

Implementasi Process Mining pada Prosedur Pengendalian Dokumen dengan Menggunakan Algoritma Heuristic Miner (Studi Kasus : Institut Teknologi Telkom)
Implementation Process Mining in Document Control Procedure Using Heuristic Miner Algorithm (Case Study : Institut Teknologi Telkom)

Wildan Khalidy¹, Angelina Prima Kurniati², Imelda Atastina³

^{1,2,3} Fakultas Informatika, Telkom Engineering School, Telkom University

Jalan Telekomunikasi No.1, Dayeuh Kolot, Bandung 40257

wildan.khalidy92@gmail.com¹, angelina@telkomuniversity.ac.id², imelda@telkomuniversity.ac.id³

Abstrak

Prosedur Pengendalian Dokumen (PPD) di Institut Teknologi Telkom merupakan suatu contoh proses bisnis yang dapat digali informasinya menggunakan teknik Process Mining. Proses bisnis tersebut sedang mengalami perubahan, akibat pergantian status perguruan tinggi, dari Institut Teknologi Telkom menjadi Telkom University sehingga sedang dilakukan pengkajian lebih lanjut terhadap proses bisnis tersebut. Salah satu tipe dari proses mining adalah Enhancement, yaitu perluasan atau peningkatan sebuah model proses yang ada dengan menggunakan informasi pada event log, hal ini akan digunakan sebagai bahan rekomendasi terhadap perubahan PPD. Algoritma Heuristic Miner merupakan salah satu algoritma yang dapat diterapkan pada Process Mining, dimana algoritma ini akan menggabungkan frekuensi antar event dengan trace dalam event log untuk membangun sebuah model proses. Tahapan yang dijalankan dalam penelitian ini adalah Discovery yaitu pembentukan model proses dari event log, Conformance yaitu pemeriksaan model proses dengan event log, dan Enhancement yang bertujuan untuk menghasilkan sebuah model proses yang baru. Selain membuat model proses baru, keluaran dari pengerjaan tugas akhir ini adalah analisis terhadap komponen threshold dalam membangun model proses.

Kata kunci: proses bisnis, process mining, discovery, conformance, enhancement, heuristic miner algorithm

Abstract

Document Control Procedure (DCP) in Institut Teknologi Telkom is an example of business process that can be extracted about its information using Process Mining technique. This business process is changing, due to the change of college status, from Institut Teknologi Telkom to Telkom University so that it's being carried out further studies on that business process. One type of process mining is Enhancement, it's the expansion or improvement of an existing process model by using the information from event log, this will be used as a recommendation in changing DCP. Heuristic Miner Algorithm is one of the algorithms that can be applied to the process mining, which this algorithm will combine the frequency between even with the event traces in event log to build a model process. In this research, several steps will be executed those are Discovery, the one that will build a model process from event log, Conformance, the one that will examine the model process with the event log, and Enhancement that has a purpose to produce a new model process. Beside those steps, the output from this research is an analysis of the threshold components while building the model process.

Keywords: business process, process mining, discovery, conformance, enhancement, heuristic miner algorithm

1. Pendahuluan

Pengendalian dokumen di Institut Teknologi Telkom (ITT) merupakan salah satu proses bisnis yang penting dalam rangka penerapan Sistem Manajemen Mutu ISO 9001:2008 [3]. Cakupan dari pengendalian dokumen di ITT sendiri mencakup pembuatan draft dokumen, pengesahan, perubahan dokumen, hingga pendistribusian dokumen di ITT. Jika hal ini tidak dikelola dengan baik maka dimungkinkan terjadi alur proses yang ambigu atau tidak sesuai dengan yang diharapkan dan dapat menurunkan mutu sistem manajemen di Institusi.

Pada Tugas Akhir ini, digunakan teknik *Process Mining* untuk mengekstraksi informasi yang terkandung dalam proses bisnis Prosedur Pengendalian Dokumen ITT [2]. Penggunaan tipe *Enhancement* bertujuan untuk membuat sebuah model proses baru dari event log yang tersedia [4], dimana sebelum itu dilakukan tahapan *Discovery* yang berguna untuk membuat sebuah model proses dari event log yang ada, untuk kemudian diperluas / *Enhancement* menjadi sebuah model proses yang baru. Penggunaan algoritma *Heuristic Miner* pada kasus ini memiliki beberapa kelebihan dibandingkan dengan algoritma *Deterministic (α Algorithm)*, contohnya *Heuristic Miner* dapat menangani data yang memiliki noise [5], kemudian algoritma ini lebih menitikberatkan pada menghitung frekuensi dari suatu proses pada event log, sehingga dapat menggambarkan *dependency* / hubungan antar *event* yang kompleks serta memang cocok dengan karakteristik dataset yang digunakan.

2. Landasan Teori

2.1 Process Mining

Process mining merupakan suatu disiplin ilmu yang berada pada disiplin ilmu kecerdasan komputasi dan *data mining* untuk pemodelan proses dan analisis [2]. Dengan kata lain *Process mining* merupakan suatu metode untuk menggali informasi yang terkandung dalam sebuah proses bisnis.

Terdapat tiga jenis *process mining*, pertama *Discovery* merupakan pembentukan sebuah model proses berdasarkan *dependency* antar *event* dalam *event log* suatu proses bisnis. Ke-dua, *Conformance Checking* adalah tipe yang menitikberatkan pada evaluasi antara suatu model proses yang dihasilkan dari tahap *discovery* dengan event lognya. Terakhir adalah *Enhancement* adalah tipe yang berfokus untuk mengembangkan, memperbaharui, atau membuat

sebuah model proses yang baru dari model proses yang sebelumnya.

2.2 Heuristic Miner Algorithm

Algoritma *Heuristic Miner* merupakan salah satu contoh algoritma dalam *Process Mining* yang digunakan dalam tahap *discovery*. Algoritma ini, lebih menitikberatkan pada penghitungan frekuensi *dependency* antar event dan keterurutannya (traces) dalam membangun suatu model proses [1]. Dalam tahapan pengerjaannya terdapat beberapa tahapan umum, yaitu :

2.2.1 Membuat Dependency Graph

Dependency Graph merupakan suatu model yang merepresentasikan *dependency* (kausalitas) antar event. Dalam membangun dependency graph terdapat tiga tahap yang dilakukan, yaitu membuat matriks *dependency*, *dependency length-one loop*, dan *dependency length-two loop* [5].

