Business Processes Discovery in Halal Restaurant Kitchen Using Graph-Based Algorithm

Adhatus Solichah Ahmadiyah
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
adhatus@if.its.ac.id

Indira Nursyamsina Hazimi
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
indirahazimi@gmail.com

Dicky Maulana Rozi
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
dickymaulanarozi@gmail.com

Rizky Andre Wibisono
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
rizkyandre.w@outlook.com

Donny Fitrado
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
donny.public@gmail.com

M. Ardika Rifqi
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
rifweb.android@gmail.com

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

Kelly Rossa Sungkono
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
kelly@if.its.ac.id

Abstract — As the halal lifestyle spreads rapidly in the culinary field, a good halal SOP in the kitchen is crucially needed because there are issues such as ingredient usage in the cooking process, cooking utensils, and storage. In Indonesia, the SOP should comply with Halal Assurance System (HAS) 23000. In generating the business process model, the existing Alpha-based algorithms work best in detecting certain tasks but fail to detect the invisible non-prime task. This paper aims to apply a graph-based algorithm to address the Alpha-based algorithm's weakness to produce a good SOP in the halal restaurant kitchen. We used event logs of the cooking process and facility treatment as the input of process model discovery. The fitness and precision results confirmed that the proposed algorithm outperforms the existing Alpha-based algorithms.

Keywords — graph, halal, process discovery, process mining.

I. INTRODUCTION

The halal lifestyle is currently popular and has become a world trend. The demand for halal meat is growing rapidly due to the fact that many countries with a majority Muslim population issue religious regulation that approve of the practice of stunning during pre-slaughter [1]. Also, purity and cleanliness in the halal lifestyle attract non-Muslim consumers' attention, especially consumers who care about food management, especially organic food [2]. Halal food consumers pay attention to halal guarantees at every eating place they visit. In general, places to eat that meet the quality of halal standards have a certification of the food served's halalness. One of the food business's challenges today is how to control cooking facilities and processes so that the food produced still meets the quality of halal standards set by the Indonesian Ulema Council (MUI).

To provide halal assurance to consumers, the food serving process, including the purchase of ingredients, management of facilities, and the process of cooking food, must be carried out in an orderly and neat manner as determined by the Indonesian Ulema Council (MUI). In addition to the procedures stipulated by the MUI, halal hygienic food can be seen from two approaches, namely the Islamic Sharia process approach, HCCP (Halal Critical Control Point), and HACCP certification (Hazard Analysis Critical Control Point). A review of various approaches to the use of HCCP, HACCP, and hybrids on food safety and halal assurance has been discussed by Marin et al. [3]. LPPOM-MUI released a Halal Assurance Systems (HAS) guide, namely HAS 23000 [4], which explains several HACCP elements, manuals, and decision trees in identifying critical points in each decision. This guideline is one of the main directions determining whether a food is halal or not in Indonesia.

In carrying out the business processes and provisions of the MUI, HCCP and HACCP must be executed in order. Given that every person has deficiencies that can neglect one of the process procedures, a management system, and a business process model are needed. This research proposes a business process model discovery that can ensure halal restaurant kitchens' conformity, including cooking food and controlling cooking utensils, and meeting the quality of halal standards. The food cooking process event log is evaluated by the discovery process, using a graph-based algorithm and compared to existing Alpha-based algorithms.

The explanation of this research is arranged as follows. Section 2 describes methods and the dataset used in the discovery process. The results and analysis of business process models is presented in Section 3. Then, section 4 explains the conclusions of this research.

II. Research Method

Fig. 1 shows our proposed method. First, the cooking process's event log, including utensil treatments and storage usage, is prepared. The event log consists of case ID, activity, and originator. A case represents a series of tasks from start to end, i.e., from cooking until serving. Meanwhile, a trace is a variation of cases that occurs in the event log. For all cases in the event log, the relationship between tasks is discovered to assemble the process model.

