Analisis dan Implementasi Process Mining Menggunakan Fuzzy Mining (Studi Kasus: Data BPI Challenge 2014)

Muhammad Agung[1], Angelina Prima Kurniati[2], S.T, M.T, Alfian Akbar Gozali S.T, M.T[3]

[1]Prodi S1 Teknik Informatika, Fakultas Informatika, Universitas Telkom
[2]Fakultas Informatika, Universitas Telkom
[3]Fakultas Ilmu Terapan, Universitas Telkom
[1]muhammadagung141293@gmail.com, [2]angelina@telkomuniversity.ac.id, [3]panggil.aku.ian@gmail.com

Abstrak

Meningkatnya fokus suatu organisasi dan perusahaan dalam mengembangkan proses bisnis untuk mendapatkan proses bisnis yang lebih efektif dan efisien menyebabkan mereka mulai melakukan pengembangan terhadap proses bisnis tersebut. Dengan proses bisnis yang lebih efektif dan efisien, kinerja sistem dari organisasi maupun perusahaan tersebut akan menjadi lebih baik. Salah satu proses bisnis yang terjadi pada *Rabobank Netherlands Group ICT* adalah proses penerimaan, perekaman, dan penanganan masalah yang dilakukan oleh bagian *service desk* dan *IT operations* yang dibantu oleh sistem *ITIL Service Management* yang disebut sebagai *HP Service Manager*. Namun pembuatan model proses yang sederhana berdasarkan data yang kompleks tidak mudah dimana model proses yang sederhana dapat memudahkan pihak analis *Rabobank Netherlands Group ICT* untuk menganalisisnya. *Process mining* dapat digunakan untuk membuat model proses tersebut sehingga proses bisnis dapat ditingkatkan kinerjanya. Untuk dapat memperoleh model tersebut, *process discovery* perlu dilakukan terlebih dahulu dari *event log* yang ada. *Event log* yang diperlukan dalam melakukan *process mining* untuk memperoleh model proses ini adalah *event log Rabobank Netherlands Group ICT*, dimana pada kasus data *real-life* biasanya data lebih kompleks atau kurang terstruktur sehingga model proses yang dihasilkan sering *"spaghetti-like"*. Oleh karena itu, diperlukan pendekatan yang dapat menghasilkan model proses yang tidak "spaghetti-like" salah satunya adalah *fuzzy mining [1]*.

1. Pendahuluan

Dalam setiap organisasi atau perusahaan, pastilah terdapat suatu proses bisnis yang merupakan suatu kumpulan kegiatan yang dilakukan secara terstruktur baik dalam menyelesaikan suatu masalah yang terjadi maupun mencapai tujuan tertentu [2]. Meningkatnya fokus mereka dalam mengembangkan proses bisnis untuk mendapatkan proses bisnis yang lebih efektif dan efisien menyebabkan banyak organisasi dan perusahaan mulai melakukan pengembangan tersebut. Pada *Rabobank Netherlands Group ICT*, proses bisnis penerimaan dan penanganan masalah yang dilakukan oleh bagian *service desk* direkam, yang dibantu oleh sistem *ITIL Service Management* yang disebut sebagai *HP Service Manager*. Namun tidak semua proses bisnis yang dimiliki adalah proses bisnis yang efektif dan efisien, oleh karena itu dilakukanlah analisis kinerja dari proses bisnis sistem penanganan masalah *customer* pada bagian *service desk* di *Rabobank Netherlands Group ICT* tersebut untuk mendapatkan wawasan untuk mengoptimalkan proses operasional tersebut [3].

Dalam rangka untuk mengatur panggilan dan pesan dari pelanggan (rekan *Rabobank*) ke *service desk* mengenai gangguan jasa *ICT,* seorang *Service Desk Agent* (SDA) akan mencatat log panggilan dan pesan serta mencoba untuk menyelesaikan permasalahan pelanggan tersebut. Namun tidak semua permasalahan yang dialami oleh pelanggan dapat dipahami dan diselesaikan oleh SDA maka dibuatlah *incident activity-record* dan memberikan masalah tersebut kepada *Assignment Group* dengan pengetahuan teknis yang lebih dalam menyelesaikan masalah gangguan layanan. *Event log incident activity* tersebut merupakan sebuah *event log* yang kompleks dimana urutan kejadian suatu *event* tidak beraturan. *Event log incident activity* ini dianalisis untuk memperoleh suatu informasi yang dapat berguna bagi *Rabobank Netherlands Group ICT*. Dalam hal tersebut, sebuah teknik yang dapat dilakukan adalah *process mining* yang menghasilkan visualisasi model proses berdasarkan *event log* yang nantinya dapat dilakukan proses lebih lanjut untuk memperoleh informasi yang diharapkan sebelumnya [4].

Terdapat beberapa pendekatan yang dapat digunakan dalam memperoleh model proses dari *event logs* yang ada. Namun tidak semua pendekatan dapat diterapkan

pada *event logs* yang terjadi di dunia nyata, seperti halnya *event logs incident activity* dari *Rabobank Netherlands Group ICT* yang kompleks. *Fuzzy mining* merupakan pendekatan yang dapat digunakan untuk menangani *event logs* yang kompleks. Model proses yang dihasilkan oleh *fuzzy mining* dapat disederhanakan untuk menghindari "*spaghetti-like*" model. Proses yang dilakukan untuk menyederhanakan model tersebut adalah *graph simplification process*. Dalam melakukan *graph simplification process*, menentukan variabel yang tepat untuk mendapatkan *conformance* yang baik juga diperlukan sehingga model yang dihasilkan dapat menggambarkan realitas dari log yang diamati dengan baik [1].

2. Landasan Teori dan Perancangan

2.1. Process Mining

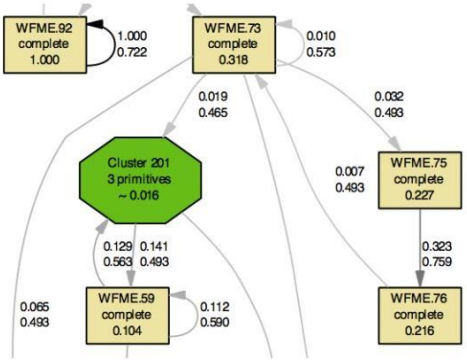
Process mining adalah sebuah penelitian yang relatif masih baru, yang berada diantara *machine learning* dan *data mining* pada satu sisi dan disisi lainnya *process mining* juga melakukan proses pemodelan serta analisis. Ide dari *process mining* adalah untuk menemukan, memantau, dan meningkatkan proses sebenarnya dengan cara mengekstraksi pengetahuan-pengetahuan dari *event log* yang tersedia. *Process mining* dapat menetapkan proses model serta hubungan antara proses aktual dan data [4]. Terdapat tiga tipe *process mining* yaitu *discovery, conformance,* dan *enhancement*.