1. Persamaan menghitung matrix dependency

$$a \Rightarrow_w b = \left(\frac{|a >_w b| - |b >_w a|}{|a >_w b| + |b >_w a| + 1} \right) \dots \text{(persamaan 1)}$$

Keterangan :

\Rightarrow

- $a \Rightarrow_w b$: Dependency antara event 'a' dan event 'b'. Nilainya berkisar antara -1 sampai 1.
- $a >_w b$: Frekuensi event 'a' mengikuti event 'b'.
- $b >_w a$: Frekuensi event 'b' mengikuti event 'a'.

2. Persamaan menghitung length-one loop

$$a \Rightarrow_w a = \left(\frac{|a >_w a|}{|a >_w a| + 1} \right) \dots \text{(persamaan 2)}$$

Keterangan :

\Rightarrow

- $a \Rightarrow_w a$: Dependency antara event 'a' dengan event 'a'. Nilainya berkisar antara -1 sampai 1.
- $a >_w a$: Frekuensi event 'a' mengikuti event 'a'.

3. Persamaan menghitung length-two loop

$$a \Rightarrow_{2w} b = \left(\frac{|a >>_w b| + |b >>_w a|}{|a >>_w b| + |b >>_w a| + 1} \right) \dots \text{(persamaan 3)}$$

Keterangan :

\Rightarrow

- $a \Rightarrow_{2w} b$: Dependency antara event 'a' dan event 'b'. Nilainya berkisar antara -1 sampai 1.
- $a >>_w b$: Frekuensi event 'a' mengikuti event 'b'.
- $b >>_w a$: Frekuensi event 'b' mengikuti event 'a'.

2.2.2 Causal Matrix

Dalam kenyataan, suatu proses dapat dilakukan / dikerjakan secara bersamaan (paralel), namun dalam suatu *event log* sangat sulit untuk ditentukan apakah proses tersebut berjalan secara sekuensial atau secara paralel. Untuk menghindari kesalahan visualisasi dalam model prosesnya, pada *Heuristic Miner* ini digunakan *Causal Matrix* untuk merepresentasikan model prosesnya [5].

Pembentukan *Causal Matrix* dilakukan setelah pembentukan *Dependency Graph*, dimana dari *dependency graph* dapat diketahui bahwa suatu event memiliki percabangan atau tidak. Dalam kausal matriks ini terdapat dua tipe / bentuk *non-observal activities*, yaitu AND dan XOR.

Non-observal activity AND memiliki arti bahwa suatu percabangan aktivitas dapat dikerjakan secara paralel / bersamaan, sedangkan *non-observal activity* XOR memiliki arti bahwa dalam suatu percabangan aktivitas hanya boleh memilih salah satu jalur saja. Dalam persamaan (4) ditunjukkan cara untuk menentukan *non-observal activity* dari suatu *dependency antar event*

$$a \Rightarrow_w b \wedge c = \left(\frac{|b >_w c| + |c >_w b|}{|a >_w b| + |a >_w c| + 1} \right) \dots \text{(persamaan 4)}$$

Keterangan

⇒

$a \Rightarrow_w b \wedge c$: Hubungan *non-observal activity* antara event 'b' dan event 'c' yang berasal dari event / percabangan 'a'.

$b >_w c$: Frekuensi event 'b' mengikuti event 'c'.

$c >_w b$: Frekuensi event 'c' mengikuti event 'b'.

$a >_w b$: Frekuensi event 'a' mengikuti event 'b'.

$a >_w c$: Frekuensi event 'a' mengikuti event 'c'.

2.3 Conformance Checking

Conformance checking merupakan tipe kedua dari *Process Mining*, dimana pada tahap ini dilakukan pengecekan / pencocokan antara model proses yang telah dibentuk dari tahap *Discovery* terhadap *event log*-nya. Terdapat beberapa pendekatan dalam melakukan *Conformance Checking* salah satunya yang akan digunakan dalam penelitian ini adalah dengan menggunakan pendekatan *Artificially Generated Negative Events* (AGNEs) yang pernah dilakukan pada penelitian sebelumnya [6] yang memungkinkan sifat-sifat *F-Measure* untuk mengevaluasi model proses. Berdasarkan pembentukan *negative event* ini, Goedertier merumuskan persamaan tabel *Confusion Matrix* agar dapat dicek keterkaitannya antara *positive*

event dan *negative event* dalam *event log* serta model proses-nya [7]:

Tabel 1 : *Confusion Matrix*

	Actual Positive	Actual Negative
Prediction Positive	True Positive (TP)	False Positive (FP)
Prediction Negative	False Negative (FN)	True Negative (TN)

Persamaan untuk mengukur Recall, Precision, dan F-Measure :

$$F\text{-Measure} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \dots \text{(persamaan 5)}$$

$$\text{Precision} = \frac{TP}{TP + FP} \dots \text{(persamaan 6)}$$

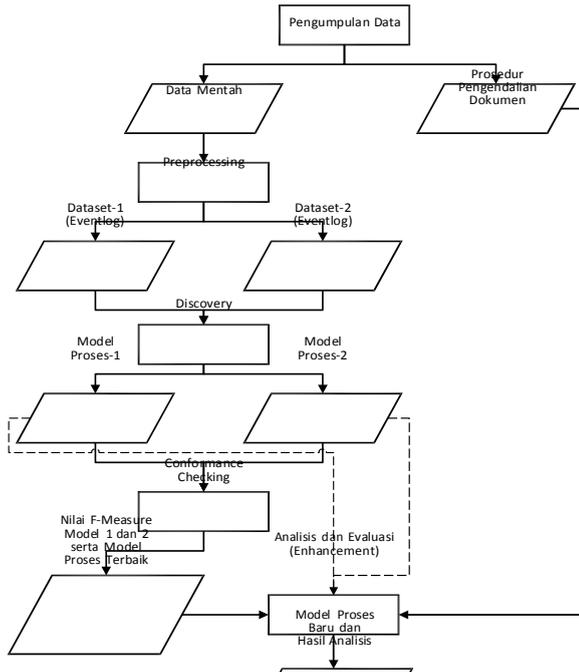
$$\text{Recall} = \frac{TP}{TP + FN} \dots \text{(persamaan 7)}$$

2.4 Enhancement

Tahapan *process mining* terakhir adalah *enhancement* yaitu proses perluasan atau penambahan model proses yang ada menggunakan informasi terkait yang terdapat di *event log* [2]. Terdapat dua jenis *enhancement*, pertama *repair*, proses memodifikasi model proses yang ada menjadi model proses yang lebih merepresentasikan kenyataan / sesuai harapan. Jenis yang kedua adalah *extension*, proses menambahkan perspektif / sudut pandang baru terhadap model proses dengan mengkorelasikan model proses dengan *event log* [2], contohnya adalah dengan menggunakan atribut "time" pada *event log* dapat diketahui *bottleneck* pada model proses / menggunakan atribut "originator" untuk menentukan hak akses setiap aktivitasnya.

