Performance Analysis of Hierarchical Process Model

Ahmad Naufal Rofiif
Informatics Systems
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
lu.pikk99@gmail.com

Muhammad Wildan M.
Informatics Systems
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
maulidaniw@gmail.com

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

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

Cahyaningtyas Sekar Wahyuni
Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
cahyaningtyas.18051@mhs.its.ac.id

***Abstract*—Each organisation has a business process to achieve organisational goals by carrying out every activity in the business process. Each activity contains information about useful human resources, time, etc. Performance analysis is the process of evaluating the performance of a process model in the business process in comparison to the goal, which was to be achieved using an event log. Using performance analysis, the bottleneck that happened in a process model is discovered. The existing researches do performance analysis on a bottom-level process model. This research improves existing performance analysis by analysing the hierarchical process model, which includes the top-level process model and bottom-level process model. The top-level process model represents a process model that contains events as the representation of a sub-processes, a group of events in the bottom-level process model. Not only provides a performance analysis, but this research also proposes a method for generating an event log of the top-level process model and an event log of bottom-level process model as the input of performance analysis automatically. The proposed method defines the double timestamp of events of the top-level process model based on the timestamp of events of the bottom-level process model. The experiment shows that not all top-level events which contain bottom-level events. The event in the top-level process model is bottleneck if the amount of its bottom-level activities that are identified as bottleneck is higher than those that are not identified as bottleneck. It is because bottleneck is identified from the total of the waiting time and the waiting time of the activity in the top-level process model is the summation of all activity in the bottom-level process model.**

Keywords— hierarchical process model, performance analysis, top-level process model, bottom-level process model.

I. INTRODUCTION

The business process model is a representation of parts or functions that are directly related to the activities or processes that exist in the company [1]–[3]. Business process model present business activities, events, flow controls, stakeholders, and their relationships [4]. Each organization has a business process to achieve organizational goals by carrying out every activity in the business process. Every activity carried out in the business process has useful human resources, time, etc. However, only a few resources evaluated in the business process. In the case of business process, optimised business processes are essential to reduce costs [5]–[7]. The business process is usually used to measure the performance of a process that exists in the company by showing the problems that are related to the process [8]–[11].

A hierarchical process model is a process model that has several levels of different levels of detail [12]. A hierarchical process model can be created by combining the processes in each department in the company [5]. One activity in the hierarchical process model can represent another process that has a higher level of detail. A hierarchical process model is a process model where each event or activity contains another process [13]. These processes can be defined as sub-models in different graphs related to the activity of this process. The top-level process model represents the full process which is composed of several sub-processes that are interconnected with each other. The bottom-level process model describes the detailed process of each sub-process in the top level process model.

Performance Analysis is the process of evaluating the performance of a particular scenario compared to the organizational objectives to be achieved. In previous papers, there have been those that performed a performance analysis of process models [14]–[16]. But in these papers only perform performance analysis on the bottom-level process model. In this paper, we present a performance analysis on not only the bottom-level but also the top-level process model. Besides performing performance analysis in the bottom-level and top-level process model, we also build a hierarchical process model and proposing a method for creating the top-level event log automatically and bottom-level event log. Our proposed method generates a double timestamp of activities in the top-

978-1-7281-3832-9/19/$31.00 ©2019 IEEE

level event log based on a single timestamp of activities in the bottom-level event log.

The experiment of this paper shows how this paper creates an event log directly from the hierarchical process model and shows the results of the performance analysis based on the process model hierarchy.

II. Literature Review

A. Hierarchical Process Model

Process model or Process modelling is a method used in an organization to improve understanding of business processes owned by the organization and to deconstruct the complexity of the organization [17]. Process modelling is an approach to describe how a business is run and usually includes graphic depictions of at least activities, events or circumstances, and control of logic flow which is a business process [18]. A business process has all the information about activities that have been executed and can represent the business process correctly [5]. A hierarchical process model can be created by combining the processes in each department in the company, thus forming a long process consisting of many sub-models [5].

