

# Process Mining Dwelling Time pada Pelabuhan Peti Kemas menggunakan Metode Alpha Miner

## *Process Mining Dwelling Time at the Container Port using the Alpha Miner Method*

Rivalno Firmansyah, Herdy Riyaldy, Bambang Jokonowo\*

<sup>1</sup>Sistem Informasi, Fakultas Ilmu Komputer, Universitas Mercu Buana, Jakarta  
Jl. Raya Meruya Selatan, Jakarta, 11650

\*e-mail: [bambang.jokonowo@mercubuana.ac.id](mailto:bambang.jokonowo@mercubuana.ac.id)

(received: 14 Juli 2021, revised: 21 November 2021, accepted: 30 Juli 2022)

### Abstrak

Permasalahan pengukuran *dwelling time* proses logistik peti kemas di pelabuhan pada negara berkembang seringkali menjadi masalah utama. Oleh karena itu, process mining sebagai bidang ilmu data yang berfokus pada analisis data *event log* digunakan untuk melakukan analisis proses bisnis. Dalam process mining, proses merupakan urutan kejadian atau aktivitas yang dijalankan untuk mencapai tujuan tertentu. *Event log* membantu sebuah organisasi untuk menemukan kesenjangan antara proses bisnis yang dirancang dengan realita proses yang terjadi. Dalam penelitian ini PM4PY sebagai library python digunakan untuk melakukan teknik process mining. Hasil perhitungan fitness pada penelitian ini menunjukkan bahwa model proses bisnis logistik peti kemas mendekati nilai 0.99.

**Kata kunci:** *Event log*, Process Mining, PM4PY.

### Abstract

*The problem of measuring the dwelling time of the container logistics process at ports in developing countries is often a major problem. Therefore, process mining as a field of data science that focuses on analyzing event log data is used to perform business process analysis. In process mining, the process is a sequence of events or activities that are carried out to achieve certain goals. Event logs help an organization to find gaps between the designed business processes and the reality of the processes that occur. In this study PM4PY as a python library is used to perform process mining techniques. The results of the fitness calculation in this study indicate that the container logistics business process model is close to 0.99.*

**Keywords:** *Process Mining, Event log, PM4PY.*

## 1 Pendahuluan

Indonesia merupakan salah satu negara berkembang sekaligus negara kepulauan terbesar di dunia dan memiliki letak yang strategis, berada di antara Samudera Hindia dan Samudera Pasifik menjadikan sektor maritim Indonesia sangat penting. Banyaknya volume kargo peti kemas Indonesia meningkat rata-rata 8,4% setiap tahunnya serta *throughput* yang tumbuh sekitar 7,7% per tahun [1]. Peningkatan kinerja pada pelabuhan Indonesia akan memberikan nilai tambah untuk mempertahankan posisi kompetitif dalam kompetisi pelabuhan internasional [2].

Salah satu alat ukur daya saing pelabuhan *Dwelling Time* yaitu waktu yang dihitung mulai dari satu peti kemas dibongkar dari kapal hingga peti kemas tersebut meninggalkan terminal pelabuhan melalui pintu utama [3]. *Dwelling Time* hanya bagian kecil dari *total cycle* yang harus dikelola oleh pelabuhan, namun *dwell time* berpengaruh besar terhadap ekonomi secara luas [4]. Dalam beberapa tahun terakhir, masalah pengukuran *Dwelling Time* proses logistik peti kemas di pelabuhan seringkali menjadi masalah utama. Polemik permasalahan *dwell time* yang tinggi di sejumlah pelabuhan bukan hal yang baru di Indonesia karena sudah menjadi sorotan para pelaku usaha dan juga pemerintah. Pemerintah telah menetapkan paling lambat waktu *dwell time* di pelabuhan selama 4 hari. Fakta yang

<http://sistemasi.ftik.unisi.ac.id>

terjadi, rata-rata dwell time di pelabuhan Indonesia pada tahun 2013 masih sekitar 10 sampai dengan 15 hari [5].

Proses bisnis pada dwelling time petikemas dapat dianalisis untuk mendapatkan kedalaman wawasan tentang realita aktivitas proses yang dicatat dalam sistem informasi. Proses ini dapat dimodelkan berdasarkan event log yang tersimpan didalam sistem informasi. Model proses kemudian bisa dianalisis dari berbagai perspektif, yaitu aliran aktivitas, jaringan sosial, peran dan kelompok, kinerja, audit dan keamanan, serta pengambilan keputusan. Pembuatan model proses bisnis dengan melakukan analisis berdasarkan event log kemudian disebut dengan process mining [6]. Process mining adalah disiplin ilmu yang relatif baru yaitu antara *computational intelligence* dan *data mining*, ide dasar process mining adalah discovery (penemuan), conformance (kesesuaian) dan enhancement (peningkatan) [7].

