

Process Mining pada Proses Pengadaan Buku dengan Algoritma *Heuristic Miner* (Studi Kasus : Perpustakaan Universitas Telkom)

Harin Veradistya Maharani¹, Angelina Prima Kurniati, ST., MT.², Imelda Atastina, S.Si, MT³

Fakultas Informatika, Universitas Telkom, Bandung

¹harinveradistya@gmail.com, ²angelina@telkomuniversity.ac.id, ³imelda@telkomuniversity.ac.id

Abstrak

Adanya SOP dan rekap menunjukkan bahwa organisasi memperhatikan proses yang terjadi. Pada pengaplikasiannya sehari-hari, rekap hanya digunakan sebagai dokumentasi proses yang dilaksanakan, tidak ada kontrol terhadap proses yang terjadi, tidak tahu apakah selama ini proses dilaksanakan sesuai dengan SOP. Oleh karena itu perlu pemodelan untuk kumpulan proses tersebut sehingga lebih mudah untuk melihat *behaviour* dari aktivitas yang dilakukan. Pada Tugas Akhir ini diaplikasikan *Process Mining* untuk proses pengadaan buku pada Unit Perpustakaan Universitas Telkom. Algoritma yang digunakan untuk memodelkan proses adalah algoritma *heuristic miner*. Proses yang dimodelkan dari sistem tersebut akan diukur kesesuaiannya dengan aktivitas yang sebenarnya terjadi dan pada akhirnya akan dihasilkan rekomendasi terhadap model proses. Hasil pengujian menunjukkan bahwa tidak semua aktivitas pada SOP dilakukan pada proses pengadaan buku. Selain itu terdapat aktivitas baru yang belum didefinisikan sebelumnya.

Kata kunci : SOP, *process mining*, *heuristic miner*, pengadaan buku, *event log*

Abstract

An Organization that defines SOP and make documentation for a process is indicate that that organization is seeing the process. They make a rule for the flow of process but there is no control for the process. The documentation is just be a documentation. Whereas the documentation can be usefull to see the real process. We can't check the documentation one by one because there is too many document. So we need to make a model to see the process behavior from the documentation. This final project use process mining to book procurement process in Library Unit of Telkom University. A model will be generated with heuristic miner algorithm. The next step is measure how conform the model and event log. The result show that there is no guarante if a higher value of threshold will produce a better model than a lower threshold. And there is no guarante too if a lower value of threshold will produce a worse model than a higher threshold. The model show that not all the activity in SOP was executed, and there are new activities that not define in SOP yet.

Keywords: SOP, *process mining*, *heuristic miner*, book procurement, *event log*

1. Pendahuluan

Proses bisnis adalah kumpulan dari proses yang mendukung proses-proses operasional dalam perusahaan/organisasi untuk mencapai *outcome* yang diinginkan [1]. Salah satu proses bisnis yang dilakukan oleh Universitas Telkom, khususnya pada unit perpustakaan adalah proses pengadaan buku.

Adanya SOP dan dokumentasi menunjukkan bahwa organisasi memperhatikan proses yang terjadi. Pada pengaplikasiannya sehari-hari, rekap tersebut hanya digunakan sebagai dokumentasi proses yang dilaksanakan, tidak ada kontrol terhadap proses yang terjadi, tidak tahu apakah selama ini proses sudah dilaksanakan sesuai dengan SOP. Masalah tersebut dapat diatasi dengan mengecek rekap satu persatu, namun hal tersebut bukan sebuah solusi ketika kasus pengadaan sangat banyak. Tidak mungkin melihat proses satu-persatu secara manual. Oleh karena itu perlu pemodelan untuk kumpulan proses tersebut sehingga lebih mudah untuk melihat *behaviour* dari aktivitas yang dilakukan.

Process Mining merupakan sebuah teknik untuk mengekstraksi data *event log*, mulai dari memodelkan proses hingga memberikan

rekomendasi terhadap hasil model proses tersebut [2]. Pada riset ini diaplikasikan *Process Mining* untuk proses pengadaan buku pada Unit Perpustakaan Universitas Telkom. Data yang diolah berasal dari rekap pengadaan buku Unit Perpustakaan Universitas Telkom.

Untuk menghasilkan model proses pengadaan buku pada Unit Perpustakaan Universitas Telkom digunakan algoritma *heuristic miner*. Hal tersebut dikarenakan model yang dihasilkan pada *heuristic miner* dapat diubah ke dalam berbagai tipe model proses. Selain itu *heuristic miner* dapat menangani data yang mengandung *noise* [5].

2. Kajian Pustaka

2.1. Process Mining

Pada analisis yang berorientasi pada data dikenal istilah *data mining*. *Data mining* merupakan proses analisis yang fokus pada keputusan yang dihasilkan berdasarkan data dan tidak mempertimbangkan proses yang terjadi [5]. *Process mining* merupakan salah satu teknik yang dikembangkan berdasarkan *data mining*, yang menjadi perbedaan adalah *process mining* fokus pada aktivitas yang terjadi. Data yang diolah merupakan *event log* yang

diekstraksi dari aktivitas yang terjadi. *Event log* merupakan data yang berisi informasi yang dapat menggambarkan *behavior* dari proses yang terjadi [5]. *Event log* bisa didapatkan dari sistem informasi maupun data dari unit yang terlibat dalam proses.

Tidak semua data dapat dijadikan *event log*, tidak semua informasi juga dibutuhkan dalam *event log*. Data pada *event log* harus berisi informasi yang berisi kumpulan aktivitas dan kasus yang terurut berdasarkan waktu sehingga data tersebut dapat menggambarkan sebuah proses yang sedang berjalan [5].