A hierarchical process model is a process model where each event or activity contains another process [13]. These processes can be defined as sub-models in different graphs related to the activity of this process. The sub model itself can represent other sub model events in the hierarchy [13], [19]. In Figure 1, we can see an example of a hierarchical process model. Figure 1 illustrates the hierarchical process model of Order Processing. At the top level, there are 3 activities of order processing which in the "Fill Orders" activity are sub-model 1 of this hierarchical process model. In sub-model one there are also 3 activities which in the "Invoice & Pgk Slip" activity are sub-model 2 of this hierarchical process model. In sub-model 2, this is the lowest level of the hierarchical process model, which also consists of 3 activities.

The hierarchical process model focuses on smaller and more manageable process activities. If an activity is still too complex, it can be divided into smaller activities until each model has a manageable size. This process reduces the risk of forgetting more complex model details and confusing interactions between models.

B. Event Log

An event log records the business processes of a company [11], [14]. An event log records business events from an information system that is run on a business process. Usually, event logs contain information about a process. Event logs can contain start time and end time of events, a sequence of events, people who have completed events, and processes that have those events. An event log could describe the events of occurring cases [6], [7]. The event log will contain any information; it depends on each organizational information [12]. By using event logs generated in information systems, actual business models can be seen more quickly.

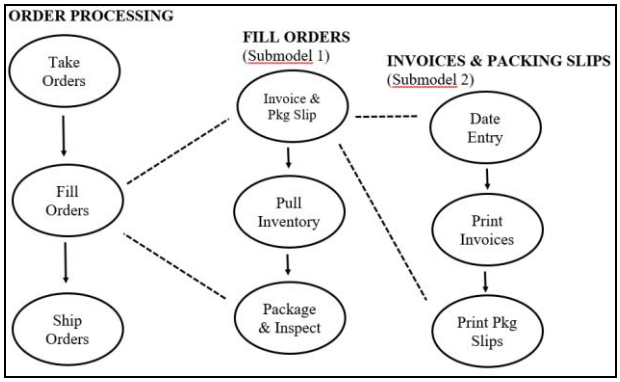

Fig. 1. Example of Hierarchical Process Model "Order Processing"

C. Performance Analysis

Performance Analysis is the process of evaluating the performance of a particular scenario compared to the organizational objectives to be achieved. Performance analysis uses the conformance checking results to measure the performance of a process. Through conformance checking, events in the log are coupled to elements in the model. This coupling allows for precise performance analysis and diagnostics. Based on the event log data linked to a model element, durations, service times, waiting times, resource usage, and more can be derived and evaluated [20]. Performance analysis aims to assess the performance of processes by using event logs [14].

By performing a performance analysis, from the event log, it can be known which process has a bottleneck, and more specifically on which events in the process are problematic. In the previous paper, performance analysis was only carried out at the bottom level process model. The result of a performance analysis from the previous paper was the identification of activities that experienced a bottleneck. To identify the bottleneck, the start time or timestamp is required for each activity in the event log. The disadvantage of performance analysis only on the bottom level is that we cannot see the bottleneck in each process. If the performance analysis is done on the top level and bottom level of the process model, besides knowing the bottleneck in each activity, we can also find out the bottleneck of each process in the top level process model.

III. Research Method

In this section, we will explain the research methodology or stages. Figure 2 shows the flowchart of the research methodology that we use. We begin the research by making a hierarchical process model based on the process in the hospital. Then, we create an algorithm to extract event logs of top-level process model and bottom-level process model automatically. Then, the last step is to perform Performance Analysis on the top level and bottom level event log using PROM.

A. Creating Hierarchical Process Model

In this stage, we make a hierarchical process model consisting of a top-level process model and bottom level process model using the Bitnami Process Maker. We create a

hierarchical process model based on the process model in the hospital. The top-level process model consists of 5 sub-models, namely: Registration, Emergency Services, Outpatient Services, Inpatient Services, Drugstore, and Payment. The top-level process model can be seen in Figure 2 and the top-level process is shown in Figure 5.

There are several bottom level process model, which are:

1. Sub-model “Registration”

This sub-model illustrates how the registration process at the hospital is carried out. Starting with checking the status of the patient whether it is an old or new patient, up to the selection of patient service. It can be seen in Figure 6.