Dalam melakukan analisis process mining terdapat beberapa tools, baik itu open source maupun komersial. Pada penelitian ini, kami menggunakan solusi process mining menggunakan tools PM4PY. PM4PY merupakan salah satu tools open source yang dibangun menggunakan bahasa pemrograman python. PM4PY menyediakan serangkaian fitur penambangan proses sebagai library dan memberikan integrasi dengan ekosistem python yang sesuai [8]. Penelitian ini juga akan membandingkan proses bisnis logistik peti kemas berdasarkan parameter yang ada untuk mencari nilai fitness dan *structure* pada *event log*.

## 2 Tinjauan Literatur

Van der Aalst [9], dalam bukunya menyimpulkan bahwa process mining dapat dilihat sebagai mata rantai yang hilang antara data mining dengan BPM berbasis model tradisional. Meskipun teknik dan *tools* process mining matang tersedia, beberapa tantangan tetap ada untuk lebih meningkatkan penerapan teknik yang disajikan. Seperti disebutkan dalam penelitian [10], tidak semua algoritma process mining dapat mendeteksi semua skenario model proses. Sehingga perlu dilakukan beberapa eksperimen dengan mencoba beberapa jenis algoritma process mining terhadap beberapa skenario model proses bisnis yang bertujuan untuk mendapatkan algoritma yang paling cocok untuk proses *discovery*. Titik awal untuk process mining adalah event log, setiap aktivitas didalam log semacam itu mengacu pada suatu aktivitas langkah yang terdefinisi dengan baik dalam beberapa proses dan terkait dengan kasus tertentu [11]. Event log yang didapatkan dari sistem informasi tidak serta merta dapat langsung dilakukan tahapan analisa. Bambang Jokonowo [12], dalam penelitiannya menyatakan bahwa ekstraksi data dari sistem informasi mutlak diperlukan untuk kebutuhan praktis proses mining. Struktur model yang dihasilkan dari log ditentukan berdasarkan hasil dari discover data kejadian atau event log. Event log digunakan untuk mengukur kinerja pada proses dwell time impor peti kemas. Ekstraksi digunakan untuk menghindari kesalahan analisis dan verifikasi. Mulai dari proses pemilihan contoh data, mengidentifikasi aktivitas dan pemilihan atribut digabungkan sesuai dengan aktivitasnya masing-masing. Riset terdahulu tentang pengecekan conformance modul keuangan dengan process mining, menyimpulkan bahwa hasil penelitian dengan process mining mendapatkan perbedaan antara model sistem informasi dengan SOP. Dimana teknik conformance berguna untuk melakukan pengecekan berdasarkan model proses dengan SOP yang ada. Conformance checking akan menghasilkan sebuah nilai fitness yang sesuai apabila nilai memiliki threshold 1 [13].

Dwelling time merupakan waktu tunggu yang dihitung mulai dari suatu peti kemas dibongkar dan diangkat dari kapal hingga keluar meninggalkan terminal pelabuhan [3],[4],[5]. Bambang Jokonowo [6], dalam penelitiannya menyatakan bahwa kinerja proses dwell time kontainer impor pelabuhan dapat diukur dengan menerapkan waktu tunggu dari semua kasus (*cases process*). Petikemas impor pelabuhan selama 3 bulan semua kasus media berdurasi 5,5 hari dan durasi rata-rata 6,07 hari. Selain itu, petikemas import pelabuhan dibandingkan antara laporan perusahaan dan hasil pengukuran proses mining, dari hasil verifikasi menunjukkan kinerja yang sama.

