

Model Discovery of Parallel Business Processes using Modified Heuristic Miner

Riyanarto Sarno, Fitrianing Haryadita, Dwi Sunaryono, Abdul Munif

Informatics Department, Faculty of Information Technology, Institut Teknologi Sepuluh Nopember, Indonesia
riyanarto@if.its.ac.id, fitrianing11@mhs.if.its.ac.id, dwi@if.its.ac.id, and munif@if.its.ac.id

Abstract—Process Mining or Process Discovery is a method to automatically discover process models from event log data. Since the process discovery is gaining attention among researchers as well as practitioners, the quality of the resulted process models is required. Business process model contains sequence and parallel traces. Many algorithms have been employed for process discovery, such as Alpha, Alpha ++ and Heuristic Miner. Both Alpha ++ and existing Heuristic Miner cannot discover processes containing parallel OR. In this paper we propose the modified Heuristic Miner which utilizes the threshold intervals to discover parallel XOR, AND, and OR. The threshold intervals are determined based on average dependency measure in dependency graph. The results show that the modified Heuristic Miner can discover OR split and join which cannot be discovered by Alpha ++ as well as the existing Heuristic Miner.

Keywords—Process Discovery; Discovery Parallel Activity OR and AND; Modified Heuristic Miner

I. INTRODUCTION

Analyzing the varieties of business process requires a business process modeling [1, 2]. Process discovery is one of the most difficult task in process mining and the large unsupervised learning task in nature due to the fact that event log rarely contains negative events [3]. The proposed process discovery is to make the business process model from event log. The structure of a business process model can be expressed as a graph which is containing a set of nodes connected by the edge [4]. There are a lot of algorithms which is used for process discovery. Alpha ++ is one of the algorithm which is used for process discovery. This algorithm uses Petri nets principal to make the model [5]. Alpha ++ algorithm can discover event log containing short loop and XOR, AND parallel process [6]. Petri Net model used in alpha ++ only can discover parallel process which is caused by XOR and AND. Because of that, Heuristic Miner appears as new algorithm.

Heuristic Miner uses Causal-Net (C-Net) and does not employ Petri Net, since Petri Net cannot model OR parallel [7]. This paper changes the parallel model of C-Net to improve readability which is shown in Table I.

C-net can model OR but the existing Heuristic Miner only can model XOR and AND. Unfortunately, a lot of models use OR for split and join. Moreover in Heuristic Miner there is no method to determine limitation of threshold value to discover business process model.

Therefore, this paper proposes modified Miner Heuristic algorithm which can discover business process models containing OR relations by determining the appropriate threshold.

TABLE I. C-NET AND PROPOSED PARALLEL MODEL

C-Net Model	Proposed Model
 XOR-Split	
 AND-Split	
 OR-Split	
 XOR-Join	
 AND-Join	
 OR-Join	

II. RELATED WORKS

A. Event log

The starting point of analysis is an event log, which is basically a set of process executions capturing the different business activities. However, typical event log will contain much more information, for instance organizational information concerning the performers of the different activities [5, 8]. Event log is divided into three parts i.e. Case, Trace, and Activity.

- Case and Trace

A case is a record of events related to a single executed process instance. Case can be described as the production process of one stuff. Whereas trace records sequence of events that belong to the same case [9]. For example in event log (1).

$$L = \{ \langle a, c, d \rangle^{45}, \langle b, c, d \rangle^{42}, \langle a, c, d \rangle^{38}, \langle b, c, e \rangle^{22} \} \quad (1)$$

In that event log:

1. Contains 4 trace .i.e. (a,c,d), (b,c,d), (a,c,d), (b,c,e)
2. Contains 147 case .i.e. (a,c,d) executed 45 time, (b,c,d) 42 time, (a,c,e) 38 time, and (b,c,e) 22 time.

- Activity

Activity is part of event log presenting sub of production process of a stuff or certain production process. Example in event log in (1), the event log contains five activities i.e. {a, b, c, d, e}..