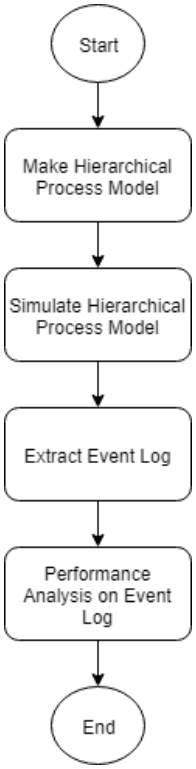

Fig. 2. Research Method Flowchart

2. Sub-model “Emergency Services”

This sub-model illustrates how the hospital handles patients in the emergency department. Starting with recording the patient's identity, physical examination, and history, up to the diagnosis of the patient whether the patient is in critical condition or not. The sub-model of emergency services is depicted in Figure 3.

3. Sub-model “Outpatient Services”

This sub-model illustrates how the hospital handles patients in outpatient services. This sub-model only consists of 3 activities, namely: See the medical record form, Passing outpatient medical records, and Give other services referrals. It can be seen in Figure 4.

4. Sub-model “Inpatient Services”

Figure 7 illustrates how the hospital handles patients in inpatient services. Starting with reference letter administration by admin officer, choosing facilities up to medical treatment and periodic checks carried out by nurses and doctors.

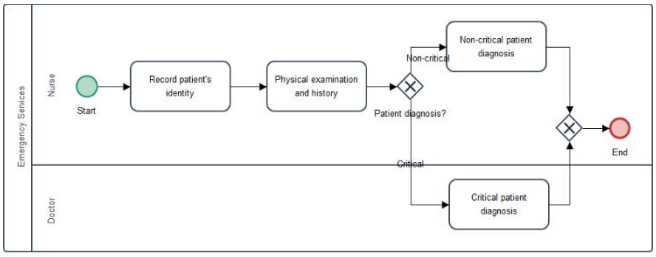

Fig. 3. Submodel Emergency Services

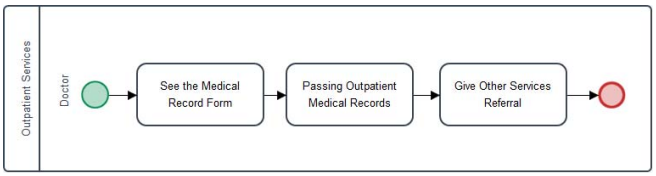

Fig. 4. Submodel Outpatient Services

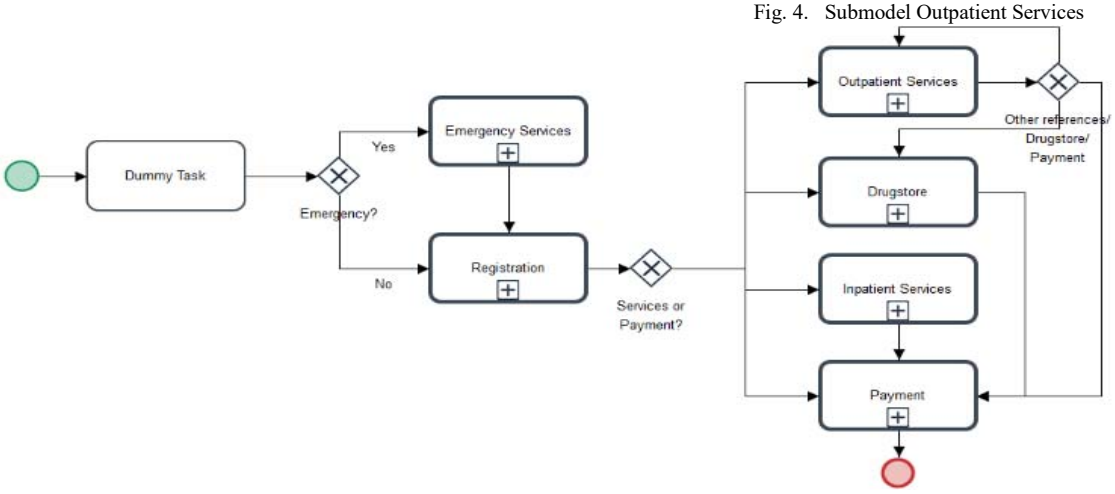

Fig. 5. Top-level process model “hospital model process”