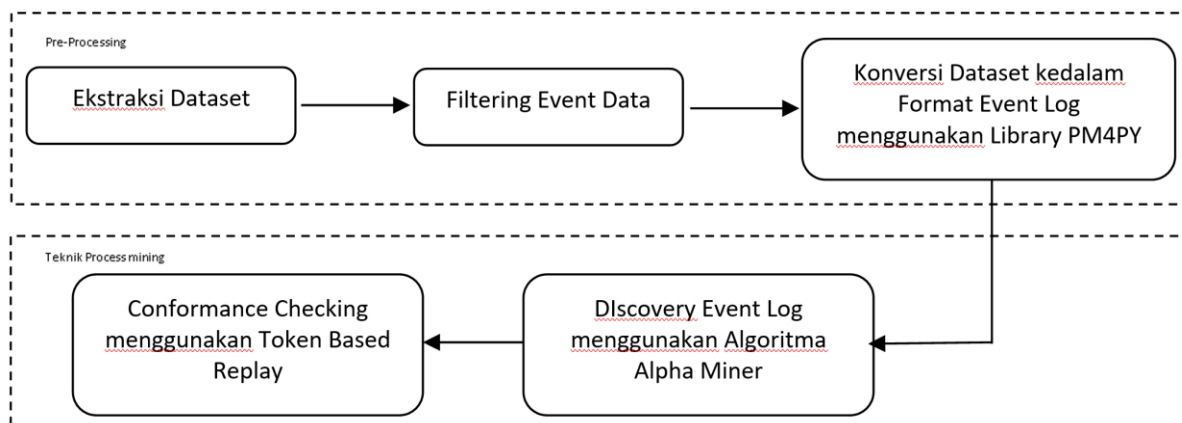
Dalam penelitian ini dilakukan 2 tahapan utama yaitu pre-processing dan melakukan Teknik proses mining yaitu discovery dan conformance checking. Pada tahapan discovery akan menggunakan algoritma alpha miner, adalah algoritma pertama yang digunakan dalam proses mining. Algoritma alpha berfokus pada proses alur kerja dan ditampilkan dalam bentuk jaringan alur kerja [14]. Setelah model proses didapatkan, maka akan dilanjutkan dengan tahapan *conformance checking* model dengan event log. Conformance checking menghubungkan peristiwa dalam event log dan model proses serta

membandingkan keduanya. Bertujuan menemukan kesamaan dan perbedaan antara model proses yang ditentukan dengan model proses yang sedang terjadi [15]. Model proses dan event log dikatakan cocok jika model proses dapat memainkan ulang (*replay*) setiap trace/sequence [16].

PM4PY mendukung serangkaian teknik penambangan proses (*discovery*, *conformance checking* dan *enhancement*), selain itu repository Github mendukung ekosistem kolaboratif tempat pengguna dapat menandakan masalah atau berkontribusi pada code [17].

### 3 Metode Penelitian

Tahapan penelitian ini, diuraikan menjadi 2 (dua) kelompok tahapan utama yaitu pertama melakukan *Pre-processing* dan kelompok tahapan ke 2 (dua) melakukan Teknik Process mining yaitu *discovery* dan *conformance checking*. Adapun tahapan penelitian tersebut dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Penelitian

#### 3.1 Pre-processing

Pada tahap Pre-processing, ekstraksi dataset digunakan untuk mengumpulkan data event (data kejadian) dalam membentuk struktur event log kami mengikuti penelitian sebelumnya [6]. Tahapan filtering data digunakan untuk menentukan start activities dan end activities, juga untuk melakukan pengelompokan data berdasarkan variant dan case. Kemudian dataset dikonversi kedalam format log yang digunakan untuk melakukan proses discovery menggunakan algoritma alpha miner.

Semua tahapan pre-processing data event log tersebut kami olah menggunakan *google collab*, *Library pandas* dan *numpy* dan *Library PM4PY*.

#### 3.2 Teknik Process Mining

##### Discovery Event Log

Kami melakukan discovery menggunakan Alpha Miner yang merupakan salah satu algoritma yang digunakan dalam penambangan proses. Cara kerja dari Algoritma ini, melakukan rekonstruksi kausalitas dari serangkaian event log. Algoritma ini didefinisikan dalam istilah *Petri Nets*, pertama kali dikemukakan oleh “van der Aalst”, seorang profesor di Departemen Matematika dan Ilmu Komputer dari Technische Universiteit Eindhoven [10].

1. Membuat sekumpulan Transisi di Workflow net ( $T_L$ )  
$$T_L = \{t \in T \mid \exists \sigma \in L \ t \in \sigma\}$$
2. Membuat Transisi Source Input Place Pada Workflow net ( $T_I$ )  
$$T_I = \{t \in T \mid \exists \sigma \in L \ t = first(\sigma)\}$$
3. Membuat Transisi Output Sink Place Pada Workflow net ( $T_O$ )  
$$T_O = \{t \in T \mid \exists \sigma \in L \ t = last(\sigma)\}$$
4. Algoritma  $\alpha$  Menentukan Transisi Mana Yang Berelasi Secara Casual