Berdasarkan paper tentang *Process Mining* yang diterbitkan oleh Weijters dkk [4], terdapat beberapa syarat dan asumsi data yang dapat digunakan untuk *event log* yaitu :

1. Setiap *event* harus mengacu pada aktivitas (proses yang sedang terjadi) dan kasus (dalam tugas akhir ini berarti satu kali pengadaaan).
2. Setiap *event* juga harus mempunyai *originator* yang merupakan eksekutor atau inisiator dari sebuah aktivitas.
3. Setiap *event* juga harus memiliki *time stamp* yang merupakan informasi waktu dieksekusinya sebuah aktivitas. *Event log* kemudian diurutkan berdasarkan urutan *time stamp* ini dari awal sampai akhir proses.

Pada contoh *event log* pada tabel 2-1, dapat disimpulkan bahwa *event log* tersebut memiliki enam *events*, empat aktivitas dan dua kasus.

Tabel 2-1 contoh *event log*

case id	activity id	originator	time stamp
case 1	A	Logistik	08-Apr-13
case 2	A	CSM	24-Mei-13
case 1	B	Logistik	12-Jul-13
case 1	C	Logistik	15-Jul-13
case 2	C	Logistik	23-Sep-13
case 1	D	CSM	24-Sep-13

Tujuan utama dari *Process Mining* adalah mengolah *event log* ke dalam sebuah model hingga memberikan rekomendasi terhadap model proses yang dihasilkan tersebut [2]. Dalam bukunya, Van der Aalst juga mengatakan terdapat tiga tipe *process mining* yaitu *Process Discovery*, *Conformance*, dan *Enhancement* [2].

Process Discovery merupakan proses pembentukan model proses dari *event log*. *Discovery* dilakukan dengan tujuan untuk mengetahui aktivitas yang sebenarnya terjadi pada proses. *Output* dari *discovery* adalah model yang menggambarkan *behavior* dari *event log* [4]. Model proses diekstraksi dari *event log*. *Event log* bisa didapatkan dari sistem informasi pada proses yang bersangkutan maupun dari data rekap.

Conformance adalah tipe *process mining* untuk membandingkan kesesuaian model proses

yang dihasilkan pada *process discovery* dengan *event log*. *Enhancement* merupakan tipe yang digunakan untuk memberikan rekomendasi terhadap proses. Konsep utama dari *enhancement* bukanlah untuk mengecek kesesuaian model proses yang ada melainkan untuk memberikan rekomendasi pada model proses yang dihasilkan pada *process discovery* [2].

2.2. Heuristic Miner

Heuristic miner merupakan salah satu algoritma untuk memodelkan proses pada *process mining* yang menggunakan pendekatan yang sama dengan algoritma *deterministic*, yaitu mengidentifikasi keterhubungan suatu aktivitas dengan aktivitas yang lain.

Heuristic miner didesain untuk melakukan *mining* pada *log* yang mengandung *noise* [2]. Algoritma ini menentukan hubungan sebab akibat, *split*, dan *join*.

Langkah-langkah pada algoritma *heuristic miner* adalah sebagai berikut [7].

1. Menghitung *dependency measure*

Ide dasar dari algoritma ini adalah menghilangkan *path* yang memiliki frekuensi yang sedikit (*infrequent*) ke dalam model yang dibangun. *Path* yang akan dibangun dihitung keterhubungannya dengan rumus *dependency measure* yang akan dijelaskan pada rumus(1). *Dependency measure* yang memenuhi *threshold* yang ditentukan akan dipilih untuk dibuat *dependency graph*.

Langkah kedua pada *process discovery* menggunakan *heuristic miner* adalah menghitung nilai dari *dependency measure* (DM) antar dua aktivitas. Jika T merupakan set aktivitas pada *event log* W , maka *dependency matrix* yang akan dibangun adalah $|T| \times |T|$. Maka rumus untuk mengisi tiap elemen dari *dependency matrix* tersebut adalah:

$$DM_{ij} = \frac{|\{w \in W \mid w \text{ is a successor of } a_i \text{ and } a_j \text{ is a predecessor of } w\}|}{|\{w \in W \mid w \text{ is a successor of } a_i\}| \cdot |\{w \in W \mid w \text{ is a predecessor of } a_j\}|}$$

Keterangan:

a_i : aktivitas yang didefinisikan sebagai predesesor

a_j : aktivitas yang didefinisikan sebagai suksesor

DM_{ij} : nilai *dependency measure* antara aktivitas a_i dan a_j

$|T|$: jumlah aktivitas yang diikuti aktivitas a_i

$|T|$: jumlah aktivitas yang diikuti aktivitas a_j

$|T|$: jumlah aktivitas yang diikuti aktivitas a_i dan a_j

Nilai DM_{ij} akan selalu berada diantara -1 dan 1. Jika nilai DM_{ij} semakin mendekati 1, maka semakin tinggi keterhubungan antara dua aktivitas tersebut, sering menyebabkan aktivitas terjadi [2].

2. Membuat *dependency matrix*

Langkah selanjutnya adalah membuat *dependency matrix*. Elemen pada *dependency*

matrix merupakan nilai dari *dependency measure*(DM) antar dua aktivitas yang telah dihitung pada langkah dua. Tabel 2-2 merupakan contoh *dependency matrix*.

Tabel 2-2 contoh *dependency matrix*

	A	B	C	D	E	F	G	H
A	0	0,8	0	0	0	0	0	0
B	-0,5	0	0,5	0,67	0,5	0	0	0
C	0	-0,5	0	0	0	0	0,5	0
D	0	-0,67	0	0	0,5	0,5	0	0
E	0	-0,5	0	-0,5	0	0,67	0	0
F	0	0	0	-0,5	-0,67	0	0,75	0
G	0	0	-0,5	0	0	-0,75	0	0,75
H	0	0	0	0	0	0	-0,75	0

Nilai tertinggi di baris i mengindikasikan aktivitas yang paling disebabkan oleh aktivitas i , sedangkan nilai tertinggi pada kolom j mengindikasikan aktivitas yang paling menyebabkan aktivitas j .

