Erick
2	Matching prices	29/04/2017 07:30:00 AM	Erick
2	Choose supplier	29/04/2017 07:35:00 AM	Erick
2	Call factory	29/04/2017 07:40:00 AM	Yurika
2	Check availability	29/04/2017 07:45:00 AM	Yurika
2	Small order	29/04/2017 07:50:00 AM	Yurika
2	Make decision	29/04/2017 07:55:00 AM	Yurika
2	Slow process	29/04/2017 08:00:00 AM	Erick
2	Queue deliver	29/04/2017 08:15:00 AM	Yurika
2	Admin process	29/04/2017 08:20:00 AM	Yurika
2	Payment	29/04/2017 08:25:00 AM	Bigane
2	Delivery	29/04/2017 08:30:00 AM	Ceto
2	Send customer	29/04/2017 08:35:00 AM	Ceto

H. YAWL

Yawl is a workflow modeling language and also an open-source workflow system [17]. The acronym yawl stands for “Yet Another Workflow Language”. The development of the yawl languages was heavily influenced by the workflow patterns initiative [18] that is mentioned earlier.

Based on systemic analysis of the constructs that is used by existing process modeling notations and workflow languages, a large collection of patterns was identified. These patterns cover all workflow perspectives. The patterns are control flow patterns, data patterns, resource patterns, change pattern, exception patterns, etc.

III. PROPOSED METHOD

A. Data Source

Data that is generated by this research is the transaction data ordering frozen chicken. The transaction is done from the time when customers were offering order until the product accepted by customer. Table I describes the result of extraction into the mxml format of some data containing:

1) *Case_id : a trial of a process or event that is done as many as n.*
2) *Activity_name: an activity that has been done.*
3) *Timestamp: the data and time of an activity performed.*

B. Data Cleaning

This section describes about data cleaning that is removing the existing noise. There are various ways to get rid of the noise that sort from the extraction of event log events.

In this study, the business process is created by using YAWL application. The extracted processes of YAWL are in .xes format, so the processes need re-extraction into .mxml format before those are analyzed by ProM. The processes are analyzed on ProM application using alpha++ and heuristic miner algorithms.

C. Models with alpha ++ and heuristic miner algorithms

After getting the result of clean-noise data, a model based on data by using algorithm alpha++ and heuristic miner in ProM is created. The models are shown in Fig. 2 and Fig. 3. Those models show that the flow of the business process model goes to the final stage.