3. Metodologi Penelitian

Berikut ini adalah gambaran umum penelitian yang akan dilakukan, ditunjukkan dalam Gambar 1.



Gambar 1 : Gambaran Umum Proses Penelitian

Berdasarkan Gambar 1, langkah pertama adalah pengumpulan data dimana akan didapatkan dua jenis data, yaitu data mentah (SQL) dan Prosedur Pengendalian Dokumen. Selanjutnya data mentah di preprocessing, sehingga dihasilkan dua jenis dataset, dataset 1 memiliki awalan aktivitas ‘A’ dan akhirnya aktivitas apapun dan dataset 2 memiliki awalan aktivitas ‘A’ dan akhiran ‘H’. Kemudian setiap dataset dibentuk dalam tahap *Discovery*, terbentuklah model proses 1 dan model proses 2 untuk kemudian dianalisis nilai recall, precision, dan f-measurenya pada tahap *Conformance Checking*. Langkah terakhir adalah *Enhancement*, memperbaharui model proses dan menganalisis informasi lain dari event log.

4. Implementasi dan Analisis
4.1 Padanan Aktivitas Eventlog dengan Probis

Pada tabel 2 ini akan dijelaskan mengenai padanan aktivitas yang ada pada event log dengan proses bisnis Pengendalian Dokumen :

Tabel 2 : Padanan Aktivitas Event log dengan Proses Bisnis

Proses Bisnis	Event Pada I-Adore	Id Event
Usulan Perubahan / Penambahan Dokumen	new document request	A
	document upload	B
Membuat Rencana Pembahasan Perubahan Dokumen	document request evaluate	C
Membuat Draft Perubahan / Penambahan Dokumen	-	-
Pembahasan Draft Perubahan / Penambahan Dokumen	-	-
Merevisi Draft Dokumen	document update	D
Pengesahan Dokumen	document request approval	E
Memperbaharui Daftar Induk	-	-
Penarikan Dokumen Kadaluarsa & Mendistribusikan Dokumen Hasil Revisi	undistributed document	F
	distribute document	G
	accept document	H
Penyimpanan Dokumen Kadaluarsa & Pemusnahan Dokumen Kadaluarsa	-	-

4.2 Discovery dan Conformance Checking

Langkah awal dalam *Discovery* adalah membentuk matrix dependency, matrix length one loop, dan matrix length two loop.

1. Matrix Frekuensi Dependency Aktivitas

Tabel 3 : Matrix Frekuensi Dependency

	A	B	C	D	E	F	G	H
A	14	159	30	4	12	0	0	1
B	7	156	0	31	0	0	70	6
C	1	0	7	11	0	0	0	0
D	4	0	1	19	0	1	19	25
E	0	0	0	1	0	0	1	0
F	0	0	0	1	0	93	3	0
G	3	0	0	32	0	2	40	54
H	0	0	0	0	0	1	2	168

2. Matrix Dependency

Tabel 4 : Matrix Dependency

	A	B	C	D	E	F	G	H
A	0	0.91	0.91	0	0.92	0	-0.75	0.5
B	-0.91	0	0	0.97	0	0	0.99	0.86
C	-0.91	0	0	0.77	0	0	0	0
D	0	-0.97	-0.77	0	-0.5	0	-0.25	0.96
E	-0.92	0	0	0.5	0	0	0.5	0
F	0	0	0	0	0	0	0.17	-0.5
G	0.75	-0.99	0	0.25	-0.5	-0.17	0	0.91
H	-0.5	-0.86	0	-0.96	0	0.5	-0.91	0

3. Matrix Length One Loop

Tabel 5 : Matrix Length One Loop

	A	B	C	D	E	F	G	H
A	0.8	0	0	0	0	0	0	0
B	0	0.99	0	0	0	0	0	0
C	0	0	0.88	0	0	0	0	0
D	0	0	0	0.95	0	0	0	0
E	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0.99	0	0
G	0	0	0	0	0	0	0.98	0
H	0	0	0	0	0	0	0	0.99

4. Matrix Length Two Loop

Tabel 6 : Matrix Length Two Loop

	A	B	C	D	E	F	G	H
A	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0
D	0	0	0	0.5	0	0	0	0
E	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0.99	0	0
G	0	0	0	0	0	0	0.75	0
H	0	0	0	0	0	0	0	0.99

4.3 Skenario dan Sub Skenario Pengujian

Tahapan selanjutnya dari *Discovery* adalah pembentukan dependency graph dan causal matrix, untuk membuatnya diperlukan penentuan nilai threshold pembentuk model proses. Dalam hal ini dilakukan sub skenario pengujian terhadap penentuan nilai threshold terbaik, sehingga akan didapatkan model proses terbaik pula. Penentuan nilai threshold terbaik dilakukan ditahap *Conformance Checking*, yaitu menentukan nilai

Recall, *Precision*, dan *F-Measure* untuk setiap sub skenario pengujian. Pada tabel 7, terdapat enam threshold yang akan dilakukan sub skenario pengujiannya beserta nilai awalnya :

Tabel 7 : Parameter Threshold

Nama Parameter Threshold	Kode Parameter Threshold	Nilai Awal
Dependency Threshold	DTH	-
Positive Observation Threshold	POT	10
Relative to Best Threshold	RBT	0.3
Loop Length-One Threshold	LOT	0.8
Loop Length-Two Threshold	LTT	0.8
AND-XOR Threshold	AXT	0.8

Pada tahap ini juga, dilakukan skenario pengujian terhadap dua jenis dataset dan terhadap aplikasi ProM (aplikasi khusus untuk *process mining*). Skenario pengujian tersebut bertujuan untuk menganalisis perbedaan antara jenis dataset serta menilai jenis dataset mana yang sebenarnya cocok untuk digunakan dalam penelitian ini. Untuk pengujian terhadap aplikasi ProM bertujuan untuk membandingkan model proses yang dihasilkan dari aplikasi yang penulis buat dengan aplikasi ProM apakah model proses yang terbentuk sama atau tidak.