Fig. 1 Proposed Method

The identified relationship from the cooking event logs are as follows. The running process depicted in the event log follows the HAS 23000 when it comes to utensil treatment, using halal ingredients, and storing or serving food.

A. Cooking process

In the cooking process, some simple tasks occur. Sequential tasks (Fig. 2): Chef drains and mashes the meat. Then, he boils water to make broth. Having finished making broth, he heat oil and sauté ground spices. Until then, he cooks the meat with other ingredients.

Fig. 2 Sequential tasks

Next, the invisible Prime Task Skip (Fig. 3) occurs when the cook helper prepares ingredients and spices to make stew. He needs to make sure all ingredients and spices are halal. If there any non-halal ingredients or spices, they must be replaced with halal ones. Then, he can use the ingredients and spices together to boil the meat.

B. Utensils Treatment

Invisible Prime Task Switch occurs from the event log (Fig. 4). Porter checks the history of cooking utensils usage. When it is contaminated with non-halal ingredients, then the officer needs to wash the utensils. Otherwise, the officer confirms halal condition so that the utensils ready to be used.

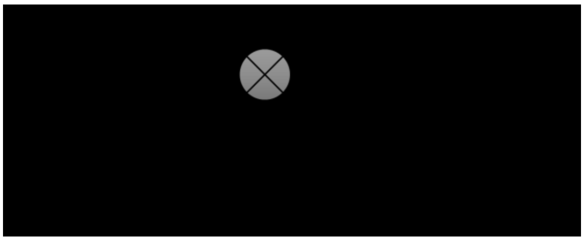

Fig. 3 Invisible Prime Task Skip

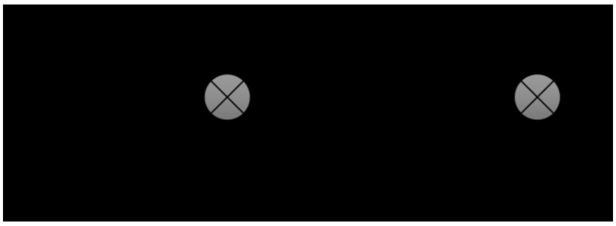

Fig. 4 Invisible Prime Task: Switch

Invisible Prime Task Redo occurs after the cook helper boils water; he sometimes checks the meet's doneness (Fig. 5).

Fig. 5 Invisible Prime Task Redo

C. Taking Orders

A unique task that occurs in the taking order process is the non-free choice task. When the waiter confirms orders, there is two option whether the customer chooses dine-in or takeaway. Then, he continues to prepare how to serve the food, either pack the food or serve it on plates. Finally, he gives the order list to the cashier. From Fig. 6, we can see even though there are choices. Actually, the first choice determines the second choice.

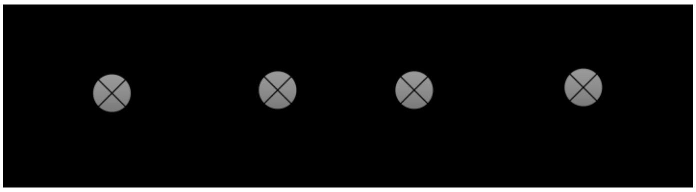

Fig. 6 Non-Free Choice Task

D. Serving Process

There is an invisible non-prime task in the serving process in stacked parallel relationships of AND and XOR relationships. As shown in Fig. 7, the cook helper checks customer preferences. If no customization, then he directly serves the standard menu. If there is customization, such as more vegetables, rice, or both, then serve the customization menu.

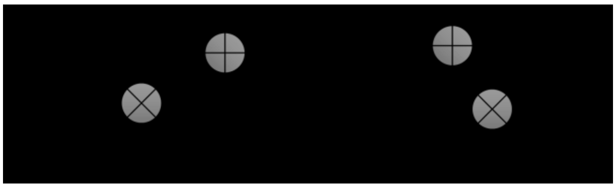

Fig. 7 Invisible Non-Prime Task

E. Graph-based Process Model Discovery Algorithm

Having identified the event logs' relationships, we employed Neo4J graph to assemble the business process model. We treat each activity from the event logs as a node (Step 1 and 2 in TABLE I). Then, we construct the edges using the previous identified relationship of tasks (subsection A to D). The edge construction is performed step by step started from the sequence relationship (Step 3 in TABLE I), continued with AND, XOR, and OR relationships using cypher query included in step 4 to 7 in TABLE I. Lastly, nodes representing invisible tasks are identified using cypher query included in step 8 to 10 in TABLE I.

