



Perbandingan Model Proses Algoritma Alpha dan Alpha++ Pada Aplikasi E-commerce

Bambang Jokonowo^{1*}, Miskah Alfiyyah Kulsum², Nita Komala³,

^{1,2,3}Sistem Informasi, Fakultas Ilmu Komputer, Universitas Mercu Buana

²41818010102@student.mercubuana.ac.id, ³41818010069@student.mercubuana.ac.id,

¹Bambang.jokonowo@mercubuana.ac.id*

Abstract

Utilization of information technology is currently growing rapidly in helping activities especially in storing an event log. The activity which is behavior of the user can be analyzed using process mining. The process mining purpose to extract information from event logs on business processes that working. Discovery technique is used in this research. The purpose of this study is to compare two algorithms applied by creating an e-commerce application that is aware of the processes. E-commerce applications require event logs to read the behavior of visitor activities against the application. This research method starts from understanding the business processes that working, then designing a website by creating the application used. Furthermore, data collection through applications that are promoted through social media. The application will be recorded user activity and formed an event log. The event log that formed then discovered using alpha and alpha++ algorithms by utilizing the ProM Lite 1.2 tools. The evaluation results show that the alpha algorithm has shortcomings, namely length one loop, length two loop and non-free choice. And the alpha++ algorithm fixed this deficiency.

Keywords: Alpha, Alpha++, Process Discovery, Process mining, Event log, E-commerce

Abstrak

Pemanfaatan teknologi informasi saat ini berkembang sangat pesat dalam membantu aktivitas terutama dalam penyimpanan sebuah *event log*. Aktivitas tersebut yang merupakan *behavior* dari *user* dapat dianalisa menggunakan proses *mining*. Proses *mining* bertujuan untuk mengekstrak informasi dari *event log* pada proses bisnis yang dijalankan. Salah satu teknik proses *mining* yang digunakan pada penelitian ini yaitu teknik *discovery*. Tujuan dari penelitian ini adalah membandingkan dua algoritma yang diaplikasikan dengan membuat aplikasi *e-commerce* yang *aware* terhadap proses. Aplikasi *e-commerce* membutuhkan *event log* untuk membaca perilaku aktivitas pengunjung terhadap aplikasi. Metode penelitian ini dimulai dari pemahaman proses bisnis yang dijalankan, kemudian merancang website dengan membuat aplikasi yang digunakan. Selanjutnya pengumpulan data melalui aplikasi yang dipromosikan melalui media *social*. Aplikasi akan merekam aktivitas *user* dan membentuk *event log*. *Event log* yang terbentuk kemudian di-*discover* menggunakan algoritma alpha dan alpha++ dengan memanfaatkan *tools* ProM Lite 1.2. Hasil evaluasi menunjukkan algoritma alpha memiliki kekurangan yaitu *length one loop*, *length two loop* dan *non-free choice*. Dan algoritma alpha++ memperbaiki kekurangan tersebut.

Kata kunci: Alpha, Alpha++, Proses Discovery, Proses Mining, Event log, E-commerce

1. Pendahuluan

Pemakaian teknologi internet menjadi salah satu hal yang penting dalam dunia bisnis karena menjadi media promosi yang bagus bagi perusahaan dan memberikan kemudahan bagi para calon pelanggan untuk melakukan transaksi pembelian barang [1]. Pada dasarnya layanan web yang dimiliki perusahaan berperan penting dalam upaya pemasaran produk [2]. Banyaknya informasi yang perlu diperbarui setiap saat dalam hubungan bisnis sudah pasti menjadi kunci utama dalam sebuah

pemasaran. Proses bisnis dapat dianalisa untuk mendapatkan pengetahuan mengenai aktivitas proses dalam sistem informasi. Proses dapat dimodelkan menggunakan *event log* yang disimpan [3]. Untuk mendukung proses bisnis, aplikasi perusahaan perlu *aware* terhadap suatu proses. Proses *aware* dalam aplikasi adalah sistem perangkat lunak yang mengelola dan menjalankan proses operasional yang melibatkan rang, aplikasi, dan/atau sumber informasi berdasarkan model proses [4].