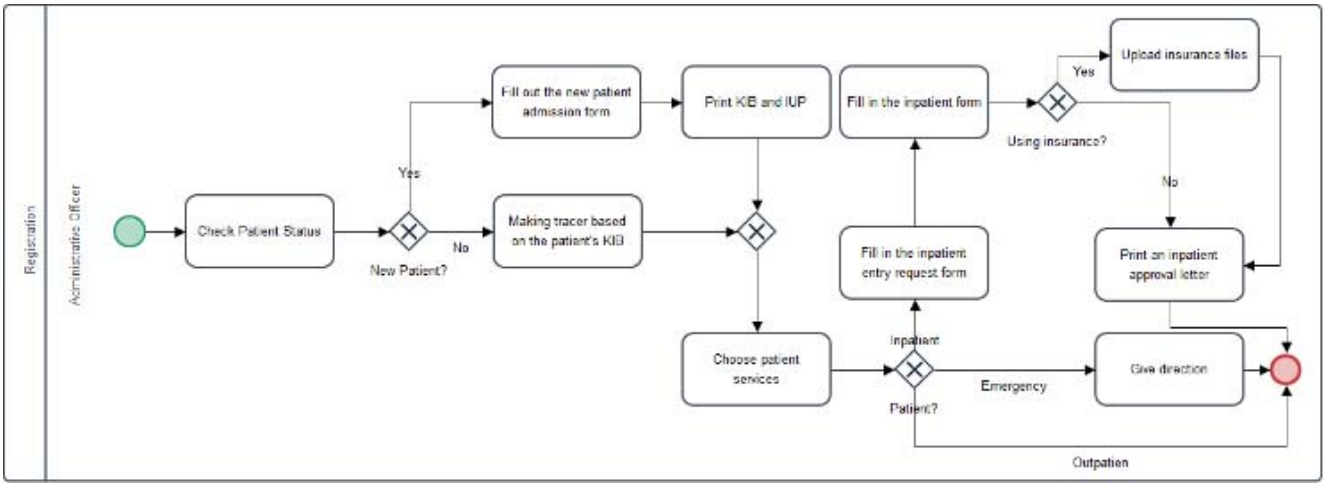

Fig. 6. Submodel Registration

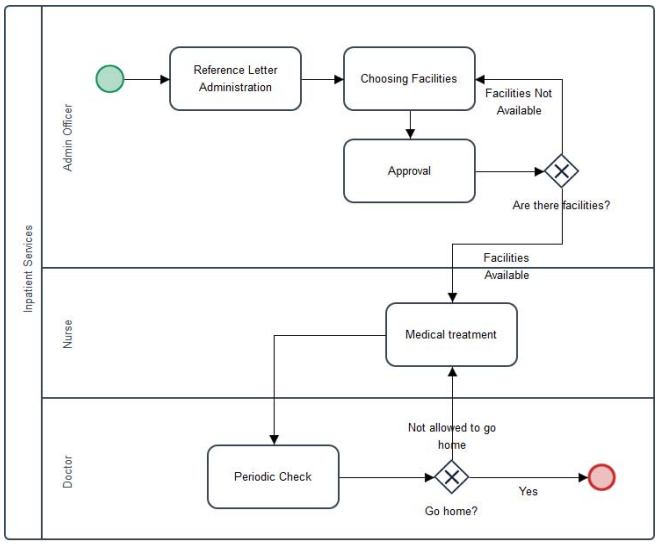

Fig. 7. Submodel Inpatient Services

5. Sub-model "Drugstore"

Figure 8 illustrates how the patients can get drugs based on a recipe from a doctor. Starting with the input doctor recipe then proceed with activities based on the services that have been selected. If the patient chooses inpatient services the drug is immediately billed to the bill of patients.

6. Sub-model "Payment"

Figure 9 illustrates how payments are made at the hospital. It starts with the cashier checking the type of payment based on the service chosen by the patient. Then, the cost of the selected service will be rechecked by adding the cost of the drug. After that, the patient can make payments based on the bill.