- $$X_L = \{(A, B) \mid A \subseteq T_L \wedge A \neq \emptyset \wedge B \subseteq T_L \wedge B \neq \emptyset \wedge \forall a \in A \forall b \in B \ a \rightarrow L^b \wedge \forall a_1, a_2 \in A \ a_1 \# L^{a_2} \wedge \forall b_1, b_2 \in B \ b_1 \# L^{b_2}\}$$
5. Menentukan Jumlah Pasti Places yang Dimiliki Oleh Workflow net  

$$Y_L = \{(A, B) \in X_L \mid \forall (A', B') \in X_L \ A \subseteq A' \wedge B \subseteq B' \Rightarrow (A, B) = (A', B')\}$$
  6. Membuat Place yang sudah Terindikasi Sebelumnya  

$$P_L = \{P_{(A,B)} \mid (A, B) \in Y_L\} \cup \{i_L, o_L\}$$
  7. Menghubungkan place yang ada dengan input dan output yang tersedia  

$$F_L = \{(a, P_{(A,B)}) \mid (A, B) \in Y_L \wedge a \in A\} \cup \{(P_{(A,B)}, b) \mid (A, B) \in Y_L \wedge b \in B\} \cup \{(i_L, t) \mid t \in T_I\} \cup \{(t, o_L) \mid t \in T_O\}$$
  8. Penggambaran ulang workflow  

$$\alpha(L) = (P_L, T_L, F_L)$$

Library Algoritma alpha miner. algoritma tersebut disediakan dalam tools PM4PY, yang merupakan salah satu tools open source yang dibangun menggunakan bahasa pemrograman python [8].

### Conformance Checking

Pada penelitian ini dilakukan Conformance Checking menggunakan Token Based Replay untuk mendapatkan nilai fitness dengan formula sebagai berikut :

Berdasarkan rumus (1) kita melakukan perhitungan fitness untuk setiap varian trace  $\sigma$  pada model proses  $N$ .

$$fitness(\sigma, N) = \frac{1}{2} \left(1 - \frac{m}{c}\right) + \frac{1}{2} \left(1 - \frac{r}{p}\right) \quad (1)$$

- m adalah jumlah token yang hilang
- c adalah jumlah token yang dipakai
- r adalah jumlah token yang tersisa
- p adalah jumlah token yang dihasilkan

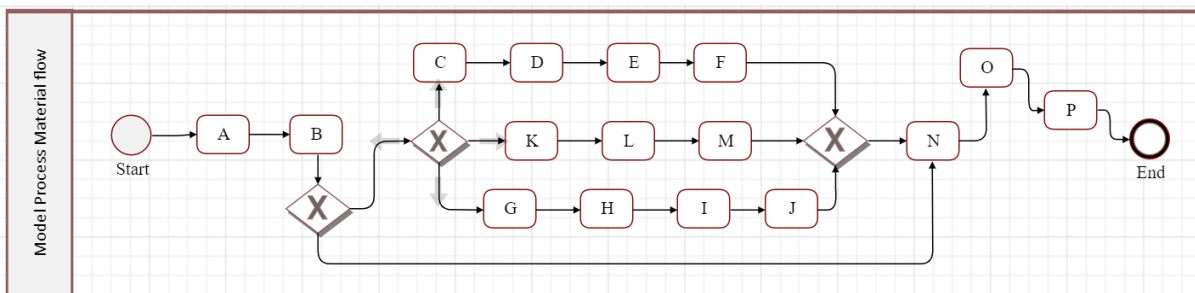
Berdasarkan rumus (2) kita melakukan perhitungan fitness untuk semua varian pada log  $L$  pada model proses  $N$ .

$$fitness(L, N) = \frac{1}{2} \left(1 - \frac{\sum_{\sigma \in L} L(\sigma) \times m_{N,\sigma}}{\sum_{\sigma \in L} L(\sigma) \times c_{N,\sigma}}\right) + \frac{1}{2} \left(1 - \frac{\sum_{\sigma \in L} L(\sigma) \times r_{N,\sigma}}{\sum_{\sigma \in L} L(\sigma) \times p_{N,\sigma}}\right) \quad (2)$$

Conformance checking adalah teknik untuk membandingkan model proses dengan event log dari proses yang sama. Tujuannya adalah untuk memeriksa apakah log peristiwa sesuai dengan model dan sebaliknya.

## 4 Hasil dan Pembahasan

Merujuk pada penelitian terdahulu [6], penelitian dimulai dengan melakukan analisa proses bisnis yang dilakukan pada import dwelling time pelabuhan. Gambaran umum proses bisnis pada import dwelling time pelabuhan dapat dilihat pada Gambar 2.



Gambar 2. Proses Bisnis Import Dwelling Time Pelabuhan

Setelah memahami domain proses bisnis yang dilakukan, kemudian penelitian dilakukan dengan melihat dan melakukan pengecekan terhadap dataset yang akan digunakan dalam penelitian ini dengan melakukan pre-processing.