B. Parallel AND

Parallel AND is used when parallel split pattern occurs. The mechanism which is allows activity in split or join to perform concurrently rather than serially is defined as Parallel Split pattern. Two or more paths which are made from split of a single path so at the same time two or more activity can be started or the activity in split shall be executed. All of algorithm in process discovery can model AND [10].

C. Conditional OR

Conditional OR is used when multiple choice pattern occurs. The alternative paths may be chosen from one to all at performance-time in Multiple Choice pattern. Technically, one path pattern may be allowed to choose in Multiple Choice. But when the process flow stops unexpectedly, this could be considered as invalid situation [11].

While mining a log from multiple choice workflow, some available discovery algorithms cannot identify the correct representation of the workflow. Some algorithms identify the model as AND-split, while other identify the model as XOR-split [12]. Alpha ++ which uses Petri net for modelling [10] cannot model OR.

D. Single Choice XOR

Single Choice XOR is used when there is only one activity to be executed [11]. All of algorithm in process discovery can model XOR.

E. Mining Parallel Activity in Existing Heuristic Miner

The existing Heuristic Miner can only discover XOR and AND, because the formula of parallel activity only has threshold of AND.

It cannot discover OR because threshold interval only divides into two regions. If threshold is zero the parallel split and join are XOR. However, if threshold is more than zero the parallel split and join are AND [13].

III. THE PROPOSED METHOD

The difference between modified Heuristic Miner and the existing Heuristic Miner is the Mining Parallel Activity. The modified method changes the parallel measure formula in the Mining Parallel Activity and determines threshold to decide split and join that will be used in the model. The formula of

Mining Parallel Activity is described in step of Heuristic Miner and the threshold of parallel activity is described in discover XOR, OR, and AND.

A. Step in Heuristic Miner Algorithm

Based on [13] there are three steps to discover model from event log using Heuristic Miner.

1) Mining Dependency Graph

Constructing *dependency graph* is the starting point of the Heuristic Miner Algorithm. The truly dependency between two event A and B (notation $A \Rightarrow_w B$) is indicated using frequency based metric. The frequency produces matrix frequency direct successor. Heuristic search use calculated \Rightarrow_w values between events in event log.

The Formula for dependency measure to decide dependency relation between activities in control-flow network.

$$a \Rightarrow_w b = \left(\frac{|a >_w b| - |b >_w a|}{|a >_w b| + |b >_w a| + 1} \right) \quad (2)$$

where the symbols mean:

$a \Rightarrow_w b$: value of depency measure from a to b

$|a >_w b|$: frequency activity a is direct followed by activity b

$|b >_w a|$: frequency activity b is direct followed by activity a

Equation (2) produce matrix dependency measure which use to make dependency graph.

A Weijters and Sofie Crude in [13, 10] described Heuristic Miner, which uses few threshold to determining dependency measure. All off the threshold are described in the short description below.

a) Relative-to-best threshold (RBT)

Relative-to-best threshold indicates the edge which we are going to accept (i.e. the edge which is inserted into the resulting control-flow network) if the difference between the value of the dependency measure in matrix dependency measure and the maximum value of the dependency measure matrix dependency measure is below the value of this threshold.

This paper suggests that this threshold value close with the average value of dependency measure in control-flow network.

- First step to get Relative-to-best threshold is to calculate the average (Avg) of positive dependency measure in matrix dependency (PDM) measure.
- Second step is to determine upper and lower limits of threshold interval using standard deviation. This paper suggest average (Avg) of positive dependency measure (PDM) as upper limit and average of dependency measure minus standard deviation (SD) as lower limit.
- The last step is to determine relative-to-best threshold. This paper suggest formula in (3) to determine relative-to the best threshold.

$$RBT = Avg PDM - \left(\frac{SD PDM}{2} \right) \quad (3)$$

b) Positive observations threshold

Positive observations threshold can control the minimum frequencies of dependency between activities. Using the relation only when the number of frequency between activities is same or above the threshold's value (i.e. there are dependency of two activity a and b so the Positive observation threshold is $|a >_w b| + |b >_w a|$). Positive to the best threshold use the minimum frequency direct successor in matrix frequency direct successor [12].