B. Simulating Hierarchical Process Model

After the hierarchy process model is successfully created, A simulation is performed to produce the event log. Every activity either in the top level process model or sub model must be executed or has been simulated to be recorded in the event log. We use Bitnami Process Maker to simulate hierarchical process models. But this simulation does not directly produce a clean event log but the activity data that is run is spread in the database.

C. Extracting Event Logs by using Proposed Algorithm

After the log data on the database is ready, we extract the event log by proposed algorithm. There are two algorithms: (1) bottom-level event log extraction algorithm, and (2) top-level event log extraction algorithm.

Figure 10 shows the pseudo code of bottom-level extraction algorithm. The first step before extracting the event log is connecting to the database. Then, take all the data needed to create the event log. After that, for every data that has been taken, add a case ID or process case for each process. Finally, create a CSV file from the data that has been prepared previously.

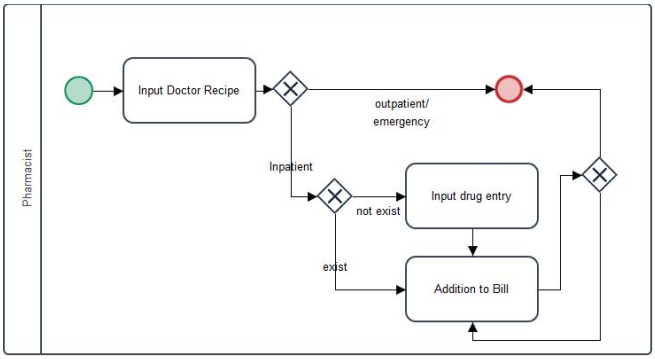

Fig. 8. Submodel Drugstore

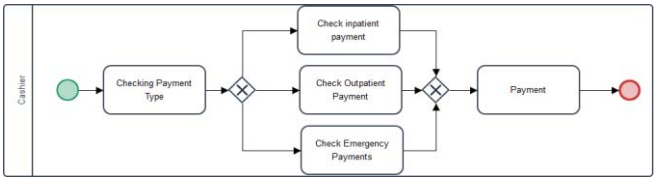

Fig. 9. Submodel Payment

```
databaseConnect()
logData = getEventLogData()

for every data in logData do
    set case ID for logData

create CSV file from logData
```

Fig. 10. Pseudo code for bottom level event log

TABLE I. EVENT LOG ON BOTTOM LEVEL

Proc. Case	Process	Task ID	Task	User	Timestamp
50	Payment	192	Make a payment	cashier	4/11/2019 12:44
50	Payment	192	Check outpatient payment	cashier	4/11/2019 12:44
50	Payment	192	Check the type of the payment	cashier	4/11/2019 12:44
50	Outpatient Services	191	Provide other service referrals	outpatient_ doctor	4/11/2019 12:43
50	Outpatient Services	191	Fill out an outpatient medical record	outpatient_ doctor	4/11/2019 12:43
50	Outpatient Services	191	See medical record forms	outpatient_ doctor	4/11/2019 12:43

where : Proc Case : case ID or process case
Task ID : the id of activity

The output of the bottom level event log extraction process can be seen in TABLE I. There are 6 columns in the event log that are created, namely: Process Case, Process, Task ID, Task, User, and Timestamp. The process case shows the case for each simulation process that is executed. The process shows the name of the submodel. The Task ID shows the case from the submodel. The task shows the name of the activity. The user shows who have done the activity. The timestamp shows when the activity starts.

```
databaseConnect()
logData = getEventLogData()
for every data in logData do
    set case ID for logData
tempID = 0
csvData
for every data in logData do
    if tempID != current process ID from logData do
        set csvData = current logData
        set tempID = current process ID from logData
    else do
        if csvData start time > current start time from
        logData do
            set csvData start time = current
            logData start time
        if csvData end time > current end time from
        logData do
            set csvData end time = current
            logData end time
create CSV file from csvData
```

Fig. 11. Pseudo code for top-level event log

In order to be able to create a top-level event log, we change the existing data on the event level bottom log according to the top level event log format. As shown in Figure 11, we remove task ID, task name, and user, then make start time and end time by using the timestamp on the bottom level event log. The timestamp on the first task will be changed to start time and the last task will be the end time for each existing submodel.