3. Menghitung *dependency measure* dan *dependency matrix* untuk *length-two loops*

Untuk kasus *trace* dapat ditangani dengan langkah kedua karena nilai DM_{ij} sudah mengindikasikan relasi *dependency*-nya [8]. Untuk kasus *length-one loops* (contoh *trace*) juga sudah dapat ditangani dengan langkah kedua yaitu dengan rumus jika $DM_{ij} > 0$.

Untuk kasus *length-two loops* (contoh *trace*) apabila ditangani dengan rumus pada langkah kedua nilai DM_{ij} akan selalu bernilai kecil sehingga dianggap tidak berelasi [8]. Untuk menghitung nilai *dependency measure* pada *length-two loops* adalah dengan menggunakan rumus:

$$DM_{ij} = \frac{DM_{ik} + DM_{kj}}{2}$$

Keterangan:

- i : aktivitas yang didefinisikan sebagai predesesor
- j : aktivitas yang didefinisikan sebagai suksesor
- DM_{ij} : nilai *dependency measure* antara aktivitas i dan j pada *length-two loops*
- DM_{ik} : jumlah aktivitas yang diikuti aktivitas i kemudian diikuti aktivitas k
- DM_{kj} : jumlah aktivitas yang diikuti aktivitas k kemudian diikuti aktivitas j

4. Membuat *dependency graph*

Tidak semua aktivitas dipasangkan satu sama lain pada *dependency graph*. Hanya aktivitas yang mempunyai hubungan sebab

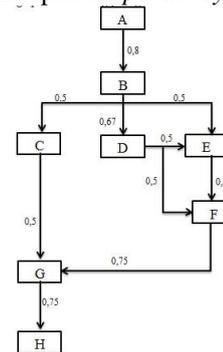
akibat yang dipasangkan. Harus ada *threshold* untuk menentukan pasangan antar aktivitas untuk mengindari ikut terhitungnya aktivitas yang merupakan *noise* [8].

Terdapat tiga *threshold* untuk membangun *dependency graph* dengan algoritma *heuristic miner*, yaitu [7]:

- a. Nilai DM_{ij} harus lebih besar atau sama dari nilai *dependency threshold* yang telah ditentukan.
- b. Nilai $|DM_{ij}|$ harus lebih besar atau sama dari nilai *positive observations threshold* yang telah ditentukan.
- c. Nilai $|DM_{ij}|$ harus lebih kecil dari nilai RTB(*relative to best*) yang telah ditentukan sebelumnya. Atau Nilai $|DM_{ij}|$ harus lebih kecil dari nilai RTB(*relative to best*) yang telah ditentukan sebelumnya.

adalah nilai *dependency measure* terbesar pada baris (suksesor dari i), sedangkan DM_{ij} adalah nilai *dependency measure* terbesar pada kolom (presedesor j)

Apabila tiga *threshold* tersebut terpenuhi, maka aktivitas i dan j dapat dipasangkan pada *dependency graph*.



Gambar 2-1 contoh *dependency graph*

5. Membuat *causal matrix*

Dependency graph yang terbentuk memang sudah menunjukkan relasi antar aktivitas. Namun untuk aktivitas yang mempunyai relasi percabangan (contoh pada gambar 3, b memiliki cabang ke c, d, dan e) tidak dapat ditentukan setelah b selesai dilakukan apakah c,d dan e harus dilakukan

secara bersama-sama(AND) atau boleh salah satu(XOR). Oleh karena itu perlu untuk

mendefinisikan jenis relasi antar aktivitas.

Cara untuk menentukan jenis relasi cukup mudah. Pada contoh *dependency graph* pada gambar 3, *output* dari aktivitas b adalah c,d,e. Apabila dua aktivitas yang merupakan aktivitas *output* dari aktivitas b (contoh c dan d) merupakan relasi AND, maka cd akan muncul pada *event log*.

Sebaliknya jika c dan d merupakan relasi XOR maka cd tidak mungkin muncul [8].

Rumus untuk menghitung *dependency relation* antara a dengan b dan c adalah:

$$\frac{p + m - c}{n}$$

Keterangan:

- p : aktivitas yang didefinisikan sebagai suksesor
- c : aktivitas yang didefinisikan sebagai predesesor 1
- m : aktivitas yang didefinisikan sebagai predesesor 2
- n : nilai *dependency relation* antara a dengan b dan c
- p : jumlah aktivitas yang diikuti aktivitas
- c : jumlah aktivitas yang diikuti aktivitas
- m : jumlah aktivitas yang diikuti aktivitas
- n : jumlah aktivitas yang diikuti aktivitas

Untuk menentukan jenis relasi AND atau XOR diberikan *threshold*, apabila nilai n di atas *threshold* yang telah ditentukan maka relasi dan merupakan relasi AND, sebaliknya apabila nilai n di bawah *threshold* yang telah ditentukan maka relasi dan merupakan relasi XOR. Untuk menentukan *input* dari aktivitas, rumus yang digunakan sama, tetapi aktivitas dan merupakan aktivitas yang dilakukan sebelum aktivitas a.

2.1. Conformance

Conformance merupakan tahap pengecekan kesesuaian *events* pada *event log* dengan aktivitas pada model [2]. Tujuan dilakukannya *conformance* adalah untuk mencari persamaan dan perbedaan antara *behavior* pada model yang telah dibangun dengan *behavior* pada data.

