Analisis Perbandingan Metode *Alpha Miner*, *Inductive Miner* dan *Causal-Net Mining* dalam Proses *Mining*

Rissa Aulia Hasyim
Jurusan Teknik Informatika
Universitas Islam Negeri Maulana Malik Ibrahim
Malang, Indonesia
17650044@student.uin-malang.ac.id

Muhammad Ainul Yaqin
Jurusan Teknik Informatika
Universitas Islam Negeri Maulana Malik Ibrahim
Malang, Indonesia
yaqinov@ti.uin-malang.ac.id

Adi Heru Utomo
Jurusan Teknologi Informasi
Politeknik Negeri Jember
Jember, Indonesia
adiheruutomo@polije.ac.id

Abstract—**Not all process mining algorithms can detect all model process scenarios, so experiments are carried out by trying 3 types of algorithms against 9 business process model scenarios in order to find the most suitable algorithm for each process model scenario. We use 3process algorithms mining including alpha miner, inductive miner and causal-net mining. We propose a solution using the ProM application to check the suitability of the 3 algorithms used against 9 scenarios. In addition, to support the results of the mining process using ProM, we measure the similarity value by comparing the process model on the dataset with the results of the mining process using ProM. Based on the similarity measurement, it is known that the experiment uses algorithm alpha miner. Figure 8 has the highest similarity level with a value of 0.89. While the smallest level of similarity is found in Figure 7 using alpha miner with a value of 0.12.**

Keywords—Process mining, discovery, alpha miner, inductive miner, causal-net mining, similarity.

Abstrak— **Tidak semua algoritma proses *mining* dapat mendeteksi semua skenario model proses, sehingga dilakukan eksperimen dengan mecoba 3 jenis algoritma terhadap 9 skenario model proses bisnis yang bertujuan untuk mendapatkan algoritma yang paling cocok untuk setiap skenario model proses. Kami menggunakan 3 algoritma proses *mining* diantaranya *alpha miner, inductive miner dan causal-net mining.* Kami mengusulkan solusi dengan menggunakan aplikasi ProM untuk mengecek kecocokan 3 algoritma yang digunakan terhadap 9 skenario. Selain itu, untuk mendukung hasil proses mining menggunakan ProM, kami mengukur nilai similarity dengan membandingkan model proses pada dataset dengan hasil proses mining menggunakan ProM. Berdasarkan hasil pengukuran *similarity* diketahui bahwa eksperimen menggunakan algortima *alpha miner*. Pada figure 8 memiliki nilai tingkat similarity paling tinggi yaitu dengan nilai 0.89. Sedangkan tingkat similarity paling kecil, didapati pada figure 7 menggunakan alpha miner dengan nilai 0.12.**

Keywords—Proses mining, discovery, alpha miner, inductive miner, causal-net mining, similarity.

PENDAHULUAN

Proses *mining* adalah proses mendapatkan informasi dari basis data dalam bentuk pola menggunakan metode tertentu. Proses *mining* merupakan cara untuk mendapatkan informasi baru dan objektif tentang proses bisnis yang sedang dilaksanakan oleh suatu organisasi. Proses *mining* dapat dilakukan dengan mengambil data nyata atau *event log*, kemudian memeriksa kesesuaian *event log* terhadap proses bisnis [1].

Ada tiga jenis proses mining yaitu: *discovery* (penemuan), *conformance* (kesesuaian) dan *enhancement* (peningkatan). Adapun penjelasan untuk masing masing jenis proses *mining* sebagai berikut [2].

1) *Discovery*
   Discovery adalah proses mining tanpa melibatkan model proses. Model proses bisnis ini hanya terbentuk dari *event log* yang sudah tersedia. Cara kerja dari teknik *discovery* adalah menganalisis suatu *event log* dan kemudian dari *event log* teknik ini akan memuat sebuah model tanpa menggunakan informasi yang apriori. Teknik ini merupakan teknik yang paling sering digunakan dari proses *mining*.
2) *Conformance*
   Conformance merupakan model proses yang terlibat. Tujuan dari *conformance* adalah untuk memeriksa apakah model yang telah ada sesuai dengan yang terjadi pada kenyataan atau sebaliknya. Aplikasi *conformance checking* sangatlah luas, dapat dilakukan pada model prosedural, model organisasi, kebijakan bisnis dan lain-lain.
3) *Enhancement*
   Enhancement juga melibatkan sebuah model proses, model yang ada diperluas dengan aspek yang baru. Tujuannya yaitu untuk memperkaya atau memperluas model yang sudah ada dengan informasi yang didapat dari *event log* di dunia nyata, dan untuk mengganti atau menambah model apriori. Sebagai contoh: menggunakan *timestamp* pada *event log* dapat memperluas model sehingga dapat menunjukkan dimana terjadi *bottleneck*, kepuasan pelayanan, waktu eksekusi dan frekuensi.

Adapun beberapa algoritma *discovery* dalam proses *mining* sebagai berikut.

a) *Alpha miner* adalah algoritma yang digunakan dalam proses *mining*, yang bertujuan untuk merekonstruksi kausalitas dari serangkaian *event log*. Algoritma *alpha miner* didefinisikan dalam istilah *Petri Nets* (*Place/Transition Nets*). Algoritma ini pertama kali dikemukakan oleh "van der Aalast", seorang profesor di Departemen Matematika dan Ilmu Komputer dari Technische Universiteit Eindhoven (TU/e) [3].
b) *Inductive Miner* adalah algoritma yang menambang pohon proses dari *event log*. Pendekatan algoritma *Inductive Miner* bekerja secara rekursif, yaitu membagi dan menaklukkan: membagi log (*split log*), kemudian membangun bagian dari proses *tree*. Kemudian proses akan dilanjutkan dengan menangani bagian log yang terbelah secara terpisah. Algoritma dikatakan bagus

DOI: https://doi.org/10.25047/jtit.v7i2.165

saat berurusan dengan log besar dan berisi perilaku unik [4].

c) *Causal-net* (C-*net*) adalah grafik di mana node mewakili aktivitas dan busur mewakili dependensi kausal. Setiap aktivitas memiliki satu set binding input yang mungkin dan satu set binding output yang mungkin [5].

d) *Heuristics Miner* adalah algoritma yang menggunakan pendekatan heuristik. Menurut Weijters [4] algoritma *heuristic miner* memiliki tiga langkah utama, yaitu: (1) membuat grafik dependensi, (2) membuat tanda input dan *output* untuk setiap aktivitas dan (3) mencari hubungan ketergantungan jarak jauh. Hasil penelitian Weber [4] menunjukkan bahwa *heuristic miner* merupakan algoritma yang tangguh dalam menangani jenis kebisingan tertentu.