Pemanfaatan teknologi informasi saat ini berkembang sangat pesat dalam membantu aktivitas terutama dalam penyimpanan sebuah data *event* [5]. Biasanya data *event* tersebut akan tersimpan dalam bentuk *event log*, merupakan proses suatu catatan kejadian yang berisi data urutan suatu peristiwa yang mengacu pada setiap kasus dan aktivitas [6][7]. *Log* ini bisa dianalisa menggunakan *process mining* [8] yang merepresentasikan proses bisnis. Oleh karena itu, bila perusahaan atau organisasi telah mempunyai aplikasi yang menyimpan log secara otomatis, maka *process mining* akan lebih praktis dilakukan [9].

Penelitian yang dilakukan oleh [10] membahas tentang algoritma alpha miner yang menggunakan *event log* data dari aplikasi pembelajaran *online*. Dari penelitian tersebut mendapatkan model proses berdasarkan aktivitas pengerjaan kuis. Hasil dari eksperimen menggunakan data *real* dan *dummy* penelitian telah dilakukan dan terbukti bahwa implementasi algoritma alpha miner dapat bekerja dengan benar pada data *event log* tersebut.

Dalam penerapan *process mining* [10][11] bahwa *event log* perlu beberapa *pre-processing* sebelum dapat dianalisis dengan *process mining tool*. Pada proses *discovery* algoritma telah dilakukan pengembangan untuk menemukan beberapa jenis *relation* [11].

Dalam penelitian ini, permasalahan yang akan diidentifikasi adalah perbandingan antara algoritma alpha dan alpha++ dalam menganalisa sebuah pola pengunjung website pada aplikasi *e-commerce*. Perilaku bisa diketahui dari aktivitas karena dibuatnya sebuah *event log* yang berguna untuk mencatat semua aktivitas yang dilakukan oleh *user* ke dalam *database*. Untuk mendukung aplikasi *e-commerce* ini, akan dicari aspek dari model proses yang didapat. Pencarian ini bisa diawali dengan melihat akses pengguna disitus web, setelah itu dilakukan *discovery* menggunakan algoritma alpha dan alpha++ untuk mengetahui dan mendeteksi perilaku *user* terhadap aplikasi.

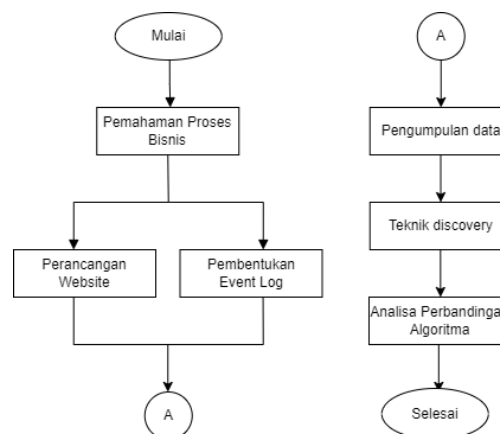
Pada penelitian ini peneliti menggunakan teknik *discovery* untuk melakukan *process mining*. Peneliti memanfaatkan *tool* ProM untuk melihat sebuah hasil dalam bentuk *petri net* serta membandingkan sebuah model proses pada *dataset* dengan hasil *process mining* dengan menggunakan algoritma alpha dan alpha++.

Berdasarkan latar belakang di atas, peneliti membuat aplikasi *e-commerce* yang memperhatikan proses *aware* guna mendapatkan sebuah *event log* untuk dapat dilakukan tahap *process mining* dalam menganalisa perbandingan menggunakan algoritma alpha dan alpha++ untuk mendapatkan model proses bisnis.

2. Metode Penelitian

Dalam metode penelitian ini akan dijelaskan beberapa tahapan melakukan proses *mining*. Tahapan penelitian

yang diusulkan dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Penelitian

2.1. Pemahaman Proses bisnis

Pada proses ini diperlukan sebuah pemahaman dalam proses bisnis. Hal ini penting untuk memahami sebuah proses bisnis pada penelitian yang berguna mengetahui setiap bagian proses yang dapat dianalisa. Dalam penelitian ini kita dapat melihat aktivitas sebuah *user* dalam mengunjungi aplikasi *e-commerce*. Aktivitas tersebut akan diakhiri dengan *end event* yang berbeda-beda. Pada dasarnya setiap *user* akan memiliki aktivitas yang beragam. Di mana *user* dapat melakukan sampai tahap transaksi, hanya melihat-lihat *product* pada aplikasi tersebut dan juga bisa memasukan *product* ke dalam keranjang tanpa melakukan *checkout product* terlebih dahulu yang dikarenakan *user* dalam menggunakan aplikasi *e-commerce* tidak memiliki *Standard Operating Procedure* (SOP).

