Process Discovery of Collaboration Business Process Containing Invisible Task in Non-Free Choice by using Modified Alpha ++

Moh. Zulfiqar Naufal Maulana
Department Of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
mznaufalmaulana@gmail.com

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

Kelly R. Sungkono
Department Of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
kelly@its.ac.id

***Abstract*— Process mining is a set of techniques that combine the scientific point of view of the data with the point of view of running processes correctly. The process mining can be performed with several algorithms, such as Alpha miner, Inductive Miner, and Fodina. Those existing algorithms only form a process model. Nevertheless, a system can be built not only a process model but also collaboration process models. Collaboration process models are several interconnected process models that are carried out by sending messages between those activities. Besides shortcoming of depicting collaboration process models, another issue is called an Invisible Task in a Non-Free Choice relationship. This issue occurs when particular conditions, i.e. skip, redo and switch, are happened in activities which are interdependent with other activities. Alpha $ is a pioneering algorithm that describes the Invisible Task in Non-Free Choice (IT-in-NFC); however, this algorithm cannot handle collaboration process models. This study proposes Modified Alpha ++ that contains rules of storing messages of activities and modifies Alpha ++ algorithm with additional Invisible Task rules to present Alpha $. The evaluation results show a process model obtained by Modified Alpha ++ is better than other algorithms, i.e. Alpha ++, Fodina and Inductive Miner. That statement is proven with the Fitness, Precision, and F-Measure values of the process model by Modified Alpha ++ are 1. On the other hand, process models of Alpha ++, Fodina and Alpha ++ obtain less than one F-Measure value.**

***Keywords*—Process Mining, Collaboration Business Process, Invisible Task, Non-Free Choice**

I. INTRODUCTION

One of the essential aspects of an organization is how to discover an event log automatically [1]. This process will provide many benefits for the company, such as detecting problems in their organizational flow and finding valuable corporate flow points to improve their managerial performance [2].

There are many discovering methods based on an existing event log in its development, such as Alpha miner [3]–[7], Inductive Miner [8]–[10], Heuristic Miner [11]–[13], Split Miner [14], [15], Fuzzy Miner [16], and Graph-Based Miner [2], [17]. These methods can also be run on several assistive tools, such as ProM [18]. Automatic process analyses, including discovering a process model based on an event log, is called process mining.

In practice, a system can execute several business processes which are interacted each other. Those processes will run considering the rules which have been made by an organisation. The system must record every activity running on the business process in an event log to analyse those processes efficiently. The event log is then used as the basis for finding out whether the system is running according to the organisation's rules. However, the existing algorithms do not address the interaction of several processes when forming a process model. The interaction cannot be detected because those algorithms only consider relationships of activities without interactions of activities on different processes.

There are also issues in discovering a process model; one is an Invisible Task (IT). IT, an additional task that is not detected in the event log, appears in a process model to facilitate particular conditions [4]. Furthermore, there is also an issue called Non-Free Choice (NFC). NFC is a specific task that is executed when passing through a particular activity. In other words, NFC is an activity that will only be carried out when certain activities have passed before IT-in-NFC is the embedding of an Invisible Task in an NFC, but the process is challenging to implement because it is not explicitly contained in an event log [4], [19].

In previous research, the using of Alpha $ brings up an IT-in-NFC in a process. The algorithm uses improved mendacious dependency and implicit dependency in its process discovery [20]. However, these algorithms have drawbacks when modelling a collaborative process model. In other words, the algorithm does not handle a case where a business process is more than one business process.

In this study, Modified Alpha ++ algorithm is proposed to describe a process model based on activities and interactions of those activities. Those interactions called messages. There are two steps of the proposed modification of Alpha ++. First, there are several rules to add sending messages on the event log. Secondly, adding Invisible Tasks is made so that Alpha ++ algorithm can run as Alpha $. Modified Alpha ++ will then be compared with Inductive Miner and Fodina algorithms, comparing the Fitness, Precision, and F-Measure values.

In this paper, several sections will be discussed, such as definitions and previous research discussed in section 0. The proposed method will be discussed in Section III. Then the research results are discussed in Section IV. until the conclusion is the conclusion contained in Section V.

II. RESEARCH METHOD

A. Collaboration Business Process

This research was carried out by applying more than one pool, which is interconnected with one another. The pool is

linked using messages sent between the two. The message is then used as a link between the two business process for further process discovery. The message is stored as an activity that is embedded between two related activities between the two business process. TABLE I is an example of an event log in which a message has been inserted.