c) Dependency threshold

Dependency threshold is used to choose dependency measure in matrix dependency which is lower than the value of the threshold. This paper suggest formula as in (4) to determine dependency threshold.

$$RBT = Avg PDM - SD PDM \quad (4)$$

2) Checking Short Loop (Length One Loop and Length Two Loop)

Activity may be execute by process in multiple time. This condition is refers as loop. Using dependency measure make Heuristic Miner does not have problem with long distance loop, but it has problem with detect short loop. Formula for Length One Loop as in (5).

$$a \Rightarrow_w a = \left(\frac{|a >_w a|}{\max\{|a >_w x| \mid x \in e\}} \right) \quad (5)$$

where the symbols mean:

$a \Rightarrow_w a$: value of *dependency Length One Loop*

$|a >_w a|$: frequency activity a is direct followed by activity a
 $\max\{|a >_w x| \mid x \in e\}$ = frequency is activity a direct followed by activity x which x is one of activity in event log.

Formula for Length Two Loop as in (6).

$$a \Rightarrow_{2w} b = \left(\frac{|a \gg_w b| + |b \gg_w a|}{|a \gg_w b| + |b \gg_w a| + 1} \right) \quad (6)$$

where the symbols mean:

$a \Rightarrow_{2w} b$: value of *dependency Length Two Loop*

$|a \gg_w b|$: frequency of aba in event log

$|b \gg_w a|$: frequency of bab in event log

3) Mining Parallel Activity

Heuristic Miner use parallel measure threshold to discover parallel activity. First step to discover parallel activity is the determination of input output activity using causal matrix (Table II).

TABLE II. CAUSAL MATRIX

Input	Activity	Output
{}	A	B, D
A	B	C
E, B	C	{}
A	D	E
D	E	D, C

If the input output is more than one activity then there is the possibility of parallel, therefor parallel measure can be counted. In the existing method the formula of parallel measure is in (7).

$$a \Rightarrow_w b \wedge c = \left(\frac{|b >_w c| + |c >_w b|}{|a >_w b| + |a >_w c| + 1} \right) \quad (7)$$

where the symbols in (7) mean: $a \Rightarrow_w b \wedge c$: *parallel measure* between b and c, split in a; $|b >_w c|$: frequency activity b is direct followed by activity c; $|c >_w b|$: frequency activity c is direct followed by activity b; $|a >_w b|$: frequency activity a is direct followed by activity b; and $|a >_w c|$: frequency activity a is direct followed by activity c

The existing method set the frequency of direct activities in the event log. However, there are indirect activities that can increase the chance of discovering AND parallel. Therefore, the modified method additionally sets the frequency of indirect activities in the event log. For example in (5), the existing Heuristic Miner calculate parallel measure of activity b and c which split in activity a. This (7) use frequency direct activity of parallel choice i.e. activity b and c ($|b >_w c|$) and activity c and b ($|c >_w b|$), whereas in parallel AND can be executed as far as all activities in split and join are executed in all trace in event log. Therefor in modified Heuristic Miner for count parallel measure activity b and c which split in a use both of frequency direct and indirect of parallel choice which split in parent activity i.e. activity b and c ($|b \gg_w c|$) and activity c and b ($|c \gg_w b|$), for XOR and OR threshold use frequency of activity in parallel choice that is not followed by another activity in parallel choice. The formula of parallel measurement in modified Heuristic Miner is written in (8).

$$a \Rightarrow_w b \wedge c = \left(\frac{|b \gg_w c| + |c \gg_w b|}{|a >_w b| + |a >_w c| + |b \gg_{not_w} c| + |c \gg_{not_w} b| + 1} \right) \quad (8)$$

where the symbols in (8) mean: $a \Rightarrow_w b \wedge c$: *parallel measure* between b and c, split in a; $|b \gg_w c|$: frequency activity b is direct and indirect followed by activity c; $|c \gg_w b|$: frequency activity c is direct and indirect followed by activity b; $|a >_w b|$: frequency activity a is direct followed by activity b; $|a >_w c|$: frequency activity a is direct followed by activity c; $|b \gg_{not_w} c|$: frequency activity b is not followed by activity c; and $|c \gg_{not_w} b|$: frequency activity c is not followed by activity b.