2.2. Perancangan Website

Perancangan website ini didasarkan pada kebutuhan dan *class diagram* yang telah dibuat, sebagai acuan untuk mengetahui struktur dari suatu sistem dengan jelas dan dapat memberikan gambaran mengenai sistem serta relasi. Dari perancangan ini beralih ke pembuatan aplikasi. Aplikasi ini menggunakan Tailwind sebagai *framework front-end* yang bisa merealisasikan tampilan *multi-device* sesuai perangkat pengguna. *Back-end* menggunakan *framework laravel* karena terdapat *library* yang cukup banyak di mana dapat mempermudah *developer* dalam pengembangan sebuah aplikasi dengan Bahasa pemrograman yang digunakan yaitu PHP.

2.3. Pembentukan Event log

Event log dibentuk untuk proses *aware* [12], adalah sebagai konteks dari semua informasi dapat digunakan untuk mendapatkan pemahaman yang lebih dalam tentang situasi dari sebuah *event* [8]. Dalam pembentukan *event log* diperlukan sebuah *logic* dalam codingan agar ketika *user* melakukan sebuah *event* akan terekam otomatis dalam *database*. Aplikasi *e-commerce* dibuat menggunakan *framework laravel*. Tiap *event*

berisi *code* yang sama tetapi ada beberapa yang memang ditambahkan sesuai kebutuhan.

2.4. Pengumpulan Data

Pada proses ini peneliti melakukan hosting aplikasi yang telah dibuat untuk mendapatkan data *event log*. Pengumpulan data berdasarkan *database* pada aplikasi *e-commerce* yang dapat diakses pada *link* berikut <http://niq-interior.my.id/>. Pengumpulan data dilakukan dalam periode 9 Oktober 2021 hingga 21 Oktober 2021. Dengan cara mempromosikan melalui media *social* seperti *Instagram* dan *Whatsapp*. Data tersebut berupa kumpulan aktivitas-aktivitas yang dilakukan oleh *user* pada saat mengakses aplikasi.

Dalam *dataset* ini terdapat atribut yaitu *id*, *users_id*, *activity*, *products_id*, *category_id*, *user_session*, *created_at*, *updated_at* dan *ip_address*. Selanjutnya dari *dataset* dilakukan ekstraksi dari hasil *export file database event log* dalam bentuk format CSV dengan data sebanyak 1380, ringkasan *dataset* ditunjukkan pada Gambar 2.

| id | users_id | activity | products_id | category_id | users_session | created_at | updated_at | ip_address |
|----|----------|----------------------|-------------|-------------|---------------|------------------|------------------|--------------|
| 1 | 0 | user menuju home | NULL | NULL | NULL | 09/10/2021 11:30 | 09/10/2021 11:30 | 127.0.0.1 |
| 2 | 0 | user melihat product | 3 | NULL | NULL | 09/10/2021 11:31 | 09/10/2021 11:31 | 127.0.0.1 |
| 3 | 0 | user menuju home | NULL | NULL | NULL | 09/10/2021 11:32 | 09/10/2021 11:32 | 114.79.4.347 |
| 4 | 0 | user menuju home | NULL | NULL | NULL | 09/10/2021 11:41 | 09/10/2021 11:41 | 121.5.219.20 |
| 5 | 0 | user menuju home | NULL | NULL | NULL | 09/10/2021 11:41 | 09/10/2021 11:41 | 121.5.219.20 |
| 6 | 0 | user melihat product | 4 | NULL | NULL | 09/10/2021 11:42 | 09/10/2021 11:42 | 121.5.219.20 |
| 7 | 0 | user menuju home | NULL | NULL | NULL | 09/10/2021 11:42 | 09/10/2021 11:42 | 121.5.219.20 |
| 8 | 0 | user menuju home | NULL | NULL | NULL | 09/10/2021 11:44 | 09/10/2021 11:44 | 121.5.219.20 |
| 9 | 0 | user menuju delivery | NULL | NULL | NULL | 09/10/2021 11:45 | 09/10/2021 11:45 | 121.5.219.20 |
| 10 | 0 | user menuju delivery | NULL | NULL | NULL | 09/10/2021 11:45 | 09/10/2021 11:45 | 121.5.219.20 |
| 11 | 0 | user menuju catalog | NULL | NULL | NULL | 09/10/2021 11:46 | 09/10/2021 11:46 | 121.5.219.20 |
| 12 | 0 | user menuju catalog | NULL | NULL | NULL | 09/10/2021 11:46 | 09/10/2021 11:46 | 121.5.219.20 |