TABLE I NEO4J CYPHER QUERY

Step	Desciption	Query
1	Load Activities	LOAD CSV with headers FROM "file:///eventlog.csv" AS line Merge (:Activity{CaseId:line.CaseID, Name:line.Activity, Time:line.Timestamp})
2	Load Case Activities	LOAD CSV with headers FROM "file:///eventlog.csv" AS line Merge (:CaseActivity {Name:line.Activity })
3	Create Sequences	MATCH (c:Activity) WITH COLLECT(c) AS Caselist UNWIND RANGE(0,Size(Caselist) - 2) as idx WITH Caselist[idx] AS s1, Caselist[idx+1] AS s2 MATCH (b:CaseActivity),(a:CaseActivity) WHERE s1.CaseId = s2.CaseId AND s1.Name = a.Name AND s2.Name = b.Name MERGE (a)-[r:SEQUENCE]->(b)
4	Create XOR Split Relations	MATCH (bef)-[r]->(aft) WHERE size((bef)-->())>1 AND size((aft)<--())=1 AND (size((aft)-->())=1 OR size((aft)-->())>1) CREATE (bef)-[:XORSPLIT]->(aft) DELETE r
5	Create XOR Join Relations	MATCH (bef)-[r]->(aft) WHERE (size((bef)-->())=1 OR size((bef)-->())>1) AND size((aft)<--())>1 CREATE (bef)-[:XORJOIN]->(aft) DELETE r
6	Create AND Split Relations	MATCH (aft1)<-[r]-(bef)-[s]->(aft2) WHERE size((bef)-->())>1 AND size((aft2)-->())=size((bef)-->()) AND size((aft1)-->())=size((bef)-->()) AND not (aft1)-[:SEQUENCE]->(bef) AND not (aft2)-[:SEQUENCE]->(bef) MERGE (aft1)<-[:ANDSPLIT]-(bef)-[:ANDSPLIT]->(aft2) DELETE r,s
7	Create AND Join Relations	MATCH (aft1)-[r]->(bef)<-[s]-(aft2) WHERE size((bef)<--())>1 AND size((aft2)-->())=size((bef)<--()) AND size((aft1)-->())=size((bef)<--()) AND not ()-[:ANDSPLIT]->(bef) MERGE (aft1)-[:ANDJOIN]->(bef)<-[:ANDJOIN]-(aft2) DELETE r,s
8	Create Invisible Task Relations	MATCH (aft2)<-[r]-(bef)-[s]->(aft1) WHERE (TYPE(s)='XORSPLIT' AND TYPE(r)='XORJOIN') CREATE (i:Invisible_Task {Name: "Inv_Task"}) WITH bef, i, aft2, r, s CALL apoc.create.relationship(bef, type(s), {}, i) YIELD rel as relation1 CALL apoc.create.relationship(i, type(r), {}, aft2) YIELD rel as relation2 DELETE r
9	Create Non-Free Choice Relations	MATCH (b)-[:XORSPLIT]->(a)-[:XORJOIN]->(d)-[:XORSPLIT]->(n) MATCH (k:Activity),(l:Activity) WHERE a.Name<>n.Name AND k.Name=a.Name AND l.Name=n.Name AND k.CaseId=l.CaseId AND k.Time<=l.Time AND size((b)-->())=2 AND size((b)<--())=1 AND size((a)<--())=1 AND size((a)-->())=1 AND size((d)<--())=2 AND size((d)-->())=2 AND size((n)-->())=1 AND size((n)<--())=1 MERGE (a)-[:NONFREECHOICE]->(n)
10	Invisible Non-Prime Task (XOR-AND)	MATCH (a)-[b]->(c:Invisible_Task)-[d]->(e) MATCH (a)-[f]->(g:Invisible_Task)-[h]->(i) MATCH (e)-[j]->(i) MATCH (i)-[k]->(e) WHERE e.Name<>i.Name AND id(c)>id(g) DETACH DELETE g DELETE d,j,k MERGE (c)-[:ANDSPLIT]->(e) MERGE (c)-[:ANDSPLIT]->(i) MATCH (a)<-[b]-(c:Invisible_Task)<-[d]-(e) MATCH (a)<-[f]-(g:Invisible_Task)<-[h]-(i) WHERE e.Name<>i.Name AND id(c)>id(g) DETACH DELETE g DELETE d MERGE (c)<-[:ANDJOIN]-(e) MERGE (c)<-[:ANDJOIN]-(i)