B. Discovering XOR, OR, and AND

This section will describe the difference between Heuristic Miner and Heuristic Miner with modification. This paper changes not only Mining Parallel activity, but also modelled split and join for parallel activity which consist of XOR, OR, and AND. Discovering XOR, OR, and AND utilizes as in (9) and (10).

$$Avg DM = \frac{\sum_{i=1}^{n_e} e_i}{n_e} \quad (9)$$

$$Avg PM = \frac{\sum_{i=1}^{n_{PM}} PM_i}{n_{PM}} \quad (10)$$

The interval for discovering XOR, OR, and AND are:

- XOR

$$\text{If Avg PM} \leq \text{Limit PDM then XOR (11)}$$

- OR

$$\text{If Limit PDM} \leq \text{Avg PM} \leq \text{Avg DM then OR (12)}$$

- AND

$$\text{If Avg DM} \leq \text{Avg PM then AND (13)}$$

where the symbols in the (11), (12), and (13) are mean: $Avg DM$: average from the arc weight of dependency graph; e_i : arc weight of dependence graph; n_e : number of edges; $Avg PM$: average from parallel measure which is calculated from (8); PM : Parallel measure from (8); n_{PM} : number of parallel measure; PDM : positive dependency measure from dependency graph; and $Limit PDM$: minimum value of positive dependency measure in matrix dependency measure.

The average of parallel measure itself have rules when the value can be averaged together. Because in one model allows to have more than one parallel measurement.

- The parallel measure which is averaged must be have same input or output activity.
- The number of inputs and outputs of parallel processes which will be calculated the average should be the same.

IV. EVALUATION

The business process model which is used in this paper is modeled by YAWL. The YAWL model is shown in Fig 1. Business process model in Fig 1 has five activity, namely A, B, C, D and E.

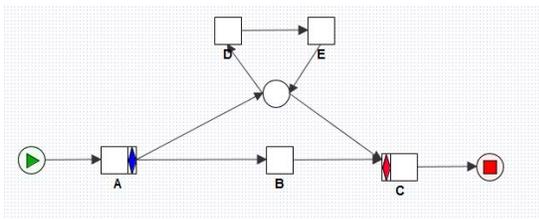


Fig. 1. YAWL OR Model

Business process model can produce event log. Using YAWL simulation Fig. 1 can produce event log (L) as in the Table III.

TABLE III. EVENT LOG

Trace	Task	Start	Finish
1	A	2015-01-28T10:22:44.299+0700	2015-01-28T10:22:51.554+0700
1	B	2015-01-28T10:23:02.283+0700	2015-01-28T10:23:08.588+0700
1	C	2015-01-28T10:23:22.387+0700	2015-01-28T10:23:32.497+0700
2	A	2015-01-28T10:23:50.422+0700	2015-01-28T10:23:57.617+0700
2	B	2015-01-28T10:24:16.002+0700	2015-01-28T10:25:23.026+0700
2	D	2015-01-28T10:25:05.487+0700	2015-01-28T10:25:07.766+0700
2	E	2015-01-28T10:25:34.266+0700	2015-01-28T10:25:40.696+0700
2	C	2015-01-28T10:25:53.035+0700	2015-01-28T10:26:01.098+0700
3	A	2015-01-28T10:26:15.640+0700	2015-01-28T10:26:21.224+0700
3	D	2015-01-28T10:26:33.942+0700	2015-01-28T10:30:49.233+0700
3	B	2015-01-28T10:30:08.440+0700	2015-01-28T10:31:32.673+0700
3	E	2015-01-28T10:31:17.733+0700	2015-01-28T10:31:41.568+0700
3	C	2015-01-28T10:31:51.686+0700	2015-01-28T10:32:04.954+0700
4	A	2015-01-28T10:32:22.175+0700	2015-01-28T10:32:28.534+0700
4	D	2015-01-28T10:32:42.904+0700	2015-01-28T10:34:47.944+0700
4	B	2015-01-28T10:33:40.243+0700	2015-01-28T10:36:11.947+0700
4	E	2015-01-28T10:35:13.082+0700	2015-01-28T10:36:05.359+0700
4	C	2015-01-28T10:36:21.772+0700	2015-01-28T10:36:27.845+0700
5	A	2015-01-28T10:36:40.186+0700	2015-01-28T10:36:47.204+0700
5	B	2015-01-28T10:37:00.812+0700	2015-01-28T10:38:42.275+0700
5	D	2015-01-28T10:37:50.609+0700	2015-01-28T10:38:31.356+0700
5	E	2015-01-28T10:38:57.875+0700	2015-01-28T10:39:06.104+0700
5	D	2015-01-28T10:39:19.783+0700	2015-01-28T10:39:25.935+0700
5	E	2015-01-28T10:39:35.812+0700	2015-01-28T10:39:43.446+0700
5	C	2015-01-28T10:39:55.256+0700	2015-01-28T10:40:01.225+0700