The business processes used in this research can be seen in Fig 5. Two business processes are interconnected. The business process is linked by sending messages between the two. The process of sending the message will be recorded in the system, carried out by the process discovery using the proposed method.

B. Invisible Task in Non-Free Choice

An Invisible Task (IT) is a process that is not recorded in an event log but appears in the business process model [17]. Several conditions cause IT to appear in a business process model. This situation differentiates the type rather than the emerging IT. Based on the Alpha # algorithm [4], various kinds of IT, including Skip, Redo, and Switch Invisible Task [21].

Fig 1 is an example of inserting an Invisible Task in a Non-Free Choice. Where activity E will never run unless activity B is executed and activity F will not run when activity C is not executed. Between E - B and C - F activities, an Invisible Task is inserted, which indicates that the process is entered into IT-in-NFC. In this study, this type of activity is found in the "Send Waste to Inecerator" activity which is a Non-Free Choice activity in the activities of "Put Waste in Safety Box", "Put Waste in Yellow Bin", "Put Waste in Brown Bin", and "Put Waste in Red Bin" (Fig 5).

C. Existing Process Discovery Algorithms

In this study, the process discovery was carried out using three methods, namely Inductive Miner, Fodina, and Modified Alpha ++. Using this algorithm, this research will compare the Fitness, Precision, and F-Measure values. Inductive miner will produce a process model in the form of a process tree that will describe the flow rather than an event log [8]. The advantage of this modeling is that we can easily identify a process model. Inductive miner can also complement an anomaly that occurs in an event log.

Fodina is a new automated discovery technique that focuses on durability and flexibility [22]. This method is proposed as a complement to the existing methods, namely heuristics. It adds the handling of encountering noisy data, the ability to find a duplicate activity, and the flexibility to configure user input.

Modified Alpha ++ is a modification of the Alpha ++ method, which currently only describes a Non-Free Choice (NFC) model. The purpose of modifying this method is to display an Invisible Task (IT) if there is an anomaly in an event log data.

TABLE I. THE EXAMPLE OF AN EVENT LOG WITH MESSAGE

Process Id	Case Id	Type	Activity	Time
Prc1	P001	Activity	A	07.00
Prc1	P001	Message	Msg 1	07.08
Prc1	P023	Activity	B	07.12
Prc1	P023	Activity	C	07.14
Prc2	P003	Activity	A	07.18
Prc2	P003	Activity	C	07.19
Prc2	P003	Activity	F	07.23

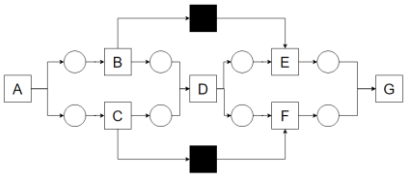

Fig 1. Invisible Task in Non-Free Choice (IT-in-NFC)

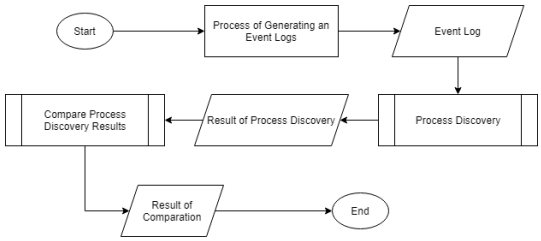

Fig 2. The Proposed Method

III. THE PROPOSED METHOD

In this section, the research flow in this paper is described. It starts from taking event logs, checking of Invisible Tasks and Non-Free Choice, then doing the process discovery using Modified Alpha ++, Inductive Miner, and Fodina algorithm, comparison of process discovery results using Fitness, Precision, and F-Measure concluding the results of these comparisons (Fig 2).

A. Process of Generating an Event Log

Many applications can be used in building a business process, such as Enterprise Resource Planning (ERP) [23], YAWL [24], and ProcessMaker [25]. This study will be using ProcessMaker as a tool to make a business process. It is built by adding the required form. From these results, the business flow that has been built is executed to completion. The process will be automatically recorded by the system, which will later be carried out using a query data retrieval process (TABLE V). The query is a join process with the BPMN event table to be able to retrieve messages from a series of procedures carried out.

BPMN is made by using two business process, namely flow of process and flow of document. The two business process will be connected using messages sent from the flow of process or vice versa. The sending of the message will be recorded, which will be used as an activity in the event log.