Gambar 2. Ringkasan Dataset

Kemudian setiap aktivitas diberi label untuk mempermudah pengelompokan pada *dataset*, tujuannya adalah agar model proses yang telah dibuat nampak lebih sederhana dengan menampilkan label tersebut. Yang ditunjukkan pada Tabel 1.

Tabel 1. Label Dataset Berdasarkan Activity

| Label | Activity |
|-------|--|
| A | User menuju home |
| B | User menuju dashboard |
| C | User menuju catalog |
| D | User melihat product |
| E | User melihat daftar keranjang |
| F | User memasukkan product ke dalam keranjang |
| G | User menuju delivery |
| H | User melakukan checkout product |
| I | User menghapus product |

2.5. Teknik Discovery

Process mining terdiri dari beberapa *task*, salah satunya adalah proses *discovery*. Ini adalah metode untuk secara langsung membentuk *process* bisnis saat ini kemudian mencatat berbagai proses yang terjadi dalam suatu bisnis. Karena *process discovery* adalah bagian dari teknik *process mining*, tujuan utamanya adalah untuk menemukan proses model dari log peristiwa yang menggambarkan perilaku terbaik dari proses bisnis yang diimplementasikan dalam suatu bisnis [13].

Dalam *process discovery* peneliti menggunakan sebuah algoritma alpha, adalah algoritma yang paling mudah

pada penambahan proses yang memanfaatkan aktivitas-aktivitas yang terdapat pada alur kerja untuk dianalisa sehingga membentuk *petri net* yang menggambarkan proses bisnis yang berhasil dianalisa [14].

Kelebihan dari algoritma alpha ialah kemudahannya untuk dijangkau serta diaplikasikan dalam *process mining*. Rangkaian eksekusi suatu model proses sebagai *input*. Catatan ini berasal dari algoritma alpha yang dapat membentuk ringkasan dari susunan serta kaitan dengan peristiwa [15].

Tabel 2. Perbandingan Alpha & Alpha++

| Algoritma | Kelebihan | Kekurangan |
|-----------|--|--|
| Alpha | - | <i>Length one loop</i> <i>Length two loop</i> <i>Invisible task</i> <i>Duplicate Task</i> <i>Implicit Task</i> <i>Non free Choice</i> |
| Alpha++ | <i>Length one loop</i> <i>Length two loop</i> <i>Non free Choice</i> | <i>Invisible task</i> <i>Duplicate Task</i> <i>Implicit Task</i> |

Algoritma didefinisikan dalam istilah *petri net*. Berikut merupakan proses pada algoritma alpha untuk mendapatkan sebuah *Workflow net / Petri net* yaitu sebagai berikut [15]:

Membentuk sekelompok transisi di *workflow net* (T_L)

$$T_L = \{t \in T \mid \exists_{\sigma \in L} t \in \sigma\} \quad (1)$$

Membentuk transisi *source input place* pada *workflow net* (T_i)

$$T_i = \{t \in T \mid \exists_{\sigma \in L} t = first(\sigma)\} \quad (2)$$

Membuat transisi *output sink place* pada *workflow net* (T_o)

$$T_o = \{t \in T \mid \exists_{\sigma \in L} t = last(\sigma)\} \quad (3)$$

Algoritma α menentukan transisi mana yang berelasi secara casual

$$X_L = \{(A, B) \mid A \subseteq T_L \wedge A \neq \emptyset \wedge B \subseteq T_L \wedge B \neq \emptyset \wedge \forall_{a \in A} \forall_{b \in B} a \rightarrow L^b \wedge \forall_{a_1, a_2 \in A} a_1 \# L^{a_2} \wedge \forall_{b_1, b_2 \in B} b_1 \# L^{b_2}\} \quad (4)$$