2.2. Fuzzy Mining

Fuzzy mining merupakan salah satu teknik dalam melakukan *process mining* yang ditujukan pada *real-life log*. Proses yang terjadi pada *real-life log* kurang terstruktur dimana pendekatan *process mining* seperti *alpha* dan *heuristic* memiliki masalah terhadap data yang kurang terstruktur tersebut. Model proses yang dihasilkan oleh pendekatan *alpha* dan *heuristic* sering "*spaghetti-like*", menampilkan semua detail tanpa membedakan proses mana yang penting dan mana tidak penting. Oleh karena itu, pendekatan *fuzzy mining* diterapkan untuk mengatasi model proses yang "*spaghetti-like*" tersebut. Untuk melakukannya, konsep peta jalan digunakan sebagai perumpamaannya untuk memvisualisasikan model yang dihasilkan [1].

Gambar 1. Konsep Peta Jalan

Dalam konsep peta jalan Gambar 2.4, dapat dilihat bahwa komponen-komponen yang memiliki peran kurang penting dalam suatu peta jalan tidak akan ditampilkan seperti jalan-jalan kecil. Beberapa bagian kota kecil yang masih dalam satu bagian kota besar akan digabungkan. Pembuatan peta jalan tersebut dan tingkat kerinciannya hanya difokuskan untuk tujuan penggunaannya saja. Konsep tersebut diterapkan pada pendekatan *fuzzy mining* dalam melakukan *process mining*. Untuk melakukan hal tersebut, diperlukan kriteria keputusan yang tepat sebagai dasar penyederhanaan dan visualisasi suatu model proses. Terdapat dua *metrics* dasar yang dapat mendukung keputusan tersebut yaitu *significance* dan *correlation*. *Significance* dapat ditentukan untuk kegiatan (*node*) dan relasi (edge) dalam suatu proses, dengan mengukur nilai *relative importance*-nya. Dengan demikian, hal ini dapat menentukan tingkat kepentingan yang dimiliki oleh suatu kegiatan. Salah satu contoh mengukur *significance* adalah dengan frekuensi, yaitu kegiatan atau relasi yang diamati lebih sering dianggap signifikan. Disisi lain, *correlation* hanya relevan untuk relasi yang lebih diutamakan daripada kegiatannya. *Correlation* mengukur seberapa erat hubungan antar dua kegiatan berikut dengan satu sama lainnya [1].



Gambar 2. Kutipan Model Proses yang Sederhana

Berikut adalah penjelasan *metrics* mengenai pengukuran *significance* dan *correlation* secara detail.

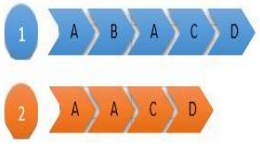
2.2.1. Log-Based Process Metrics

Pentingnya perilaku (*behavior*) dalam proses diperoleh dengan mengukur *significance* dan *correlation*.

Significance yang diukur untuk kegiatan atau *event* dalam proses disebut sebagai *unary significance*, sedangkan *significance* yang diukur untuk hubungan atau relasi dalam proses disebut sebagai *binary significance*. *Correlation* hanya diperkirakan untuk hubungan atau relasi antara dua buah aktivitas atau *event*. *Correlation* mengukur seberapa erat satu kelas *event* terkait dengan kelas *event* lainnya. Dengan demikian, *correlation* sering disebut sebagai *binary correlation*. Ketiga metrik tersebut (*unary significance*, *binary significance*, dan *binary correlation*) akan memberikan informasi untuk membangun model awal (*initial model*) dan akan digunakan untuk menyederhanakan model proses secara adaptif [7].

a. *Unary Significance Metrics*

Unary significance menggambarkan *relative importance* dari kelas *event*, yang akan direpresentasikan sebagai *node* dalam model proses. Salah satu metrik untuk *unary significance* adalah *frequency significance*, yaitu semakin sering kelas *event* tertentu diamati dalam log maka akan semakin signifikan pula kelas *event* tersebut.

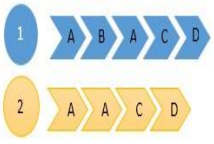


Event Class	A	B	C	D
Frequency	4	1	2	2
Significance	1.0	0.25	0.5	0.5

Gambar 3. Mengukur *Unary Frequency Significance*

b. *Binary Significance Metrics*

Binary significance menggambarkan *relative importance* dari hubungan atau relasi antar dua kelas *event*, yaitu *edge* dalam model proses. Tujuannya adalah untuk memperkuat dan memisahkan perilaku (*behavior*) yang diamati dalam proses. Seperti *unary significance*, metrik *frequency significance* juga digunakan. Semakin sering dua kelas *event* diamati setelah satu sama lain maka semakin signifikan pula hubungan atau relasinya.

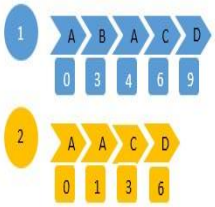


Relation	A→B	B→A	A→C	C→D	A→A
Frequency	1	1	2	2	1
Significance	0.5	0.5	1	1	0.5

Gambar 4. Mengukur *Binary Frequency Significance*

c. *Binary Correlation Metrics*

Binary correlation digunakan untuk mengukur jarak *event* dalam hubungan atau relasinya, yaitu seberapa erat hubungan dua *event* berikut *event-event* lainnya. Jarak, dalam domain proses, bisa disamakan dengan besarnya perubahan konteks antara dua eksekusi aktivitas. Selanjutnya aktivitas yang terjadi memiliki konteks yang lebih mirip (misalnya *event* yang dijalankan oleh orang yang sama atau dalam jangka waktu yang pendek) akan dievaluasi untuk menjadi lebih berkorelasi (*highly correlated*). *Binary correlation* adalah pendorong utama untuk keputusan antara agregasi (*aggregation*) atau abstraksi (*abstraction*) perilaku yang kurang signifikan. *Proximity correlation* mengevaluasi kelas *event* yang terjadi segera setelah satu sama lain. Hal ini penting untuk mengidentifikasi kelompok *event* yang sesuai dengan satu aktivitas logis, karena ini biasanya dilaksanakan dalam jangka waktu yang pendek.



