Identifying Bottlenecks of Hierarchical Process Model by Using Discrete Event Simulation and Process Mining

Abdullah Faqih Septiyanto
Department of Informatics Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
abdullahfaqih161@gmail.com

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

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

2022 IEEE 8th International Conference on Computing, Engineering and Design (ICCED) | 978-1-6654-5389-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICCED56140.2022.10010450

***Abstract*— Identification of bottlenecks is a challenge for researchers to detect and deal with as soon as possible so as not to affect the implementation of business processes in the actual situation. The complexity of business processes can make it difficult for analysts to identify bottlenecks in a business process. Process mining is a technique to detect bottlenecks using event logs. However, many studies still use event logs from the system resource activities. Therefore, identification cannot be made before implementation. The researcher proposes bottleneck detection using the Discrete Event Simulation (DES) hierarchical process model. The hierarchical process model is divided into Top-level and Bottom-level process models. DES is used to simulate a hierarchical process model that can generate event logs. The event log was analyzed using ProM with the Inductive Miner method and performance analysis. This research indicates that activities that occur bottlenecks at the Top-level can represent a group of activities that occur bottlenecks at the Bottom-level. Therefore, the bottleneck identification results in the hierarchical process model can provide an overview of bottlenecks in general (Top-level) and specifically (Bottom-level).**

Keywords—Bottleneck, Business Process, Discrete Event Simulation, Inductive Miner, Process Mining

I. INTRODUCTION

The development of business processes is rapid and complex, following the company's growth in running a business to achieve its own goals and expected goals by collaborating between companies [1]. One of the problems that often occur in applying business processes is the occurrence of bottlenecks. Identifying bottlenecks is critical to detect whether business processes can run by the Standard Operation Procedure (SOP). Company managers can evaluate the business processes used and handle which activities are bottlenecks to increase system productivity [2].

One technique to detect business process problems or bottlenecks is process mining [3]. Process mining is extracting business process knowledge using event logs [4]. The event log is a list of workflows that occur within the company, consisting of the unique Case ID code of one process, the Activity that was executed, the execution time of the activity (Timestamp), and the Agent who performed the activity (Resource) [5]. Process mining is applied using ProM software. ProM is an open-source framework used by researchers for process mining [6].

Previous research about identification bottlenecks [7][8] used ProM to detect bottlenecks overcome with various solutions by adjusting the analyzed case studies. Another case conducted by [9], was the identification of bottlenecks in the agile development business process. Based on this research, bottleneck analysis is still carried out on one overall process model. The weakness of the analysis is that it is difficult to see where the bottleneck problem is broadly in a process with a lot of activity.

Previous research about hierarchical models [10], applied a hierarchical mining process to find process models based on event logs from several distributed departmental sources. Another research [6] used a hierarchical process model to detect bottlenecks in the process model by making Top-level (TL) and Bottom-level (BL) event logs. The following research [11] proposes a causal hierarchical approach to capture two levels of causal maps, namely the general level and the subsystem level. Based on previous research, applying the hierarchical process model still uses event logs from resource activities. The weakness of using logs from running resource activities is that bottleneck detection is only analyzed when the process has been run. Thus, there is trial and error with activity actors (resources) to detect bottlenecks.

This research proposed identify bottlenecks in the Hierarchical Process Model (HPM) using Discrete Event Simulation (DES) and process mining. Waste Handling SOP modeling using DES is carried out to be simulated according to the actual situation. Therefore, analysts can perform various process model options that can be simulated to see the possibility of bottlenecks. This can help analysts not trial and error with activity actors (resources) to detect bottlenecks. They are next, modeling HPM from the Waste Handling SOP. The hierarchical level of the process model is divided into TL and BL. Both process models are described in DES and simulated to generate BL event log. Based on the event log, the time setting is obtained that will be used in TL modeling in DES to generate TL event log. Process mining is carried out based on the two event logs to identify bottlenecks in each event log using Inductive Miner and performance analysis. Process mining results at the TL and BL are analyzed to see where the bottleneck problems are in general and specifically.