Ketika melakukan *conformance*, peting untuk melihat perbedaan antara model yang dibuat dengan *event log* dari dua sisi: (a) model yang dibuat “salah” dan tidak mencerminkan kenyataan dan (b) aktivitas pada *event log* yang menyimpang dari model. Untuk mengukur akurasi model dengan aktivitas pada *event log* dilakukan dengan menghitung nilai *fitness*. Sesuai dengan ide *fitness* yaitu menghitung proporsi *behavior* pada *event log* dengan model, maka *fitness* paling cocok untuk melakukan *conformance* [2].

Perhitungan *fitness* dilakukan dengan melakukan token *replay* untuk setiap *trace*. Model yang dilakukan *conformance* adalah *petri net*. Pada *petri net* transisi yang sedang aktif dikatakan telah mengkonsumsi satu token dari tiap *place input* dan memproduksi satu token dari tiap *place output*. Untuk melakukan token *replay* digunakan empat *counter* yaitu p (*produced token*), c (*consumed token*), m (*missing token*), dan r (*remaining token*). Rumus *fitness* adalah sebagai berikut

$$-(\frac{p}{n}) - (\frac{c}{n})$$

Keterangan:

- t : *trace*
- p : *produced token*
- c : *consumed token*

r : *remaining token*

Semakin tinggi nilai *fitness* maka tingkat kesamaan model dengan aktivitas yang terjadi

juga tinggi. Namun rumus (4) merupakan rumus untuk menghitung nilai *fitness* pada

single case. Pendekatan yang sama juga dapat dilakukan untuk menghitung *fitness* pada *event log* yang mempunyai banyak *case*. Hal tersebut dapat dilakukan dengan menghitung jumlah semua token yang diproduksi, dikonsumsi, *missing*, dan *remaining* kemudian *fitness* dihitung dengan rumus yang sama [2]. Untuk banyak *case*, rumus (4) dapat didefinisikan seperti pada rumus (5).

$$-(\frac{\sum p}{\sum n}) - (\frac{\sum c}{\sum n})$$

Keterangan:

- p : *missing token* ketika melakukan *replay trace*
- c : *consumed token* ketika melakukan *replay trace*
- r : *remaining token* ketika melakukan *replay trace*
- p : *produced token* ketika melakukan *replay trace*
- f : frekuensi *trace*

2.2. Enhancement

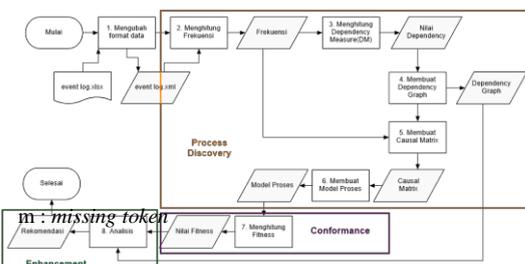
Ide dari *enhancement* adalah menambah atau memperbaiki proses model berdasarkan proses yang terjadi pada *event log* [2]. Salah satu tipe *enhancement* adalah *repair*. *Repair* memodifikasi model proses yang lebih mencerminkan realitas. Jika sebuah aktivitas pada model digambarkan sekuensial tetapi pada kenyataannya dapat terjadi dalam urutan apapun, model dapat diperbaiki agar lebih mencerminkan urutan yang sebenarnya terjadi.

Tipe lain dari *enhancement* adalah *extension*. Pada *extension*, dilakukan penambahan perspektif baru pada model proses. Contohnya *extension* dengan menambah proses baru pada model proses berdasarkan hasil *discovery* dan *performance*.

3. Analisis Kebutuhan

3.1. Gambaran Umum

Data yang akan diolah merupakan data rekap pengadaan buku Unit Perpustakaan Universitas Telkom. Pada tugas akhir ini akan dibuat sistem untuk membantu memodelkan proses. Proses yang dimodelkan dari sistem tersebut akan diukur kesesuaiannya dengan aktivitas yang sebenarnya terjadi dan pada akhirnya akan dihasilkan rekomendasi terhadap model proses. Gambar 3-1 adalah alur pengerjaan dari Tugas Akhir ini.



Gambar 3-1
gambaran
umum
pengerjaan

3.2. Analisis Kebutuhan Aplikasi

Berdasarkan gambaran umum pengerjaan maka aplikasi harus dapat:

- Membaca file xml
- Menghitung nilai *dependency measure* antar aktivitas
- Membuat *dependency graph*
- Membuat *causal matrix*
- Mampu membuat model proses
- Menghitung nilai *fitness*

3.3. Analisis Input dan Output Aplikasi

Input dari aplikasi yang dibuat adalah data yang berformat .xml. Data diperoleh dari rekam pengadaan buku Unit Perpustakaan Universitas Telkom. Aplikasi dapat menampilkan file yang diinput. Setelah aplikasi membaca file yang telah diinput, inputan berikutnya adalah nilai DM *threshold*, PO *threshold*, RTB *threshold* untuk membuat *dependency graph*. *Output* setelah menentukan DM, PO, dan RTB *threshold* adalah *dependency graph*. Selanjutnya *input threshold* untuk menentukan jenis relasi pada *causal matrix*, setelah itu aplikasi menampilkan *causal matrix* dan model proses dari data *input* dan semua *threshold* sebelumnya. *Output* aplikasi selanjutnya adalah nilai *fitness* dari model proses yang dihasilkan.

4. Pengujian

4.1. Event log

Data yang digunakan adalah *event log* yang berasal dari rekam pengadaan buku Unit Perpustakaan pada Logistik Universitas Telkom. Rekam yang diambil adalah rekam pengadaan buku pada tahun 2004, 2009, 2010, 2011, 2012, dan 2013. Hal itu dikarenakan hanya rekam tersebut yang ditemukan dalam arsip penyimpanan Unit Logistik Universitas Telkom. *Event log* sebelum diubah menjadi xml terdiri dari 101 pengadaan, 1143 baris, dan 14 jenis aktivitas.