Relation	A→B	B→A	A→C	C→D	A→A
Duration	3	1	2	3	1
Correlation	0.33	1.0	0.5	0.33	1.0

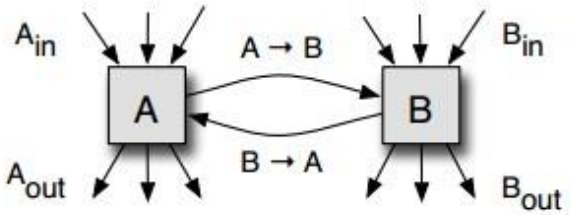
Gambar 5. Mengukur *Proximity Correlation*

2.2.2. *Graph Simplification Process*

Semua kelas *event* yang ditemukan dalam *log* akan dijabarkan ke dalam *node* aktivitas, dimana tingkat kepentingannya dinyatakan oleh *unary significance*. Untuk setiap relasi antar *event* yang diamati, sebuah *edge* berarah yang sesuai akan ditambahkan ke dalam model proses. *Edge* tersebut telah dijelaskan oleh *binary significance* dan *correlation* dari urutan relasi yang diwakilinya. Setelah itu, terdapat tiga metode penyederhanaan untuk model proses, yang secara berurutan akan menyederhanakan aspek-aspek tertentu dari model proses tersebut. Dua fase pertama adalah conflict *resolution* dan *edge filtering* yaitu menghapus *edge* (relasi) antar *nodes* aktivitas, sedangkan fase terakhir adalah *node aggregation* dan *abstraction* yaitu menghapus dan/atau klaster *node* yang kurang signifikan [1]. Berikut adalah penjelasan mengenai fase-fase tersebut.

a. *Binary Conflict resolution*

Setiap dua buah *node* dalam *initial model* yang dihubungkan oleh *edges* di kedua arahnya maka hal tersebut dikatakan sebagai konflik.



Gambar 6. Evaluasi Relative Significance

Gambar 6 menunjukkan contoh dari dua kegiatan A dan B dalam konflik. *Relative significance* untuk edge A → B dapat ditentukan sebagai berikut.

$$rel(A,B) = \frac{1}{2} \cdot \frac{sig(A,B)}{\sum X \in N^{sig(A,X)}} + \frac{1}{2} \cdot \frac{sig(A,B)}{\sum X \in N^{sig(X,B)}} \quad (1)$$

Dimana N adalah himpunan *node* dalam proses, dan *sig* diatas maksudnya adalah *binary significance*. $rel: N\ x\ N \rightarrow R_0^+$ adalah sebuah relasi yang menetapkan setiap pasangan *node* A, B $\in$ N terhadap *relative importance* dari urutan relasi mereka. Jika *relative significance* dari kedua relasi yang mengalami konflik (rel(A,B) dan rel(B,A)) melebihi nilai *preserve threshold*, maka A dan B menandakan bahwa mereka bersifat *length-2-loop* sehingga boleh ditampilkan dalam model proses yang dibuat. Namun, jika satu saja relasi yang mengalami konflik tersebut memiliki nilai *relative significance* dibawah dari *preserve threshold*, maka relasi tersebut akan dihilangkan.

b. *Edge Filtering*

Berbeda dengan *binary conflict resolution*, *edge filtering* akan menghilangkan *edges* yang kurang penting secara global dan hanya meninggalkan *edges* yang memiliki *high significance* saja. Pendekatan ini mengevaluasi setiap *edges* dengan menggunakan *UtilityR util(A,B)* yang merupakan jumlah dari *binary significance* dan *binary correlation*. Parameter *UtilityR*, $ur \in [0,1]$ menentukan nilainya. *Utility* untuk *edge* A → B didefinisikan sebagai berikut [7]:

$$util(A,B) = ur * sig(A,B) + (1 - ur) * cor(A,B) \quad (2)$$

Dimana *sig* menjadi fungsi yang menetapkan *binary significance* suatu relasi, dan *cor* menjadi fungsi yang menetapkan *binary correlation*. Nilai *UtilityR* berkisar dari 0 sampai 1. *Edges* yang memiliki nilai *UtilityR* tertinggi akan ditampilkan. Sebuah nilai *UtilityR* yang lebih besar akan menampilkan *edge* dengan *binary significance* yang tinggi, ketika nilai *UtilityR* yang lebih kecil akan mencadangkan *edge* dengan *binary correlation* yang tinggi. Keputusan akan *edges* mana saja yang akan ditampilkan ditentukan oleh parameter *EdgeCutOff* $\in [0,1]$. Untuk setiap *node* Y, nilai *UtilityR* untuk setiap *edge* yang masuk X → Y akan dinormalisasi ke [0,1], sehingga *edge* dengan nilai terkecil akan diberikan nilai 0 sedangkan *edge* dengan nilai terbesar akan diberikan nilai 1. Semua *edges* yang dinormalisasi apabila nilai *UtilityR* melebihi (1 – *EdgeCutOff*) maka *edges* tersebut akan dipertahankan (ditampilkan dalam model proses). Untuk *edge* yang keluar akan diproses dengan cara yang sama untuk setiap *node*. *EdgeCutOff* yang rendah akan bertindak sebagai penguat (*amplifier*), membantu untuk membedakan *edge* yang paling penting. *EdgeCutOff* yang tinggi akan mempertahankan lebih banyak *edges* [7].

c. *Node Aggregation and Abstraction*

Proses *node aggregation* dan *abstraction* akan mengelompokkan (*cluster*) kelompok *node* yang *less-significance* namun sangat berkorelasi (*highly correlated*). *Node* yang terisolasi dan kurang signifikan akan dihapus. Parameter *NodeCutOff* digunakan untuk menghapus *node* tersebut. Setiap *node* yang memiliki nilai *unary significance* dibawah batas dari parameter *NodeCutOff* akan menjadi korban (*victim node*). Penanganan terhadap *victim node* yaitu dengan membuat *inisial cluster*. Untuk setiap *victim node*, cari *node* tetangga dengan korelasi yang paling tinggi (*node* yang memiliki hubungan atau *edge*). Jika tetangganya adalah sebuah *node cluster*, tambahkan *victim node* ke dalam *cluster*. Jika tidak, sebuah *cluster* baru akan dibuat dan *victim node* akan menjadi anggota pertama dalam *cluster* tersebut [7].