Tidak semua algoritma proses *mining* dapat mendeteksi semua skenario model proses, sehingga dilakukan eksperimen dengan mecoba 3 jenis algoritma terhadap 9 skenario [6] model proses bisnis yang bertujuan untuk mendapatkan algoritma yang paling cocok untuk setiap skenario model proses. Kami menggunakan 3 algoritma proses *mining* diantaranya *alpha miner, inductive miner dan causal-net mining.*

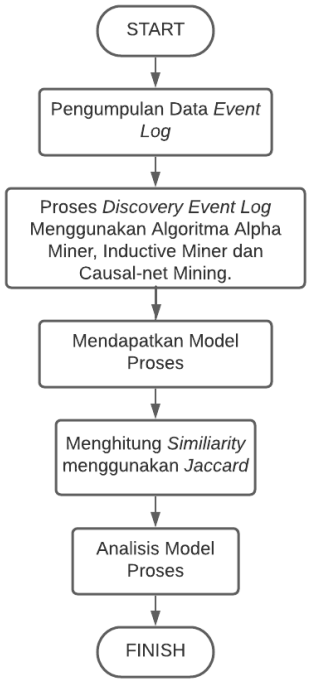
Beberapa penelitian terkait proses *mining* yang sudah pernah dilakukan sebelumnya. Antara lain, Accorsi [7] menjelaskan potensi proses *mining* sebagai dasar untuk audit keamanan proses bisnis dan sistem manajemen proses bisnis yang sesuai. Secara khusus, ini berfokus pada proses *discovery* sebagai sarana untuk merekonstruksi struktur terkait proses dari *event log*, seperti aliran kontrol proses, jaringan sosial, dan aliran data. Berdasarkan informasi ini, analisis keamanan untuk menentukan kepatuhan terhadap persyaratan keamanan dan privasi dapat diotomatiskan. Selain itu, Aalst [8] melakukan penelitian dalam upaya pengujian *conformance* untuk mengukur '*fit*' antara *event log* dan beberapa model proses yang telah ditentukan sebelumnya. Dalam makalah ini, peneliti menunjukkan bahwa analisis Delta dan pengujian *confermance* dapat digunakan untuk menganalisis penyelarasan bisnis selama *actual event* dicatat dan *user* memiliki kendali atas proses tersebut. Selanjtnya, Caldeira [9] melakuka penelitian yang bertujuan untuk memberikan wawasan baru tentang proses pengembangan perangkat lunak dengan menganalisis cara pengembang menggunakan IDE mereka. Berdasarkan teknik proses *mining* seperti proses *discovery* dan *confermance checking*, perspektif yang hilang ini diharapkan akan memungkinkan penemuan pola pengkodean, pencarian perilaku programmer, dan deteksi penyimpangan dari proses yang ditentukan. peneliti berharap dapat memberikan saran untuk peningkatan proses perangkat lunak individu.

Pada penelitian ini, kami meggunakan *discovery* untuk melakukan proses *mining*, karena diketahui bahwa proses *discovery* dibangun dari bawah ke atas khusus untuk RPA - itulah sebabnya proses *discovery* memberikan yang lebih cepat, lebih fleksibel, pendekatan yang lebih komprehensif, dan lebih hemat biaya untuk secara otomatis mengidentifikasi dan menganalisis proses kerja otomatis.

Kami mengusulkan solusi dengan menggunakan aplikasi ProM untuk mengecek kecocokan 3 algoritma yang digunakan terhadap 9 skenario. Selain itu, untuk mendukung hasil proses *mining* menggunakan ProM, kami mengukur nilai *structural similarity* dengan membandingkan model proses pada dataset dengan hasil proses *mining* menggunakan ProM.

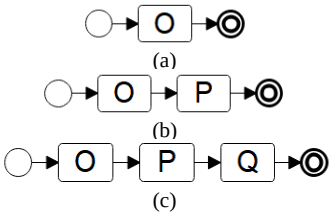
METODE PENELITIAN

Metodologi penelitian yang diimplementasikan terbagi kedalam beberapa tahapan, ditunjukkan pada prosedur penelitian pada Gambar. 1 sebagai berikut.

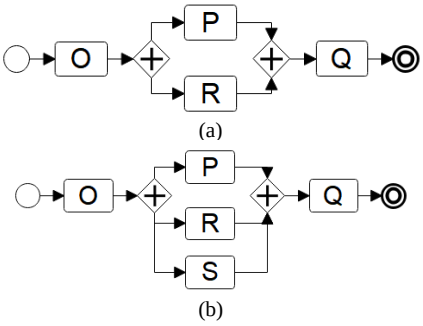


Gambar. 1. Prosedur Penelitian

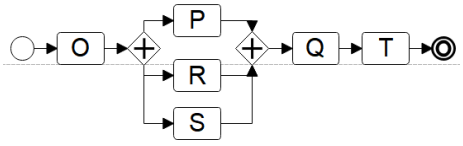
Pada penelitian ini kami menggunakan dataset yang terdiri dari beberapa model proses bisnis [6].