Menetapkan hasil pasti *places* yang didapat oleh *workflow net*

$$Y_L = \{(A, B) \in X_L \mid \forall_{(A', B') \in X_L} A \subseteq A' \wedge B \subseteq B' \Rightarrow (A, B) = (A', B')\} \quad (5)$$

Membuat *place* yang sudah terindikasi sebelumnya

$$P_L = \{P_{(A, B)} \mid (A, B) \in Y_L\} \cup \{i_L, o_L\} \quad (6)$$

Menyambungkan *place* yang tersedia dengan *input* dan *output* yang ada

$$F_L = \{(a, P_{(A,B)}) | (A, B) \in Y_L \wedge a \in A\} \cup \{(P_{(A,B)}, b) | (A, B) \in Y_L \wedge b \in B\} \cup \{(i_L, t) | t \in T_I\} \cup \{(t, O_L) | t \in T_O\} \quad (7)$$

Penggambaran ulang *workflow net*

$$\alpha(L) = (P_L, T_L, F_L) \quad (8)$$

Selanjutnya merupakan pendefinisian algoritma alpha++ sebagai berikut [16]:

Places yang menghubungkan transisi *length one loop* diidentifikasi dan termasuk dalam L_W

$$X_W = \{(A, B, C) | A \subseteq T' \wedge B \subseteq T' L1L \wedge \forall_{a \in A} \forall_{c \in C} (a > W c \wedge \neg(c \Delta W A)) \wedge \forall_{b \in B} \forall_{c \in C} (c > W b \wedge \neg(c \Delta W b)) \wedge \forall_{a \in A} \forall_{b \in B} a \# W b \wedge \forall \forall_{a_1, a_2 \in A} a_1 \# W a_2 \wedge \forall_{b_1, b_2 \in B} b_1 \# W b_2\} \quad (9)$$

$$L_W = \{(A, B, C) \in X_W | \forall_{(A', B', C') \in X_W} A \subseteq A' \wedge B \subseteq B' \wedge C \subseteq C' \Rightarrow (A, B, C) = (A', B', C')\} \quad (10)$$

Semua transisi *length one loop* dihapus dari input log dan kemudian diproses menggunakan algoritma alpha

$$W^{-L1L} = \emptyset \quad (11)$$

For each $\sigma \in W$ do:

$$(a) \sigma' = \sigma$$

(b) For each $t \in L1L$ do: $i.\sigma' :=$
 $eliminate\ Task(\sigma', t)$

$$(c) W^{-L1L} := W^{-L1L} \cup \sigma' \quad (12)$$

Semua urutan *implicit relation* terdeteksi

$$(P_W^{-L1L}, T_W^{-L1L}, F_W^{-L1L}) = a(W^{-L1L}) \quad (13)$$

Algoritma alpha menemukan WF-net berdasarkan W^{-L1L} dan menghubungkan urutan tersebut

$$P_W = P_W^{-L1L}, T_W = T_W^{-L1L} \cup L1L, F_W = F_W^{-L1L} \cup F_{L1L} \quad (14)$$