Hasil pre-processing adalah kami memahami data dan melakukan pengecekan missing value pada dataset event log. Setelah itu melakukan pemilihan atribut yang digunakan untuk penambangan proses, adapun beberapa atribut yang digunakan dalam penambangan proses yaitu case, event dan time stamp. Kemudian dataset dikonversi kedalam format Event log menggunakan Library PM4PY.

### Ekstraksi Dataset

Dataset diekstraksi dari transaksi import dwelling time pelabuhan dengan sebanyak 531582, ringkasan dataset ditunjukkan pada Tabel 1.

**Tabel 1. Ringkasan Dataset**

CaseID	LABEL	Time_Stamps	Group_Activities	CTR_SIZE	CTR_TYPE	ORI_CTR	:GROSS	CUSTOMS_FINAL	CTF	CUSTOMS_BHD	TYPE	CUSTOMS_PERMIT	DOC	TYPE
3042664	A	03/01/2015 07:40	berth discharge	40	DRY	FCL	10	NULL	NULL	NULL	NULL	NULL		
3042664	B	03/01/2015 08:21	yard discharge	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL		
3042664	N	31/07/2015 10:23	delivery	NULL	NULL	NULL	NULL	SPPB	NULL	NULL	NULL	NULL		
3042664	O	31/07/2015 19:21	delivery	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL		
3042664	P	31/07/2015 19:39	delivery	NULL	NULL	NULL	NULL	NULL	FCL	NULL	NULL	NULL		

Kemudian dilakukan pelabelan terhadap dataset berdasarkan aktivitas untuk mempermudah pengelompokan pada dataset, ditunjukkan pada Tabel 2.

**Tabel 2. Pelabelan Dataset Berdasarkan Aktivitas**

Label	Aktivitas
A	Discharge
B	Stack Date
C	Job BHD
D	Move to BHD LOC
E	Job BHD Complete
F	Return BHD move to Yard
G	Job Change Status
H	Move to CFS
I	JComplete Striping
J	Return RS Move to Yard
K	Quarantine Job BHD
L	Move to TPFT
M	Quarantine Job BHD Complete
N	Job DEL Date
O	Truck IN (Ext)
P	Truck OUT

Pada tahapan ekstraksi ini, data yang diekstraksi mengacu pada penelitian terdahulu [6], kemudian dilakukan seleksi data sesuai kebutuhan analisis yang dilakukan. Adapun atribut yang digunakan meliputi case, label dan Time Stamp. Ditunjukkan pada Gambar 3.

```
events = event_df[['CaseID', 'LABEL', 'Time_Stamps']]

events.rename(columns={'CaseID': 'case',
                      'LABEL': 'label',
                      'Time_Stamps': 'datetime'}, inplace=True)
```

**Gambar 3. Seleksi Data Sesuai Kebutuhan Analisis**

### Filtering Event Data

Kemudian dilakukan tahapan filtering data berdasarkan start activities dan end activities pada proses import dwelling time, dimana aktivitas awal dalam proses import dwelling time dimulai dengan aktivitas dengan label “A” dan aktivitas akhir dengan label “P”. Tahapan filtering ini ditunjukkan pada Gambar 4 dan 5.

```
from pm4py.algo.filtering.pandas.start_activities import start_activities_filter
log_start = start_activities_filter.get_start_activities(events)
df_start_activities = start_activities_filter.apply(events, ["A"])
```

**Gambar 4. Filtering on Start Activities**

```
from pm4py.algo.filtering.pandas.end_activities import end_activities_filter
end_activities = end_activities_filter.get_end_activities(df_start_activities)
filtered_df = end_activities_filter.apply(df_start_activities, ["P"])
```

**Gambar 5. Filtering on End Activities**

Tahapan selanjutnya dilakukan proses filtering untuk melakukan pengelompokan pada data berdasarkan variant, ditunjukkan pada Gambar 6. Kemudian didapatkan hasil sebanyak 41 variant dan 102330 case, ditunjukkan pada Tabel 3.

```
from pm4py.statistics.traces.generic.pandas import case_statistics
variants_count = case_statistics.get_variant_statistics(filtered_df)
variants_count = sorted(variants_count, key=lambda x: x['case:concept:name'], reverse=True)
```