II. PRELIMINARY

A. Hierarchical Process Model (HPM)

The Hierarchical Process Model (HPM) is a process model in which several levels have different details [12]. In some instances, the process model analyst must decompose the model into several submodels [12]. Thus, the model can be analyzed in general and specifically. HPM collection discovery technique [13] is shown to find processes at different levels that exhibit the relationship between parent and child processes. The child process is a detail of the parent process [14].

B. Discrete Event Simulation (DES)

Discrete Event Simulation (DES) is a system modeling as a sequence of discrete events [15] that can comply with the Standard Operation Procedure (SOP) within a company by meeting boundaries, rules, and managerial decisions [16]. The DES model can be built using one software, namely AnyLogic. In AnyLogic, several elements are often used to construct a DES model: source, service, resource pool, select output, and sink. One of the advantages of DES is that it can easily and practically adapt to the real world and can handle the complexity and dependencies of several components involved in the system [17]. DES is a popular simulation technique often used by many researchers in modeling systems for production planning, supply management, and maintenance [18]. Angel [17], simulation can provide many conditions and improvements that are difficult to implement in actual experiments. Analysts can apply various system handling conditions flexibly and identify their impact without causing system failure [19] or knowing that the system needs upgrading [20].

C. Bottleneck Analysis

One technique to identify bottlenecks is by using process mining. The process mining results can provide congestion, decisions, and resource use [21]. An activity is bottlenecked if it has a longer waiting time than other business processes [8]. Process mining can be implemented using ProM. ProM has been widely used to identify bottlenecks in event logs. However, various methods are used to detect and handle bottlenecks in several studies, according to the case studies analyzed [22][23]. Based on previous research, a bottleneck is an important thing that must be seen and handled immediately to avoid the system not running according to the predetermined time estimate.

III. METHODOLOGY

The method proposed in this study is to identify bottlenecks in the SOP Hierarchical Process Model of Waste Handling simulated using DES. The flow of the proposed method is depicted in Fig. 2.

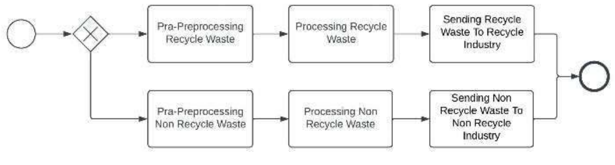

Fig. 1. Top-level Hierarchical Process Model of Waste Handling

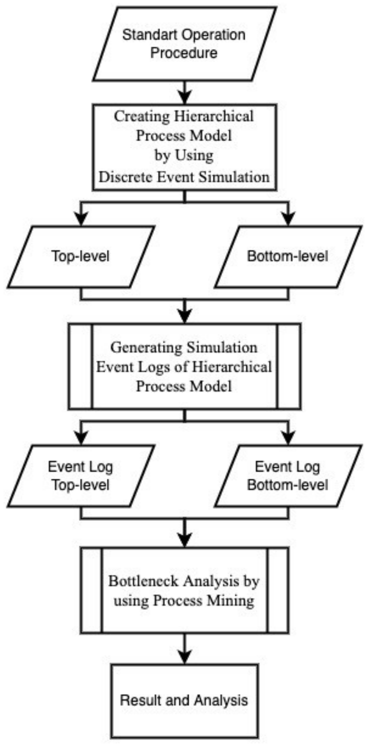

Fig. 2. Proposed Method

A. Creating Hierarchical Process Model by using Discrete Event Simulation

At this stage, the SOP Hierarchical Process Model of Waste Handling is made into a TL and BL process model using BPMN. Based on the Waste Handling SOP, the TL process model consists of 6 submodels, namely Pra-Preprocessing Recycle Waste, Processing Recycle Waste, Sending Recycle Waste To Recycle Industry, Pra-Preprocessing Non Recycle Waste, Processing Non Recycle Waste, Sending Non Recycle Waste To Non Recycle Industry. TL process model is depicted in Fig. 1.

The details of the activities of each submodel are described as follows:

1. Pre-Preprocessing Recycle Waste

This submodel describes the flow of activities before the processing of recycled waste. The process starts with the stakeholders placing the waste in the waste bin, sending the waste to the recycling area, and inputting the waste delivery time. The Pre-Preprocessing Recycle Waste submodel is depicted in Fig. 3

2. Processing Recycle Waste

This submodel describes the flow of activities before processing non-recyclable waste. The process starts from the stakeholders placing the waste in a non-recyclable waste bin, sending non-recycled waste to a waste incinerator, and inputting the delivery time for non-recycled waste. The submodel is depicted in Fig. 4.