B. Process Discovery

At this stage, the process carried out is to reveal the Invisible Task via previously introduced rules. The steps taken to insert the Invisible Task can be seen in the following illustration. In the first stage of the event log obtained (TABLE II), a mapping is carried out using a matrix footprint (TABLE III). in the footprint matrix there is a → symbol which means delivery, the ← sign means receiving, and the # sign means not related to each other. The matrix results are then used to form a path illustration to be used as an illustration of how the path is drawn (Fig 3). The path results are then carried out by mapping the Petri net path to determine which process relationships contain the Invisible Task (TABLE IV). These results are obtained from the path

illustrative process in the previous stage. When it is known which activity relationship has an Invisible Task, then a new activity called an Invisible Task (IT) is inserted (Fig 4).

Based on the rules, this research modifies the existing event log data by adding a new activity, which we call IT. The activity is inserted into the event log as a type of Skip Invisible Task, because there is a trace in the event log that passes the actual path. In the event log that you have, some activities are skipped. The event log modification results will then be executed using the Alpha ++ algorithm to show Non-Free Choice in the event log data.

TABLE II. EVENT LOG FOR INVISIBLE TASK IDENTIFICATION

CaseID	Activity	Time	CaseID	Activity	Time
P001	A	08:20	P002	A	08:32
P001	B	08:22	P002	C	08:36
P001	C	08:23	P002	D	08:39
P001	D	08:25	P002	F	08:41
P001	E	08:26	P002	G	08:44
P001	G	08:29			

TABLE III. MATRIX FOOTPRINT

	A	**B**	**C**	**D**	**E**	**F**	**G**
A	#	→	→	#	#	#	#
B	←	#	→	#	#	#	#
C	←	←	#	→	#	#	#
D	#	#	←	#	→	→	#
E	#	#	#	←	#	#	→
F	#	#	#	←	#	#	→
G	#	#	#	#	←	←	#

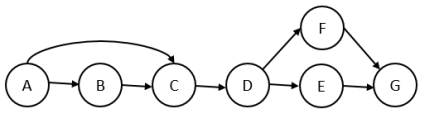

Fig 3. Path Illustration

TABLE IV. MAPPING FOR CHECKING INVISIBLE TASK

	From	**To**	**Petri net**
P1	{A}	{B}	A → B
P2	{B}	{C}	B → C
P3	{C}	{D}	C → D
P4	{D}	{E}	D → E
P5	{E}	{G}	E → G
P6	{D}	{F}	D → F
P7	{F}	{G}	F → G
P8	{A}	{ }	A →
P9	{ }	{C}	→ C

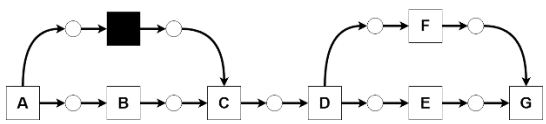

Fig 4. Result of Invisible Task

The addition of activity (IT) is based on the type of skip Invisible Task. In the generated event log, the "Put Waste in Purple Bin" activity passes the "Shed Waste" activity. However, the other four activities immediately executed the "Send Waste to Inecerator" activity. Based on these facts, new activities were then inserted to mark the Invisible Task.

C. Compare Process Discovery Result

There are several aspects that can be done to measure the quality of a process model. In this study, using three quality measurement parameters, namely Fitness (Eq. 1), Precision (Eq. 2), and F-Measure (Eq. 4). Before we get the F-Measure value, first we do the calculation the value of Recall (3). Fitness measure calculates how many processes can be described in a model. Meanwhile, Precision measure calculates the number of traces formed compared to the recorded process model.

$$\text{Fitness(k)} = \frac{n_{cim}}{n_{cil}} \quad (1)$$

where:

n_{cim} = Obtained from the total trace of the event log that was successfully drawn in the model

n_{cil} = Obtained from the total trace contained in the event log

$$\text{Precision(k)} = \frac{n_{tcil}}{n_{tom}} \quad (2)$$

where:

n_{tcil} = The total trace which in the event log drawn into the model

n_{tom} = The total trace that can be modeled

$$\text{Recall(k)} = \frac{n_{tcil}}{n_{tomc}} \quad (3)$$

where:

n_{tcil} = The total trace which in the event log drawn into the model

n_{tomc} = The total trace depicted in the model

$$\text{F} - \text{Score(k)} = \frac{2(\text{Recall} \times \text{Precision})}{(\text{Recall} + \text{Precision})} \quad (4)$$

where:

Recall = The result value of recall measurement

Precision = The result vakue of Precision measurement

IV. RESULT AND DISCUSSION

A. Data Event Log

The event log used is the result of extraction from the BPMN model that has been designed using the ProcessMaker. The data used in this paper is synthetic. The process of retrieving event log data is carried out by running a query contained in TABLE V. Fig 5 is a BPMN model that has been built through the ProcessMaker. There are two business process, namely flow of waste and flow of document. The two business process then communicate via messages sent. The message sending process is recorded by the system for further use in the process discovery.