**Gambar 6. Filtering Dataset Berdasarkan Variant**

**Tabel 3. Variant**

Frequency	Reference	Trace
96904	$\sigma_1$	A,B,N,O,P
3734	$\sigma_2$	A,B,C,D,E,F,N,O,P
772	$\sigma_3$	A,B,K,L,M,N,O,P
238	$\sigma_4$	A,B,C,D,E,N,F,O,P
150	$\sigma_5$	A,B,C,D,F,E,N,O,P
132	$\sigma_6$	A,B,C,E,N,O,P
121	$\sigma_7$	A,N,B,O,P
103	$\sigma_8$	A,B,C,D,E,N,O,P
26	$\sigma_9$	A,B,K,M,N,O,P
22	$\sigma_{10}$	A,B,N,C,D,O,P
14	$\sigma_{11}$	A,B,N,G,H,I,J,O,P
11	$\sigma_{12}$	A,B,O,N,P
11	$\sigma_{13}$	A,B,N,C,O,P
11	$\sigma_{14}$	A,B,K,M,N,L,O,P
10	$\sigma_{15}$	A,B,C,E,D,N,O,P
8	$\sigma_{16}$	A,N,O,P
8	$\sigma_{17}$	A,B,C,D,E,F,C,D,E,F,N,O,P
7	$\sigma_{18}$	A,B,C,E,D,F,N,O,P
6	$\sigma_{19}$	A,B,K,M,L,N,O,P
4	$\sigma_{20}$	A,B,G,I,N,O,P
3	$\sigma_{21}$	A,C,B,D,E,F,N,O,P
3	$\sigma_{22}$	A,B,N,C,O,D,P
3	$\sigma_{23}$	A,B,G,H,I,N,J,O,P
3	$\sigma_{24}$	A,B,G,H,I,J,N,O,P
2	$\sigma_{25}$	A,B,N,G,I,O,P
2	$\sigma_{26}$	A,B,N,G,I,H,J,O,P

2	$\sigma_{27}$	A,B,K,L,M,C,D,E,F,N,O,P
2	$\sigma_{28}$	A,B,C,N,O,P
2	$\sigma_{29}$	A,B,C,E,D,N,F,O,P
2	$\sigma_{30}$	A,B,C,D,E,N,O,F,P
2	$\sigma_{31}$	A,B,C,D,E,F,O,N,P
2	$\sigma_{32}$	A,B,C,D,E,F,K,L,M,N,O,P
2	$\sigma_{33}$	A,B,C,C,D,D,E,E,F,F,N,O,P
1	$\sigma_{34}$	A,B,N,G,H,I,O,P
1	$\sigma_{35}$	A,B,N,C,D,E,F,O,P
1	$\sigma_{36}$	A,B,E,C,N,O,P
1	$\sigma_{37}$	A,B,C,E,D,C,D,E,F,F,N,O,P
1	$\sigma_{38}$	A,B,N,G,H,I,O,P
1	$\sigma_{39}$	A,B,C,D,E,F,C,E,N,O,P
1	$\sigma_{40}$	A,B,C,D,E,F,C,D,F,E,N,O,P
1	$\sigma_{41}$	A,B,C,D,E,C,F,D,E,N,F,O,P

### Konversi Dataset menjadi Event Log

Kemudian dataset dikonversi kedalam format log, untuk nantinya dilakukan proses discovery menggunakan algoritma alpha miner, ditunjukkan pada Gambar 7.

```
from pm4py.objects.conversion.log import converter as log_converter

log_1 = log_converter.apply(filtered_df)
```

**Gambar 7. Konversi Dataset Kedalam Format Log**

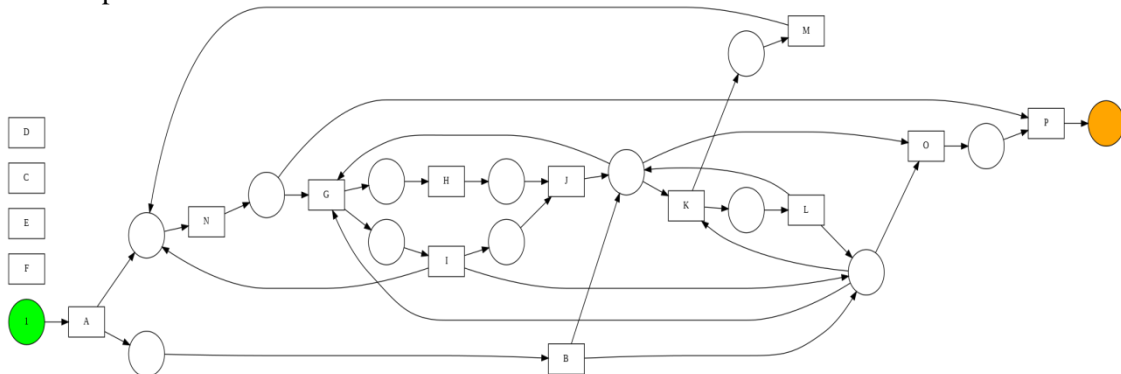
Pengolahan proses dataset dengan algoritma alpha dapat dilihat pada Gambar 8.

```
from pm4py.algo.discovery.alpha import algorithm as alpha_miner

net, initial_marking, final_marking = alpha_miner.apply(log_1)
```

**Gambar 8. Proses Discovery Algoritma Alpha Miner**

Hasil penambangan proses menggunakan algoritma alpha miner akan divisualisasikan dalam bentuk petri net untuk menampilkan struktur yang terdapat pada dataset. Hasil petri net yang dihasilkan dapat dilihat pada Gambar 9.