3. Sending Recycle Waste To Recycle Industry

This submodel describes the flow of the waste recycling process activities. The process starts from cleaning the waste, dividing the waste, which is only carried out on the waste that requires the activity, drying the waste, packing the waste, and entering the waste processing log. The submodel is depicted in Fig. 5

4. Pre-Preprocessing Non Recycle Waste

This submodel describes the flow of non-recycling waste processing activities. The process starts from burning waste, packing burning ash, and inputting ash processing logs. The submodel is depicted in Fig. 6.

5. Processing Non Recycle Waste

This submodel describes the flow of sending recycled waste to the recycling industry. The process starts from taking the waste by the industry, inputting the delivery time, and sending the waste to the recycling industry. The submodel is depicted in Fig. 7.

6. Sending Recycle Waste To Non Recycle Industry

This submodel describes the flow of non-recycled waste delivery activities to non-recycled industries. The process is almost the same as Fig. 7, but the process is carried out for non recycle waste. The submodel is depicted in Fig. 8.

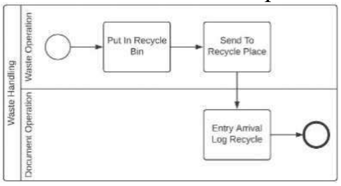

Fig. 3. Submodel Pra-Preprocessing Recycle Waste

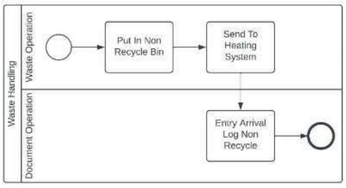

Fig. 4. Submodel Pra-Preprocessing Non Recycle Waste

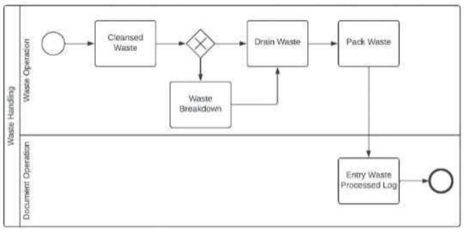

Fig. 5. Submodel Processing Recycle Waste

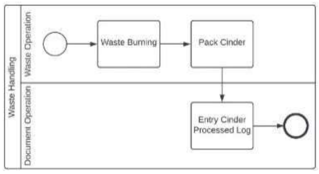

Fig. 6. Submodel Processing Non Recycle Waste

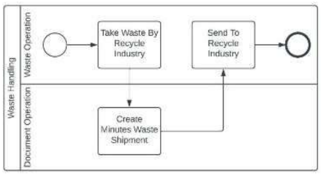

Fig. 7. Submodel Sending Recycle Waste To Recycle Industry

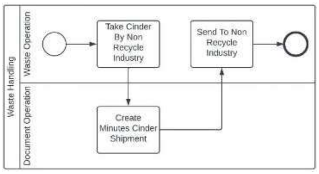

Fig. 8. Submodel Sending Non Recycle Waste To Non Recycle Industry

Based on the SOP Hierarchical Process Model of Waste Handling, the TL and BL process models are described using DES therefore that they can be simulated to generate event logs. The TL DES is illustrated in Fig. 9, and BL DES is depicted in Fig. 10.

In Fig. 9, each submodel activity is described using the service in DES, while in Fig. 9, service on DES represents each submodel. The DES model was created using AnyLogic software. Both DES models are simulated to generate event logs for each model. Before being simulated, all services are given an estimated processing time of each data using triangular. Triangular is the time format AnyLogic to determine the minimum, maximum, and most likely time in a triangular structure (min, most, max). In BL DES, each service is given an estimated time according to the Waste Handling SOP. Details of time estimation for BL DES are depicted in Table I.