The output of the top level event log extraction process can be seen in TABLE II. There are four columns in the event log that are created, namely: Process Case, Activity, StartTime, and EndTime. Same as the bottom level event log, the process case shows the case for each simulation process that is executed. Activity shows the name of the submodel. StartTime shows when the first activity in the submodel is executed. EndTime shows when the last activity in the submodel is executed.

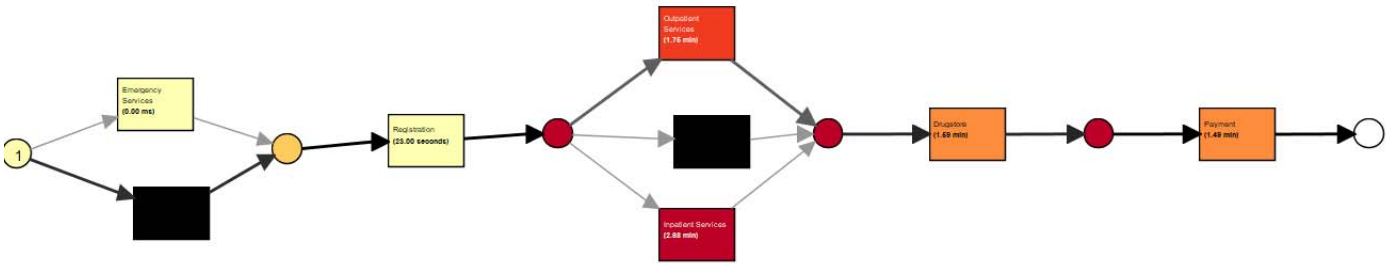

Fig. 12. Result of Performance Analysis in Top Level Process Model.

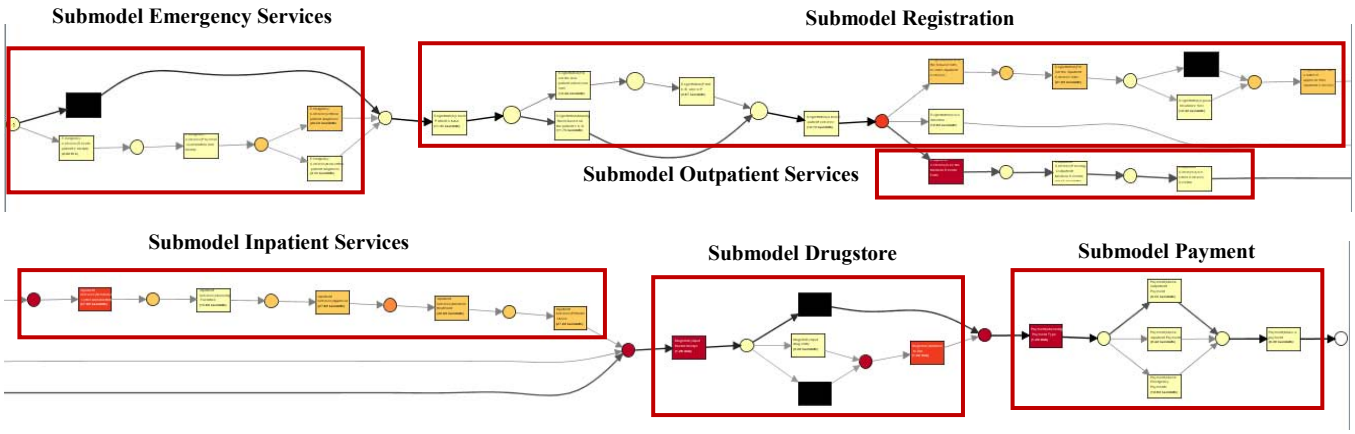

Fig. 13. Result of Performance Analysis in Bottom Level Process Model

D. *Performance Analysis on Event Log*

Performance Analysis performed in the top-level process model and bottom level process model. The top-level process model is obtained from processing the event log in TABLE II and the bottom-level process model is from the event log in TABLE I. The performance analysis shows the activities that have bottleneck, so analyst knows the activities that should be improved to reducing the processing time.