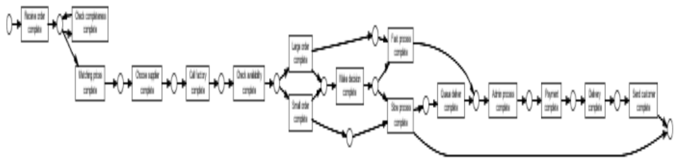


Fig. 2. A model by alpha ++ algorithm

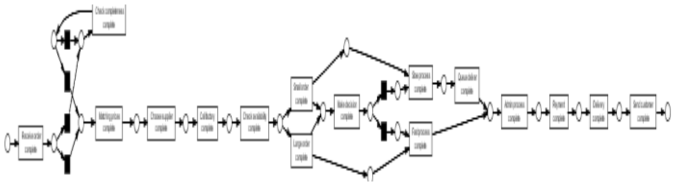


Fig. 3. A model by heuristic miner algorithm

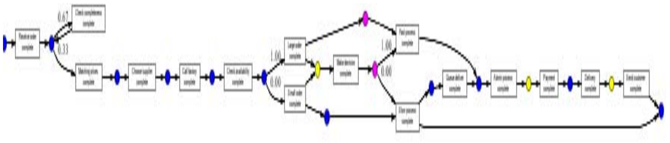


Fig. 4. Bottleneck in the first case with alpha++ algorithm

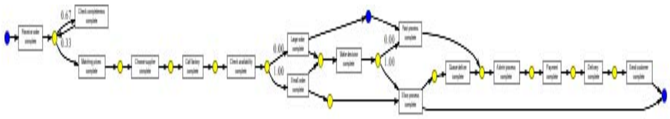


Fig. 5. Bottleneck in the second case with alpha++ algorithm

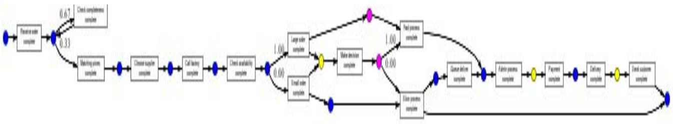


Fig. 6. Bottleneck in the first case with heuristic miner algorithm

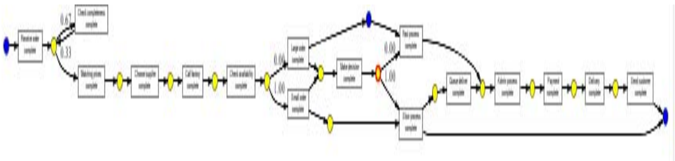


Fig. 7. Bottleneck in the second case with heuristic miner algorithm

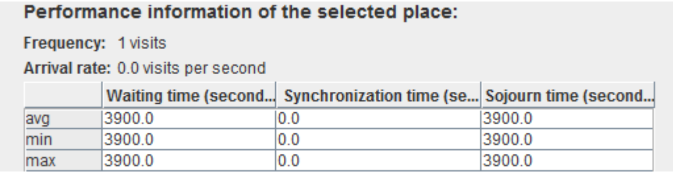
IV. RESULT AND ANALYSIS

In this research, there are 2 used cases for modeling different business process. The first case is the a case where the transaction will run faster caused by quantity of order that is bigger than the specified limit. The second case will experience a slow process where the transaction will run longer due to the quantity of order that is less than the specified limit. In the case, there are two decisions conditions where the customer can wait for the delivery process or customer may order cancelation.

A. Bottleneck identification

This section describes the identification stages of the bottleneck. This identification is based on two cases. Two cases are shown below.

1) *Receive order >> check completeness >> check fix status >> matching price >> choose supplier >> call factory >> check availability >> large order >> make decision >> fast process >> admin process >> payment >> delivery >> send customer.*
2) *Receive order >> check completeness >> check fix status >> matching prices >> choose supplier >> call factory >> check availability >> small order >> make decision >> slow process >> queue deliver >> admin process >> payment >> delivery >> send customer*



Performance information of the selected place:

Frequency: 1 visits

Arrival rate: 0.0 visits per second

	Waiting time (second...	Synchronization time (se...	Sojourn time (second...
avg	3900.0	0.0	3900.0
min	3900.0	0.0	3900.0
max	3900.0	0.0	3900.0

Fig. 8. Total tokens in the first case

Fig. 4 until Fig. 7 show the overviews of the business process model based on alpha++ algorithm and heuristic miner algorithm along with the location of the bottleneck. Those pictures show the existence of a magenta-colored task that means the average waiting time is high and making it an obstacle. The yellow color signifies a reasonably fast waiting time, while the blue color signifies a fast wait time.

Based on the both cases, the bottleneck of each algorithm will be different for each algorithm. The difference is happening in the first case where the occurrence of bottlenecks in the task of fast process and make decisions are shown in magenta color. The bottleneck calculation only calculates the time difference required between transactions connected with the place. Then, the average waiting time between places will be compared. Fig. 8 shows the total execution time in the first case.

The occurrence of the long waiting time makes the decision process that causes a decision based on the quantity of orders. It also affects whether to in the process of delivery quickly or long that requires waiting until the order quota is fulfilled. Because of this process, the bottleneck occurs between the process of decision-making.

By analyzing the both cases, it will be seen that better bottleneck determination on the heuristic miner algorithm. This is due to the fact that the possibility of bottleneck increasing. The bottleneck searching is viewed through analysis performance with Petri net in ProM. The purpose of performance analysis is to provide a means of assessing the process performance of event logs. With the performance analysis using Petri net, it is possible to see places that are considered jams.