The event log generated in the process will then be carried out process discovery. There seven traces are generated. There is a message sending between the flow of waste and the flow of documents in each trace. The message is an entity that connects the flow of waste with the flow of document. It is saved into a new activity that has the same Case Id in the trace. TABLE VI is the event log that was successfully generated through the process mentioned earlier. The results of the event log will be process discovery using predetermined methods. The process discovery results will be compared by measuring the Fitness, Precision, and F-Measure value.

TABLE V. SQL QUERY TO TAKE AN EVENT LOG

No.	SQL Query
1	SELECT IF(bpmn_lane.LAN_NAME IS NULL, (
2	SELECT LAN_NAME FROM bpmn_lane
3	INNER JOIN bpmn_bound ON bpmn_lane.LAN_UID = bpmn_bound.BOU_ELEMENT
4	INNER JOIN app_cache_view ON bpmn_bound.ELEMENT_UID = app_cache_view.TAS_UID
5	WHERE app_cache_view.DEL_FINISH_DATE = message_application.MSGAPP_THROW_DATE
6	), bpmn_lane.LAN_NAME) as ProcessName,
7	message_application.APP_UID as CaseID,
8	message_application.EVN_UID_THROW as Activity,
9	message_application.MSGAPP_THROW_DATE as Timestamp,
10	IF(users.USR_FIRSTNAME IS NULL, (
11	SELECT USR_FIRSTNAME FROM users
12	INNER JOIN app_cache_view ON users.USR_UID = app_cache_view.PREVIOUS_USR_UID
13	WHERE app_cache_view.DEL_FINISH_DATE = message_application.MSGAPP_THROW_DATE
14	LIMIT 1
15	),
16	users.USR_FIRSTNAME) AS Originator,
17	(SELECT task.TAS_TITLE FROM task INNER JOIN app_history ON task.TAS_UID = app_history.TAS_UID WHERE app_history.HISTORY_DATE > message_application.MSGAPP_THROW_DATE limit 1) As MessageTo
18	FROM message_application
19	INNER JOIN bpmn_event ON message_application.EVN_UID_THROW = bpmn_event.EVN_UID
20	INNER JOIN app_cache_view ON
21	(IF (bpmn_event.EVN_TYPE = 'INTERMEDIATE', message_application.MSGAPP_THROW_DATE = app_cache_view.DEL_DELEGATE_DATE, message_application.MSGAPP_THROW_DATE = app_cache_view.APP_FINISH_DATE))
22	INNER JOIN task ON app_cache_view.TAS_UID = task.TAS_UID
23	LEFT JOIN bpmn_bound ON task.TAS_UID = bpmn_bound.ELEMENT_UID
24	LEFT JOIN bpmn_lane ON bpmn_bound.BOU_ELEMENT = bpmn_lane.LAN_UID
25	LEFT JOIN users ON app_cache_view.USR_UID = users.USR_UID
26	GROUP BY Timestamp
27	UNION ALL
28	SELECT bpmn_lane.LAN_NAME as ProcessName, app_history.APP_UID as CaseID, task.TAS_TITLE AS Activity,
29	app_history.HISTORY_DATE AS Timestamp, users.USR_FIRSTNAME AS Originator, NULL as MessageTo
30	FROM bpmn_lane
31	INNER JOIN bpmn_bound ON bpmn_lane.LAN_UID = bpmn_bound.BOU_ELEMENT
32	INNER JOIN task ON bpmn_bound.ELEMENT_UID = task.TAS_UID
33	INNER JOIN app_history ON app_history.TAS_UID = task.TAS_UID
34	INNER JOIN users ON app_history.USR_UID = users.USR_UID
35	GROUP BY app_history.APP_UID, app_history.DEL_INDEX
36	ORDER BY Timestamp