2.3. Conformance Checking

Seperti yang telah dijelaskan sebelumnya, *conformance checking* dapat digunakan untuk mengecek apakah model proses sudah sesuai dengan *event log*-nya. Penulis menggunakan pengukuran *recall*, *precision*, dan *f-measure* pada *conformance* ini yang berdasarkan matrik *artificial negative event*. Teknik dalam menginduksi *negative events* relatif mudah. *Negative events* merekam bahwa pada posisi tertentu dalam sebuah rentetan *event*, *event* tertentu tidak dapat terjadi. Pada setiap posisi dalam *event trace* di *log* akan diperiksa mana *negative event* yang dapat direkam pada posisi tersebut. Berikut adalah matrik dari *artificial negative event* [9].

Tabel 1. Matrik *Artificial Negative Events*

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

- TP : Ketika *actual node* positif dan *prediction node* positif.
- FP : Ketika *actual node* negatif dan *prediction node* positif.
- FN : Ketika *actual node* positif dan *prediction node* negatif.

- TN : Ketika *actual node* negatif dan *prediction node* negatif.

Actual node diperoleh dari *trace* suatu *event log*, dimana kumpulan *event* yang ada pada *trace* tersebut merupakan *positive event* sedangkan *event* lain yang tidak terjadi pada *trace* tersebut merupakan *negative event*. *Prediction node* diperoleh dari model dimana urutan *event* yang mungkin terjadi berdasarkan model tersebut merupakan *prediction positive* sedangkan urutan *event* yang tidak mungkin terjadi merupakan *prediction negative* [9].

2.3.1. Recall

Perhitungan *recall* sudah tidak diragukan lagi sebagai perhitungan yang paling penting bagi evaluasi model proses yang dihasilkan. Sebuah metrik *recall* mencerminkan seberapa banyak perilaku *event* yang ada dalam *event log* ditangkap oleh model. Untuk setiap algoritma *process discovery*, sangat penting untuk suatu model memiliki nilai *recall* yang baik karena tujuan utama dalam *process discovery* adalah model yang merepresentasikan perilaku *control-flow* suatu *event* dalam *event log*. Berikut adalah persamaan dari *recall* (r_B^P) berdasarkan *artificial negative event*.

$$r_B^P = \left(\frac{\sum_{i=1}^{k} n_i TP_i}{\sum_{i=1}^{k} n_i TP_i + \sum_{i=1}^{k} n_i FN_i} \right) \quad (3)$$

Pada persamaan diatas, *k* adalah jumlah grup proses yang berbeda pada *event log*. Indeks *i* berjalan menghampiri semua grup proses tersebut. n_i adalah jumlah *event* dalam satu grup proses yang sama. TP dan FN adalah jumlah TP dan jumlah FN berdasarkan *artificial negative event* sebelumnya. *Recall* memiliki *range* nilai dari 0 sampai 1 dimana semakin besar nilai *recall* maka model yang dihasilkan semakin baik [9].

2.3.2. Precision

Tantangan utama bagi setiap *process discovery* teknik adalah dengan membuat model proses yang akurat,sembari menemukan model yang seimbang antara *underfitting* dan *overfitting*. Evaluasi pengukuran *precision* akan mengecek apakah model proses yang dibuat tidak *underfit* terhadap perilaku *event* yang ada pada *event log*. Berikut adalah persamaan dari *precision* (p_B) berdasarkan *artificial negative event* [9].

$$p_B = \left(\frac{\sum_{i=1}^{k} n_i TP_i}{\sum_{i=1}^{k} n_i TP_i + \sum_{i=1}^{k} n_i FP_i} \right) \quad (4)$$

Persamaan *precision* hampir sama dengan persamaan *recall* akan tetapi pada persamaan *precision* yang digunakan adalah FP. Selain itu, proses perhitungan sama dengan perhitungan *recall*. *Precision* memiliki *range* nilai dari 0 sampai 1 dimana semakin besar nilai *precision* maka model yang dihasilkan semakin baik.

2.3.3. F-measure

Dalam bidang *machine learning* dan *data mining*, *f-measure* sering digunakan sebagai standar keseimbangan antara *precision* dan *recall* untuk mengevaluasi titik pengklasifikasi. Nilai *precision* (p_B) dan *recall* (r_B^P) yang diperoleh sebelumnya akan digunakan dalam perhitungan *f-measure*. Berikut adalah persamaan dari *f-measure* (F) [9].

$$F = 2 \times \frac{p_B \times r_B^P}{p_B + r_B^P} \quad (7)$$

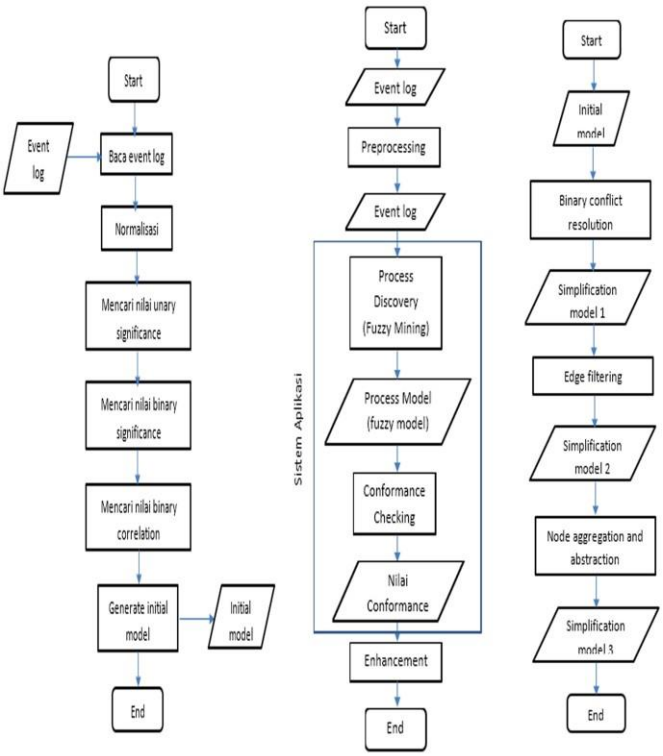
F-measure memiliki range nilai dari 0 sampai 1 dimana semakin besar nilai *f-measure* maka keseimbangan antara *precision* dan *recall* semakin baik.

2.4. Enhancement

Tahap terakhir dalam *process mining* adalah *enhancement*. Proses *enhancement* dapat dilakukan dalam dua bentuk yaitu *repair* atau *extension*. *Repair* adalah memodifikasi model untuk lebih mencerminkan realitas sedangkan *extension* adalah menambahkan perspektif baru berdasarkan event log yang tersedia maupun dari model proses yang dihasilkan [4].