**Gambar 9. Hasil Discovery Algoritma Alpha Miner**

Dari model proses yang dihasilkan aktivitas C, D, E, F tidak masuk dalam struktur model proses, hal ini terjadi karena data event log yang ada memiliki aktivitas short loops, yaitu diantaranya pada trace 33 “A,B,C,C,D,D,E,E,F,F,N,O,P”.

Conformance Checking dapat dilakukan karena berbagai alasan. Pertama-tama, ini dapat digunakan untuk mengaudit proses dan melihat apakah realitas sesuai dengan beberapa model normatif dan deskriptif. Penyimpangan mungkin mengarah pada kecurangan, inefisiensi dan prosedur yang dirancang dengan buruk. Kedua, Conformance Checking dapat dilakukan untuk mengevaluasi hasil proses mining [11].

Dalam penelitian ini, conformance checking dilakukan menggunakan token based replay. Token based replay akan mencocokkan trace dengan model proses, mulai dari aktivitas awal untuk menemukan transisi mana yang dieksekusi dan aktivitas mana terdapat token yang hilang untuk proses yang diberikan. Token based replay akan menghitung nilai fitness dengan threshold 1, jika nilai fitness mendekati 1 maka variant tersebut mendekati sempurna. Dan jika nilai fitness menjauhi dari 1, maka variant proses tersebut belum mendekati sempurna.

Conformance checking dengan token based replay ditunjukkan pada Gambar 10.

```
from pm4py.algo.conformance.tokenreplay import algorithm as token_replay

replayed_traces = token_replay.apply(log_1, net, initial_marking, final_marking)
replayed_traces
```

**Gambar 10. Conformance Checking**

Adapun hasil dari conformance checking dengan token based replay pada penelitian ini dapat dilihat pada Tabel 4.

**Tabel 4. Hasil Conformance Checking**

Variant	Consumed Token	Missing Token	Produced Token	Remaining Token	Fitness
A, B, N, O, P	8	0	8	0	1.0
A, B, C, D, E, F, N, O, P	8	0	8	0	1.0
A, B, K, L, M, N, O, P	12	0	13	1	0.96
A, B, C, D, E, N, F, O, P	8	0	8	0	1.0
A, B, C, D, F, E, N, O, P	8	0	8	0	1.0
A, B, C, E, N, O, P	8	0	8	0	1.0
A, B, C, D, E, N, O, P	8	0	8	0	1.0
A, B, K, M, N, O, P	11	2	11	2	0.81
A, B, C, D, E, F, C, D, E, F, N, O, P	8	0	8	0	1.0
A, B, C, D, E, F, C, D, F, E, N, O, P	8	0	8	0	1.0

Tahapan akhir dalam penelitian ini adalah membandingkan hasil behavior yang terkandung dalam event log dan behavior yang terkandung dalam model proses, untuk melihat kecocokan keduanya. Dalam penelitian ini evaluasi antara event log dan model proses dilakukan menggunakan replay fitness. Bertujuan untuk menghitung seberapa besar behavior dalam event log yang diterima oleh model proses. Evaluasi event log dan model proses ditunjukkan pada Gambar 11, Adapun hasil dari evaluasi ini ditunjukkan pada Tabel 5.

```
from pm4py.algo.evaluation.replay_fitness import algorithm as replay_fitness_evaluator

fitness = replay_fitness_evaluator.apply(log_1, net, initial_marking, final_marking, variant=replay_fitness_evaluator.Variants.TOKEN_BASED)
```

**Gambar 11. Evaluasi Event Log dan Model Proses**

**Tabel 5. Hasil Evaluasi Event Log dan Model Proses**

Average Trace Fitness	0.999610282059692
Log Fitness	0.9993923001223441
Percentage Of Fitting Traces	99.16349066744846

Untuk lebih lengkap hasil dari pengolahan data event log tersebut dapat dilihat pada GitHub berikut:

<https://github.com/bjokonowo/ConformancePelabuhanPetikemas/blob/main/AlphaMiner.ipynb>



## 5 Kesimpulan

Dari hasil perhitungan fitness didapatkan bahwa model proses dan event log menghasilkan nilai 0.99, yang mengartikan bahwa model proses sesuai dengan terjadinya realita transaksi yang tercatat kedalam event log. Dengan demikian event log sesuai dengan model proses yang dihasilkan. Penelitian ini berfokus pada 1 teknik process discovery, menggunakan algoritma alpha miner. Maka pada pengembangan penelitian berikutnya dapat dilakukan process discovery menggunakan alpha plus yang dapat mengatasi terjadinya short loops.