The test results revealed that the algorithm plays a role in an accuracy bottleneck. The more precise a business process the more accurate the calculation of the bottleneck. The alpha ++ algorithm does not take into account the frequency of modeling business processes. The heuristic miner calculates frequency, so that it can detect any noise in the data. The heuristic miner algorithm describes the most dominant transactions.

The calculation results and the output of the ProM in determining the bottleneck are calculated based on the token in place. The algorithms follow to determine the bottleneck that occurs related to the exact or not algorithm. The last step is calculating the average waiting time of each place.

B. Conformance checking

Conformance Checking is one of the plug-in in ProM Tools. By using the analysis, it will see the performance value of the model that is formed. Calculation of the value of the accuracy between log and process model is done by repeat the log process contained in the model and calculate the inaccuracy of the tokens. The equation of the calculation is shown in (1) [19].

$$f = \frac{1}{2}\left(1 - \frac{\sum_{i=1}^{k} n_i m_i}{\sum_{i=1}^{k} n_i c_i}\right) + \frac{1}{2}\left(1 - \frac{\sum_{i=1}^{k} n_i r_i}{\sum_{i=1}^{k} n_i p_i}\right) \qquad (1)$$

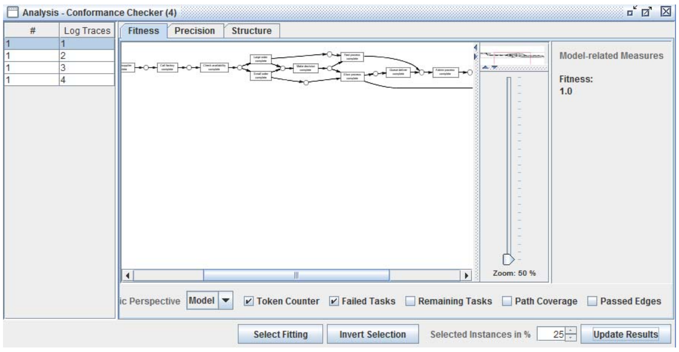
In (1), for all values i, mi ≤ ci and ri ≤ pi and fitness = 0 ≤ f ≤ 1. For ci and pi, the values are not possible 0 because tokens are always present at the beginning and end of the process at least 1 token. Fig. 9 shows the conformance checking in ProM.

C. *Dotted Chart Analysis*

To get an overview of the event log, this research performed a dotted chart analysis [20], i.e. plotting the log over time where each event from the log is represented by a dot in the chart. It enables visually examining the log and lets humans quickly identify patterns, that are impossible to see from browsing the log in text format and hard to detect from standard statistics calculated from the log (unless one of the searched patterns is already know). Fig. 10 shows the example of dotted chart analysis.

D. *Fraud Detection with heuristic miner*

After doing bottleneck analysis using alpha++ algorithm and heuristic miner, the fraud analysis from every activity that exist in the business process is done by using heuristic miner algorithm. Table II and Table III show the dependency matrix of each activity value and total of matrix dependency by using heuristic miner.



Analysis - Conformance Checker (4)

#	Log Traces
1	1
1	2
1	3
1	4

Model-related Measures

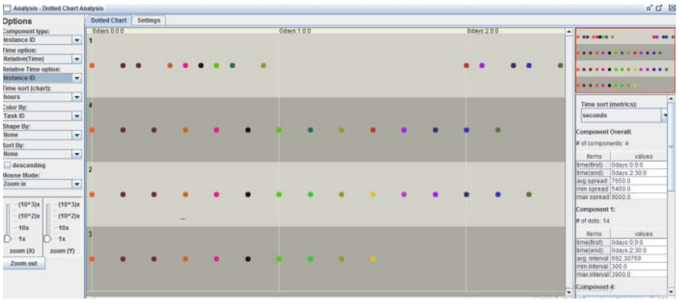
Fitness:
1.0

ic Perspective: Model

☑ Token Counter ☑ Failed Tasks □ Remaining Tasks □ Path Coverage □ Passed Edges

Selected Instances in %: 25

Fig. 9. Conformance checking



Analysis - Dotted Chart Analysis

Options

Component type: Instance ID

Time option: Relative(Time)