$$L = \{(A,B,C), (A,B,D,E,C), (A,D,B,E,C), (A,D,E,B,C), (A,B,D,E,D,E,C)\}$$

A. Mining Using Modified Heuristic Miner

Event log Table III is discovered using modified Heuristic Miner as follows:

1) Mining Dependency Graph

The first step is to create matrix dependence relation frequency activity in event log. The event log above can produce matrix described in Table IV.

TABLE IV. MATRIX FREQUENCY DIRECT SUCCESSOR

Activity	A	B	C	D	E
A	0	3	0	2	0
B	0	0	2	2	1
C	0	0	0	0	0
D	0	1	0	0	4
E	0	1	3	1	0

The second step is to count dependency measure using (2). From the Formula can produced matrix dependency measure like in Table V.

TABLE V. MATRIX DEPENDENCY MEASURE

Activity	A	B	C	D	E
A	0	0.75	0	0.667	0
B	0	0	0.667	0.25	0
C	0	0	0	0	0
D	0	-0.25	0	0	0.5
E	0	0	0.75	-0.5	0

The third step is to determine the threshold. This step needs average and standard deviation of positive dependency measure calculated from the matrix dependency.

- Average of positive dependency measure = 0.64
- Standard deviation of positive dependency measure = 0.19

Average and standard deviation of positive dependency measure in matrix dependency is used to make interval of Relative-to-best threshold. Relative-to-the Best Threshold Suggested Interval is average of positive dependency measure minus standard deviation of dependency measure.

With the threshold are:

- Value of Relative-to-best threshold is calculated using (3).

$$\text{Relative-to-best threshold} = 0.54$$

- Positive to the best threshold use the minimum dependency of matrix frequency direct successor.

$$\text{Positive observations threshold} = 1$$

- Value of Dependency threshold is calculated using (4).

$$\text{Dependency threshold} = 0.45$$

These threshold can indicate the dependency relation between activities which will be accept. The relation of activity which has dependency measure more than or same with dependency threshold, the relation of the activity which has frequency more than or same value of Positive observations threshold, and the relation of activity which has difference value with the max dependency measure is lower than RTB.

Dependency graph can be modelled as in Fig. 2.

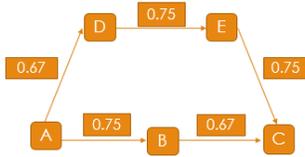


Fig. 2. Dependency Graph

2) Checking Short Loop (Length One Loop dan Length Two Loop)

Equation (5) uses to calculate matrix Length One Loop. This model does not have short loop because all of value in frequency short loop are zero. Length Two Loop needs matrix frequency for Length Two Loop to calculate Dependency Length Two Loop. Matrix frequency Length Two Loop (Table VI).

TABLE VI. MATRIX FREQUENCY LENGTH TWO LOOP

Activity	A	B	C	D	E
A	0	0	0	0	0
B	0	0	0	0	0
C	0	0	0	0	0
D	0	0	0	0	1
E	0	0	0	1	0

Then using Length Two Loop Formula in (6), we can get a matrix as shown in Table VII.