Konsep sub skenario pengujian nilai threshold adalah dengan menginputkan perbedaan nilai pada setiap threshold, dimana untuk setiap fase threshold diambil satu nilai yang merepresentasikan nilai *F-Measure*, *Precision*, dan *Recall* terbesar, untuk kemudian nilai tersebut digunakan pada pencarian nilai threshold selanjutnya. Jumlah pengujian setiap parameter thresholdnya adalah 10 kali pengujian. Mulai dari nilai terendah hingga yang paling tinggi.

4.4 Hasil Pengujian dan Analisis Sub Skenario Uji

Dari sub skenario pengujian diatas, maka dapat ditentukan nilai threshold terbaik yang dapat digunakan untuk membentuk model proses terbaik juga, jika nilai recall, preicison, dan f-measurenya sama maka bebas untuk memilih nilai thresholdnya. Berikut hasil pengujian dari setiap sub skenario pengujian :

1. Sub Skenario Pengujian 1

Tabel 8 : Hasil Perhitungan Sub Skenario Uji 1

No	DTH	Recall	Precision	F-Measure
1	0.1	0.9249	0.9881	0.9554
2	0.2	0.9249	0.9881	0.9554
3	0.3	0.9249	0.9881	0.9554
4	0.4	0.9249	0.9881	0.9554
5	0.5	0.9249	0.9881	0.9554
6	0.6	0.9249	0.9881	0.9554
7	0.7	0.9249	0.9881	0.9554
8	0.8	0.9155	0.988	0.9504
9	0.9	0.9155	0.988	0.9504
10	1	0.5884	0.9663	0.7314
Nilai DTH terbaik yang diambil				0.7

2. Sub Skenario Pengujian 2

Tabel 9 : Hasil Perhitungan Sub Skenario Uji 2

No	POT	Recall	Precision	F-Measure
1	1	0.9325	0.9882	0.9596
2	4	0.93	0.9891	0.9586
3	7	0.9249	0.9881	0.9554
4	10	0.9249	0.9881	0.9554
5	13	0.9052	0.987	0.9443
6	16	0.9052	0.987	0.9443
7	19	0.9052	0.987	0.9443
8	21	0.9052	0.987	0.9443
9	24	0.9052	0.987	0.9443
10	27	0.8839	0.9885	0.9333
Nilai POT terbaik yang diambil				1

3. Sub Skenario Pengujian 3

Tabel 10 : Hasil Perhitungan Sub Skenario Uji 3

No	RBT	Recall	Precision	F-Measure
1	0.01	0.6482	0.9669	0.7761
2	0.06	0.696	0.9714	0.8109
3	0.11	0.9155	0.988	0.9504
4	0.16	0.9206	0.989	0.9536
5	0.21	0.9206	0.989	0.9536
6	0.26	0.9325	0.9882	0.9596
7	0.31	0.9325	0.9882	0.9596
8	0.36	0.9325	0.9882	0.9596
9	0.41	0.9325	0.9882	0.9596
10	0.46	0.9325	0.9882	0.9596
Nilai RBT terbaik yang diambil				0.26

4. Sub Skenario Pengujian 4

Tabel 11 : Hasil Perhitungan Sub Skenario Uji 4

No	LOT	Recall	Precision	F-Measure
1	0.1	0.9325	0.9882	0.9596
2	0.2	0.9325	0.9882	0.9596
3	0.3	0.9325	0.9882	0.9596
4	0.4	0.9325	0.9882	0.9596
5	0.5	0.9325	0.9882	0.9596
6	0.6	0.9325	0.9882	0.9596
7	0.7	0.9325	0.9882	0.9596
8	0.8	0.9325	0.9882	0.9596
9	0.9	0.9129	0.9816	0.946
10	1	0.7293	0.7274	0.7284
Nilai LOT terbaik yang diambil				0.8

5. Sub Skenario Pengujian 5

Tabel 12 : Hasil Perhitungan Sub Skenario Uji 5

No	LTT	Recall	Precision	F-Measure
1	0.1	0.9325	0.9882	0.9596
2	0.2	0.9325	0.9882	0.9596
3	0.3	0.9325	0.9882	0.9596
4	0.4	0.9325	0.9882	0.9596
5	0.5	0.9325	0.9882	0.9596
6	0.6	0.9325	0.9882	0.9596
7	0.7	0.9325	0.9882	0.9596
8	0.8	0.9325	0.9882	0.9596
9	0.9	0.9325	0.9882	0.9596
10	1	0.9325	0.9882	0.9596
Nilai LTT terbaik yang diambil				0.8

6. Sub Skenario Pengujian 6

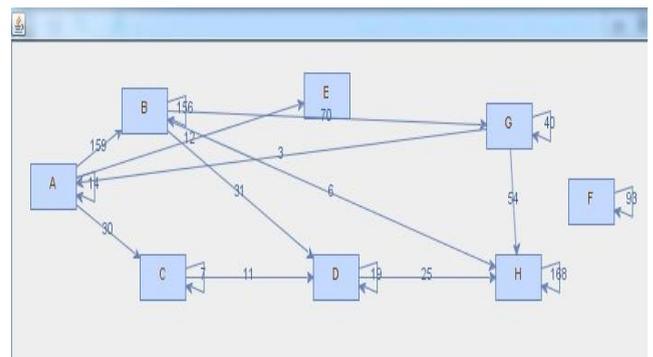
Tabel 13 : Hasil Perhitungan Sub Skenario Uji 6

No	AXT	Recall	Precision	F-Measure
1	0.1	0.9787	0.9948	0.9867
2	0.2	0.9787	0.9948	0.9867
3	0.3	0.9787	0.9948	0.9867
4	0.4	0.9787	0.9948	0.9867
5	0.5	0.9787	0.9948	0.9867
6	0.6	0.9342	0.9883	0.9605
7	0.7	0.9334	0.9882	0.96
8	0.8	0.9334	0.9882	0.96
9	0.9	0.9334	0.9882	0.96
10	1	0.9308	0.9846	0.957
Nilai AXT terbaik yang diambil				0.5

Dari hasil sub skenario uji diatas dapat dinyatakan bahwa nilai-nilai threshold pembentuk model proses terbaik dalam penelitian ini untuk dataset 1 adalah DTH = 0.7, POT = 1, RBT = 0.26, LOT = 0.8, LTT = 0.8, dan AXT = 0.5.