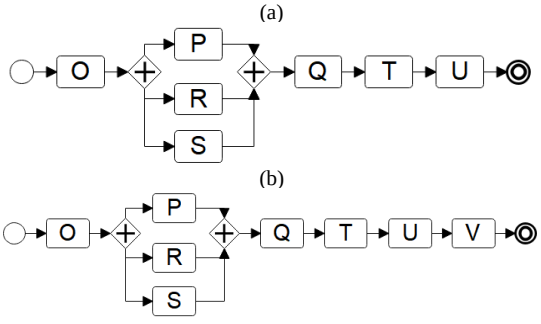


Gambar. 2. Model proses bisnis sequence



Gambar. 3. Model proses bisnis dengan cabang AND



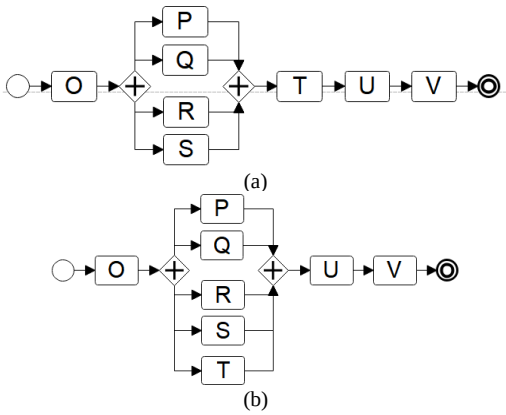


(a)

(b)

(c)

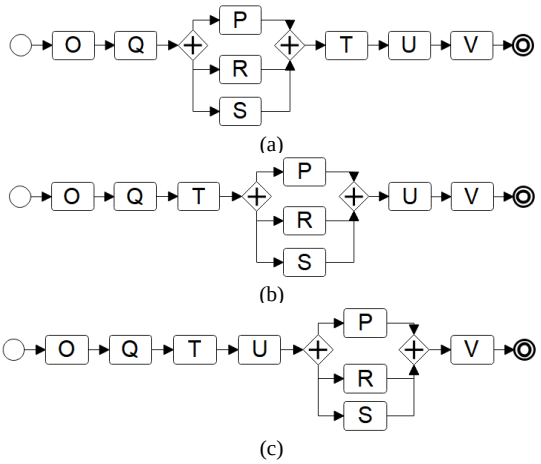
Gambar. 4. Model proses bisnis *sequence* dengan penambahan kegiatan



(a)

(b)

Gambar. 5. Model proses bisnis dengan variasi jumlah cabang.

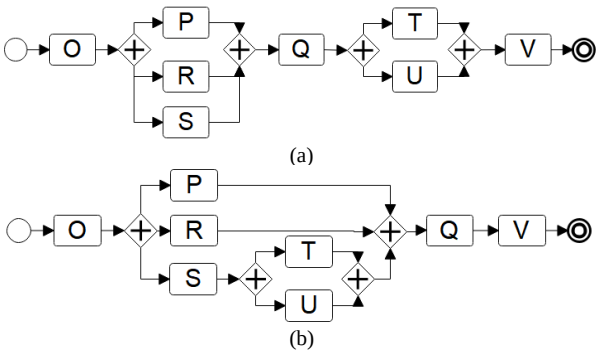


(a)

(b)

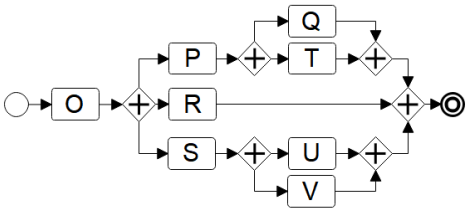
(c)

Gambar. 6. Model proses bisnis dengan variasi posisi bercabang.



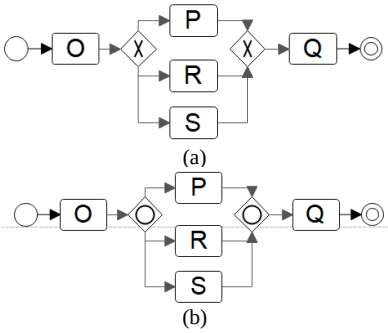
(a)

(b)



(c)

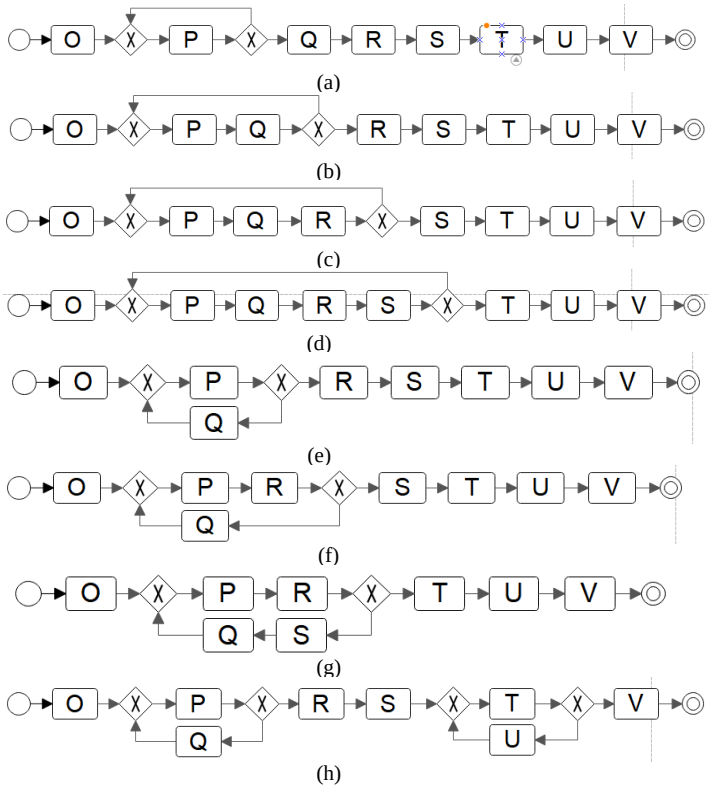
Gambar. 7. Model proses bisnis dengan variasi kedalaman percabangan.



(a)

(b)

Gambar. 8. Model proses bisnis dengan variasi jenis percabangan.



(a)

(b)

(c)

(d)

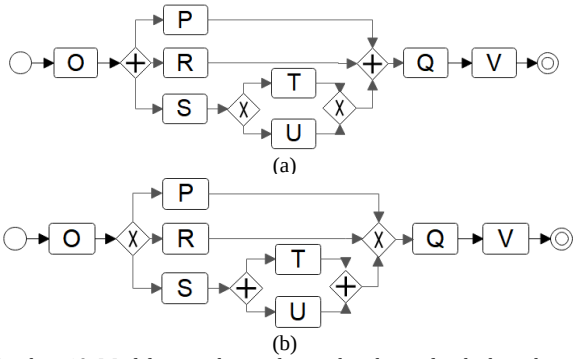
(e)

(f)

(g)

(h)

Gambar. 9. Model proses bisnis dengan variasi perulangan.