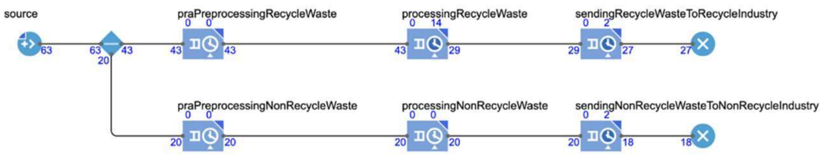

Fig. 9. DES of Top-Level Waste Handling

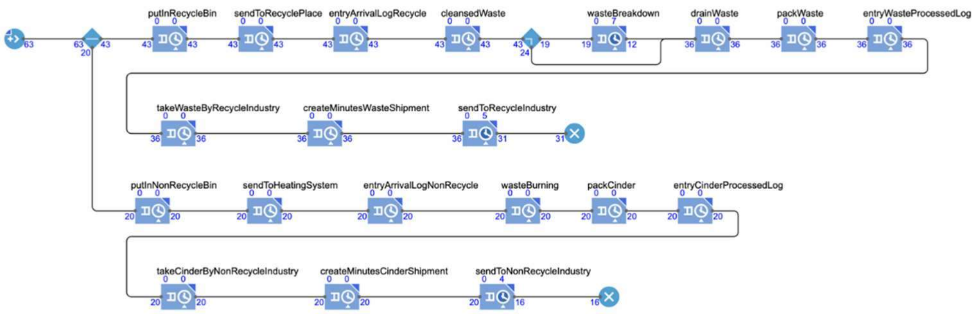

Fig. 10. DES of Bottom-Level Waste Handling

TABLE I. DETAIL DES OF BOTTOM-LEVEL

Service	T/M
putInRecycleBin	(30, 60, 90)
sendToRecyclePlace	(0.5, 1, 1.5)
entryArrivalLogRecycle	(0.5, 1, 1.5)
cleansedWaste	(0.5, 1, 1.5)
wasteBreakdown	(480, 720, 960)
drainWaste	(0.5, 1, 1.5)
packWaste	(0.5, 1, 1.5)
entryWasteProcessedLog	(0.5, 1, 1.5)
takeWasteByRecycleIndustry	(0.5, 1, 1.5)
createMinutesWasteShipment	(30, 60, 90)
sendByRecycleIndustry	(30, 60, 90)
putInNonRecycleBin	(30, 60, 90)
sendToHeatingSystem	(5, 10, 15)
entryArrivalLogNonRecycle	(0.5, 1, 1.5)
wasteBurning	(0.5, 1, 1.5)
packCinder	(4, 5, 7)
entryCinderProcessedLog	(0.5, 1, 1.5)
takeCinderByNonRecycleIndustry	(5, 8, 12)
createMinutesCinderShipment	(0.5, 1, 1.5)
sendToNonRecycleIndustry	(30, 60, 90)

where:
T/M = Triangular/Minutes

In DES TL, to provide an estimate of service time requires event logs from DES BL simulation results. Because the TL is a grouping of each service, the minimum, maximum, and most likely time calculations are carried out for each activity in the BL event log. The pseudocode to calculate the estimated time value of the TL DES service is described in Table II.

TABLE II. PSEUDOCODE SET TRIANGULAR VALUE DES TOP-LEVEL

Pseudocode Triangular Value DES Top Level

```
exportLogDESBottomLevel()
listTopLevel = getListActivityTopLevel()
db = importLogtoDatabase()
add new column = level
eventLog = getListDatalog()
for eventLog
    if eventLog.block in listTopLevel
        update eventLog.level = listToplevel.activity
    endif
endfor
for eventLog
    listRange = eventLog.stopdate - eventLog.startdate
endfor
most = avg(listRange)
max = max(listRange)
min = min(listRange)
```

Table II shows the pseudocode to get the minimum, maximum, and most likely time for each submodel from the BL DES simulation event log. Log data from the simulation results are imported into the database, and data processing is carried out using Standard Query Language (SQL). Therefore, the minimum, maximum, and most likely values for each submodel are depicted in Table III.

TABLE III. DETAIL DES OF TOP-LEVEL

Service	T/M
praPreprocessingRecycleWaste	(0.5, 21, 87)
processingRecycleWaste	(0.5, 70, 930)
sendingRecycleWasteToRecycleIndustry	(0.5, 20.5, 82)
praPreprocessingNonRecycleWaste	(0.5, 24, 89)
processingNonRecycleWaste	(0.5, 2.5, 7)
sendingNonRecycleWasteToNonRecycleIndustry	(0.6, 22.5, 82)

Table III shows a list of time estimates used in the TL DES model for each service representing each submodel. Furthermore, the simulation is carried out to generate an event log.