Melakukan eliminasi pada *implicit* yang terjadi dalam langkah

$$Treat\ each\ a \mapsto_{W^1} b \in ID_{W^1} as\ a \rightarrow W\ b\ and\ ID_{W^2} = \{(a, b) | a \in T' \wedge b \in T' \wedge a \mapsto_{W^2} b\} \quad (15)$$

$$ID_{W^2} := eliminateRDBByRule1(ID_{W^2}) \quad (16)$$

$$X_W = \{(A \cup A_2, B \cup B_2) | P_{W-L1L} \wedge A_2 \cup B_2 \neq \emptyset \wedge A \cap A_2 = \emptyset \wedge B \cap B_2 = \emptyset \wedge \forall_{A \in A} \forall_{B \in B_2} (a \mapsto_{W^1} b \vee a \mapsto_{W^2} b) \wedge \forall_{a \in A_2} \forall_{b \in B \cup B_2} (a \mapsto_{W^1} b \vee a \mapsto_{W^2} b) \wedge \forall_{a_2 \in A} \forall_{a^2 \in A_2} (a_2 \#_{W^1} a_1 \wedge a_2 \gg W a_1) \wedge \forall_{b_1 \in B} \forall_{b_2 \in B_2} (b_1 \#_{W^1} b_2 \wedge b_1 \gg W b_2)\} \quad (17)$$

$$Y_W = \{(A, B) | ((A, B) \in X_W \vee P_{(A,B)} \in P_{W-L1L}) \wedge \forall_{(A', B') \in X_W} \vee P_{(A', B')} \in P_{W-L1L} (A \subseteq A' \wedge B \subseteq B' \Rightarrow (A, B) = (A', B'))\} \quad (18)$$

Semua *place* yang melibatkan relasi \mapsto_{W^3} diturunkan dan termasuk dalam Z_W

$$Treat\ each\ a \mapsto_{W^2} b \in ID_{W^2} as\ a \rightarrow W\ b\ and\ ID_{W^3} = \{(a, b) | a \in T' \wedge b \in T' a \mapsto_{W^3} b\} \quad (19)$$

Pada rumus ini diterapkan untuk mengurangi *implicit* dependensi yang berlebihan pada ID_{W^3}

$$ID_{W^3} := eliminateRDBByRule2(ID_{W^3}) \quad (20)$$

Semua *places* yang melibatkan *implicit* dependensi diturunkan berdasarkan relasi \mapsto_{W^3}

$$X_W = \{(A, B) | A \subseteq T' \wedge B \subseteq T' \wedge \forall_{a \in A} \forall_{b \in B} a \mapsto_{W^3} b \wedge \forall_{a_1, a_2 \in A} a_1 \#_{W^3} a_2 \wedge \forall_{b_1, b_2 \in B} b_1 \#_{W^3} b_2\} \quad (21)$$

$$Z_W = \{(A, B) \in X_W | \forall_{(A', B') \in X_W} A \subseteq A' \wedge B \subseteq B' \Rightarrow (A, B) = (A', B')\} \quad (22)$$

Semua *place* yang melalui tahap *discovery* dikumpulkan lalu *length one loop* dan *places* juga ditambahkan ke WF-net

$$P_W = \{P_{(A,B)} | (A, B) \in Y_W \cup Z_W\} - \{P_{(A,B)} | \exists_{(A', B', C') \in L_W} A' = A \wedge B' = B\} \cup \{P_{(A \cup C, B \cup C)} | (A, B, C) \in L_W\} \quad (23)$$

$$T_W = T_W^{-L1L} \cup L1L \quad (24)$$

$$F_W = \{(a, P_{(A,B)}) | (A, B) \in P_W \wedge a \in A\} \cup \{(P_{(A,B)}, b) | (A, B) \in P_W \wedge b \in B\} \quad (25)$$

WF-net dengan *non free choice* dan *place* yang telah dihasilkan ditambahkan ke net yang ditemukan oleh algoritma alpha++

$$\alpha^{++}(W) = (P_W, T_W, F_W) \quad (26)$$

3. Hasil dan Pembahasan

Pada penelitian ini akan menjelaskan mengenai perbandingan algoritma alpha dan alpha++ dari data pengunjung aplikasi *e-commerce*. Data yang akan dianalisa adalah data yang masuk, data didapatkan berdasarkan *database* pada aplikasi seperti yang sudah disebutkan pada bagian pengumpulan data diatas. Kemudian data tersebut pada tahap awal akan dilakukan *pre-processing* data dan dilakukan pemberian label pada setiap aktivitas.

Dalam mengolah data untuk mendapatkan model proses yang sesuai dengan kasus pada penelitian ini yaitu, menganalisa pengunjung aplikasi *e-commerce* dengan menggunakan algoritma alpha dan alpha++ digunakanlah sebuah *tool* ProM Lite versi 1.2.