Berikut bentuk dependency graph dan causal matrix yang terbentuk setelah dilakukan input nilai threshold :

1. Dependency Graph



Gambar 2 : Dependency Graph

2. Causal Matrix

Tabel 14 : Causal Matrix

Aktivitas	Input	Output
A	A	$(A \wedge B) \vee (A \wedge C) \vee E$
B	$A \wedge B$	$B \vee (D \wedge G) \vee (D \wedge H) \vee (H \wedge D)$
C	$A \wedge C$	$C \wedge D$
D	$B \wedge C \wedge D$	$D \wedge H$
E	A	-
F	F	F
G	$B \wedge G$	$A \vee (G \wedge H)$
H	$D \wedge G \wedge H$	H

4.5 Hasil Pengujian dan Analisis Skenario Pengujian

Tahapan ini menganalisis hasil pengujian terhadap Skenario Pengujian dataset 1 dan dataset 2 serta pengujian terhadap aplikasi ProM. Nilai threshold pembangun model proses yang digunakan diambil dari hasil Sub Skenario pengujian 1 dimana nilai thresholdnya adalah $DTH = 0.7$, $POT = 1$, $RBT = 0.26$, $LOT = 0.8$, $LTT = 0.8$, dan $AXT = 0.5$.

Tujuan dilakukan skenario pengujian ini adalah untuk mengetahui perbandingan antara dataset 1 yang diawali oleh aktivitas 'A' dan diakhiri oleh aktivitas apapun (dokumen belum selesai) dengan dataset 2 yang diawali oleh aktivitas 'A' dan diakhiri oleh aktivitas 'H' (dokumen sudah selesai) dan menganalisis jenis dataset mana yang lebih baik untuk diterapkan dalam penelitian, serta apakah model proses yang dihasilkan dari aplikasi yang telah penulis buat sudah sama seperti aplikasi ProM.

1. Perbandingan Jenis Dataset

Setelah dilakukan pengujian terhadap dua jenis dataset maka dihasilkan nilai recall, precision, dan f-measurenya sebesar 0.9787 (recall), 0.9948 (precision), dan 0.9867 (f-measure) untuk dataset 1 sedangkan untuk dataset 2 memiliki nilai sebesar 0.9716 (recall), 0.9875 (precision), dan 0.9795 (f-measure). Maka jenis dataset pada skenario pengujian 1 / dataset dengan "dokumen belum selesai" lebih baik 0.0072 pada perbedaan nilai f-measurenya. Namun jika diuji dengan batas threshold terbaik dari dataset 2 maka jenis dataset 2 memiliki nilai yang lebih besar dibanding dataset 1 dengan nilai untuk dataset 1 sebesar 0.9667 (recall), 0.9921 (precision), dan 0.9792 (f-measure) dan untuk dataset 2 sebesar 0.9741 (recall), 0.9875 (precision), dan 0.9808 (f-measure). (Hasil nilai threshold terbaik dataset 2 dapat dilihat di lampiran 1). Sehingga kesimpulannya adalah tidak dapat ditentukan jenis dataset mana yang lebih baik dalam penelitian ini, dikarenakan masing-masing dataset memiliki karakteristik yang berbeda.

2. Perbandingan Program

Perbandingan antara model proses yang dihasilkan dari aplikasi penulis dengan hasil model proses yang dihasilkan dari aplikasi ProM ditunjukkan pada tabel 15.

Tabel 15 : Perbandingan Hasil Model Proses

No	Dependency Penulis	Hasil Aplikasi	Dependency ProM	Hasil Aplikasi
	Aktivitas 1	Aktivitas 2	Aktivitas 1	Aktivitas 2
1	A	A	A	A
2	A	B	A	B
3	A	C	A	C
4	A	E	A	D
5	B	B	A	E
6	B	D	B	A
7	B	G	B	B
8	B	H	B	D
9	C	C	B	G
10	C	D	B	H
11	D	D	C	C
12	D	H	C	D
13	F	F	D	A
14	G	A	D	D
15	G	G	D	G
16	G	H	D	H
17	H	H	E	E
18			F	F
19			F	G
20			G	A
21			G	D
22			G	G
23			G	H
24			H	H

Dari hasil pemaparan model proses yang terbentuk antara aplikasi yang dibuat penulis dan hasil dari aplikasi ProM terdapat tujuh perbedaan dependency yang terbentuk pada model proses yaitu $A \rightarrow D$, $B \rightarrow A$, $D \rightarrow A$, $D \rightarrow G$, $E \rightarrow E$, $F \rightarrow G$ dan $G \rightarrow D$. Hal ini disebabkan karena pada proses pembentukan model proses di aplikasi ProM terdapat satu buah dependency yang ditambahkan sedangkan dalam aplikasi yang penulis buat tidak ditambahkan, yaitu dependency *Long Distance Threshold* (LDT). LDT merupakan dependency yang menghitung relasi antar event yang tidak berdekatan (*non-local behavior*) / tidak berurutan sesuai dengan event log sehingga dimungkinkan terbentuknya beberapa dependency seperti diatas.

4.6 Enhancement

Pada tahap ini dilakukan proses perluasan / penambahan terhadap model proses yang telah dihasilkan. Terdapat dua jenis enhancement, pertama *repair* dan kedua *extension*, kemudian ditambah dengan rekomendasi untuk perubahan Prosedur Pengendalian Dokumen.

1. Repair

Pada jenis repair ini penulis mencocokkan model proses hasil *discovery* dengan proses bisnis Prosedur Pengendalian Dokumen,

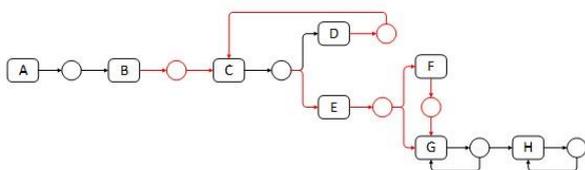
apakah telah menggambarkan alur aktivitas yang sesuai kebutuhan. Dalam proses bisnisnya terdapat beberapa kemungkinan pola yang terbentuk.