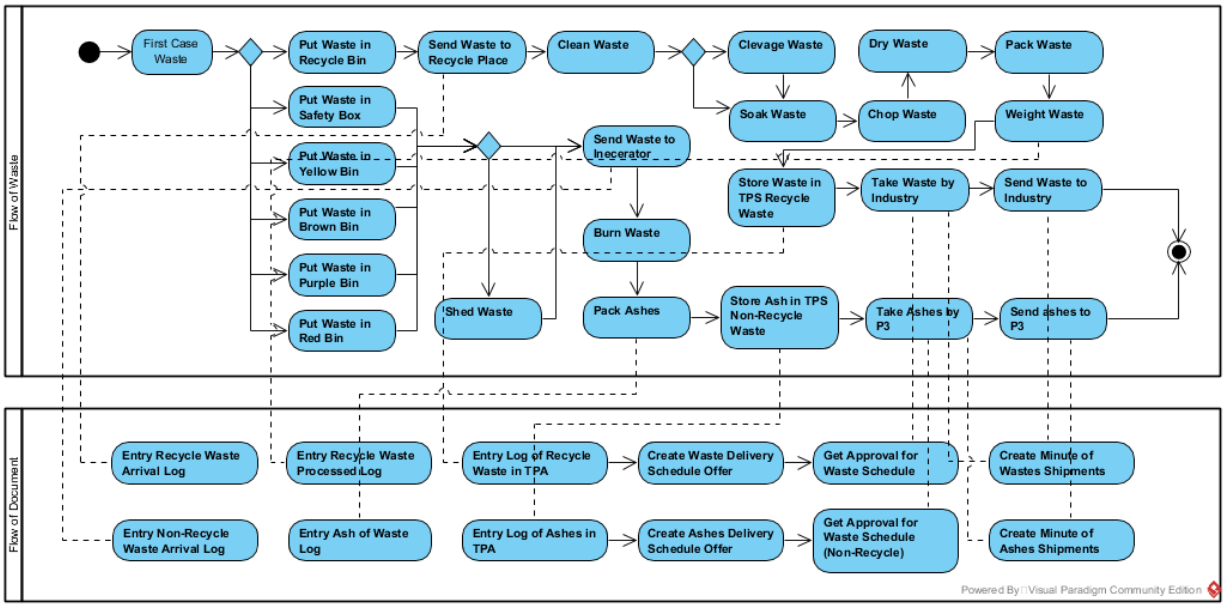

Fig 5. Model for Process Discovery

TABLE VI. EVENT LOG

CaseId	ProcessName	Activity	Timestamp	Originator	MessageTo
P001	Flow of Waste	First Case Waste	01/12/2020 05.09	MOHAMMAD	
P001	Flow of Waste	Put Waste in Recycle Bin	01/12/2020 05.10	MOHAMMAD	
P001	Flow of Waste	Send Waste to Recycle Place	01/12/2020 05.11	MOHAMMAD	
P001	Flow of Waste	5193167885fa96ae83a0e48049183827	01/12/2020 05.12	MOHAMMAD	Clean Waste
P001	Flow of Document	6753772595fb0cf2a8a05f6066126316	01/12/2020 05.28	NAUFAL	Send Waste to Industry
P001	Flow of Waste	Send Waste to Industry	01/12/2020 05.29	MOHAMMAD	

B. Evaluation

At this stage, a comparison is made of the event log results generated in the previous process. The evaluation process is carried out on event log data that can capture messages from the system compared to event log data that has been modified based on the proposed method. Process discovery is carried out using two algorithms, namely Modifed Alpha ++ and inductive miner. The results of the process discovery , then the comparison will be carried out. In this comparison process, a comparison is made with three measuring methods. The three measuring methods include Fitness, Precision, and F-Measure.

TABLE VII. CODE FOR RESULT PROCESS DISCOVERY

Code	Activity	Type
A	First Case Waste	Activity
B	Put Waste in Recycle Bin	Activity
C	Put Waste in Purple Bin	Activity
D	Put Waste in Safety Box	Activity
E	Put Waste in Yellow Bin	Activity
F	Put Waste in Brown Bin	Activity
G	Put Waste in Red Bin	Activity
H	Send Waste to Recycle Place	Activity
I	5193167885fa96ae83a0e48049183827	Message
J	Clean Waste	Activity
K	Entry Recycle Waste Arrival Log	Activity
L	Cleavage Waste	Activity
M	Soak Waste	Activity
N	Chop Waste	Activity
O	Dry Waste	Activity
P	Pack Waste	Activity
Q	Weight Waste	Activity
R	Entry Recycle Waste Processed Log	Activity
S	Store Waste in TPS Recycle Waste	Activity
T	Entry Log of Recycle Waste in TPA	Activity
U	Create Waste Delivery Schedule Offer	Activity
V	Get Approval for Waste Schedule	Activity
W	Take Waste by Industry	Activity
X	Create Minute of Wastes Shipments	Activity
Y	6753772595fb0cf2a8a05f6066126316	Message
Z	Send Waste to Industry	Activity
AA	Shed Waste	Activity
AB	Send Waste to Incenerator	Activity
AC	2113291195fa97aa6e53cc7019499914	Message
AD	Entry Non-Recycle Waste Arrival Log	Activity
AE	Burn Waste	Activity
AF	Pack Ashes	Activity
AG	2968385505fa97c54220574061868579	Message
AH	Entry Ash of Waste Log	Activity
AI	Store Ashes in TPS Non-Recycle Waste	Activity
AJ	Entry Log of Ashes in TPA	Activity
AK	Create Ashes Delivery Schedule Offer	Activity
AL	Get Approval for Waste Schedule (Non-Recycle)	Activity
AM	7702926755fb0d043edb183094239109	Message
AN	Take Ashes by P3	Activity
AO	Create Minutes of Ashes Shipments	Activity
AP	7323430325fb0d0e3f0e1f9083199339	Message
AQ	Send Ashes to P3	Activity
IT	Invisible Task	Invisible Task