(a)

(b)

Gambar. 10. Model proses bisnis dengan kombinasi logika bercabang.

DOI: https://doi.org/10.25047/jtit.v7i2.165

Nilai kesamaan pada *similarity* struktur diukur dengan membandingkan aspek yang serupa pada model proses BPMN, seperti *label task activity*, *connector*, dan *gateway* (percabangan) yang sama. Model proses secara struktural dihitung similaritynya menggunakan rumus *Jaccard Coefficient Similarity* pada (1).

$$Jacc\ sim = \frac{x \cap y}{x \cup y} \qquad (1)$$

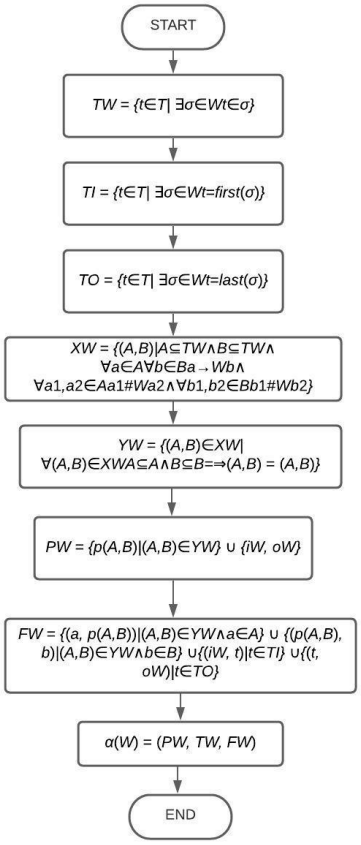
Berdasarkan dataset pada figure 2 sampai 10, didapatkan *event log* yang ditunjukan pada tabel. 1 berikut.

TABEL 1. EVENT LOG

No	Figure	*Event Log*
1	2a	O
2	2b	OP
3	2c	OPQ
4	3a	OPQR, ORPQ
5	3b	OPRSQ, ORPSQ, ORSPQ, OSPRQ, OSRPQ, OPSRQ
6	4a	OPRQST, ORPSQT, ORSPQT, OSPRQT, OSRPQT, OPSRQT
7	4b	OPRQSTU, ORPSQTU, ORSPQTU, OSPRQTU, OSRPQTU, OPSRQTU
8	4c	OPRQSTUV, ORPSQTUV, ORSPQTUV, OSPRQTUV, OSRPQTUV, OPSRQTUV
9	5a	OPQRSTUV, OPQSRTUV, OPRQSTUV, OPRSQTUV, OPSQRTUV, OPSRQTUV, OQPRSTUV, OQPSRTUV, OQRPSTUV, OQRSPTUV, OQSPRTUV, OQSRPTUV, ORPQSTU, ORPSQTUV, ORQPSTUV, ORQSPTUV, ORSPQTUV, ORSQPTUV, OSPQRTUV, OSPRQTUV, OSQPRTUV, OSQRPTUV, OSRPQTUV, OSRQPTUV
10	5b	OPQRST, OPQRTS, OPQSRT, OPQTRS, OPQTSR, OPRQST, OPRQTS, OPRSQT, OPRSTQ, OPRTQS, OPRTSQ, OPSQRT, OPSQTR, OPSRQT, OPSRTQ, OPSRTQ, OPSTQR, OPSTRQ, OPTQRS, OPTQSR, OPTRQS, OPTRSQ, OPTSQR, OPTSRQ, OQPRST, OQPRTS, OQPSRT, OQPSTR, OQPTRS, OQPTSR, OQRPST, OQRPTS, OQRSPT, OQRSTP, OQRTPS, OQRTSP, OQSPRT, OQSPTR, OQSRPT, OQSRTP, OQSTPR, OQSTRP, OQTPRS, OQTPSR, OQTRPS, OQTRSP, OQTSPR, OQTSRP, ORPRST, ORPQTS, ORPSQT, ORPSTQ, ORPTQS, ORPTSQ, ORQPST, ORQPTS, ORQSTP, ORQSPT, ORQTSP, ORQTPS, ORSPQT, ORSPTQ, ORSQPT, ORSQTP, ORSTPQ, ORSTQP, ORTPQS, ORTPSQ, ORTQPS, ORTQSP, ORTSPQ, ORTSQP, OSPQRT, OSPQTR, OSPRQT, OSPRTQ, OSPTQR, OSPTRQ, OSQPRT, OSQPTR, OSQRPT, OSQRTP, OSQTRP, OSQTPR, OSRPQT, OSRPTQ, OSRQPT, OSRQTP, OSRTPQ, OSRTQP, OSTPQR, OSTPRQ, OSTQPR, OSTQRP, OSTRPQ, OSTRQP, OTPQRS, OTPQSR, OTPRQS, OTPRSQ, OTPSQR, OTPSRQ, OTQPRS, OTQPSR, OTQRPS, OTQRSP, OTQSPR, OTQSRP, OTRPQS, OTRPSQ, OTRQPS, OTRQSP, OTRSPQ, OTRSQP, OTSPQR, OTSPRQ, OTSQPR, OTSQRP, OTSRPQ, OTSRQP.
11	6a	OQPRSTUV, OQRPSTUV, OQSRPTUV, OQRSPTUV, OQSPRTUV, OQPSRTUV.
12	6b	OQTPRSUV, OQTRPSUV, OQTSRPUV, OQTRSPUV, OQTSPRUV, OQTPSRUV.
13	6c	OQTUPRSV, OQTURPSV, OQTUSRPV, OQTURSPV, OQTUSPRV, OQTUPSRV.
14	7a	OPQSRUVT, OQPSRUVT, OQSPRUVT, OSPQRUVT, OSQPRUVT, OPSQRUVT, OPQSRVUT, OQPSRVUT, OQSPRVUT, OSPQRVUT, OSQPRVUT, OPSQRVUT
15	7b	OTQSUVRT, OQPSUVRT, OQSUVPRT, OSUVPQRT, OSUVQVRT, OPSUVQRT, OPQSVURT, OQPSVURT, OQSVUPRT, OSVUPQRT,OSVVUQPRT, OPSVUVQRT.
16	7c	OPRTQSUV, OPTRQSUV, OPRTQSVU, OPRQSVU, OQPRTSUV, OQPTRSUV, OQPRTSVU, OQPTRSVU, OQSUVPRT, OQSVUPRT, OQSUVPTR, OQSVUPTR, OSUVPRTQ, OSVUPRTQ, OSUVPTRQ, OSVUPTRQ, OSUVQPRT, OSVUQPRT, OSUVQPTR, OSVUQPTR, OPRTSUVQ, OPTRSUVQ, OPRTSVUQ, OPTRSVUQ.
17	8a	OPQ, ORQ, OSQ.
18	8b	OPQ, ORQ, OSQ, OPRSQ, OPSRQ, ORPSQ, ORSPQ, OSRPQ, OSPRQ, OPRQ, OPSQ, ORPQ, ORSQ, OSPQ, OSRQ.
19	9a	OPPPQRSTUV
20	9b	OPQPQRSTUV
21	9c	OPQRPQRSTUV, OPQRSTUV.
22	9d	OPQRSPQRSTUV, OPQRSTUV.
23	9e	OPQPRSTUV.
24	9f	OQRPQRSTUV.
25	9g	OQRSPQRTUV.
26	9h	OQPQRSUVUT.
27	10a	OPQSURT, OPQSVRT, OPSQRT, OQPSURT, OQPSVRT, OQSPRT, OSUQPRT, OSVQPRT, OSUPQRT, OSVPQRT.
28	10b	OPRS, OQRS, OTUVRS, OTVURS.