2.5. Perancangan Sistem

Sistem yang akan dibangun pada tugas akhir ini adalah sistem yang akan menerapkan metode *fuzzy mining* dalam membuat sebuah model proses dari *event log incident activity* pada *service desk* di *Rabobank Netherlands Group ICT*. Masukan dari sistem ini berupa data *event log* yang sudah dilakukan *preprocessing* secara manual dan keluaran dari sistem ini berupa sebuah model proses (*fuzzy model*).



Gambar 7. Perancangan Sistem

Gambar 7 merupakan perancangan sistem secara umum. Pada gambar 7 terdapat tiga buah *flowchart* diantaranya perancangan sistem secara umum (tengah), pembuatan *initial model* (kiri), dan *graph simplification process* (kanan). Pembuatan *initial model* dan *graph simplification process* dilakukan pada saat *process discovery* (*fuzzy mining*).

3. Pengujian dan Analisis

3.1. Tujuan Pengujian

Pengujian yang dilakukan adalah pengujian terhadap program aplikasi dalam menerapkan metode *fuzzy mining* untuk memperoleh *fuzzy model* serta melihat *conformance* yang dilakukan oleh sistem sudah benar sesuai dengan teori yang digunakan. Untuk mendapatkan *fuzzy model* yang bagus, penetapan *threshold* yang digunakan dalam *graph simplification* sangat menentukan oleh karena itu diperlukan beberapa pengujian terhadap masing-masing *threshold*. Acuan dalam menetapkan *threshold* yang akan digunakan adalah nilai *conformance* yang dihasilkan dari *threshold* tersebut. Selain itu, dilakukan proses *enhancement* jenis *extension* dalam menganalisis *originator* suatu *event* dengan membuat daftar *originator* untuk semua *event* berdasarkan data yang digunakan.

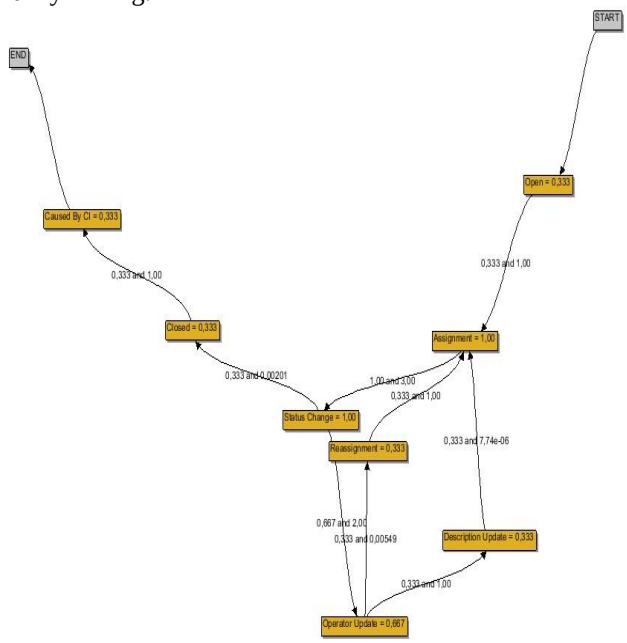
3.2. Hasil Pengujian Nilai *Conformance* Sistem

Digunakan data contoh sebagai bahan pengujian. Berikut adalah data yang digunakan dalam pengujian skenario pertama ini.

```
Incident ID,DateStamp,IncidentActivity_Number,IncidentActivity_Type
IM0043584,14/03/2014 11:26,001A6800653,Open,TEAM0008
IM0043584,14/03/2014 11:27,001A6800656,Assignment,TEAM0008
IM0043584,14/03/2014 11:27,001A6800657,Status Change,TEAM0008
IM0043584,14/03/2014 11:27,001A6800658,Operator Update,TEAM0008
IM0043584,14/03/2014 14:29,001A6802138,Reassignment,TEAM0008
IM0043584,14/03/2014 14:29,001A6802139,Assignment,TEAM0008
IM0043584,14/03/2014 14:29,001A6802140,Status Change,TEAM0008
IM0043584,14/03/2014 14:29,001A6802141,Operator Update,TEAM0008
IM0043584,14/03/2014 14:29,001A6802142,Description Update,TEAM0008
IM0043584,17/03/2014 6:55,001A6802734,Assignment,TEAM0192
IM0043584,17/03/2014 6:55,001A6802735,Status Change,TEAM0192
IM0043584,17/03/2014 14:12,001A6814149,Closed,TEAM0192
IM0043584,17/03/2014 14:12,001A6814150,Caused By CI,TEAM0192
```

Gambar 8. Data Pengujian Nilai *Conformance*

Dari data tersebut dibuatlah sebuah model proses, gambar 9 adalah model proses yang dihasilkan oleh fuzzy mining.



Gambar 9. Model Proses Data Pengujian Nilai *Conformance*

Dibuat inisialisasi untuk setiap nama *event* yang terjadi berdasarkan gambar 8 (*Open* = A, *Assignment* = B, *Status Change* = C, *Operator Update* = D,

Reassignment = E, *Description Update* = F, *Closed* = G, *Caused By CI* = H). Setelah itu cari aktual dan prediksi dari *trace* IM0043584 (ABCDEBCDFBCGH).

Tabel 2. Aktual *Trace* IM0043584

Aktual Positif	A	B	C	D	E	B	C	D	F	B	C	G	H
Aktual Negatif	B	A	A	A	A	A	A	A	A	A	A	A	A
	C	C	B	B	B	C	B	B	B	C	B	B	B
	D	D	D	C	C	D	D	C	C	D	D	C	C
	E	E	E	E	D	E	E	E	D	E	E	D	D
	F	F	F	F	F	F	F	F	E	F	F	E	E
	G	G	G	G	G	G	G	G	G	G	G	F	F
	H	H	H	H	H	H	H	H	H	H	H	H	G

Tabel 3. Prediksi *Trace* IM0043584

Prediksi Positif	A	B	C	D/G	E/F	B	C	D/G	E/F	B	C	D/G	H
Prediksi Negatif	B	A	A	A	A	A	A	A	A	A	A	A	A
	C	C	B	B	B	C	B	B	B	C	B	B	B
	D	D	D	C	C	D	D	C	C	D	D	C	C
	E	E	E	E	D	E	E	E	D	E	E	E	D
	F	F	F	F	G	F	F	F	G	F	F	F	E
	G	G	G	H	H	G	G	H	H	G	G	H	F
	H	H	H			H	H			H	H		G