F. Case Study

In a halal restaurant kitchen, the business processes that occur include the purchase of cooking tools and ingredients that comply with halal standards, cleaning of cooking utensils, and the cooking process. The cooking business process in a halal restaurant kitchen also needs to follow MUI's halal standards. What needs to be considered in this business process is the storage and use of cooking utensils. If stored together with haram food ingredients or used to process haram food ingredients, the cooking utensils need to be washed according to halal standards. The same is true for the food ingredients and spices used. If stored together with haram food ingredients, these ingredients and or spices need to be replaced according to halal standards.

The case study is a business process in cooking Indonesian dish namely Empal Gepuk. Usually, this dish use beef, sliced in the fiber and half-cooked direction. After that, it is removed and marinated with coconut milk mixture. Then, the meat is boiled again and fried when it will be served. Warm rice and distinct chili sauce are usually served together with "empal gepuk". The event log consists of 26 activities, described in TABLE II..

An event log from the case studies described before is used for this research. The event log contains CaseID as an instance of the process, Activity as the name of the activity being carried out, Timestamp to record the event's time, Originator as the person or actor of the activity. Overall, the event log used consists of 72 cases. describes the event log fragments of the cooking process of "empal gepuk" used in this research. In this table, the Activity column is simplified using the Activity Code described in TABLE II.

TABLE II ACTIVITY IN THE "EMPAL GEPUK" COOKING EVENT LOG

Activity Code	Activity	Originator
a	Check cooking utensils	Porter
b	Move the cooking utensils	Porter
c	Check the history of cooking utensils usage	Porter
d	Wash tools	Porter
e	Halal Tools Usage Confirmation	Porter
f	Boil water	Cook Helper
g	Check doneness of the meat	Cook Helper
h	Prepare ingredients and spices for the stew	Cook Helper
j	Replace non-halal ingredients and spices with halal ingredients	Cook Helper
i	Boil the meat and ingredients	Cook Helper
k	Drain and mash the meat	Chef
l	Boils water to make broth	Chef
m	Heat oil and sauté ground spices	Chef
n	Cook the meat with other ingredients	Chef
o	Check customer preferences	Cook Helper
p	No customization	Cook Helper
q	Add vegetables	Cook Helper
r	Add rice	Cook Helper
s	Plating	Cook Helper
t	Confirm orders	Waiter
u	Accept take-away orders	Waiter
v	Accept dine-in orders	waiter
w	Prepare the serving of food	waiter
x	packing food	waiter
y	Serve on plates	waiter
z	Forward order lists to the cashier	waiter

III. RESULT AND ANALYSIS

A. Result

In this research, halal kitchen SOP containing sequential and parallel relationships, non-free choice, and invisible non-prime task has been successfully discovered. To evaluate the performance of the proposed graph-based algorithm, comparison against the existing Alpha-based algorithm was conducted. Process model of the invisible non-prime task discovered by graph-based algorithm, Alpha, Alpha++, and Alpha# are displayed in Fig. 9, Fig. 10, and Fig. 11 respectively. The complete business process model generated The complete result of the Graph-based algorithm is displayed in Fig. 12.