Dengan menggunakan *event log* pada tabel 1, dilakukan proses *mining* dengan menggunakan tiga algoritma yaitu algoritma *alpha miner*, *inductive miner* dan *causal-net mining*. Penerapan algoritma alpha miner didefinisikan pada definisi 1 [10].

a. Algoritma *Alpha Miner*

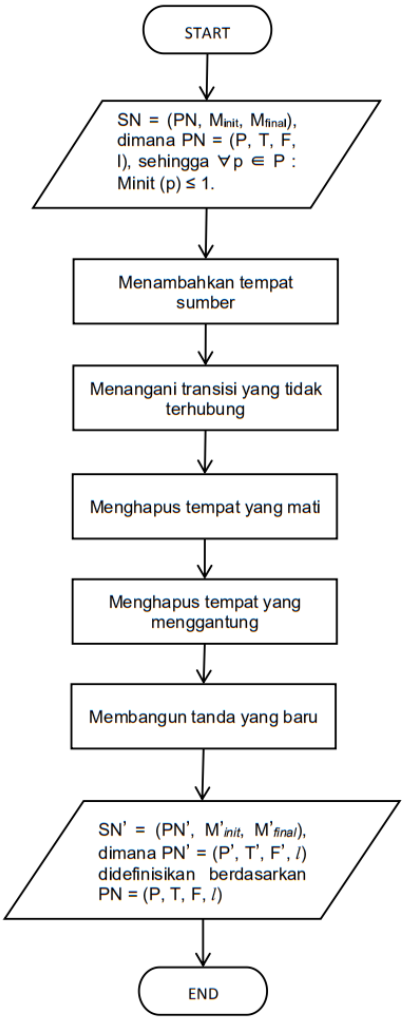


Gambar. 11. Algoritma *Alpha Miner*

Algoritma *inductive miner* pada Gambar 11 menjelaskan tahapan-tahapan sebagai berikut [10].

1. Membuat sekumpulan transisi dari *event log* pada *Workflow net* (T_L)
2. Membuat sekumpulan transisi output dari source place pada *Workflow net* (T_I)
3. Membuat sekumpulan transisi input dari *sink place* pada *Workflow net* (T_O)
4. Pada langkah ke-4 dan ke-5, algoritma *alpha miner* membuat set (XW dan YW, masing-masing) yang digunakan untuk menentukan tempat jaring alur kerja yang ditemukan. Pada Langkah 4, algoritma *alpha miner* menemukan transisi mana yang terkait secara kausal. Jadi, untuk setiap tupel (A, B) di XW, setiap transisi dalam himpunan A secara kausal berhubungan dengan semua transisi di himpunan B, dan tidak ada transisi dalam A (atau B) yang mengikuti satu sama lain dalam beberapa urutan pengaktifan. Batasan ke elemen dalam himpunan A dan B ini memungkinkan penambangan yang benar dari konstruksi AND-split / join dan OR-split / join. Perhatikan bahwa OR-split / join membutuhkan fusi tempat.
5. Pada Langkah 5, algoritma *alpha miner* menyempurnakan set XW dengan hanya mengambil elemen terbesar sehubungan dengan set inklusi. Faktanya, Langkah 5 menetapkan jumlah yang tepat dari tempat yang dimiliki jaring yang ditemukan (tidak termasuk tempat sumber iW dan tempat pembuangan).
6. *Place* yang sebelumnya diidentifikasi dibuat.
7. Masing-masing *place* dihubungkan dengan transisi input/output yang bersesuaian.
8. Algoritma kemudian mengembalikan *Workflow net* yang diperoleh dari langkah-langkah sebelumnya.

b. Algoritma *Inductive Miner*



Gambar. 12. Algoritma Inductive Miner

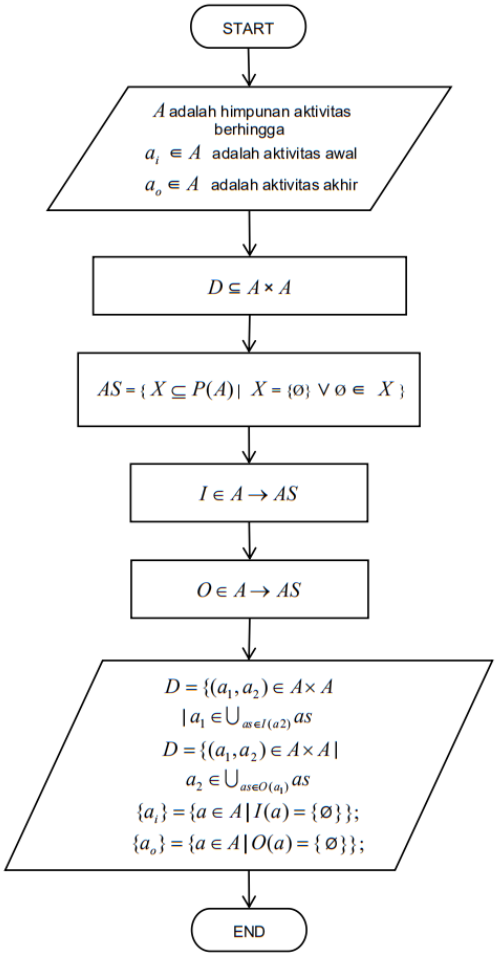
Algoritma *inductive miner* pada Gambar 12 menjelaskan tahapan-tahapan sebagai berikut [11].