## Referensi

- [1] Dwi Sukmawati Syafaaruddin, "Evaluation of Container Terminal Efficiency Performance in Indonesia : Future Investment," p. 104, 2015.
- [2] N. Oktavia, M. Marimin, and Y. Arkeman, "Strategi Peningkatan Kinerja Operasional Bongkar Muat Peti Kemas: Studi Kasus di PT Jakarta International Container Terminal," *J. Apl. Bisnis dan Manaj.*, 2020, doi: 10.17358/jabm.6.2.344.
- [3] P. Ricardianto, A. Suhalis, and D. P. Sirait, "Integrasi antara Dwelling Time dan Bongkar Muat Peti Kemas Pelabuhan Tanjung Priok Integration Between Dwelling Time and Loading-Unloading at Tanjung Priok Port," *J. Manaj. Transp. Dan Logistik*, 2018.
- [4] S. Rafi and B. Purwanto, "Dwelling Time Management (Antara Harapan dan Kenyataan Di Indonesia).," *J. Manaj. Bisnis Transp. dan Logistik*, 2016.
- [5] A. R. Tentowi, T. S. Sumadikara, and R. Panggabean, *Politik Hukum Tata Kelola Kepelabuhanan Nasional*. 2016.
- [6] B. Jokonowo, R. Sarno, S. Rochimah, and B. Priambodo, "Process mining: Measuring Key Performance Indicator Container Dwell Time," *Indones. J. Electr. Eng. Comput. Sci.*, 2019, doi: 10.11591/ijeecs.v16.i1.pp401-411.
- [7] W. Van Der Aalst *et al.*, "Process mining manifesto," in *Lecture Notes in Business Information Processing*, 2012, doi: 10.1007/978-3-642-28108-2\_19.
- [8] A. Berti, S. J. Van Zelst, and W. Van Der Aalst, "PM4Py Web Services: Easy Development, Integration and Deployment of Process Mining Features in any Application Stack," in *CEUR Workshop Proceedings*, 2019.
- [9] W. Van der Aalst, *Process mining: Data science in action*. 2016.
- [10] R. A. Hasyim, M. A. Yaqin, and A. H. Utomo, "Analisis Perbandingan Metode Alpha Miner, Inductive Miner dan Causal-Net Mining dalam Proses Mining," *J. Teknol. Inf. dan Terap.*, 2020, doi: 10.25047/jtit.v7i2.165.
- [11] W. M. P. Van Der Aalst and H. M. W. Verbeek, "Process Discovery and Conformance Checking using Passages," *Fundam. Informaticae*, 2014, doi: 10.3233/FI-2014-1006.
- [12] B. Jokonowo, R. Sarno, and S. Rochimah, "Extracting Audit Trail Data of Port Container Terminal For Process Mining," in *Proceedings of 2019 International Conference on Information and Communication Technology and Systems, ICTS 2019*, 2019, doi: 10.1109/ICTS.2019.8850984.
- [13] H. I. Amin, A. Rachmadi, and N. Y. Setiawan, "Pengecekan Conformance Modul Proses Keuangan Dengan Process Mining pada Rumah Sakit Medika Mulia di Kabupaten Tuban," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 9, pp. 2874–2881, 2018.
- [14] Y. A. Effendi and R. Sarno, "Parallel Process Discovery using A New Time-Based Alpha++ Miner," *IJUM Eng. J.*, 2020, doi: 10.31436/iiumej.v21i1.1173.
- [15] A. H. Kusuma, G. Gunawan, and J. Santoso, "Conformance Checking Process Mining SAP Modul SD (Sales and Distribution) dengan Metode Heuristic Miner," *Sisfo*, 2020, doi: 10.24089/j.sisfo.2020.01.001.
- [16] N. F. Fahrudin, "Proses Mining untuk Optimasi Proses Bisnis," *J. Ilm. Teknol. Infomasi Terap.*, 2020, doi: 10.33197/jitter.vol6.iss2.2020.365.
- [17] A. Berti, S. J. Van Zelst, and W. M. P. Van Der Aalst, "Process Mining For Python (PM4Py): Bridging the Gap Between Process and Data Science," *arXiv*. 2019.