Dari tabel 2 dan 3 dapat dilakukan perhitungan TP, FP, dan FN untuk mendapatkan nilai *precision* dan *recall* dilanjutkan dengan *f-measure*. Berdasarkan data yang ada pada tabel-tabel diatas, TP = 13, FP = 0, dan FN = 0. Sehingga:

$$p_B = \left(\frac{13}{13+0}\right) = 1$$

$$r_B^P = \left(\frac{13}{13+0}\right) = 1$$

$$F = 2 \times \frac{1 \times 1}{1+1} = 1$$

Gambar 10. Precision, Recall, dan F-measure *Trace* IM0043584

3.3. Hasil Pengujian *Threshold*

Pada pengujian ini, data yang digunakan adalah data *Detail Incident Activity*.csv dari *BPI Challenge 2014* sebanyak 100.000 baris data yang terbaru. Dilakukan proses pencarian *conformance* terbaik untuk masing-masing *threshold* terlebih dahulu (*preserve threshold*, *edge cutoff*, *utility ratio*, dan *node cutoff*). Pemilihan *threshold* dimulai dari yang terkecil dan terus naik dengan kelipatan 0,05 hingga *threshold* mencapai nilai 1. Pada *preserve threshold*, semakin besar nilai *threshold* maka akan semakin rendah *conformance* yang dihasilkan begitu pula sebaliknya. Pada *edge cutoff* (nilai *utility ratio* yang digunakan adalah nilai tengahnya), semakin besar nilai *threshold* maka akan semakin rendah *conformance* yang dihasilkan begitu pula sebaliknya. Pada *utility ratio* (*edge cutoff* yang digunakan adalah *edge cutoff* dengan *conformance* terbaik berdasarkan pencarian *edge cutoff* sebelumnya), semakin bersar nilai *threshold* maka akan semakin besar juga *conformance* yang dihasilkan begitu pula sebaliknya. Pada *node cutoff*, semakin besar nilai *threshold* maka akan semakin rendah *conformance* yang dihasilkan. Namun pada *node cutoff*, conformance yang dihasilkan dapat naik dan turun karena ada kemungkinan dimana *threshold* tertentu membuat lebih banyak *node* yang terisolasi maupun *node* dengan anggota *cluster* hanya *node* itu sendiri sehingga *node-node* tersebut akan dihilangkan dari model proses yang dapat berpengaruh pada nilai TP, FP, dan FN.

Berdasarkan pengujian diatas, untuk *event log BPI Challenge 2014* ini, secara umum penggunaan *threshold* yang rendah akan menghasilkan *conformance* yang lebih baik kecuali dalam penggunaan parameter *utility ratio* yang berlaku sebaliknya. Sehingga jika ingin menggunakan kombinasi *threshold* pada *graph simplification process* dapat dilakukan dengan menggunakan *preserve threshold* yang rendah, *utility ratio* yang tinggi, *edge cutoff* yang rendah, dan *node cutoff* yang rendah. Berdasarkan kesimpulan tersebut dilakukan pengujian *conformance* dengan kombinasi *threshold* dan diperoleh kombinasi *threshold* terbaik yaitu *preserve threshold* 0,05, *utility ratio* 0,85, *edge cutoff* 0,05, dan *node cutoff* 0,05.

Tabel 4. *Conformance* Kombinasi *Threshold* dengan *Preserve Threshold* Diubah

PT	EC	UR	NC	Precision	Recall	F-Measure
0,05	0,05	0,85	0,05	0,906	0,882	0,893
0,1	0,05	0,85	0,05	0,870	0,843	0,856
0,15	0,05	0,85	0,05	0,830	0,783	0,806
0,2	0,05	0,85	0,05	0,686	0,657	0,671
0,25	0,05	0,85	0,05	0,642	0,612	0,627
0,3	0,05	0,85	0,05	0,642	0,612	0,627
0,35	0,05	0,85	0,05	0,535	0,513	0,524
0,4	0,05	0,85	0,05	0,535	0,513	0,524
0,45	0,05	0,85	0,05	0,446	0,430	0,438
0,5	0,05	0,85	0,05	0,436	0,399	0,417
0,55	0,05	0,85	0,05	0,436	0,399	0,417
0,65	0,05	0,85	0,05	0,268	0,264	0,266
0,7	0,05	0,85	0,05	0,268	0,264	0,266
0,75	0,05	0,85	0,05	0,268	0,264	0,266
0,8	0,05	0,85	0,05	0,268	0,264	0,266
0,85	0,05	0,85	0,05	0,268	0,264	0,266
0,9	0,05	0,85	0,05	0,268	0,264	0,266
0,95	0,05	0,85	0,05	0,187	0,184	0,185
1	0,05	0,85	0,05	0,187	0,184	0,185

Tabel 5. *Conformance* Kombinasi *Threshold* dengan *Edge Cutoff* Diubah

PT	EC	UR	NC	Precision	Recall	F-Measure
0,05	0,05	0,85	0,05	0,906	0,882	0,893
0,05	0,1	0,85	0,05	0,863	0,835	0,849
0,05	0,15	0,85	0,05	0,808	0,760	0,783
0,05	0,2	0,85	0,05	0,665	0,631	0,648
0,05	0,25	0,85	0,05	0,601	0,564	0,582
0,05	0,3	0,85	0,05	0,542	0,508	0,525
0,05	0,35	0,85	0,05	0,328	0,326	0,327
0,05	0,4	0,85	0,05	0,328	0,326	0,327
0,05	0,45	0,85	0,05	0,328	0,326	0,327
0,05	0,5	0,85	0,05	0,328	0,326	0,327
0,05	0,55	0,85	0,05	0,413	0,401	0,407
0,05	0,6	0,85	0,05	0,413	0,401	0,407
0,05	0,65	0,85	0,05	0,408	0,374	0,391
0,05	0,7	0,85	0,05	0,408	0,374	0,391
0,05	0,75	0,85	0,05	0,408	0,374	0,391
0,05	0,8	0,85	0,05	0,122	0,122	0,122
0,05	0,85	0,85	0,05	0,122	0,122	0,122
0,05	0,9	0,85	0,05	0,032	0,032	0,032
0,05	0,95	0,85	0,05	0,032	0,032	0,032
0,05	1	0,85	0,05	0,032	0,032	0,032