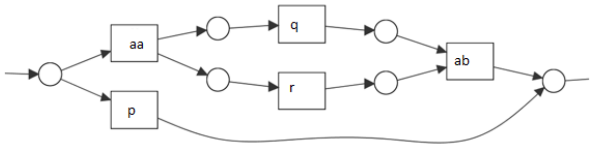

Fig. 8 Result of Invisible Non-Prime Task on Alpha Algorithm

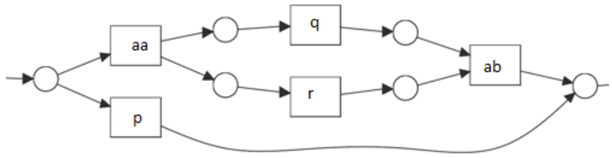

Fig. 9 Result of Invisible Task Non-Prime Task on Alpha++

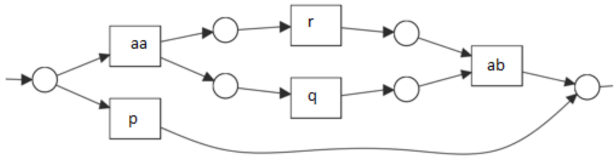

Fig. 10 Result of Invisible Task Non-Prime Task on Alpha#

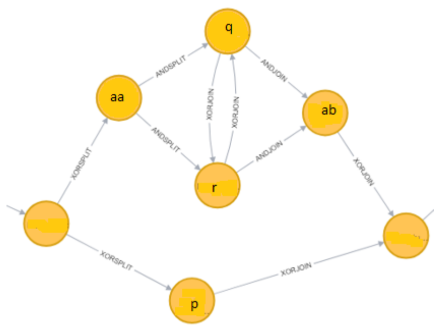

Fig. 11 Result of Invisible Task Non-Prime Task on Graph-based Algorithm

The performance each method is evaluated using Fitness and Precision. Fitness represents the degree of similarity of the generated cases of the process model compared to the cases found in the event log formulated in (1). Meanwhile, precision represents the process model's ability to discover traces found in the event log, formulated in (2). Variable ce represents total cases in the process model, te represents total cases in the event log, ct represents total traces discovered by the process model, and tm represents total traces in the event log.

$$Fitness = \frac{ce}{te} \quad (1)$$

$$Precision = \frac{ct}{tt} \quad (2)$$

Comparison between the proposed method and the existing Alpha algorithm is tabulated in TABLE III.

TABLE III COMPARISON RESULTS

Algorithm	Invisible Non-Prime Task	Fitness	Precision
Alpha#	failed	0.929	0.903
Alpha	failed	0.986	0.784
Alpha++	failed	0.986	0.948
Graph-based	succeed	1	1

B. Discussion

Based Fig. 8, Fig. 9, and Fig. 10, the generated process model of Alpha, Alpha++, and Alpha# are failed to construct invisible non-prime task. Meanwhile, the Graph-based algorithm successfully discover it. Moreover, from the complete generation of process model as displayed in Fig. 12, the graph-based algorithm are able to construct all types of relationship correctly. It shows the superiority of the graph-based algorithm. Apart from the visual comparison, Fitness and Precision values show its superiority against the existing Alpha-based algorithms. Even though the improvement in the Fitness and Precision values is small, it significantly effects the structure of the generated process model.

From the results obtained, fitness and precision values are compared between the Alpha# algorithm with Alpha and the Alpha++ algorithm, which bring different values, seen in TABLE III. To visualize the proposed business process model entirely, Neo4j is used and then compared with the previous algorithm.