4.2. Parameter Pengujian

Parameter yang digunakan sebagai alat ukur dari pengujian yang dilakukan adalah nilai *fitness*. *Fitness* merupakan parameter pengujian yang mengukur *behavior event log* berdasarkan model [2]. Berdasarkan pengertiannya *fitness* merupakan kriteria kualitas model yang paling relevan dengan *conformance*, yaitu mengecek kesesuaian model yang dihasilkan dengan *event log*. Oleh karena itu *fitness* digunakan dalam pengujian ini.

4.3. Skenario Pengujian

DM *threshold* yang tinggi akan menghasilkan model dengan nilai *dependency measure* yang tinggi, sehingga hanya set aktivitas yang kuat yang dimodelkan. PO *threshold* yang tinggi akan menghasilkan model

dengan set aktivitas yang memiliki frekuensi tinggi. RTB *threshold* memungkinkan untuk memasangkan ekstra *set* aktivitas.

Pengujian dilakukan dengan menggunakan *threshold* yang berbeda-beda untuk mengetahui *threshold* yang paling tepat untuk menghasilkan model yang paling sesuai dengan *event log*. Pengujian dilakukan dengan mencari nilai PO dan DM terbaik berdasarkan nilai *fitness*. Nilai RTB tidak diubah dengan pertimbangan apabila RTB *threshold* diset terlalu besar maka terlalu banyak aktivitas *infrequent* dan/atau tidak berelasi yang ikut termodelkan. Sedangkan apabila terlalu kecil maka akan ada kemungkinan muncul *set* aktivitas yang tidak terhubung dengan model utama. Sehingga RTB *threshold* 0,05 merupakan *threshold* yang paling ideal [9].

4.4. Analisis Hasil Pengujian

Berdasarkan hasil pengujian, Nilai *threshold* yang tinggi akan menghasilkan model yang hanya menggambarkan aktivitas yang berelasi dengan kuat. Akan tetapi penggunaan nilai *threshold* yang tinggi dapat menghilangkan aktivitas yang *infrequent* dan dianggap sebagai *noise* sehingga model menjadi tidak lengkap. Nilai *threshold* yang tinggi juga dapat menghilangkan aktivitas yang hanya terjadi pada kondisi tertentu karena dianggap sebagai *noise* dan tidak relevan.

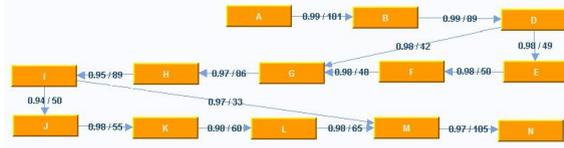
Nilai *threshold* yang rendah akan menghasilkan model yang lengkap, akan tetapi aktivitas yang *infrequent* dan *noise* dapat ikut termodelkan. Nilai *threshold* yang rendah juga dapat menampilkan aktivitas yang hanya terjadi pada kondisi tertentu.

Tabel 4-1 Hasil Skenario Pengujian

Skenario	DM	PO	RTB	Fitness
1	0.9	42	0.05	0.91
2	0.9	32	0.05	0.94
3	0.9	22	0.05	0.94
4	0.9	12	0.05	0.94
5	0.9	2	0.05	0.88
6	0.8	42	0.05	0.91
7	0.8	32	0.05	0.94
8	0.8	22	0.05	0.94
9	0.8	12	0.05	0.944
10	0.8	2	0.05	0.82
11	0.7	42	0.05	0.91
12	0.7	32	0.05	0.94
13	0.7	22	0.05	0.94
14	0.7	12	0.05	0.94
15	0.7	2	0.05	0.80
16	0.6	42	0.05	0.91
17	0.6	32	0.05	0.94
18	0.6	22	0.05	0.94

Skenario	DM	PO	RTB	Fitness
19	0,6	12	0,05	0,94
20	0,6	2	0,05	0,80

Dapat dilihat pada tabel 4-1, nilai *threshold* paling tinggi didapatkan jika nilai *DM threshold* berada di rentang = 0,9-0,6, *PO threshold* = 32-12, dan *RTB threshold* = 0,05. Hal tersebut mengindikasikan bahwa pada *threshold* tersebut, model yang dihasilkan cukup akurat untuk menggambarkan aktivitas yang sesungguhnya terjadi pada pengadaan buku Unit Perpustakaan Universitas Telkom.

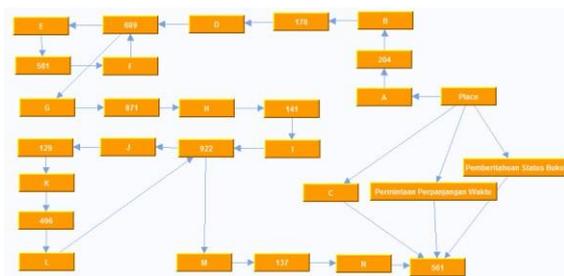


Gambar 4-1 *dependency graph* *DM threshold* = 0,9-0,6, *PO threshold* = 32-12, dan *RTB threshold* = 0,05

Activity	Input	Output
A		B
B	A	D
D	B	(E XOR G);
E	D	F
F	E	G
G	(D XOR F);	H
H	G	I
I	H	(J XOR M);
J	I	K
K	J	L
L	K	M
M	(I XOR L);	N
N	M	
Pemberitahuan Status Buku		
C		
Permintaan Perpanjangan Waktu		

Activity	Input	Output
A		B
B	A	D
D	B	(E XOR G);
E	D	F
F	E	G
G	(D XOR F);	H
H	G	I
I	H	(J XOR M);
J	I	K
K	J	L
L	K	M
M	(I XOR L);	N
N	M	
P		
C		
O		

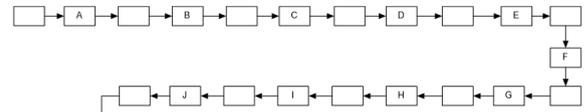
Gambar 4-2 *causal matrix* *DM threshold* = 0,9-0,6, *PO threshold* = 30-20, dan *RTB threshold* = 0,05



Gambar 4-3 model proses *DM threshold* = 0,9-0,6, *PO threshold* = 30-20, dan *RTB threshold* = 0,05

Jika dibandingkan dengan model proses berdasarkan SOP, terdapat beberapa perbedaan

antara keduanya. Model proses berdasarkan SOP dapat dilihat pada gambar 8.