- a. Dokumen baru tanpa update
[A→B→C→E→G→H]
- b. Dokumen baru dengan update
[A→B→C→D→C→E→G→H]
- c. Dokumen lama tanpa update
[A→B→C→E→F→G→H]
- d. Dokumen lama dengan update
[A→B→C→D→C→E→F→G→H]

Dari pemaparan beberapa kemungkinan skema proses bisnis yang ada maka pada tahap repair ini, digabungkan antara hubungan yang ada pada model proses dengan yang ada pada proses bisnis, dimana penggabungan aktivitas hanya akan menghapus dependency yang tidak sesuai dengan proses bisnis serta menambahkan jika belum ada pada model proses. Tabel 15 menunjukkan hubungan aktivitas yang mengalami perubahan dalam model proses dan Gambar 3 menampilkan visualisasi model prosesnya :

Tabel 16 : Hubungan Aktivitas Repair

Dependency yang Dihapus		Dependency yang Ditambah	
Aktivitas 1	Aktivitas 2	Aktivitas 1	Aktivitas 2
A	A	B	C
A	C	C	D
A	E	C	E
B	B	D	C
B	D	E	F
B	G	E	G
B	H	F	G
C	C		
D	D		
D	H		
F	F		



Gambar 3 : Model Proses Repair

2. Extension

Pada bagian ini akan dilakukan penggalian / penambahan informasi lain dari model proses yang telah dibentuk, tujuannya adalah sebagai bahan masukan / rekomendasi terhadap model proses yang telah terbentuk. Dalam penelitian ini informasi yang akan digali sebagai berikut :

- a. Penentuan *Bottleneck*
Informasi mengenai waktu yang dibutuhkan suatu dokumen dalam

melewati setiap tahapannya. Tujuannya adalah untuk dapat menentukan *bottleneck* yang dapat terjadi pada model proses. Dalam penelitian ini *bottleneck*

dikategorikan kedalam dua jenis, yaitu short dan long. Cara menghitungnya adalah dengan mengakumulasikan waktu / timestamp dari setiap dependency aktivitas dibagi dengan jumlah dependency aktivitasnya.

- o *Short Bottleneck*
Batas waktu minimal *bottleneck*

antar setiap relasi aktivitas. Dimana rentang waktunya berkisar antara waktu normal dan long *bottleneck*.

- o *Long Bottleneck*
Batas waktu minimal *bottleneck* antar setiap relasi aktivitas. Dimana rentang waktunya pada rentang setelah *short Bottleneck*.

Pada Tabel 16 dan 17 menunjukkan hasil perhitungan rata-rata waktu eksekusi antara *dependency* aktivitas satu dengan yang lainnya (dalam satuan menit dan hari).

Tabel 17 : Waktu Dependency Antar Aktivitas (menit)

	A	B	C	D	E	F	G	H
A	21173	11	55171	105086	960	0	0	378798
B	16562	1	0	49900	0	0	110091	619057
C	256	0	4846	255900	0	0	0	0
D	32122	0	2	18373	0	190	177384	12158
E	0	0	0	21536	0	0	402945	0
F	0	0	0	10124	0	0	2	0
G	201470	0	0	8109	0	5892	472924	2077
H	0	0	0	0	0	72	288754	20614

Tabel 18 : Waktu Dependency Antar Aktivitas (hari)

	A	B	C	D	E	F	G	H
A	14.7	0.007	38.31	72.97	0.667	0	0	263
B	11.5	0	0	34.65	0	0	76.45	430
C	0.17	0	3.36	177.7	0	0	0	0
D	22.3	0	0.01	12.75	0	0.13	123.1	844
E	0	0	0	14.95	0	0	279.8	0
F	0	0	0	7	0	0	0.001	0
G	140	0	0	5.63	0	4.09	328.4	1.44
H	0	0	0	0	0	0.05	200.5	14.3

Dalam penelitian ini perhitungan *bottleneck* tidak berdasarkan model proses yang dihasilkan dalam tahap *discovery*, namun berdasarkan model proses yang akan dijadikan rekomendasi (Gambar 3). Hasil analisis *bottleneck* disajikan pada tabel 18.

Tabel 19 : Tabel Bottleneck

No	Aktivitas		Batas Short Bottleneck	Batas Long Bottleneck	Keterangan
	1	2			
1	A	B	5 menit	15 menit	Short Bottleneck
2	B	C	3 hari	7 hari	Normal
3	C	D	30 hari	90 hari	Long Bottle Neck
4	C	E	3 hari	7 hari	Normal
7	E	F	7 hari	14 hari	Normal
8	E	G	7 hari	14 hari	Long Bottle Neck
9	F	G	7 hari	14 hari	Normal
10	G	G	7 hari	14 hari	Long Bottle Neck
11	G	H	7 hari	14 hari	Normal
12	H	H	7 hari	14 hari	Long Bottle Neck

Berdasarkan hasil analisis bottle neck maka proses $A \rightarrow B$ termasuk short bottle neck, penyebabnya lebih dipengaruhi oleh *bandwidth* sistem yang lama. Proses $C \rightarrow D$ termasuk long bottle neck, ini mengindikasikan bahwa proses evaluasi memakan waktu yang lama. Kemudian proses $E \rightarrow G$ mengindikasikan bahwa proses persetujuan dokumen memakan waktu yang lama agar dapat didistribusikan. Kemudian proses $G \rightarrow G$ mengisyaratkan bahwa proses distribusi dokumen terhadap unit memakan waktu yang cukup lama. Dan proses $H \rightarrow H$ mengindikasikan waktu target distribusi (unit) dalam menerima dokumen cukup lama.

b. Penentuan kevalidan user

Extension selanjutnya adalah informasi mengenai kevalidan aktor / user yang berperan dalam Prosedur Pengendalian Dokumen, apakah telah sesuai dengan perannya masing-masing. Dalam penelitian ini diperiksa kesesuaian antara originator dalam event log dengan data user dalam proses bisnis.