B. Generating Simulation Event Logs of Hierarchical Process Model

At this stage, the event log is obtained from the simulation results of each TL and BL DES. Event log retrieval is used to be analyzed using process mining. Event log retrieval using the pseudocode described in Table IV.

Table IV shows the pseudocode for processing logs from TL and BL DES simulations to be analyzed using process mining. The pseudocode uses the sort by agent and startdate. The log data from the DES simulation results can be sorted by agent and processing start time from the service. Therefore, the process flow is by the SOP of Waste Handling. The result of the event log exported to a CSV file at the BL is depicted in Table V, and the TL is depicted in Table VI.

TABLE IV. PSEUDOCODE EXPORT EVENT LOG FROM DES

Pseudocode Export Event Log DES

```
simulationDES()
eventLog = export table flowchart_process_states_log
importLogtoDatabase()
Sort by agent asc
Sort by start_date asc
Export result to csv
```

TABLE V. EVENT LOG OF DES BOTTOM-LEVEL

Agent	Block	Start	Stop
<population>[0] : 1726	putInRecycleBin	20/03/20 07.00	20/03/20 08.07
<population>[0] : 1726	sendToRecyclePlace	20/03/20 08.07	20/03/20 08.08
<population>[0] : 1726	entryArrivalLogRecycle	20/03/20 08.08	20/03/20 08.09
<population>[0] : 1726	cleansedWaste	20/03/20 08.09	20/03/20 08.11
<population>[0] : 1726	wasteBreakdown	20/03/20 08.11	20/03/20 18.59
<population>[0] : 1726	drainWaste	20/03/20 18.59	20/03/20 19.00
<population>[0] : 1726	packWaste	20/03/20 19.00	20/03/20 19.01
<population>[0] : 1726	entryWasteProcessedLog	20/03/20 19.01	20/03/20 19.02
<population>[0] : 1726	takeWasteByRecycleIndustry	20/03/20 19.02	20/03/20 19.03
<population>[0] : 1726	createMinutesWasteShipment	20/03/20 19.03	20/03/20 19.04
<population>[0] : 1726	sendByRecycleIndustry	20/03/20 19.04	20/03/20 19.52

TABLE VI. EVENT LOG OF DES TOP-LEVEL

Agent	Block	Start	Stop
<population>[0] : 586	praPreprocessingRecycleWaste	20/03/20 07.00	20/03/20 07.47
<population>[0] : 586	processingRecycleWaste	20/03/20 07.47	20/03/20 10.46
<population>[0] : 586	sendingRecycleWasteToRecycleIndustry	20/03/20 10.46	20/03/20 10.51

<population >[0] : 610	praPreprocessingRecycleWaste	21/03/20 07.00	21/03/20 07.56
<population >[0] : 610	processingRecycleWaste	21/03/20 07.56	21/03/20 13.54

<population >[0] : 610	sendingRecycleWasteToRecyclelndustry	21/03/20 13.54	21/03/20 14.18