TABLE VIII. THE RESULT OF EVALUATION

Algorithm	Fitness	Precision	F-Measure
Alpha ++	1	0.250	0.387
Inductive Miner	1	0.051	0.095
Fodina	1	0.115	0.203
Modified Alpha ++	1	1	1

This research has been carried out using Alpha ++ and Inductive Miner methods on event logs that have not been edited and have not recorded messages and event logs that have been modified and have recorded and displayed messages. TABLE VII is an acronym for the process model that has been process discovery.

In Fig 6 is the result of discovering process using Alpha ++ method. The result of the event log that has not been recorded and does not display a message. The process discovery results show that there are 6 traces that can be drawn, 1 trace that cannot be drawn, 18 traces that appear in the discovery model but are not recorded in the event log.

In Fig 7 is the result of discovering process using Fodina method. The result of the event log that has not been recorded and does not display a message. The results of the process discovery that have been carried out show that 3 traces are drawn and recorded in the event log, 43 traces that appear in the discovery model but are not recorded in the event log, and 1 traces that do not appear in the discovery model but are in the event log data.

In Fig 8 is the result of discovering process using Inductive Miner method. The result of the event log that has not been recorded and does not display a message. The results of the process discovery that have been carried out show that 5 traces are drawn and recorded in the event log, 93 traces that appear in the discovery model but are not recorded in the event log, and 2 traces that do not appear in the discovery model but are in the event log data.

In Fig 9 is the result of discovering process using Modified Alpha ++ method. The result of the process discovery carried out on event log data, which can record business process collaboration and has been modified to display an IT-in-NFC using Alpha ++. The results show that the 7 traces in the event log can be shown in the discovery model properly and no other traces appear that are not recorded in the event log used.

The results of the four process discovery es are then carried out by measuring the Fitness value, Precision and F-Measurement (

TABLE *VIII*). In the first process discovery (Fig 6) the Fitness value obtained is 1, with a Precision value of 0.25, and an F-Measure value of 0.387. In the second process discovery (Fig 8), a Fitness value of 1 is obtained, with a Precision value of 0.051, and an F-Measure value of 0.095. In the third process discovery (Fig 7), the Fitness value is 1, with a Precision value of 0.057, and the F-Measure value is 0.107. In the last process discovery (Fig 9), the Fitness value is 1, with a Precision value of 1, and the F-Measure value is 1. Therefore, the Modified Alpha ++ algorithm is the best algorithm for drawing a collaboration business process.