1. Langkah 0: Menambahkan tempat sumber. Tambahkan tempat baru i ∈ P, transisi awal baru t ∗ (perhatikan bahwa t ∗ tidak memiliki label, karena t ∗ / ∈ dom (l)) dan hubungkan dengan busur (i, t ∗). Untuk setiap tempat p ∈ P, sehingga Minit (p) = 1, tambahkan busur (t ∗, p).
2. Langkah 1: Menangani transisi yang tidak terhubung. Untuk setiap transisi t ∈ T, sehingga • t = ∅, tambahkan tempat p, dihubungkan dengan t dengan busur masuk dan keluar. Tambahkan busur dari transisi awal t ∗ ke tempat p.
3. Langkah 2: Menghapus tempat mati. Hapus setiap tempat p ∈ P dan transisi dari p • bersama dengan busur datang, jika tidak ada jalur dari i ke p. Ulangi Langkah 2 sampai tidak ada lagi tempat mati.
4. Langkah 3: Menghapus tempat gantung. Hapus semua tempat p ∈ P, sehingga | p • | = 0, bersama dengan busur insiden.

DOI: https://doi.org/10.25047/jtit.v7i2.165

5. Langkah 4: Membangun tanda baru. Misalkan P adalah himpunan tempat yang dihasilkan, dan P ∗ ⊆ P adalah himpunan tempat yang ditambahkan pada Langkah 1. Kemudian tanda awal dan akhir M init dan M akhir didefinisikan sebagai berikut: Untuk semua p ∈ P, sehingga p = i, M init (p) = 0, M init (i) = 1, untuk semua p ∈ P ∗ menyatakan bahwa M f inal (p) = 1, dan untuk semua p ∈ (P ∩ P) jumlah token dipertahankan, yaitu, M final (p) = Mfinal (p). Tempat sumber tidak berisi token apa pun dalam penandaan akhir, yaitu M f inal (i) = 0.

c. Algoritma *Causal-net Mining*



Gambar. 13. Algoritma *Causal-net mining*

Algoritma *causal-net mining* pada figure 13 menjelaskan tahapan-tahapan sebagai berikut [5].

A adalah himpunan aktivitas berhingga.

$a_i \in A$ adalah aktivitas awal.

$a_o \in A$ adalah aktivitas akhir.

$D \subseteq A \times A$ adalah relasi ketergantungan.

$AS = \{X \subseteq P(A) \mid X = \{\emptyset\} \vee \emptyset \in X\};$

$I \in A \rightarrow AS$ mendefinisikan himpunan kemungkinan ikatan masukan per aktivitas

$O \in A \rightarrow AS$ mendefinisikan himpunan ikatan keluaran yang mungkin per aktivitas.

seperti,

$$D = \{(a_1, a_2) \in A \times A \mid a_1 \in \bigcup_{as \in I(a2)} as$$

$$D = \{(a_1, a_2) \in A \times A \mid a_2 \in \bigcup_{as \in O(a_1)} as$$

$$\{a_i\} = \{a \in A \mid I(a) = \{\emptyset\}\};$$

$$\{a_o\} = \{a \in A \mid O(a) = \{\emptyset\}\};$$

HASIL DAN PEMBAHASAN

Proses *discovery* yang telah dilakukan dengan menggunakan tools ProM. Pada tabel. 2 ditampilkan hasil eksperimen menggunakan ProM dan didapati bahwa terdapat 4 skenario yang tidak cocok dengan model proses pada dataset [7]. Pada algoritma *inductive miner* dan *causal-net mining* didapati bahwa hasil *discovery* berupa BPMN. Sedangkan, pada algoritma *alpha miner* didapati hasil *discovery* berupa *petri net*. Oleh karena itu, *petri net* dikonversi ke dalam bentuk BPMN menggunakan format file .pnml.

Kami menggunakan figure 7(b) dan dibandingkan dengan hasil *discovery* figure 7(b) menggunakan algoritma inductive miner untuk diukur similarity strukturalnya. Notasi BPMN terdiri dari *activity, connector* dan *gateaway* sehingga dari perbandingan model tersebut didapatkan elemen irisan sebagai berikut:

Label task activity = 8
Connector = 11
Gateaway = 2

sedangkan, gabungan dari kedua model, yaitu:

Label task activity = 12
Connector = 19
Gateaway = 4

Angka yang didapat, dimasukkan ke dalam rumus perhitungan similarity pada (1)

$$Jacc\ sim = \frac{8+11+2}{12+19+4} = \frac{21}{35} = 0.6$$

Dengan menggunakan cara yang sama, didapatkan hasil perhitungan *similarity* secara keseluruhan yang ditunjukan pada tabel. 2 dan tabel. 3 berikut.