Gambar 4-4 model proses pengadaan barang dan jasa 5 juta-100 juta berdasarkan SOP

Berdasarkan model tersebut seharusnya semua aktivitas selalu dilakukan secara berurutan. Model berdasarkan SOP memiliki nilai *fitness* 0,82 atau 82%. Nilai tersebut dapat dilihat dari dua sudut pandang yaitu:

- *Event log* mempunyai *fitness* 0,82 atau 82% yang berarti sekitar 18% *event* dari seluruh *event* menyimpang, atau
- Model memiliki *fitness* 0,82 atau 82% yang berarti model tidak dapat menjelaskan 18% dari *behavior*.

Model hasil *discovery* memiliki *fitness* 0,94 sehingga dapat disimpulkan model hasil *discovery* lebih dapat menggambarkan *event log*. Model hasil *discovery* dapat dilihat pada gambar 4-21, aktivitas EF (penerbitan SPPH dan penerimaan SPH) dapat berulang. Setelah aktivitas D (Pengecekan permintaan barang/jasa) dapat dilakukan aktivitas E (penerbitan SPH) atau langsung G (pelaksanaan klarifikasi dan negosiasi harga, sedangkan aktivitas E dan F tidak dilakukan). Setelah aktivitas I (penetapan kontrak) dapat dilakukan aktivitas J (penerbitan bast) atau langsung M (pembayaran, aktivitas J dan K/penerimaan FPS tidak dilakukan). Sementara aktivitas C yang merupakan aktivitas perumusan harga perkiraan sendiri tidak berelasi dengan aktivitas manapun pada model ini. Selain itu terdapat aktivitas baru yang belum terdefiniskan pada SOP yaitu aktivitas permintaan perpanjangan waktu dan pemberitahuan status buku, namun aktivitas tersebut juga tidak berelasi dengan aktivitas manapun.

4.5. Analisis Rekomendasi Model Hasil *Discovery*

Dengan menggunakan *threshold* yang lebih rendah yaitu *DM threshold* berada di rentang = 0,7-0,6, *PO threshold* = 1 (skenario 15 dan 20), dan *RTB threshold* = 0,05 ditemukan bahwa aktivitas C dilakukan setelah aktivitas B dan sebelum aktivitas D. Dari 101 pengadaan, aktivitas BC hanya dilakukan sebanyak 11 kali dan aktivitas CD dilakukan sebanyak 10 kali.

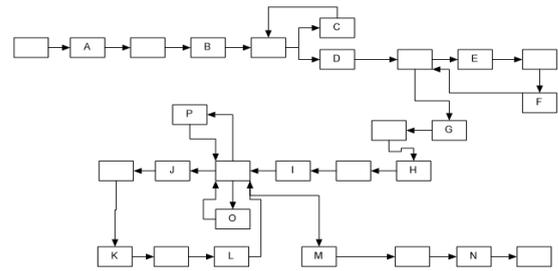
Pada skenario 15 dan 20 juga ditemukan relasi dari aktivitas permintaan perpanjangan waktu dan pemberitahuan status buku, keduanya dilakukan setelah aktivitas I dan sebelum aktivitas J yaitu setelah penetapan kontrak dan sebelum BAST. Dari 101 pengadaan aktivitas I-permintaan perpanjangan waktu dilakukan sebanyak 3 kali, aktivitas permintaan perpanjangan waktu-J 2 kali, aktivitas I-Pemberitahuan status buku 4 kali, dan aktivitas pemberitahuan status buku-J 4 kali. Walaupun frekuensi kemunculannya sangat kecil, berdasarkan nilai *dependency measure* yang cukup tinggi aktivitas ini bukan *noise*. Aktivitas ini merupakan aktivitas yang terjadi pada kondisi tertentu sehingga tidak terdefinisi pada SOP.

Skenario 15 dan 20 memang menghasilkan model yang lengkap, semua aktivitas dapat tergambar termasuk aktivitas baru yang tidak ditemukan pada model dengan *fitness* yang lebih tinggi. Namun model ini terlalu beresiko untuk digunakan karena terlalu banyaknya aktivitas *infrequent* maupun *noise* yang ikut termodelkan. Untuk menyiasatinya dapat menggunakan kombinasi dari beberapa model yang terbentuk.

Pada model dengan *fitness* 0,94 ditemukan bahwa setelah aktivitas B hanya dilakukan aktivitas D. Pada skenario 15 dan 20, setelah aktivitas B dilakukan aktivitas C dan D, namun setelah aktivitas C dilakukan aktivitas D lagi sehingga pada model ini dapat terbentuk *pattern* BCDD. Pada model dengan *fitness* 0,94(misalnya skenario 2) *pattern* D tidak dapat *looping* sehingga lebih baik jika setelah aktivitas B dilakukan aktivitas C atau D. Sehingga aktivitas D tidak akan mengalami *looping*.

Karena merupakan aktivitas kondisional, aktivitas permintaan perpanjangan waktu dan pemberitahuan status buku dapat dilakukan namun tidak menjadi masalah apabila tidak dilakukan sehingga relasi dengan I dan J adalah XOR.