IV. RESULT & ANALYSIS

A. *Result of Performance Analysis*

Figure 12 shows the result of performance analysis from the top level process model. There is 6 activity in this process model, the 2 rectangle with the black color is not identified as an activity. It is a path from start event or activity to another activity. And each activity has a color that describes in figure 14. This color describes the waiting time of the activity to start the processing activity. The high waiting time has a color of red maroon (the right color in the Figure 14.). There is 1 activity that has high waiting time that is 'Inpatient Services'. 1 activity has a medium-high waiting time that is 'Outpatient Services', 2 activity have medium waiting time that is 'Drugstores' and 'Payment' and 2 activity have low waiting time that is 'Emergency Services' and 'Registration'. Those activity that identifies as high and medium-high is bottleneck activities.

Figure 13 shows the result of performance analysis from the bottom level process model. There is 5 activities that has high and medium-high waiting time. That are 'Outpatient Services|See medical record forms', 'Inpatient Services|Reference Letter Administration', 'Drugstore| Input Doctor Recipe', 'Drugstore| Addition to Bill' and 'Payment|Check' the type of the payment'. And that activities is identified as bottleneck activities. The other activities identify as an activity that has low or medium-low waiting time.

TABLE II. EVENT LOG ON TOP LEVEL

Proc. Case	Activity	StartTime	EndTime
29	Registration	4/8/2019 6:44	4/8/2019 6:45
29	Outpatient Services	4/8/2019 6:45	4/8/2019 6:45
29	Drugstore	4/8/2019 6:46	4/8/2019 6:46
29	Payment	4/8/2019 6:47	4/8/2019 6:47
30	Emergency Services	4/8/2019 6:50	4/8/2019 6:51
30	Registration	4/8/2019 6:51	4/8/2019 6:52
30	Drugstore	4/8/2019 6:52	4/8/2019 6:52
30	Payment	4/8/2019 6:53	4/8/2019 6:53

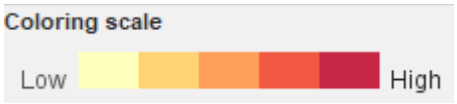

Fig. 14. Coloring Scale on Bottleneck

B. *Bottleneck Identification*

A bottleneck is an event in the process model that has a long waiting time than the other processes or an event with maroon red and red color in the result of performance analysis. Based on the top level process model. There is 2 sub model that has longer waiting time that is 'Drugstore' and 'Payment'. In the bottom level process model, activities that have long waiting time is 'Outpatient Services|See medical record forms', 'Inpatient Services|Reference Letter Administration', 'Drugstore| Input Doctor Recipe', 'Drugstore| Addition to Bill' and 'Payment|Check the type of the payment'.

Based on the results, we know that if there is an event at a bottom level which is a bottleneck, the sub model of that event not or the activity in the top-level process model cannot be immediately identified as a bottleneck. The sub model 'Outpatient services' and 'Inpatient services' has events that are identified as bottleneck, however this sub model is not identified as bottleneck. It is because they have more events in the bottom-level process model that are not identified as bottlenecks. In the top-level process model, waiting time is calculated by adding the waiting for every event in those sub model or in the bottom-level process model. So, if an event has

a very long time for waiting, its sub model or activity in the top-level process model can be identified as a bottleneck.

V. CONCLUSION

This research identifies bottlenecks events from a hierarchical process model based on the top level process model and the bottom level process model. For identify bottleneck first we have to simulate the process model to get the event log. After that, we should extract the top level and bottom level event log in the hierarchical process model. So we could perform that event log for performance analysis. After performing performance analysis on the top-level process model we find that there is 2 bottleneck sub models and for the bottom level there are 5 bottleneck events.

Based on the result in the conclusion, it can be seen that the activity in the top-level process model is bottleneck if the amount of its bottom-level activities that are identified as bottleneck is higher than those that are not identified as bottleneck. It is because bottleneck is identified from the total of the waiting time and the waiting time of the activity in the top-level process model is the summation of all activity in the bottom-level process model. An activity is identified as bottleneck if the total of waiting time is high.