TABLE VII. MATRIX DEPENDENCY LENGTH TWO LOOP

Activity	A	B	C	D	E
A	0	0	0	0	0
B	0	0	0	0	0
C	0	0	0	0	0
D	0	0	0	0	0.667
E	0	0	0	0.667	0

From Table VII, we can know that Length Two Loop DED is contained in the model. Hence the model can be described in Fig. 3.

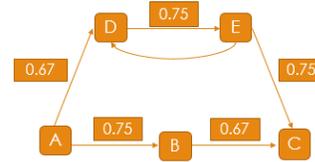


Fig. 3. Dependency Graph with Length Two Loop

3) Mining Parallel Activity

The first thing to do is to create causal matrix (Table VIII).

TABLE VIII. CAUSAL MATRIX

Input	Activity	Output
{}	A	B, D
A	B	C
E, B	C	{}
A	D	E
D	E	D, C

Since there are input output which have more than one activity, then the parallel activities are $a \rightarrow w^{b^d}$, $c \rightarrow w^{b^e}$, and $e \rightarrow w^{d^c}$. In this case activity E has input D besides output D. and when short loop checking DED is indicated as short loop, so E to D do not use split because include in short loop, so $e \rightarrow w^{d^c}$ does not use split. Parallel measure is calculated using (8).

$$a \rightarrow w^{b^d} = 0.57$$

$$c \rightarrow w^{b^e} = 0.57$$

Then check both of the parallel measurement can be averaged or not. Because the input output has one same activity and the number of input output activity is same, the parallel measurement can be averaged.

Determining XOR, OR, and AND use (11), (12), (13). Equation (11), (12), and (13) need average of dependency measure in dependency graph, limit of dependency measure in dependency graph. Average of dependency measure using (9) = 0.72. Limit of dependency measure using (10) = 0.25. The average of parallel measure is 0.57. Equation (12), (13), and (14) show split and join that used in this model is OR. Therefore the model become as shown in Fig. 4.

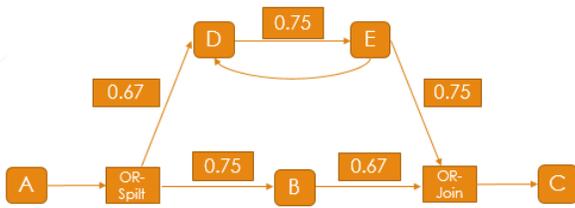


Fig. 4. Final Model from Modified Heuristic Miner

B. Mining Using Existing Heuristic Miner

Event log Table 3 is discovered using existing Heuristic Miner with same RTB, PO, and DT will produce same dependency graph like Fig. 2 and same causal matrix like Table 8, but the calculation of parallel measure is different. Existing Heuristic Miner use (7) to calculate parallel measure.

$$a \rightarrow w^{b^d} = 0.67$$

$$c \rightarrow w^{b^e} = 0.33$$

Using the rule of existing Heuristic Miner, if the parallel measure is more than zero then the split and join which is used is AND. Therefore the model which is discovered using existing Heuristic Miner become as shown in Fig. 5.

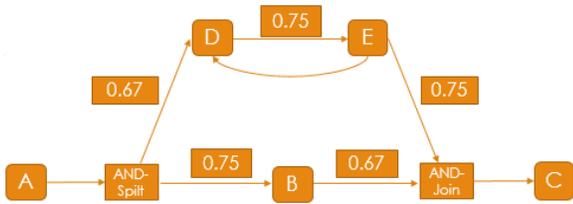


Fig. 5. Final Model from Existing Heuristic Miner

The experiment model which is produced event log use split and join OR but existing Heuristic Miner cannot discover split and join OR therefore final model from existing Heuristic Miner using split and join AND. Process discovery which use existing Heuristic Miner is fail because split and join which is use from experiment model and model from process discovery event use existing Heuristic Miner is different.