Rekomendasi perbaikan berdasarkan model dengan *fitness* terbaik pada pengujian dapat dilihat pada gambar 4-23. Model proses pada gambar 4-23 memiliki *fitness* 0,95 (lebih baik dari hasil *discovery*). Pada model ini aktivitas C, O, dan P dapat dijelaskan tanpa mengikutsertakan terlalu banyak *noise* dan aktivitas yang *infrequent*.



Gambar 4-5 Rekomendasi model proses

Tabel 4-2 Kode aktivitas

Kode	Aktivitas	Kode	Aktivitas
A	Identifikasi kebutuhan barang/jasa	I	Penetapan kontrak
B	Pengajuan permintaan pengadaan barang/jasa	J	Penerbitan BAST
C	Perumusan daftar harga referensi	K	Penerimaan FPS
D	Pengecekan permintaan barang/jasa	L	Penerbitan Permohonan Pembayaran
E	Penerbitan SPPH	M	Pembayaran
F	Penerimaan SPH	N	Pertanggungjawaban pihak terkait
G	Pelaksanaan klarifikasi dan negosiasi	O	Permintaan Perpanjangan Waktu
H	Penerbitan Pakta Integritas	P	Pemberitahuan Status Buku

4.6. Analisis Rekomendasi Model Proses Pengadaan Buku

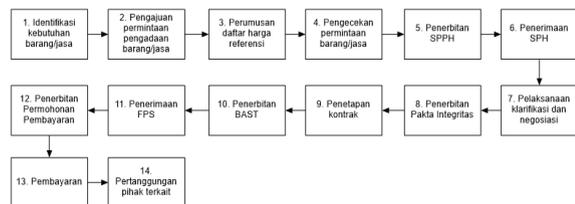
Rekomendasi untuk model proses pengadaan buku disesuaikan dengan SOP, aktivitas yang terjadi, fleksibilitas, dan kesederhanaan proses. Berdasarkan aktivitas yang dilaksanakan pada Proses Pengadaan Buku, aktivitas C yang merupakan perumusan daftar harga referensi jarang dilakukan. Perumusan daftar harga referensi merupakan aktivitas yang penting untuk dilakukan karena dokumen daftar harga referensi digunakan sebagai acuan dalam pada saat pelaksanaan

klarifikasi dan negosiasi harga. Namun pada kenyataannya proses bisa berjalan tanpa ada perumusan daftar harga referensi. Sebaiknya apabila perumusan harga referensi tidak dilakukan maka unit terkait harus ikut dalam pelaksanaan klarifikasi dan negosiasi harga.

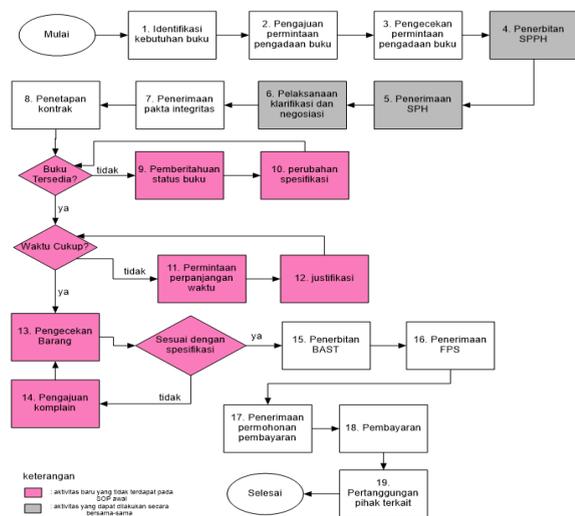
Penerbitan SPPH dan penerimaan SPPH sebaiknya tetap dilakukan agar Unit Logistik dan Terkait dapat melakukan persiapan pada saat negosiasi, namun tidak menjadi masalah apabila tidak dilakukan karena harga yang ditawarkan oleh *supplier* masih dapat dinegosiasi pada saat klarifikasi dan negosiasi harga. Rekomendasi untuk aktivitas ini adalah aktivitas penerbitan SPPH dan SPH tetap dilakukan namun aktivitas tersebut dapat dilakukan bersamaan dengan pelaksanaan klarifikasi dan negosiasi harga.

Aktivitas permintaan perpanjangan waktu dan pemberitahuan status buku sebaiknya didefinisikan ke dalam SOP bahwa setelah penetapan kontrak, apabila terdapat pemberitahuan status buku sebaiknya dilakukan justifikasi terhadap permintaan tersebut. Pada saat serah terima sebaiknya tetap dibuat BAST untuk mengecek kondisi dan kesesuaian spesifikasi buku, sangat tidak direkomendasikan langsung melakukan pembayaran.

Rekomendasi SOP untuk pengadaan buku dapat dilihat pada gambar 4-25. Gambar 4-24 merupakan SOP awal dari pengadaan barang/jasa berdasarkan rekaman.



Gambar 4-6 Flow proses pengadaan buku 5juta-100juta rupiah



Gambar 4-7 Rekomendasi flow proses pengadaan buku 5juta-100juta rupiah

Aktivitas berwarna magenta merupakan aktivitas baru yang tidak terdapat pada SOP awal, sedangkan proses berwarna abu-abu merupakan aktivitas yang dapat dilakukan secara bersamaan. Penjelasan lebih lanjut mengenai rekomendasi SOP dapat dilihat pada tabel 4-3.