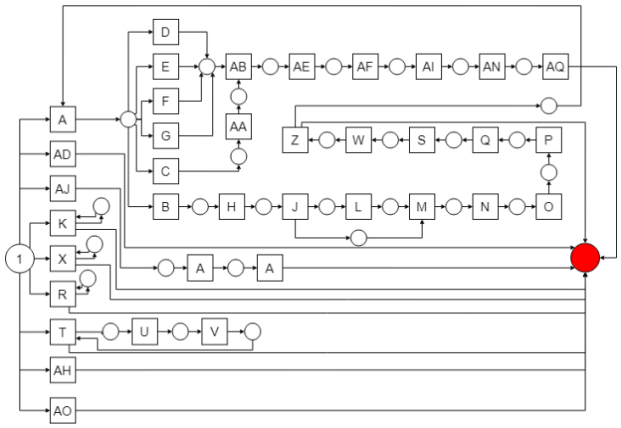

Fig 6. Obtained Process Model by using Alpha ++

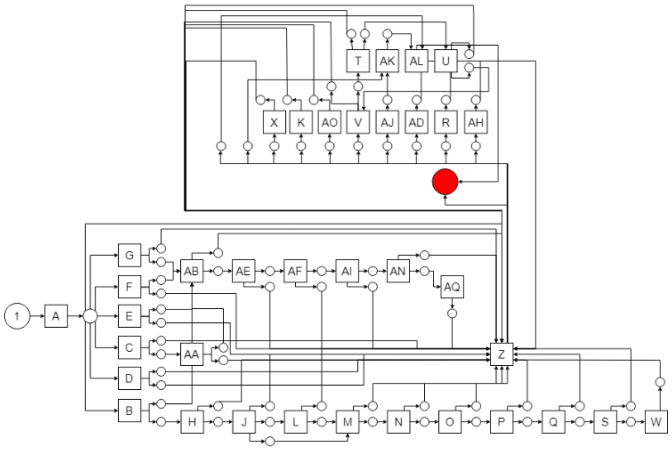

Fig 7. Obtained Process Model by using Fodina

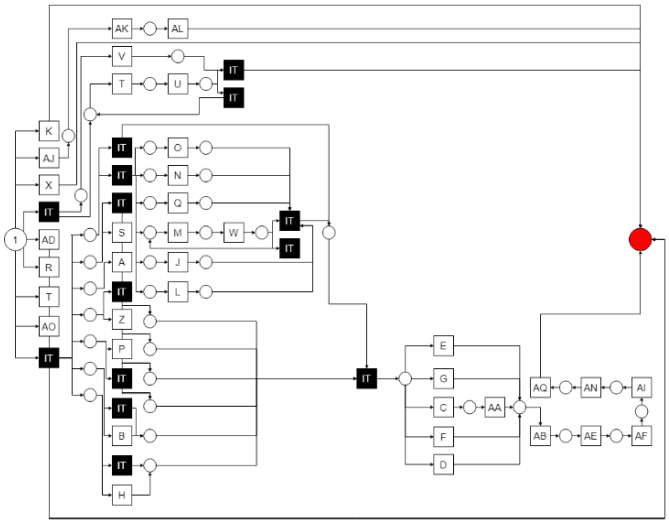

Fig 8. Obtained Process Model by using Inductive Miner

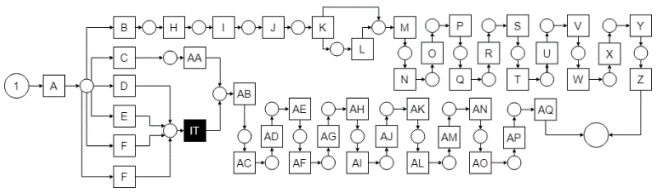

Fig 9. Obtained Process Model by using Modified Alpha ++

V. CONCLUSION

In this study, the emphasis is on collaborating business processes that previously only consisted of one business process into more than one business process. From the evaluation, all of methods produce 1 Fitness value. In addition, Modified Alpha ++ method has a Precision value of 1 and a F-Measure value of 1. On the other hand, Alpha ++, Inductive Miner and Fodina obtain 0.25, 0.051, and 0.051 Precision, respectively. Those methods also get less than 1 F-Measure, i.e., 0.387 for Alpha ++, 0.095 for Inductive Miner and 0.107 for Fodina. That happens because Modified Alpha ++ can display all messages recorded in the event log, while other comparison methods cannot display the recorded messages, so that it cannot describe the collaboration process in the two approaches. Therefore, a process model formed by Modified Alpha ++ is better than process models obtained by other process discovery algorithms.

For the future work, this research will expand Modified Alpha ++ algorithm, so this algorithm can separate business processes if more than one business process is obtained. The aim of separating business processes is showing collaboration of processes with more realistic.

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] W. van der Aalst, "Process mining: discovering and improving Spaghetti and Lasagna processes," in *2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, 2012, no. January, hal. 1–7.

[2] I. Waspada, R. Sarno, dan K. R. Sungkono, "An improved method of parallel model detection for graph-based process model discovery," *International Journal of Intelligent Engineering and Systems*, vol. 13, no. 2, hal. 127–139, 2020.

[3] W. Van Der Aalst, T. Weijters, dan L. Maruster, "Workflow mining: Discovering process models from event logs," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 9, hal. 1128–1142, 2004.

[4] L. Wen, J. Wang, W. M. P. Van Der Aalst, B. Huang, dan J. Sun, "Mining process models with prime invisible tasks," *Data and Knowledge Engineering*, vol. 69, no. 10, hal. 999–1021, 2010.

[5] L. Hakim, R. Sarno, dan 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, hal. 31–40, 2019.