Tabel 6. *Conformance* Kombinasi *Threshold* dengan *Utility Ratio* Diubah

PT	EC	UR	NC	Precision	Recall	F-Measure
0,05	0,05	0	0,05	0,269	0,242	0,255
0,05	0,05	0,05	0,05	0,360	0,323	0,341
0,05	0,05	0,1	0,05	0,508	0,479	0,493
0,05	0,05	0,15	0,05	0,549	0,515	0,531
0,05	0,05	0,2	0,05	0,640	0,598	0,618
0,05	0,05	0,25	0,05	0,752	0,714	0,732
0,05	0,05	0,3	0,05	0,812	0,765	0,788
0,05	0,05	0,35	0,05	0,862	0,821	0,841
0,05	0,05	0,4	0,05	0,867	0,841	0,854
0,05	0,05	0,45	0,05	0,879	0,851	0,865
0,05	0,05	0,5	0,05	0,882	0,855	0,868
0,05	0,05	0,55	0,05	0,882	0,855	0,868
0,05	0,05	0,6	0,05	0,890	0,862	0,876
0,05	0,05	0,65	0,05	0,890	0,862	0,876
0,05	0,05	0,7	0,05	0,890	0,862	0,876
0,05	0,05	0,75	0,05	0,895	0,869	0,882
0,05	0,05	0,8	0,05	0,902	0,875	0,888
0,05	0,05	0,85	0,05	0,906	0,882	0,893
0,05	0,05	0,9	0,05	0,906	0,882	0,893
0,05	0,05	0,95	0,05	0,906	0,882	0,893
0,05	0,05	1	0,05	0,751	0,744	0,747

Tabel 7. *Conformance* Kombinasi *Threshold* dengan *Node Cutoff* Diubah

PT	EC	UR	NC	Precision	Recall	F-Measure
0,05	0,05	0,85	0,05	0,906	0,882	0,893
0,05	0,05	0,85	0,1	0,713	0,709	0,711
0,05	0,05	0,85	0,15	0,713	0,709	0,711
0,05	0,05	0,85	0,2	0,713	0,709	0,711
0,05	0,05	0,85	0,25	0,713	0,709	0,711
0,05	0,05	0,85	0,3	0,713	0,709	0,711
0,05	0,05	0,85	0,35	0,713	0,709	0,711
0,05	0,05	0,85	0,4	0,713	0,709	0,711
0,05	0,05	0,85	0,45	0,857	0,840	0,848
0,05	0,05	0,85	0,5	0,857	0,840	0,848
0,05	0,05	0,85	0,55	0,545	0,542	0,544
0,05	0,05	0,85	0,6	0,679	0,669	0,674
0,05	0,05	0,85	0,65	0,314	0,313	0,314
0,05	0,05	0,85	0,7	0,314	0,313	0,314
0,05	0,05	0,85	0,75	0,314	0,313	0,314
0,05	0,05	0,85	0,8	0,314	0,313	0,314
0,05	0,05	0,85	0,85	0,314	0,313	0,314
0,05	0,05	0,85	0,9	0,314	0,313	0,314
0,05	0,05	0,85	0,95	0,314	0,313	0,314
0,05	0,05	0,85	1	0,314	0,313	0,314

3.4. Hasil Pengujian Hubungan Threshold dengan Karakteristik Data

Berikut adalah tabel yang menjelaskan perbedaan antara model proses yang sudah disimplifikasi dan *initial model* berdasarkan *node* dan *edge*.

Tabel 8. Pengaruh *Threshold* Terhadap *Node* dan *Edge*

Model Proses	Node	Edge
Initial Model	37	457
Model Proses PT 0,05	37	243
Model Proses EC 0,05 & UR 0,5	36	153
Model Proses EC 0,05 & UR 0,85	36	140
Model Proses NC 0,05	15	172
Model Proses PT 0,05, EC 0,05, UR 0,85, & NC 0,05	13	37

Pada tabel 8, dapat dilihat perbedaan antara jumlah *node* dan *edge* yang ditampilkan pada model proses. Berikut adalah analisis keterkaitan *threshold* terhadap data yang digunakan yaitu data *Detail Incident Activity*.csv dari *BPI Challenge 2014* sebanyak 100.000 baris data yang terbaru.

- *Initial model* merupakan model awal yang hanya merubah *event* menjadi *node* dan relasi menjadi *edge* sehingga semuanya ditampilkan pada model tersebut.
- Model proses dengan PT 0,05 memiliki *node* yang sama dengan *initial model* namun *edge* yang berbeda dengan pemotongan sebanyak 214 *edges*. Berdasarkan subbab 2.4.2 di bagian *binary conflict resolution*, *preserve threshold* melakukan penyeleksian *edge* sehingga *node* pada model proses tidak berubah.
- Model proses EC 0,05 dan UR 0,5 memiliki *node* dan *edge* yang berbeda dari *initial model*. Perubahan *edge* terjadi karena *edge cutoff* dan *utility ratio* adalah bagian dari *edge filtering* yang merupakan proses penyeleksian *edge* namun terjadi pemotongan *node* sebanyak 1 dikarenakan pada saat melakukan proses penyeleksian *edge* terdapat *node* yang terisolasi karena proses tersebut sehingga *node* yang terisolasi itu dihilangkan oleh proses *abstraction*.
- Model proses EC 0,05 dan UR 0,85 memiliki *node* dan *edge* yang berbeda dari *initial model*. Perubahan *edge* terjadi karena *edge cutoff* dan *utility ratio* adalah bagian dari *edge filtering* yang merupakan proses penyeleksian *edge* namun terjadi pemotongan *node* sebanyak 1 dikarenakan pada saat melakukan proses

penyeleksian *edge* terdapat *node* yang terisolasi karena proses tersebut sehingga *node* yang terisolasi itu dihilangkan oleh proses *abstraction*.

- Model proses NC 0,05 memiliki *node* dan *edge* yang berbeda dari *initial model*. Berdasarkan subbab 2.4.2 di bagian *node aggregation dan abstraction*, *node cutoff* melakukan penyeleksian *node* yang berpengaruh terhadap jumlah *node* yang akan ditampilkan. Selain itu terdapat proses agregasi suatu *node* yang berdampak terhadap berkurangnya jumlah *node*. Dengan dihilangkannya suatu *node*, maka relasi yang berhubungan dengan *node* tersebut juga akan dihilangkan sehingga seiring berkurangnya jumlah *node* yang ditampilkan maka jumlah *edge* yang ditampilkan juga ikut berkurang.
- Model proses PT 0,05, EC 0,05, UR 0,85, dan NC 0,05 memiliki *node* dan *edge* yang berbeda dari *initial model*. Pada model proses ini, banyak *node* dan *edge* yang diseleksi karena kombinasi *threshold* yang digunakan. Semakin banyak *threshold* yang digunakan maka akan semakin banyak konten (*node* dan *edge*) yang akan diseleksi.