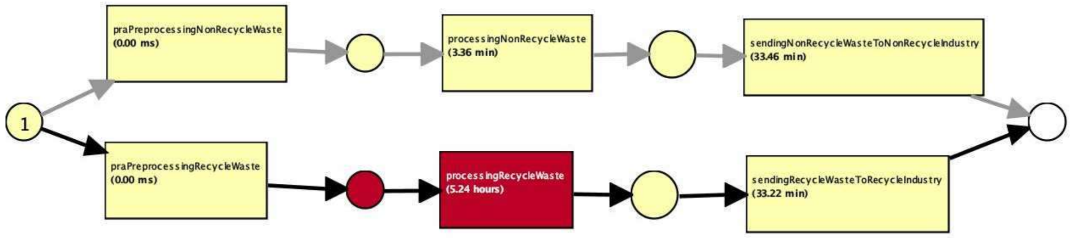

Fig. 11. Petri Net of Top-Level Process Model

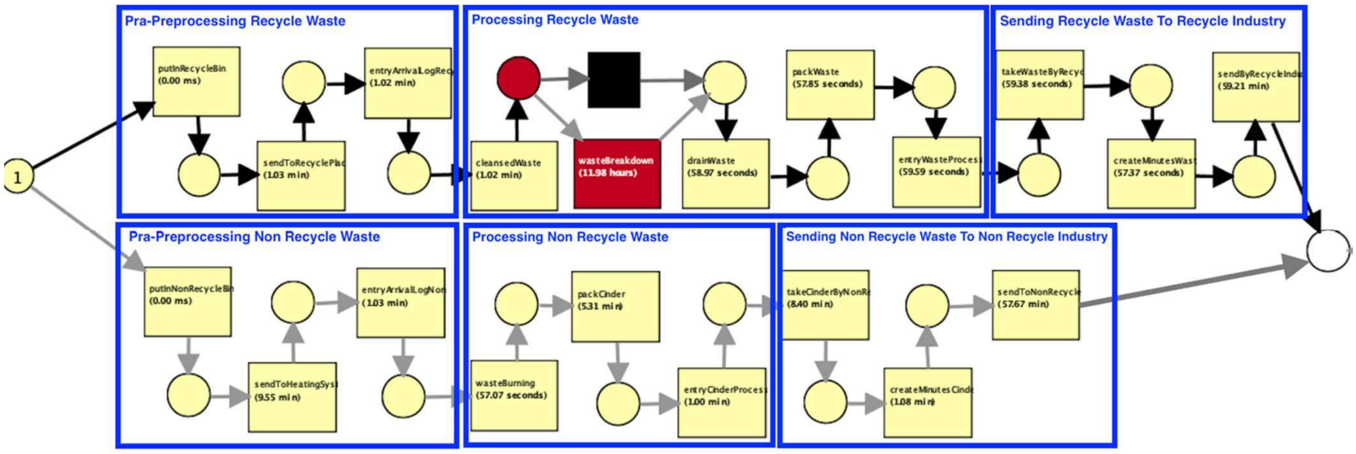

Fig. 12. Petri Net of Bottom-Level Process Model

C. *Bottleneck Analysis by using Process Mining*

Based on the results of the TL and BL event logs, then they are analyzed using process mining. Process mining is carried out to find out if there are bottlenecks in implementing SOP from Waste Handling. Process mining is done using ProM. Both event logs are imported into ProM and selected for CaseID, Activity, StartDate, and EndDate. Thus, the Petri Net can be described according to the event log. Table V and Table VI show column agents as CaseID, block as Activity, start as StartDate, and end as EndDate. Then process discovery is carried out using inductive mining and performance analysis to produce a Petri Net that can show bottlenecks from the TL and BL event log.

IV. RESULT AND DISCUSSION

A. *Result*

After the mining process using TL and BL event logs is carried out, the TL Petri Net results are obtained as illustrated in Fig. 11, and the BL illustrated in Fig. 12.

Fig. 11 shows the Petri Net of the TL process model. The results of the Petri net consist of 6 activities according to the total submodel formed from the SOP Hierarchical Process Model of Waste Handling. There are different colors for each activity. The color indicates the high waiting time for each activity. The higher the waiting time, the redder the color displayed, the lower the waiting time, the yellower the color of activity in the Petri Net. One activity has a high waiting time value, namely processingRecycleWaste with the red color, and another yellow activity, which indicates that the activity has a low waiting time value. Thus, there is a bottleneck in the processingRecycleWaste activity in the TL process model.

Fig. 12 shows the Petri Net of the BL process model. The results of the Petri Net consist of 20 activities that offer only one activity with a high waiting time value, namely wasteBreakdown. In contrast, other activities have a low waiting time value. Thus, there is a bottleneck in the wasteBreakdown activity in the BL process model.

B. *Analysis*

Based on the TL and BL Petri Net results, the two Petri Net have bottleneck activities. The bottleneck that occurs at the BL is that the wasteBreakdown activity is included in the processingWasteRecycle submodel, where there is a bottleneck in the processingWasteRecycle activity at the TL. Results show that if a bottleneck occurs at the BL, it can affect the Top -level. Bottlenecks at the BL also affect the results from the TL. Thus, identifying bottlenecks in the hierarchical process model can overview bottlenecks in general (TL) and specifically (BL).