From TABLE III, algorithm Alpha# obtained a fitness value of 0.929 and precision with 0.903 of the total trace in the event log (te) of 70 and mining process trace (tt) of 72, with the trace of the event log (ct) is 65. The fitness and precision value of Alpha# is close to 1 due to Alpha# reveal three invisible tasks, and there is none disconnected part. The result from Alpha has a fitness value of 0.986 and precision of 0.784 from the total *trace* in the event log (te) of 70 and mining process trace (tt) of 88, with trace from event log (ct) has a result of 69. When compared to Alpha#, Alpha has a fitness value that is closer to 1 but with a precision that is far from the Alpha# result. The Alpha causes this does not show any invisible tasks and has a disconnected part. Results of Alpha++ have the value of fitness 0.986 and precision of 0.958 of the total trace in the event log (te) of 70 and mining process trace (tt) of 72, as well as the trace of the event log (ct) have the results of 69. Alpha++ does not display any invisible tasks like the Alpha algorithm but does not have a disconnected part like Alpha# does.

Neo4j reaches the fitness and precision of 1 from the total trace in the event log (te) of 72, and mining process trace ™ of 72 and traces from event log (ct) have a result of 72. As the results of Alpha#, Alpha, and Alpha++ already explained, it can be concluded that Neo4j is the best algorithm, which owns a fitness and precision value of 1. Compared to Alpha#, Alpha, and Alpha++, this result is caused by Neo4j has many variants.

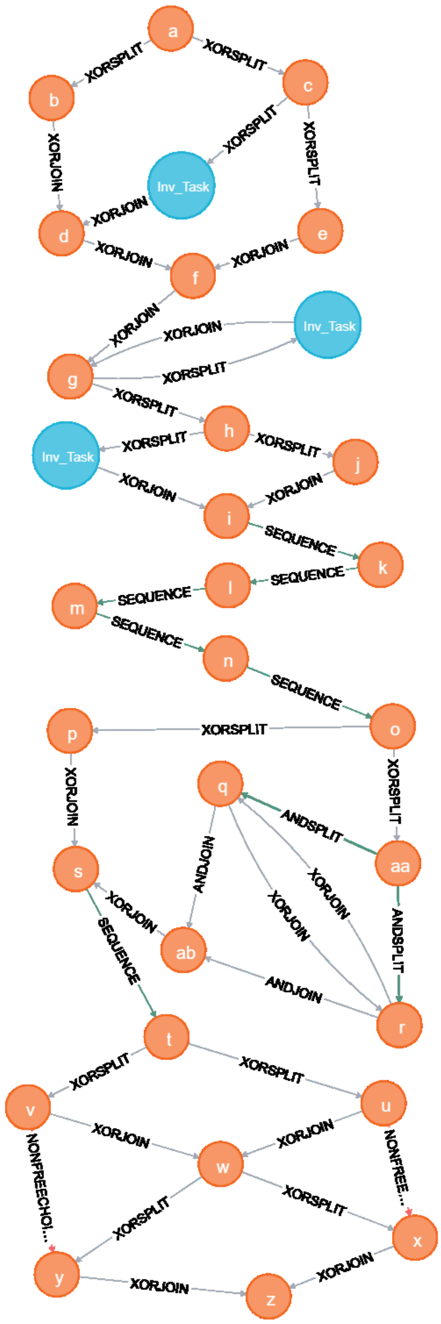

Fig. 12. Graph Visualization with Neo4j

IV. CONCLUSION

This research employed a graph-based algorithm to discover the halal restaurant kitchen's business process, which can ensure cooking process, utensils treatment, and serving meet halal standards stated in Halal Assurance System (HAS) 23000. In specific, the business process contains invisible non-prime tasks in the form of XOR and OR relationships.

The graph-based algorithm has successfully discovered the business process model containing various relationships that the existing Alpha-based algorithms can only construct some of them. However, future research needs to be conducted on solving the time complexity issue for the large size of the event logs.

V. ACKNOWLEDGMENT

This research was funded by AUN/SEED-Net under Special Program for Research Against COVID-19 (SPRAC) and the Ministry of Research and Technology/National Research and Innovation Agency Republic of Indonesia (Ristek-BRIN) and the Indonesian Ministry of Education and Culture under Newton Institutional Links - Kerjasama Luar Negeri (KLN) Program and under Penelitian Terapan Unggulan Perguruan Tinggi (PTUPT) Program managed by Institut Teknologi Sepuluh Nopember (ITS).