Relative Time option: Instance ID

Time sort (chart): hours

Color By: Task ID

Shape By: None

Sort By: None

□ descending

Mouse Mode: Zoom in

Time sort (metrics): seconds

Component Overall:
# of components: 4

items	values
time(first)	0days:0:0:0
time(end)	0days:2:30:0
avg spread	7650.0
min spread	5400.0
max spread	9000.0

Component 1:
# of dots: 14

items	values
time(first)	0days:0:0:0
time(end)	0days:2:30:0
avg. interval	692.30769
min interval	300.0
max interval	3900.0

Component 4:

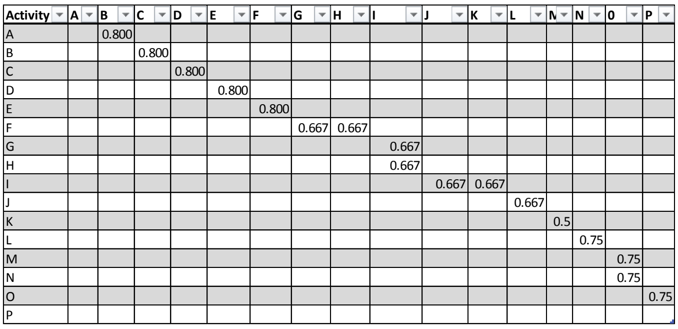
Fig. 10. Dotted chart analysis

After the values between each business process activity are connected, then the next step is to find the standard deviation and the average value. Based on Table III, the standard deviation value is 0.0803 and the average value is 0.8113. Utilizing the average value and the standard deviation value, relative to best threshold and dependency threshold are calculated. The calculation of those thresholds is described in (2) and (3).

RBT (Relative to best threshold)
= Avg PDM - (SD PDM/2)
= 0.8113 - (0.0803/2)
= 0.77115 (2)

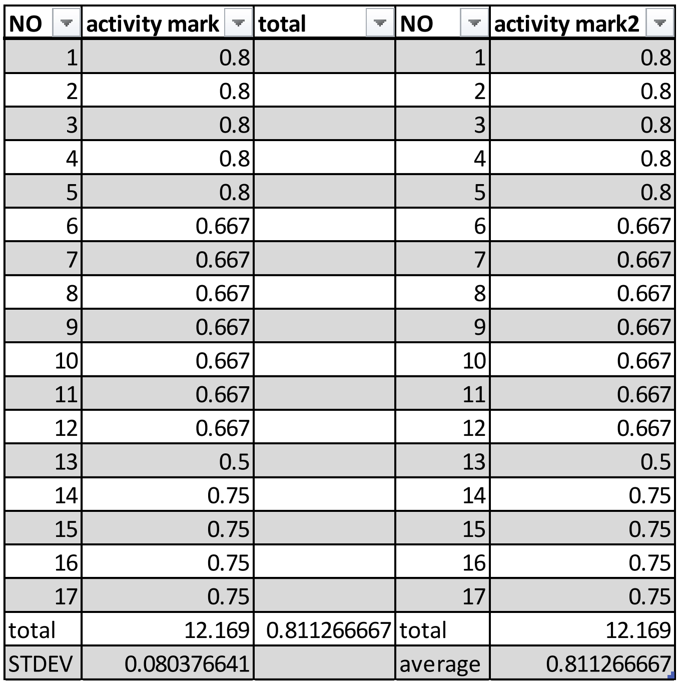
Dependency threshold
= Avg PDM - SD PDM
= 0.8113 – 0.08038
= 0.73089 (3)

TABLE II. MATRIX DEPENDENCY HEURISTIC MINER



Activity	A	B	C	D	E	F	G	H	I	J	K	L	M	N	0	P
A		0.800														
B			0.800													
C				0.800												
D					0.800											
E						0.800										
F							0.667	0.667								
G									0.667							
H									0.667							
I										0.667	0.667					
J												0.667				
K													0.5			
L														0.75		
M															0.75	
N															0.75	
O																0.75
P																