V. CONCLUSION

In this research, the bottleneck identification in the hierarchical model process is divided into two levels of the process model, namely TL and BL. Researchers created a simulation model using DES to identify bottlenecks to analyze the possibilities that could cause a bottleneck to occur with various process model options that can be simulated and generate event logs to be analyzed using process mining. One of the mining process methods is performance analysis to identify bottlenecks in activities based on event logs. Therefore, one bottleneck activity is found at the TL and one activity at the BL. Based on these results, it shows that DES can model and simulate hierarchical process models and can generate event logs that can be analyzed using process mining. The results of the Petri net at the TL show that submodels or activities that occur bottlenecks can represent the occurrence of bottlenecks in activities in the BL. Thus, bottleneck identification in the hierarchical process model can show the location of the problem in outline or specifically.

Future work in this research results from bottleneck analysis, which continues to be the basis for optimization. Therefore, the optimization process can be carried out on activities that occur in bottlenecks. Because the results in this research only arrived at identifying the bottleneck.

ACKNOWLEDGMENT

This research was funded by Lembaga Pengelola Dana Pendidikan (LPDP) under Riset Inovatif-Produktif (RISPRO) Invitation Program, the Indonesian Ministry of Education and Culture under Penelitian Terapan Unggulan Perguruan Tinggi (PTUPT) Program, and Institut Teknologi Sepuluh Nopember (ITS) under project scheme of the Publication Writing and Intellectual Property Rights Incentive Program (PPHKI).

REFERENCES

[1] A. F. Septiyanto, R. Sarno, and K. R. Sungkono, "Mining Collaboration Business Process Containing Invisible Task by Using Modified Alpha," in *2021 13th International Conference on Information & Communication Technology and System (ICTS)*, Oct. 2021, pp. 90–95, doi: 10.1109/ICTS52701.2021.9608963.

[2] M. Subramaniyan, A. Skoogh, J. Bokrantz, M. A. Sheikh, M. Thürer, and Q. Chang, "Artificial intelligence for throughput bottleneck analysis – State-of-the-art and future directions," *Journal of Manufacturing Systems*, vol. 60, no. June, pp. 734–751, Jul. 2021, doi: 10.1016/j.jmsy.2021.07.021.

[3] M. Siek and R. M. G. Mukti, "Business process mining from e-commerce event web logs: Conformance checking and bottleneck identification," *IOP Conference Series: Earth and Environmental Science*, vol. 729, no. 1, p. 012133, Apr. 2021, doi: 10.1088/1755-1315/729/1/012133.

[4] W. Van Der Aalst, "Process Mining," *Commun. ACM*, vol. 55, no. 8, pp. 76–83, Aug. 2012, doi: 10.1145/2240236.2240257.

[5] D. R. Ferreira, F. Szimanski, and C. G. Ralha, "Mining the low-level behaviour of agents in high-level business processes," *International Journal of Business Process Integration and Management*, vol. 6, no. 2, p. 146, 2013, doi: 10.1504/IJBPIM.2013.054678.

[6] A. N. Rofiif, M. M. Wildan, K. R. Sungkono, R. Sarno, and C. S. Wahyuni, "Performance Analysis of Hierarchical Process Model," in *2019 International Seminar on Application for Technology of Information and Communication (iSemantic)*, Sep. 2019, pp. 171–177, doi: 10.1109/ISEMANTIC.2019.8884259.

[7] N. Raitubu, K. R. Sungkono, R. Sarno, and C. S. Wahyuni, "Detection of Bottleneck and Social Network in Business Process of Agile Development," in *2019 International Seminar on Application for Technology of Information and Communication (iSemantic)*, Sep. 2019, pp. 208–213, doi: 10.1109/ISEMANTIC.2019.8884341.

[8] Y. Caesarita, R. Sarno, and K. R. Sungkono, "Identifying bottlenecks and fraud of business process using alpha ++ and heuristic miner algorithms (Case study: CV. Wicaksana Artha)," in *2017 11th International Conference on Information & Communication Technology and System (ICTS)*, Oct. 2017, vol. 2018-Janua, pp. 143–148, doi: 10.1109/ICTS.2017.8265660.