REFERENCES

[1] I. Tangkawarow and J. Waworuntu, "A Comparative of business process modelling techniques," in *IOP Conference Series: Materials Science and Engineering*, 2016, vol. 128, no. 1, p. 12010.

[2] K. R. Sungkono, R. Sarno, and N. F. Ariyani, "Refining business process ontology model with invisible prime tasks using SWRL rules," in *2017 11th International Conference on Information Communication Technology and System (ICTS)*, 2017, pp. 215–220. http://doi.org/10.1109/ICTS.2017.8265673.

[3] K. R. Sungkono and R. Sarno, "CHMM for discovering intentional process model from event logs by considering sequence of activities," in *2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, 2017, pp. 1–6. http://doi.org/10.1109/EECSI.2017.8239194.

[4] Y. Nagm-Aldeen, M. A. Abdel-Fattah, and A. El-Khedr, "A Literature Review of Business Process Modeling Techniques," *International Journal*, vol. 5, no. 3, 2015.

[5] R. Sarno and Y. A. Effendi, "Hierarchy Process Mining from Multi-source Logs," *TELKOMNIKA*, vol. 15, no. 4, pp. 1960–1975, 2017. http://doi.org/10.12928/TELKOMNIKA.v15i4.6326.

[6] 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. http://doi.org/10.15866/irecos.v11i6.9555.

[7] 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, 2018. http://doi.org/10.1016/j.procs.2017.12.139.

[8] 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 (IJIES)*, vol. 12, no. 3, pp. 31–40, 2019.

[9] R. Sarno, K. R. Sungkono, R. Johanes, and D. Sunaryono, "Graph-Based Algorithms for Discovering A Process Model Containing Invisible Tasks," *Intelligent Networks and Systems Society (INASS)*, vol. 12, no. 2, pp. 85–94, 2019.

[10] K. R. Sungkono, R. Sarno, and N. F. Ariyani, "Refining business process ontology model with invisible prime tasks using SWRL rules," in *2017 11th International Conference on Information & Communication Technology and System (ICTS)*, 2017, pp. 215–220. http://doi.org/10.1109/ICTS.2017.8265673.

[11] K. R. Sungkono and R. Sarno, "Constructing Control-Flow Patterns Containing Invisible Task and Non-Free Choice Based on Declarative Model," *International Journal of Innovative Computing, Information and Control (IJICIC)*, vol. 14, no. 4, 2018.

[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 *International Conference on Advanced Information Systems Engineering*, 2011, pp. 33–48.

[13] 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, pp. 146–166, 2013.

[14] 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)," *Proceedings of the 11th International Conference on Information and Communication Technology and System, ICTS 2017*, vol. 2018-Janua, pp. 143–148, 2018. http://doi.org/10.1109/ICTS.2017.8265660.

[15] S. Son, B. N. Yahya, M. Song, S. Choi, J. Hyeon, B. Lee, Y. Jang, and N. Sung, "Process mining for manufacturing process analysis: a case study," in *Proceeding of 2nd Asia Pacific Conference on Business Process Management, Brisbane, Australia*, 2014.

[16] P. Badakhshan and A. Alibabaei, "Using process mining for process analysis improvement in pre-hospital emergency," in *Middle East North Africa Conference for Information Systems, Paris*, 2018.

[17] W. Bandara, G. G. Gable, and M. Rosemann, "Factors and measures of business process modelling: model building through a multiple case study," *European Journal of Information Systems*, vol. 14, no. 4, pp. 347–360, 2005.

[18] T. H. Davenport, "The coming commoditization of processes," *Harvard business review*, vol. 83, no. 6, pp. 100–108, 2005.

[19] A. Cahyapratama and R. Sarno, "Application of Analytic Hierarchy Process (AHP) and Simple Additive Weighting (SAW) Methods In Singer Selection Process," in *International Conference on Information and Communications Technology (ICOIACT) Application*, 2018, no. Mcdm, pp. 234–239.

[20] M. Leemans, "Hierarchical process mining for scalable software analysis," 2018.