TABLE III. TOTAL OF MATRIKS DEPEDENCY



NO	activity mark	total	NO	activity mark2
1	0.8		1	0.8
2	0.8		2	0.8
3	0.8		3	0.8
4	0.8		4	0.8
5	0.8		5	0.8
6	0.667		6	0.667
7	0.667		7	0.667
8	0.667		8	0.667
9	0.667		9	0.667
10	0.667		10	0.667
11	0.667		11	0.667
12	0.667		12	0.667
13	0.5		13	0.5
14	0.75		14	0.75
15	0.75		15	0.75
16	0.75		16	0.75
17	0.75		17	0.75
total	12.169	0.811266667	total	12.169
STDEV	0.080376641		average	0.811266667

TABLE IV. FRAUD WITH ALPHA++ ALGORITHM

Defrauding	Fitness	Precision	Structure
Alpha ++	1	0.77	1

The fitness value >= 0.88 means there is no indication of business process in fraud, and there is fraud indication if the fitness value of a business process is detected < 0.88. Table IV shows the comparison of fraud checks using alpha++ algorithm.

V. CONCLUSION

This research identifies bottlenecks and fraud in a business process by using Heuristic miner and alpha++ algorithms. Based on the evaluation, there are three conclusions that can be derived.

The first conclusion is the accuracy of a calculation of the bottleneck is depended on the precise of a business process. The accuracy is higher if the business process is more precise. Alpha++ algorithm does not consider the frequency of modeling business processes. In contrast, heuristic miner algorithm calculates the frequency, so that it can detect any noise in the data.

The second conclusion is the occurrence of the long waiting time because the decision-making process based on the number of orders. It also affects whether it will be in the fast or slow delivery process which requires waiting time until the order is fulfilled. Because of this process, then the bottleneck occurs between the decision-making process..

The last conclusion is the performance of using Heuristic miner and alpha++ algorithms. Heuristic miner can model a business process appropriately and it has a higher performance than alpha++ algorithm because it can calculate the frequency, whereas alpha++ algorithm can not generate the frequency calculation.

REFERENCES

[1] O. Allani and S. A. Ghannouchi, "Verification of BPMN 2.0 Process Models: An Event Log-based Approach," *Procedia Computer Science*, vol. 100, pp. 1064–1070, 2016.
Available: http://doi.org/10.1016/j.procs.2016.09.282

[2] R. Sarno, E. W. Pamungkas, D. Sunaryono, and Sarwosri, "Business Process Composition Based on Meta Models," in *2015 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, 2015, pp. 315–318.
Available: https://doi.org/10.1109%2Fisitia.2015.7219998

[3] Hermawan and R. Sarno, "Developing Distributed System with Service Resource Oriented Architecture," *TELKOMNIKA (Telecommunication, Computing, Electronics and Control)*, vol. 10, no. 2, pp. 389–399, 2012.
Avaulable: https://doi.org/10.12928/telkomnika.v10i2.815

[4] R. Sarno, W. A. Wibowo, Kartini, Y. A. Effendi, and K. R. Sungkono, "Determining Model Using Non-Linear Heuristics Miner and Control-Flow Pattern," *TELKOMNIKA (Telecommunication, Computing, Electronics and Control)*, vol. 14, no. 1, pp. 349–360, 2016.
Available: https://doi.org/10.12928/telkomnika.v14i1.3257

[5] S. Huda, R. Sarno, T. Ahmad, and H. A. Santoso, "Identification of Process-based Fraud Patterns in Credit Application," in *2014 2nd International Conference on Information and Communication Technology (ICoICT)*, 2014, pp. 84–89.
Available: https://doi.org/10.1109/ICoICT.2014.6914045