[9] M. A. Razi, K. R. Sungkono, R. Sarno, and C. S. Wahyuni, "Detecting Bottleneck and Fraud in Agile Development by using Petri net Performance and Trace Clustering," in *2019 International Seminar on Application for Technology of Information and Communication (iSemantic)*, Sep. 2019, pp. 214–218, doi: 10.1109/ISEMANTIC.2019.8884226.

[10] R. Sarno and Y. A. Effendi, "Hierarchy Process Mining from Multi-Source Logs," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 15, no. 4, p. 1955, Dec. 2017, doi: 10.12928/telkomnika.v15i4.6326.

[11] R. Suresh, A. Sivaram, and V. Venkatasubramanian, "A hierarchical approach for causal modeling of process systems," *Computers & Chemical Engineering*, vol. 123, pp. 170–183, Apr. 2019, doi: 10.1016/j.compchemeng.2018.12.017.

[12] R. P. J. C. Bose, E. H. M. W. Verbeek, and W. M. P. Van Der Aalst, "Discovering hierarchical process models using ProM," in *Lecture Notes in Business Information Processing*, 2012, vol. 107 LNBIP, pp. 33–48, doi: 10.1007/978-3-642-29749-6_3.

[13] G. Greco, A. Guzzo, and L. Pontieri, "Mining taxonomies of process models," *Data and Knowledge Engineering*, vol. 67, no. 1, pp. 74–102, 2008, doi: 10.1016/j.datak.2008.06.010.

[14] R. Conforti, M. Dumas, L. García-Bañuelos, and M. La Rosa, "BPMN Miner: Automated discovery of BPMN process models with hierarchical structure," *Information Systems*, vol. 56, pp. 284–303, 2016, doi: 10.1016/j.is.2015.07.004.

[15] K. Agalianos, S. T. Ponis, E. Aretoulaki, G. Plakas, and O. Efthymiou, "Discrete event simulation and digital twins: Review and challenges for logistics," *Procedia Manufacturing*, vol. 51, no. 2019, pp. 1636–1641, 2020, doi: 10.1016/j.promfg.2020.10.228.

[16] M. H. Nili, H. Taghaddos, and B. Zahraie, "Integrating discrete event simulation and genetic algorithm optimization for bridge maintenance planning," *Automation in Construction*, vol. 122, no. December 2020, p. 103513, 2021, doi: 10.1016/j.autcon.2020.103513.

[17] A. Fiallos and X. Ochoa, "Discrete event simulation for student flow in academic study periods," *12th Latin American Conference on Learning Objects and Technologies, LACLO 2017*, vol. 2017-Janua, pp. 1–7, 2017, doi: 10.1109/LACLO.2017.8120908.

[18] M. Jahangirian, T. Eldabi, A. Naseer, L. K. Stergioulas, and T. Young, "Simulation in manufacturing and business: A review," *European Journal of Operational Research*, vol. 203, no. 1, pp. 1–13, 2010, doi: 10.1016/j.ejor.2009.06.004.

[19] A. Fajar and R. Sarno, "Optimization of time and cost of multi organization business processes in a port container terminal," *International Journal of Intelligent Engineering and Systems*, vol. 12, no. 6, pp. 259–271, 2019, doi: 10.22266/ijies2019.1231.25.

[20] A. Fajar and R. Sarno, "Asynchronous agent-based simulation and optimization of parallel business," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 17, no. 4, pp. 1731–1739, 2019, doi: 10.12928/TELKOMNIKA.v17i4.10846.

[21] W. van der Aalst, A. Adriansyah, and B. van Dongen, "Replaying history on process models for conformance checking and performance analysis," *WIREs Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 182–192, Mar. 2012, doi: 10.1002/widm.1045.

[22] H. Tang, "A new method of bottleneck analysis for manufacturing systems," *Manufacturing Letters*, vol. 19, pp. 21–24, Jan. 2019, doi: 10.1016/j.mfglet.2019.01.003.

[23] J. Tang, L. Li, Y. Liu, and K.-Y. Lin, "Automatic Identification of Bottleneck Tasks for Business Process Management using Fusion-based Text Clustering," *IFAC-PapersOnLine*, vol. 54, no. 1, pp. 1200–1205, 2021, doi: 10.1016/j.ifacol.2021.08.142.