3.5. Analisis Originator

Dalam melakukan analisis *originator*, penulis melakukan analisis berdasarkan tabel *originator* yang dibuat berdasarkan atribut *assignment group* pada data *BPI Challenge* 2014 yang digunakan oleh penulis.

No.	Event	Frekuensi Event	Assignment Group	Frekuensi Assignment Group
1	Affected CI Change	30	TEAM9999	1
2	Analysis / Research	207	TEAM0024, TEAM0197, TEAM0188, TEAM0172, TEAM0224, TEAM0028, TEAM0034, TEAM0209, TEAM0180, TEAM0212, TEAM0148, TEAM0187, TEAM0207, TEAM0217, TEAM0202, TEAM0165, TEAM0081, TEAM0178, TEAM0063, TEAM0071, TEAM0127, TEAM0166, TEAM0227, TEAM0104, TEAM0226, TEAM0176, TEAM0134, TEAM0163, TEAM0186, TEAM0114, TEAM9999, TEAM0079, TEAM0017	33

Gambar 11. Contoh Tabel *Originator Event*

No.	Assignment Group	Event
1	TEAM0002	Assignment, Caused By CI, Closed, Description Update, Impact Change, Open, Quality Indicator Fixed Reassignment, Reopen, Status Change, Update, Update from customer, Urgency Change
2	TEAM0005	Assignment, Operator Update, Reassignment

Gambar 12. Contoh Tabel *Event* yang Dikerjakan *Assignment Group*

Berikut adalah hasil analisis yang dapat penulis temukan berdasarkan tabel *originator* yang dibuat:

- Pembagian ranah kerja suatu tim kurang diatur dengan baik. Contohnya seperti TEAM0197 yang melakukan 22 jenis *event* dan terdapat tim yang hanya melakukan 1 jenis *event* saja seperti TEAM0238 yaitu *reassignment*.
- Dilakukan perhitungan standar deviasi untuk mengetahui apakah beban kerja setiap tim yang ada pada *event* yang sama seimbang atau tidak. Sebagai contoh diambil *event* dengan frekuensi yang besar seperti *assignment*, *closed*, dan *status change*. Pada *assignment's event*, dari 18.595 frekuensi kejadian yang ditangani 131 tim diperoleh rata-rata pengerjaan 141,9466 per tim dengan standar deviasi 334,3000567. Pada *closed's event*, dari 11.951 frekuensi kejadian yang ditangani 116 tim diperoleh rata-rata pengerjaan 103,0258621 per tim dengan standar deviasi 223,4974355. Pada *status change's event*, dari 11.487 frekuensi kejadian yang ditangani 94 tim diperoleh rata-rata pengerjaan 122,2021277 per tim dengan standar deviasi 326,5173209. Berdasarkan perbandingan nilai standar deviasi dengan rata-rata tersebut dapat dilihat bahwa persebaran beban kerja pada suatu *event* masih kurang seimbang.

Sebaiknya dilakukan pembagian ranah kerja yang jelas untuk setiap tim dalam proses bisnis yang dilakukan. Selain itu, pembagian beban kerja yang seimbang pada tim dengan *event* yang sama perlu dilakukan.

4. Kesimpulan

Berdasarkan hasil pengujian yang diperoleh dan analisis yang telah dilakukan dapat diberikan kesimpulan sebagai berikut:

1. Metode *fuzzy mining* dapat diterapkan pada *event log* yang kompleks seperti *event log BPI Challenge* 2014 untuk memperoleh model proses (*fuzzy model*).
2. Penggunaan *threshold* pada *Graph simplification process* akan menghasilkan *conformance* yang baik dengan menggunakan *preserve threshold* yang rendah, *utility ratio* yang tinggi, *edge cutoff* yang rendah, dan *node cutoff* yang rendah. Berdasarkan pengujian, model proses dengan *conformance* terbaik yaitu *preserve threshold* 0.05, *utility ratio* 0.85, *edge cutoff* 0.05, dan *node cutoff* 0.05.
3. Pembagian ranah kerja kurang diatur baik yang menyebabkan suatu tim mengerjakan *event* yang sangat banyak sedangkan terdapat tim lainnya yang hanya mengerjakan satu *event* saja. Selain itu, beban kerja suatu tim pada *event* yang sama tidak seimbang. Oleh karena itu, sebaiknya dilakukan pembagian

ranah kerja yang jelas untuk setiap tim dalam proses bisnis yang dilakukan dan pembagian beban kerja yang seimbang pada tim dengan *event* yang sama perlu dilakukan.

5. Daftar Pustaka

[1] C. W. Gunther and W. M. V. d. Aalst, "Fuzzy Mining – Adaptive Process Simplification Based on Multi-Perspective Metrics".

[2] B. Andersson, I. Bider, P. Johannesson and E. Perjons, "Towards a Formal Definition of Goal-Oriented Business," *Business Process Management Journal,* vol. 11, 2005.

[3] B. V. Dongen and A. Adriansyah, "Process Mining: Fuzzy Clustering and Performance Visualization," *Springer,* 2010.

[4] W. V. D. Aalst, Process Mining Discovery, Conformance and Enhancement of Business Processes, Germany: Springer, 2011.

[5] L. Mardhatillah, M. Er and R. P. Kusumawardani, "Identifikasi Bottleneck pada Hasil Ekstraksi Proses Bisnis ERP dengan Membandingkan Algoritma Alpha++ dan Heuristic Miner," *JURNAL TEKNIK ITS* , vol. Vol.1, pp. A322-A327, 2012.

[6] F. Famili, W. M. Shen, R. Weber and E. Simoudis, "Data Pre-Processing and Intelligent Data Analysis," *Intelligent Data Analysis,* 1997.

[7] J. Xia, "Automatic Determination of Graph Simplification Parameter Values for Fuzzy Miner," 2010.

[8] C. W. Gunther, "Process Mining in Flexible Environments," 2009.

[9] J. D. Weerdt, M. D. Backer, J. Vanthienen and B. Baesens, "A Robust F-measure for Evaluating Discovered Process Models," 2011.

[10] S. Aksoy and R. M. Haralick, "F eature Normalization and Likelihood-based Similarity Measures for Image Retrieval," 2000.

[11] Y. C. Wang, Q. C. Zhao and D. Z. Zheng, "BOTTLENECKS IN PRODUCTION NETWORKS: AN OVERVIEW," 2005.

[12] A. Adriansyah, "Performance Analysis of Business Processes from Event Logs and Given Process Models," *Master's Thesis, EINDHOVEN UNIVERSITY OF TECHNOLOGY*, 2009.