5. Kesimpulan

Berdasarkan hasil analisis dan pengujian yang telah dilakukan, tugas akhir ini memiliki kesimpulan sebagai berikut:

1. Nilai *threshold* yang tinggi akan menghasilkan model yang hanya menggambarkan aktivitas yang berelasi dengan kuat. Akan tetapi penggunaan nilai *threshold* yang tinggi dapat menghilangkan aktivitas yang *infrequent* dan dianggap sebagai *noise* sehingga model menjadi tidak lengkap. Nilai *threshold* yang tinggi juga dapat menghilangkan aktivitas yang hanya terjadi pada kondisi tertentu karena dianggap sebagai *noise* dan tidak relevan.
2. Nilai *threshold* yang rendah akan menghasilkan model yang lengkap, akan tetapi aktivitas yang *infrequent* dan *noise* dapat ikut termodelkan. Nilai *threshold* yang rendah juga dapat menampilkan aktivitas yang hanya terjadi pada kondisi tertentu.
3. Nilai *Threshold* yang disarankan untuk pengadaan buku Unit Perpustakaan Universitas Telkom adalah dengan rentang *DM threshold* = 0,9-0,6 *PO threshold* = 32-12 dan *RTB threshold* = 0,05 dengan nilai *fitness* = 0,94.
4. Terdapat beberapa perbedaan antara model proses berdasarkan SOP dengan model proses hasil *discovery* yaitu:
 - a. Pada model dari SOP seharusnya semua aktivitas(A-N) selalu dilakukan secara berurutan.
 - b. Pada model hasil *discovery*, terdapat pelompatan aktivitas penerbitan SPPH, penerimaan SPH, penerbitan BAST, dan penerimaan FPS.
5. Kekurangan pada model proses berdasarkan SOP yaitu:
 - a. Model berdasarkan SOP hanya memiliki *fitness* 0,82 sehingga belum bisa menggambarkan proses dengan baik.
 - b. Model proses belum dapat menjelaskan aktivitas kondisional seperti aktivitas permintaan perpanjangan waktu dan pemberitahuan status buku.
 - c. Model proses belum dapat menjelaskan pelompatan aktivitas
6. Kekurangan pada model proses hasil *discovery* adalah model tersebut belum dapat menjelaskan aktivitas perumusan daftar harga referensi, permintaan perpanjangan waktu dan pemberitahuan status buku.
7. Rekomendasi terhadap model proses hasil *discovery* adalah:

- a. Perumusan daftar harga referensi dilakukan sebelum pengecekan permintaan barang/jasa dan setelah pengajuan dengan relasi XOR agar tidak terjadi *looping* pada pengajuan permintaan barang/jasa.
 - b. Aktivitas permintaan perpanjangan waktu dan pemberitahuan status buku sebelum penerbitan BAST dan setelah penetapan kontrak dengan relasi XOR karena merupakan aktivitas kondisional.
8. Rekomendasi SOP untuk proses pengadaan buku adalah:
 - a. Proses bisa berjalan tanpa ada perumusan daftar harga referensi. Sebaiknya apabila perumusan harga referensi tidak dilakukan maka unit terkait harus ikut dalam pelaksanaan klarifikasi dan negosiasi harga.
 9. Aktivitas permintaan perpanjangan waktu dan pemberitahuan status buku sebaiknya didefinisikan ke dalam SOP bahwa setelah penetapan kontrak, apabila terdapat pemberitahuan status buku sebaiknya dilakukan justifikasi terhadap permintaan tersebut.

6. Saran

Untuk pengembangan yang lebih baik, terdapat beberapa hal yang dapat dikembangkan yaitu:

1. Perekapan pengadaan buku sebaiknya dilakukan secara terkomputerisasi untuk menghasilkan rekap data yang lebih bersih.
2. Penelitian dapat dikembangkan dengan melakukan *mining* pada perspektif lain seperti *organizational perspective*.

7. Daftar Pustaka

- [1] Will M.P. Van Der Aalst, *Process Mining : Discovery, Conformance and Enhancement of Business Processes*. New York: Springer, 2011.
- [2] Thomas H. Davenport and James E. Short, *The New Industrial Engineering: Information Technology and Business Process Redesign*. Cambridge, Massachusetts: sloan school of management-massachusetts institute of technology, 1990.
- [3] W.M.P. van der Aalst et al., "Business Process Mining: An Industrial Application," pp. 713-732, 2007.
- [4] R.S. Mans, M.H. Schonenberg, M. Song, W.M.P. van der Aalst, and P.J.M. Bakker, "Application of Process Mining in Healthcare," *BIOSTEC 2008*, pp. 425-438, 2008.
- [5] Philip Weber, Behzad Bordbar, and Peter Tino, "A Principled Approach to Mining From Noisy Log Using Heuristic Miner," *IEEE*, pp. 119-126, 2013.
- [6] Wil M.P. van der Aalst, "Decomposing Petri Nets for Process Mining(A Generic Approach)," Technische Universiteit Eindhoven, September 2012.
- [7] Saravanan.M.S and Rama.R.J Sree, "A Role of Heuristic Miner Algorithm in the Business Process," *ISSN:2229-6093*, pp. 340-344.
- [8] A.J.M.M. Weijters, W.M.P Van der Aalst, and A.K. Alves de Medeiros, "Process Mining with the HeuristicMiner," , Eindhoven, The Netherlands.
- [9] Stefanie Rinderle-Ma, Shazia Sadiq, and Frank Leymann, Eds., *Business Process Management Workshops*. Ulm, Germany: Springer, 2009.
- [10] Administrator. (2008, Juni) <http://digilib.ittelkom.ac.id/>. [Online]. http://digilib.ittelkom.ac.id/index.php?view=article&catid=2%3Ainformatika&id=2%3Asejarah-dan-visi-misi&option=com_content&Itemid=29
- [11] StB Prof. Dr. Nick Gehrke and Michael, Dipl.-Wirt.-Inf. Werner, "Process Mining," *WISU - die Zeitschrift für den Wirtschaftsstudenten*, 2013.
- [12] the Process Mining Group. (2009, May) Process Mining research tools application. [Online]. www.processmining.org