Tabel 20 : Daftar Kevalidan User I-Adore

Id Event	Id User Event	User Pada Proses Bisnis	Keterangan
A	3,5,23,24,26,42,62,70,71,77,83,91,98,101,119,120,128,255,529,553	Pemilik Proses	Valid
B	3,5,23,26,62	Pemilik Proses	Tidak Valid
C	3	QMR	Tidak Valid
D	1,63	Pemilik Proses	Tidak Valid
E	3	QMR	Tidak Valid
F	3,62	PPD	Tidak Valid
G	3,62	PPD	Tidak Valid
H	28,33,36,37,42,60,67,74,77,86,96,103,104,112,119,120,121,123,125,147,168,255,550,556,606	Pemilik Proses / Target	Valid

Dilihat dari kevalidan user yang menggunakan sistem I-Adore, terdapat kejanggalan dimulai dari aktivitas “document upload” (B) yang seharusnya dilakukan Pemilik Proses namun pada log dilakukan oleh Administrator dan PPD, kemudian pada “document request evaluate” (C) dan “document request approval” (E) dimana aktivitas tersebut seharusnya dilakukan

oleh QMR namun pada log dilakukan oleh PPD, kemudian pada aktivitas document update (D) seharusnya dilakukan oleh Pemilik Proses namun pada log dilakukan oleh Administrator, lalu yang terakhir adalah aktivitas “undistributed document” (F) dan “distribute document” (G) yang seharusnya dilakukan oleh PPD namun terdapat id Administrasi yang juga melakukannya.

4.7 Rekomendasi

Setelah dilakukan proses *Enhancement* dapat ditentukan beberapa rekomendasi yang akan diajukan kepada pihak SPM berdasarkan data-data yang dihasilkan dalam penelitian ini. Berikut rekomendasi yang dapat diberikan dari hasil penelitian ini :

4.7.1 Rekomendasi Model Proses

Dari model proses yang dihasilkan pada tahap *Discovery* terdapat beberapa relasi antar aktivitas yang secara proses bisnis tidak diperbolehkan, ditunjukkan pada tabel 20. Untuk itu berikut beberapa rekomendasi yang dapat diberikan terkait hal tersebut :

Tabel 21 : Tabel Rekomendasi Model Proses

Aktivitas		Rekomendasi
1	2	
A	A	Simpan data-data dokumen, terutama untuk aktivitas new request document. Lakukan pengecekan terhadap nama dokumen / nomor dokumen, jika sudah ada nama yang sama maka keluarkan notifikasi.
A	C	Jadikan aktivitas upload document hanya satu-satunya aktivitas setelah aktivitas new request document. Serta pada menu document request evaluate hanya akan muncul nama-nama document hasil dari document upload dan hasil update saja.
A	E	Begitu pula dengan ini, setelah proses new document request menu yang muncul hanyalah menu untuk upload document tidak untuk yang lain.
B	B	Mungkin aktivitas ini terjadi akibat <i>bandwidth</i> jaringannya yang lambat, karena berdasarkan tabel dependency

		waktu rata-rata pemrosesan B→B adalah 1 menit, sehingga dimungkinkan user menekan tombol update beberapa kali. Untuk untuk rekomendasinya adalah tambahkan <i>bandwidth</i> pada jaringannya.
B	D	Sama seperti poin A-C, menu update hanya akan membaca nama dokumen yang telah di evaluasi.
B	G	Sama seperti poin A-C dan B-D, menu distribute document hanya akan menerima dokumen dari aktivitas document request approval.
B	H	Sama seperti poin A-C dan B-D, menu distribute document hanya akan menerima dokumen dari aktivitas document request approval.
C	C	Sama seperti poin sebelumnya, dimana jika dokumen telah di evaluasi maka dokumen tersebut tidak bisa di evaluasi lagi, harus menunggu proses perbaikan / update terlebih dahulu. Dapat juga dihilangkan nama dokumen yang telah dilakukan evaluate, sehingga setelah dievaluate nama document tidak akan muncul kembali.
D	D	Setiap nama dokumen yang telah diupdate akan dikirim ke document evaluate dan setelah dikirim nama document dapat dihilangkan dari menu update.
D	H	Sama seperti poin sebelumnya, caranya adalah dengan hanya menampilkan menu untuk mengirim dokumen hasil update ke menu evaluate saja tidak untuk yang lain.
F	F	Jika ini kesalahan dari sistem, maka jika tombol undistribut telah ditekan maka request yang dikirim cukup dibaca sekali saja. Dikhawatirkan relasi ini diakibatkan oleh sistem "lag" karena dilihat dari tabel 34, rata-rata waktu prosesnya adalah 0 menit. Sehingga memungkinkan user untuk meng- <i>click</i> tombolnya berulang kali.

		Rekomendasinya adalah tambahkan <i>bandwidth</i> untuk jaringannya.
--	--	---

Sehingga bentuk rekomendasinya adalah create new document → document upload → document request evaluate → document update → document request approval → undistributed document → distribute document → accept document. Kemudian ditambahkan dengan looping length one pada distribute document → distribute document dan accept document → accept document. Untuk looping length two terjadi pada document request evaluate → document update → document request evaluate.

4.7.2 Rekomendasi Penanganan *Bottleneck*

Dari analisis *bottleneck* terdapat lima dependency aktivitas yang termasuk kedalam kategori *bottleneck*.

- a. Proses A ke B
Hal ini mungkin disebabkan oleh *bandwidth* internet yang kurang, sehingga proses pengaksesannya memakan waktu yang lebih lama. Sehingga rekomendasinya adalah ditambah *bandwidth* untuk mengakses ke sistemnya.
- b. Proses C ke D
Hal ini disebabkan proses evaluasinya dilakukan oleh sedikit orang, sehingga memakan waktu yang lama. Rekomendasinya adalah dengan menambah jumlah evaluatornya.
- c. Proses E ke G
Hal ini disebabkan oleh jumlah QMR yang terbatas sehingga terjadi penumpukan dokumen yang harus di approval. Jadi rekomendasinya adalah menambah jumlah QMR.
- d. Proses G ke G
Hal ini mungkin terjadi dikarenakan kurang koordinasinya antara pemilik proses dan PPD dalam menentukan target pendistribusian. Rekomendasinya adalah dengan terlebih dahulu membuat list target mana saja yang akan mendapat distribusi antara pihak-pihak yang terkait, sehingga ketika dokumen telah di approval, PPD dapat mendistribusikan dengan cepat.
- e. Proses H ke H
Hal ini mungkin disebabkan karena target dari distribusi tidak mengetahui / jarang membuka sistem. Rekomendasi untuk proses ini adalah berikan notifikasi setiap harinya untuk unit

yang mendapat distribusi dokumen agar segera menerima / *accept* dokumennya.