V. CONCLUSION

It has been shown that the modified Heuristic Miner can discover business process which contains parallel OR, which cannot be discovered by the existing Heuristic Miner. The algorithm of the modified Heuristic Miner to discover a model is similar to that of the Heuristic Miner; whereas the difference is in the mining of parallel activities. The modified Heuristic Miner uses both direct and indirect activities. The parallel activity are taken from the causal matrix which has been counted then parallel measure is counted using (8). Hence, the threshold intervals of average parallel measurement is used to decide the types of parallel split and join. The results show that

the modified Heuristic Miner can discover correct models which contain parallel XOR, AND, and OR.

Our future research will focus on the limitation of handling noise in both the modified and existing Heuristic Miners, which are about noise, model of noise and the maximum number of noise which can be handled by the modified Heuristic Miner.

ACKNOWLEDGEMENT

We would like to thank Institut Teknologi Sepuluh Nopember and DIKTI - Directorate General of Higher Education - Ministry of Education and Culture of Indonesia for supporting the research

REFERENCES

- [1] Riyanarto Sarno, B. Sanjoyo, A. Mukhlash and H.M. Astuti, "Petri Net model of ERP business process variation for Small and Medium Enterprises", *Journal of Theoretical and Applied Information Technology*, Vol. 54, No.1, pp. 31-38, 2013.
- [2] Riyanarto Sarno, Djeni C.A., Mukhlash, I., Sunaryono D., "Developing A Workflow Management System for Enterprise Resource Planning," *Journal of Theoretical and Applied Information Technology*, Vol. 72, No.3, pp. 412-421, February 2015.
- [3] S. Goedertier, J. De Weerd, D. Martens, J. Vanthienen, and B. Baesens, "Process Discovery in Event Log: An Application in the Telecom Industry," *Applied Soft Computing*, vol. 11, no. 2, pp. 1697- 1710, 2011.
- [4] Riyanarto Sarno, Hari Ginardi, Endang Wahyu Pamungkas, Dwi Sunaryono, "Clustering of ERP Business Process Fragments", *Proceeding IEEE International conference on computer, control, informatics, and its applications*, pp. 319-324, 2013.
- [5] Lijie Wen, W.M.P van der Aalst, Jianmin Wang, and Jianguang Sun, *Mining Process Models with Non-Free-Choice Construct*, School of Software Tsinghua University, 100084, Beijing, China, 2012.
- [6] Weber Philip, "Principled Approach to Mining From Noisy Log Using Heuristics Miner", *IEEE Symposium on Computational Intelligence and Data Mining*, 2013.
- [7] W.M.P van der Aalst, A. Adriansyah, and B.F. van Dongen, "Causal Nets: A Modeling Language Tailored Towards Process Discovery", In J.P. Katoen and B. Koenig, editors, *22nd International Conference on Concurrency Theory (CONCUR 2011)*, pp. 28-42, 2011.
- [8] M. Song and W.M.P. van der Aalst, "Towards Comprehensive Support for Organizational Mining," *Decision Support Systems*, vol. 46, no. 1, pp. 300-317, 2008.
- [9] van der Aalst, W. M. P. *Process Mining - Discovery, Conformance and Enhancement of Business Processes*, Netherlands: Springer, pp. 95-125, 2011.
- [10] Cnude Sofie DE, "Improving the quality of the Heuristics Miner in ProM 6.2", 2014.
- [11] Weske Mathias, *Business Process Management- Concepts, Languages, Architectures*, Springer, pages 128-135, 2011.
- [12] Riyanarto Sarno, Putu Linda Indita Sari, Hari Ginardi, Dwi Sunaryono, Imam Mukhlash, "Decision Mining for Multi Choice Workflow Patterns," in *2013 International Conference on Computer, Control, Informatics and Its Applications*, pp. 337-342, 2013.
- [13] A. Weijters, W. van der Aalst, A.A. de Medeiros, "Process mining with the Heuristics Miner-algorithm", *BETA Working Paper Series, WP 166*, Eindhoven University of Technology, Eindhoven, 2006.