[6] Y. A. Effendi dan R. Sarno, "Conformance Checking Evaluation of Process Discovery Using Modified Alpha ++ Miner Algorithm," in *2018 International Seminar on Application for Technology of Information and Communication (iSemantic) Conformance*, 2018, hal. 435–440.

[7] Y. U. A. M. E. Ffendi dan R. I. S. Arno, "Parallel Process Discovery Using A New Time-Based Alpha++ Miner," *IIUM Engineering Journal*, vol. 21, no. 1, hal. 126–141, 2020.

[8] S. J. J. Leemans, D. Fahland, dan W. Van Der Aalst, "Discovering Block-Structured Process Models from Event Logs - A Constructive Approach," in *International Conference on Applications and Theory of Petri Nets and Concurrency*, 2013, no. January.

[9] R. Adhim, M. A. Shiddiq, F. F. Ghizbunaza, dan M. A. Yaqin, "Process Discovery pada Event Log Permainan Hay Day menggunakan Algoritma Inductive Miner," *Seminar Nasional Inovasi dan Aplikasi Teknologi di Industri 2019*, no. February, hal. 66–73, 2019.

[10] R. Ghawi, "Process Discovery using Inductive Miner and Decomposition," 2016.

[11] A. J. M. M. Weijters, W. M. P. van der Aalst, dan A. K. A. de Medeiros;, "Process Mining with the Heuristics Miner Algorithm," *Beta working papers*, no. January 2006, 2006.

[12] A. J. M. M. Weijters dan J. Ribeiro, "Flexible Heuristics Miner (FHM)," in *2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, 2014.

[13] K. R. Sungkono, U. E. N. Rochmah, dan R. Sarno, "Heuristic Linear Temporal Logic Pattern Algorithm in Business Process Model," *International Journal of Intelligent Engineering and Systems*, vol. 12, no. 4, hal. 31–40, 2019.

[14] A. Augusto, R. Conforti, M. Dumas, dan M. La Rosa, "Split Miner : Discovering Accurate and Simple Business Process Models from Event Logs," in *2017 IEEE International Conference on Data Mining (ICDM)*, 2017.

[15] A. Augusto, R. Conforti, M. Dumas, dan M. La, "Split Miner : Automated Discovery of Accurate and Simple Business Process Models from Event Logs," *Knowledge and Information Systems*, vol. 59, hal. 251–284, 2018.

[16] C. W. Gunther dan W. Van Der Aalst, "Fuzzy Mining – Adaptive Process Simplification Based on Multi-perspective Metrics," in *International Conference on Business Process Management*, 2007, hal. 328–343.

[17] R. Sarno, K. R. Sungkono, R. Johanes, dan D. Sunaryono, "Graph-Based Algorithms for Discovering a Process Model Containing Invisible Tasks," *International Journal of Intelligent Engineering and Systems*, vol. 12, no. 2, hal. 85–94, 2019.

[18] A. J. M. M. Weijters dan W. Van Der Aalst, "The ProM Framework : A New Era in Process Mining Tool Support," in *International Conference on Application and Theory of Petri Nets*, 2005, hal. 444–454.

[19] L. Wen, W. M. P. Van Der Aalst, J. Wang, dan J. Sun, "Mining process models with non-free-choice constructs," *Data Mining and Knowledge Discovery*, vol. 15, no. 2, hal. 145–180, 2007.

[20] Q. Guo, L. Wen, J. Wang, Z. Yan, dan P. S. Yu, "Mining invisible tasks in non-free-choice constructs," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015, vol. 9253, hal. 109–125.

[21] R. Sarno dan K. Rossa, "A survey of graph-based algorithms for discovering business processes," *International Journal of Advances in Intelligent Informatics*, vol. 5, no. 2, hal. 137–149, 2019.

[22] S. K. L. M. vanden Broucke dan J. De Weerdt, "Fodina: A robust and flexible heuristic process discovery technique," *Decision Support Systems*, vol. 100, hal. 109–118, 2017.

[23] R. Sarno, H. Ginardi, E. W. Pamungkas, dan D. Sunaryono, "Clustering of ERP Business Process Fragments," in *2013 International Conference on Computer, Control, Informatics and Its Applications*, 2013, hal. 319–324.

[24] W. Van Der Aalst dan A. Ter, "YAWL : Yet Another Workflow Language," *Information Systems*, vol. 30, no. 4, hal. 245–275, 2005.

[25] D. Majekodunmi, *The ProcessMaker Workflow Designer*. Berkeley, CA: Apress, 2018.