4.7.3 Rekomendasi Penanganan User

Rekomendasi terakhir yang dapat dibentuk dari hasil penelitian ini adalah tentang pembagian tugas / peran tiap user dalam sistem. Dapat dilakukan dengan memisahkan interface yang berbeda-beda untuk setiap tipe user sehingga tidak dimungkinkan user lain untuk mengakses / mengerjakan tugas yang berbeda, kemudian penggunaan login dan logout diperketat kembali, pendaftaran user baru yang tidak memungkinkan seseorang memiliki dua akun / lebih, serta limitasi peran administrator dalam sistem karena administrator dalam prosedur pengendalian dokumen ini tidak dijelaskan secara rinci apa peran dan hak nya dalam sistem.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Berdasarkan pada pengujian yang telah dilakukan pada bab 4 dari buku tugas akhir ini, diperoleh kesimpulan sebagai berikut:

1. Tahapan pengerjaan dari Algoritma Heuristic adalah
 - a. Membuat matrix dependency, matrix length-one loop, dan matrix length-two loop.
 - b. Membuat dependency graph dengan menentukan nilai-nilai threshold pembangunannya.
 - c. Membangun causal matrix dengan membuat tabel non-observal activity terlebih dahulu.
 - d. Dapat dibentuk kedalam petrinet untuk memudahkan dalam visualisasi.
2. Bentuk model proses terbaik yang dihasilkan pada penelitian ini adalah model proses yang memiliki threshold sebagai berikut DTH = 0.7, POT = 1, RBT = 0.26, LOT = 0.8, LTT = 0.8, dan AXT = 0.5.
3. Parameter yang mempengaruhi pembangunan model proses adalah nilai-nilai threshold yaitu DTH, POT, RBT, LOT, dan AXT. Dimana threshold tersebut menilai dependency yang terdapat pada Matrix Dependency, Matrix Length One Loop, dan Matrix Length Two Loop. Sedangkan threshold LTT tidak mempengaruhi pembangunan model proses, ini dikarenakan nilai-nilai dependency pada matrix length-two loop telah ada di matrix lain seperti di matrix dependency dan matrix length-one loop. Sehingga meskipun nilai pada matrix length-two

loop tidak melewati threshold LTT hal tersebut tidak merubah nilai recall, precision, dan f-measurenya, karena sudah tertangani oleh matrix dependency ataupun matrix length-one loop.

4. Bentuk model proses baru yang diajukan kepada SPM untuk Prosedur Pengendalian Dokumen di Telkom University adalah yang sesuai dengan model proses hasil repair yang memiliki 8 buah aktivitas, yaitu create new document → document upload → document request evaluate → document update → document request approval → undistributed document → distribute document → accept document. Kemudian ditambahkan dengan looping length one pada distribute document → distribute document dan accept document → accept document. Untuk looping length two terjadi pada document request evaluate → document update → document request evaluate.
5. Analisis lain yang dapat dilakukan pada tahap Enhancement adalah analisis bottle neck, terdapat lima relasi proses yang mengalami bottle neck yaitu $A \rightarrow B$, $C \rightarrow D$, $E \rightarrow G$, $G \rightarrow G$, $H \rightarrow H$. Sehingga perlu ditindaklanjuti oleh pihak SPM agar waktu pemrosesannya menjadi lebih cepat.
6. Masih terdapat ambiguitas antara user pengguna sistem i-adore pada event log dengan user yang seharusnya menggunakannya berdasarkan Prosedur Pengendalian Dokumen, ambiguitas tersebut terdapat pada aktivitas document upload, document request evaluate, document update, document request approval, undistributed document, dan distribute document. Untuk selanjutnya harus ditentukan peran setiap usernya dengan jelas.
7. Tidak dapat ditarik kesimpulan untuk jenis dataset mana yang lebih cocok untuk diterapkan dalam penelitian ini dikarenakan masing-masing dataset memiliki karakteristiknya sendiri.

5.2 Saran

Setelah dilakukannya proses penelitian ini, berikut beberapa saran yang dapat penulis berikan untuk penelitian selanjutnya :

1. Karena algoritma ini mampu menangani noise, maka dapat dilakukan penelitian yang lebih lanjut mengenai seberapa besar algoritma Heuristic miner dapat menangani noise maupun incomplete trace dll.
2. Membandingkan algoritma heuristic miner dengan algoritma lain dalam menilai kemampuannya menangani data noise, incomplete trace dll.

3. Jika ingin mengetahui akurasi dari model proses dapat digunakan fitness pada proses conformance checkingnya [10].
4. Jika ingin mendapatkan nilai conformance yang lebih akurat dapat menggunakan nilai desimal yang lebih teliti dari penelitian ini.

Daftar Pustaka

- [1] StB Prof. Dr. Nick Gehrke and Michael Werner, Dipl.-Wirt.-Inf., "PROCESS MINING".
- [2] W. M. P. v. d. Aalst, Process Mining : Discovery, Conformance and Enhancement of Business Processes.
- [3] F. Syamsu, Prosedur Pengendalian Dokumen.
- [4] W. v. d. Aalst, "Process Mining: Making Knowledge Discovery Process Centric".
- [5] W. v. d. A. a. A. A. d. M. A.J.M.M. Weijters, "Process Mining with the HeuristicsMiner Algorithm".
- [6] Z. R. Arief, "Process Mining Pada Proses Registrasi Mahasiswa dengan Menggunakan Algoritma Alpha," 2014.
- [7] D. M. J. V. ., B. B. Stijn Goedertier, "Robust Process Discovery with Artificial Negative Events".
- [8] M. D. B. J. V. B. B. Jochen De Weerd, "A Robust F-measure for Evaluating Discovered Process Models".
- [9] A. W. a. W. v. d. A. A.K. Alves de Medeiros, "Using Genetic Algorithms to Mine Process Models: Representation, Operators and Results".
- [10] A. R. a. W. v. d. Aalst, "Conformance Checking of Processes Based on Monitoring Real Behavior".
- [11] S. M. I. Wil van der Aalst, "Service Mining: Using Process Mining to Discover, Check, and Improve Service Behavior".