[6] S. Huda, R. Sarno, and T. Ahmad, "Fuzzy MADM Approach for Rating of Process-Based Fraud," *Journal of ICT Research and Applications*, vol. 9, no. 2, pp. 111–128, 2015.
Available: https://doi.org/10.5614/itbj.ict.res.appl.2015.9.2.1

[7] R. Sarno, Y. A. Effendi, and F. Haryadita, "Modified Time-Based Heuristics Miner for Parallel Business Processes," *International Review on Computers and Software (IRECOS)*, vol. 11, no. 3, pp. 249–260, 2016.
Available: https://doi.org/10.15866/irecos.v11i3.8717

[8] R. Sarno and K. R. Sungkono, "Coupled Hidden Markov Model for Process Mining of Invisible Prime Tasks," *International Review on Computers and Software (IRECOS)*, vol. 11, no. 6, pp. 539–547, 2016.
Available: https://doi.org/10.15866/irecos.v11i6.9555

[9] R. Sarno and K. R. Sungkono, "Hidden Markov Model for Process Mining of Parallel Business Processes," *International Review on Computers and Software (IRECOS)*, vol. 11, no. 4, pp. 290–300, 2016.
Available: https://doi.org/10.15866/irecos.v11i4.8700

[10] L. Wen, W. M. P. van der Aalst, J. Wang, and J. Sun, "Mining process models with non-free-choice constructs," *Data Mining and Knowledge Discovery*, vol. 15, no. 2, pp. 145–180, 2007.
Available: https://doi.org/10.1007/s10618-007-0065-y

[11] A. Burattin and A. Sperduti, "Automatic determination of parameters' values for Heuristics Miner++," in *Evolutionary Computation (CEC), 2010 IEEE Congress*, 2010, pp. 1–8.
Available: http://doi.org/10.1109/CEC.2010.5586208

[12] H. Ritchi, "Identifikasi Pengendalian Aplikasi Dalam Analisis Proses Bisnis," *Abstrak*, 2009.

[13] M. N. Lacheheub and R. Maamri, "BPRN: A Formal Model of Resources Consumption in Business Process Based on an Extension of Petri-Net and Workflow model," *Procedia Computer Science*, vol. 83, pp. 171–178, 2016.
Available: http://doi.org/10.1016/j.procs.2016.04.113

[14] W. Es-Souf, E. Yahia, and L.Roucoules,"On the Use of Process Mining and Machine Learning to Support Decision Making in Systems Design," in *IFIP International Conference on Product Lifecycle Management*, 2016, pp. 56–66.
Available: https://doi.org/10.1007/978-3-319-54660-5_6

[15] R. S. Mans, D. Wismeijer, H. A. Reijers, and M. van Genuchten, "Mining Processes in Dentistry," *IHI '12 Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, pp. 379–388, 2012.
Available: http://doi.org/10.1016/j.procs.2016.07.066

[16] N. Tax, N. Sidorova, R. Haakma, and W. M. P. V. Der Aalst, "Log-based Evaluation of Label Splits for Process Models," *Procedia Computer Science*, vol. 96, pp. 63–72, 2016.
Available: http://doi.org/10.1016/j.procs.2016.08.096

[17] A. H. M. ter Hofstede, W. M. P. van der Aalst, M. Adams, and N. Russell, *Modern Business Process Automation: YAWL and its support environment*. Springer Science & Business Media, 2009.

[18] W. M. P. Van der Aalst, A. H. M. Ter Hofstede, B. Kiepuszewski, and A. P. Barros, "Workflow patterns," *Distributed and Parallel Databases*, vol. 14, no. 1, pp. 5–51, 2003.
Available: http://doi.org/10.1023/A:1022883727209

[19] A. Rozinat and W. M. P. van der Aalst, "Conformance checking of processes based on monitoring real behavior," *Information Systems*, vol. 33, no. 1, pp. 64–95, 2008.
Available: http://doi.org/10.1016/j.is.2007.07.001

[20] W. M. P. Van Der Aalst, *Process Mining Discovery, Conformance and Enhancement of Business Processes*. Germany: Springer, 2011.