REFERENCES

[1] A. Fuseini, T. Knowles, P. Hadley and S. Wotton, "Halal stunning and slaughter: Criteria for the assessment of dead animals," *Meat Science,* vol. 119, pp. 132-137, 2016.

[2] H. N. Lubis, N. F. Mohd-Naim, N. N. Alizul and M. U. Ahmed, "Current trusted technology and innovations in halal food analysis," *Trends in Food Science & Technology,* vol. 58, pp. 55-68, 2016.

[3] H. Lubis, N. Mohd-Naim, N. Alizul and M. Ahmed, "From market to food plate: Current trusted technology and innovations in halal food analysis," *Trends in Food Science & Technology,* vol. 58, pp. 55-68, 2016.

[4] MU Indonesia, "General Guidelines of Halal Assurance System," *Food, Drugs and Cosmetics,* 2008.

[5] L. Hakim, R. Sarno and K. R. Sungkono, "Modified Alpha++ Algorithm for Discovering the Hybrid of Non-Free Choice and Invisible Task of Business Processes," *International Journal of Intelligent Engineering and Systems,* vol. 12, no. 3, pp. 31-40, 2019.

[6] Y. A. Effendi and R. Sarno, "Parallel Process Discovery Using A New Time-based Alpha++ Miner," *IIUMEJ,* vol. 21, no. 1, pp. 126-141, 2020.

[7] Leno, A. Armas-Cerventes, M. Dumas, M. L. Rosa and F. M. Maggi, "Discovering Process Maps from Event Streams," *International Conference on the Software and Systems Process,* pp. 86-95, 2018.

[8] S. Park and Y. Kang, "A Study of Process Mining-based Business Process Innovation," *Procedia Computer Science,* vol. 91, pp. 734-743, 2016.

[9] A. Augusto, R. Conforti, M. Dumas, F. M. Maggi, A. Marrella, M. Mecella, A. Soo and M. La Rosa, "Automated Discovery of Process Models from Event Logs: Review and Benchmark," *IEEE Transactions on Knowledge and Data Engineering,* vol. 31, no. 4, pp. 686-705, 2018.

[10] S. J. Leemans, D. Fahland and W. M. van der Aalst, "Scalable Process Discovery with Guarantees," *In Enterprise Business-Process and Information Systems Modeling,* pp. 85-101, 2015.

[11] H. Darmawan, R. Sarno, A. S. Ahmadiyah, K. R. Sungkono and C. S. Wahyuni, "Anomaly Detection Based on Control-flow Pattern of Parallel Business Processes," *TELKOMNIKA,* vol. 16, no. 6, pp. 2808-2815, 2018.

[12] R. Sarno and K. R. Sungkono, "Coupled Hidden Markov Model for Process Discovery of Non-Free Choice and Invisible Prime Tasks," *Procedia Computer Science,* vol. 124, pp. 134-141, 2017.

[13] R. Sarno, F. Sinaga and K. R. Sungkono, "Anomaly Detection in Business Processes Using Process Mining and Fuzzy Association Rule Learning," *Journal of Big Data 7,* pp. 2-5, 2020.

[14] S. J. Lemans, D. Fahland and W. M. van der Aalst, "Discovering Block-Structured Process Models from Incomplete Event Logs," *International Conference on Applications and Theory of Petri Nets and Concurrency,* pp. 4-7, 2014.

[15] H. I. Amin, A. Rachmadi and N. Y. Setiawan, "Pengecekan Conformance Modul Proses Keuangan Dengan Process Mining Pada Rumah Sakit Medika Mulia Di Kabupaten Tuban," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer,* vol. 2, no. 9, pp. 2874-2881, 2018.

[16] V. Naderifar, S. Sahran and Z. Shukur, "A Review on Conformance Checking Technique for the Evaluation of Process Mining Algorithms," *TEM Journal,* vol. 8, no. 4, pp. 1232-1241, 2019.