TABEL 2. HASIL SIMILARITY KESELURUHAN

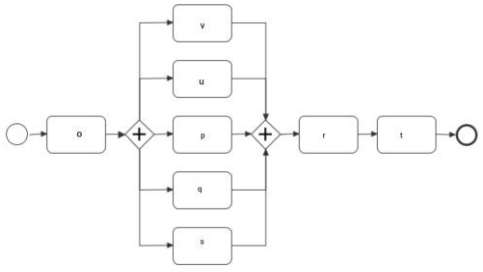
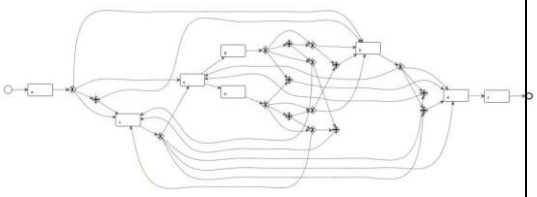
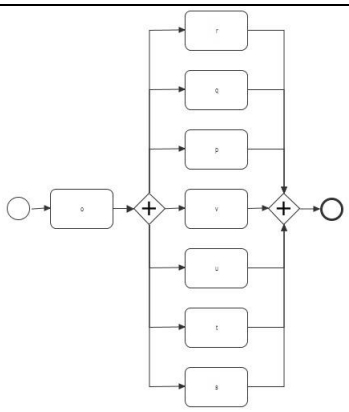
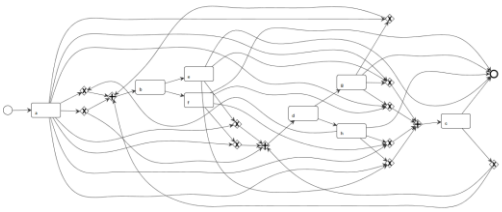
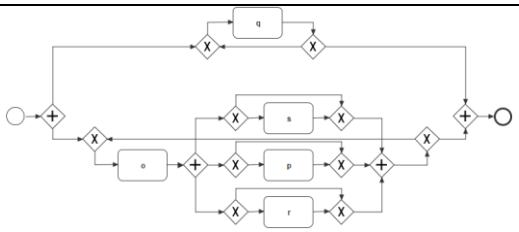
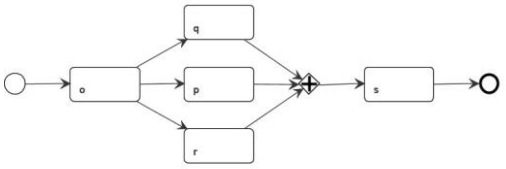
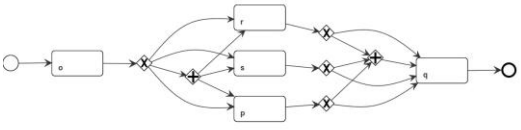
Dataset	*Similarity*		
	Inductive Miner	*Alpha Miner*	*Causal-net Mining*
Fig 2a	1	1	1
Fig 2b	1	1	1
Fig 2c	1	1	1
Fig 3a	1	1	1
Fig 3b	1	1	1
Fig 4a	1	1	1
Fig 4b	1	1	1
Fig 4c	1	1	1
Fig 5a	1	1	1
Fig 5b	1	1	1
Fig 6a	1	1	1
Fig 6b	1	1	1

Fig 6c	1	1	1
Fig 7a	1	1	1
Fig 7b	0.6	0.18	0.26
Fig 7c	0.33	0.12	1
Fig 8a	1	1	1
Fig 8b	0.37	0.89	0.22
Fig 9a	1	Error	1
Fig 9b	1	Error	1
Fig 9c	0.43	1	0.43
Fig 9d	0.35	1	0.34
Fig 9e	1	1	1
Fig 9f	1	1	1
Fig 9g	1	1	1
Fig 9h	1	1	1
Fig 10a	0.62	0.75	0.29
Fig 10b	1	1	1

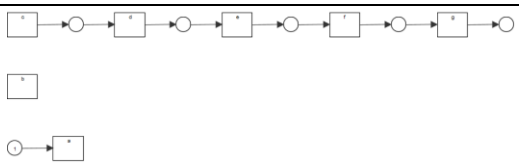
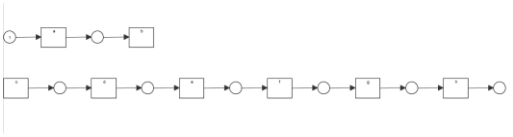
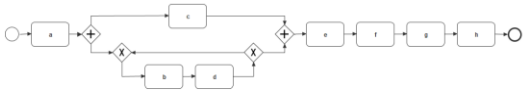
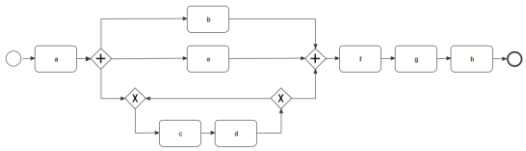
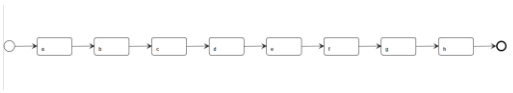
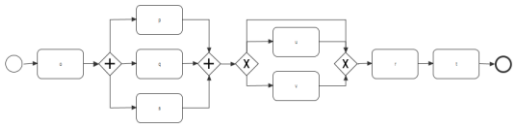
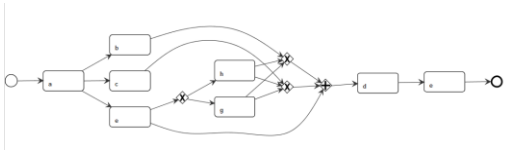
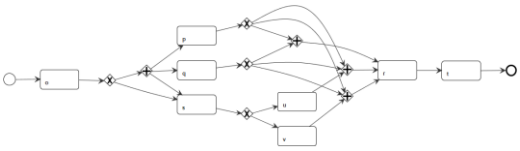
Pada tabel 2 diketahui bahwa terdapat perbedaan similarity pada setiap figure. Usai menyelesaikan perhitungan *similarity*, didapatkan bahwa ketika nilai *similarity* antara dua model proses mendekati atau sama dengan 1, maka semakin mirip pula kedua model proses tersebut. Sebaliknya, ketika nilai *similarity* antara dua model proses dibawah 1, maka semakin rendah tingkat kemiripan kedua model proses.

Figure dengan nilai similarity 1 merupakan model proses dari hasil *discovery* yang cocok dengan model prose pada dataset. Sedangkan, model proses dengan nilai *similarity* dibawah 1 merupakan model proses dari hasil *discovery* yang tidak cocok dengan dataset [7]. Terdapat 17 model proses dengan nilai *similarity* dibawah 1. Model proses dengan nilai *similarity* dibawah 1 dijabarkan pada tabel. 3 berikut.

TABEL 3. HASIL SIMILARITY YANG TIDAK COCOK

Data Set	Keterangan	Sim
Gbr 7b	 *Inductive Miner*	0.6
		0.18
	Alpha Miner	
	 Causal Net Mining	0.26
Gbr 7c	 Inductive Miner	0.33
	 Alpha Miner	0.12
Gbr 8b	 Inductive Miner	0.37
	 Alpha Miner	0.89
	 Causal Net Mining	0.22
Gbr 9a		Error

DOI: https://doi.org/10.25047/jtit.v7i2.165 ©2020 J-TIT

	 Alpha Miner	
Gbr 9b	 Alpha Miner	Error
Gbr 9c	 Inductive Miner	0.43
	 Causal Net Mining	0.43
Gbr 9d	 Inductive Miner	0.35
	 Causal Net Mining	0.34
Gbr 10a	 Inductive Miner	0.62
	 Alpha Miner	0.75
	 Causal Net Mining	0.29

Berdasarkan hasil penelitian pada tabel. 3 setelah kami lakukan perbandingan, diketahui bahwa setiap model proses hasil *discovery* memiliki nilai *similarity* yang beragam. Nilai *similarity* yang didapatkan dipengaruhi oleh 6 komponen, antara lain, jumlah anak panah yang sama, jumlah aktivitas yang sama, jumlah *gateaway* yang sama, jumlah anak panah kedua model proses setelah digabungkan, jumlah aktivitas kedua model proses setelah digabungkan, dan terakhir jumlah *gateaway* kedua model proses setelah digabungkan. Pada algoritma *alpha miner* menghasilkan model roses dalam bentuk *petri net*, oleh karena itu kami mengubahnya menjadi bentuk BPMN dengan format .pnml menggunakan software Wopad.

Berdasarkan hasil pengukuran *similarity* diketahui bahwa eksperimen menggunakan algortima *alpha miner*. Pada figure 8 memiliki nilai tingkat *similarity* paling tinggi yaitu dengan nilai 0.89. Sedangkan tingkat *similarity* paling kecil, didapati pada figure 7 menggunakan *alpha miner* dengan nilai 0.12. Selain itu, ada pula figure 9a dan 9b dengan hasil *similarity* yang error dikarenakan saat melakukan *discovery* mengunakan algoritma alpha miner, model proses yang dihasilkan terpecah menjadi 2 bagian, sehingga tidak bisa dideteksi nilai *similarity* nya.

KESIMPULAN

Berdasarkan pembahasan yang telah diuraikan sebelumnya, kami menyimpulkan bahwa pada penelitian ini kami menemukan bahwa, nilai tingkat similarity tertinggi didapati pada figure 8 menggunakan *alpha miner* dengan nilai 0.89. Nilai tingkat similarity yang rendah didapati pada figure 7 menggunakan algoritma *alpha miner* dengan nilai 0.12. Gambar 2 sampai 6 cocok dengan semua algoritma karena model proses hasil proses mining menggunakan semua algoritma sama persis dengan model proses pada dataset. Gambar 7 paling cocok menggunakan algoritma *inductive miner* dengan tingkat *similarity* 0.6. Gambar 8 paling cocok menggunakan algoritma *aplha miner* dengan tingkat similarity 0.89. Gambar 9 paling cocok menggunakan algoritma *inductive miner* dengan tingkat similarity 0.35. Gambar 10 paling cocok dengan algoritma *alpha miner* dengan tingkat similarity 0.75.

REFERENSI

[1] W. N. Rumana, "Pemodelan Dan Simulasi Proses Bisnis Coloured Petri Nets Modeling and Simulation Production Business Process of Steel Industry Through," pp. 1–93, 2015.

[2] A. A. Hermawan, "Business Process Context Analysis Based on 'Event Log,'" J. Penelit. dan Pengemb. Komun. dan Inform., vol. 4, no. 3, p. 122699, 2014.

[3] P. Porouhan, N. Jongsawat, and W. Premchaiswadi, "Process and deviation exploration through Alpha-algorithm and Heuristic miner techniques," Int. Conf. ICT Knowl. Eng., pp. 83–89, 2014, doi: 10.1109/ICTKE.2014.7001540.

[4] I. Nuritha and E. R. Mahendrawathi, "Structural Similarity Measurement of Business Process Model to Compare Heuristic and Inductive Miner Algorithms Performance in Dealing with Noise," Procedia Comput. Sci., vol. 124, pp. 255–263, 2017, doi: 10.1016/j.procs.2017.12.154.

[5] W. van der Aalst, A. Adriansyah, and B. van Dongen, "Causal Nets: A Modeling Language Tailored towards Process Discovery," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 6901 LNCS, pp. 28–42, 2011, doi: 10.1007/978-3-642-23217-6.

[6] M. A. Yaqin, R. Sarno, and S. Rochimah, "Measuring Scalable Business Process Model Complexity Based on Basic Control Structure," Int. J. Intell. Eng. Syst., vol. 13, no. 6, pp. 52–65, 2020, doi: 10.22266/ijies2020.1231.06.

[7] R. Accorsi, T. Stocker, and G. Müller, "On the exploitation of process mining for security audits: The process discovery case," Proc. ACM Symp. Appl. Comput., pp. 1462–1468, 2013, doi: 10.1145/2480362.2480634.

[8] W. M. P. van der Aalst, "Business alignment: Using process mining as a tool for Delta analysis and conformance testing," Requir. Eng., vol. 10, no. 3, pp. 198–211, 2005, doi: 10.1007/s00766-005-0001-x.

[9] J. Caldeira and F. B. e Abreu, “Software Development Process Mining:,” 10th Int. Conf. Qual. Inf. Commun. Technol. Softw., pp. 254–259, 2016, doi: 10.1109/QUATIC.2016.51.

[10] A. K. A. de Medeiros, B. F. van Dongen, W. M. P. van der Aalst, and A. J. M. M. Weijters, “Process mining: Extending the a algorithm to mine short loops,” BETA Work. Pap. Ser., no. June, 2004.

[11] A. A. Kalenkova, W. M. P. van der Aalst, I. A. Lomazova, and V. A. Rubin, “Process mining using BPMN: relating event logs and process models,” Softw. Syst. Model., vol. 16, no. 4, pp. 1019–1048, 2017, doi: 10.1007/s10270-015-0502-0.

DOI: https://doi.org/10.25047/jtit.v7i2.165