3.1. Pre-processing Data

Sebelum digunakan untuk analisa, dataset akan diproses pada tahap *pre-processing*. Pada tahap ini data dilakukan *sort by ip_address* agar data tersebut lebih teratur. Setelah dilakukan tahap *pre-processing*, atribut yang digunakan yaitu *ip_address*, *label*, *timestamp*, *products_id* dan *activity*.

| ip_address | label | created_at | products_id | activity |
|-----------------|-------|------------------|-------------|----------------------|
| 69.171.231.112 | A | 13/10/2021 13:05 | NULL | user menuju home |
| 69.171.231.112 | A | 13/10/2021 14:48 | NULL | user menuju home |
| 69.171.231.112 | A | 13/10/2021 14:50 | NULL | user menuju home |
| 69.171.231.112 | A | 13/10/2021 14:51 | NULL | user menuju home |
| 69.171.231.112 | A | 13/10/2021 14:57 | NULL | user menuju home |
| 69.171.231.112 | A | 13/10/2021 17:36 | NULL | user menuju home |
| 103.119.230.243 | A | 20/10/2021 3:33 | NULL | user menuju home |
| 103.119.230.243 | D | 20/10/2021 3:35 | 6 | user melihat product |
| 103.119.230.243 | D | 20/10/2021 3:35 | 2 | user melihat product |
| 103.119.230.243 | A | 20/10/2021 3:35 | NULL | user menuju home |
| 103.119.230.243 | A | 20/10/2021 5:56 | NULL | user menuju home |
| 103.119.230.243 | D | 20/10/2021 5:56 | 6 | user melihat product |

Gambar 3. Pre-processing Dataset

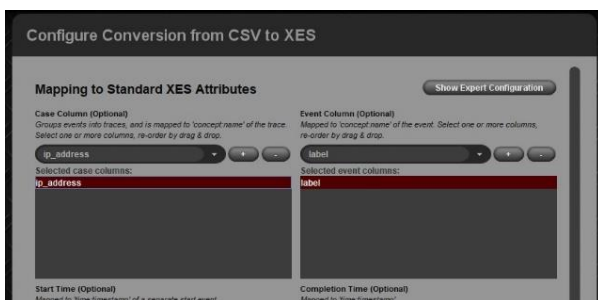
3.2. Analisis Data

Hal pertama yang dilakukan yaitu melakukan *import* file dengan format CSV kemudian diubah menjadi format file XES (*eXtensible Event Stream*) yang merupakan format standar yang digunakan oleh sebagian besar *tool* proses *mining* [17]. Ditunjukkan pada Gambar 4.



Gambar 4. Import File

Setelah melakukan *import* data, hal yang perlu dilakukan selanjutnya adalah menentukan *case column* yaitu *ip_address* dan *event column* yaitu *label*.



Gambar 5. Mapping to Standard XES Attributes

Pada penelitian ini column yang digunakan adalah *ip_address* sebagai *caseID* karena pada penggunaannya *users_id* tidak dapat terdeteksi jika *default* nilainya 0 di mana nilai tersebut terlihat di saat *user* tidak melakukan login dan register. Maka dari itu *ip_address* menjadi *caseID* untuk dapat menangkap aktivitas pengunjung aplikasi *e-commerce*.

| Class | Occurrences (absolute) | Occurrences (relative) |
|-------|------------------------|------------------------|
| A | 593 | 48.967% |
| D | 152 | 12.449% |
| B | 151 | 12.367% |
| E | 112 | 9.173% |
| C | 67 | 5.407% |
| G | 54 | 4.423% |
| F | 42 | 3.44% |
| H | 41 | 3.358% |
| I | 9 | 0.737% |

Gambar 6. Log Summary Activity

Selanjutnya *log summary* menampilkan detail seluruh kejadian dari *event log* berdasarkan *class* yang ada dan mengurutkannya dari *class* terbanyak hingga terkecil. Dalam penelitian ini *class* terbanyak yaitu A (*User menuju home*) dan *class* terkecil yaitu I (*User menghapus product*).

| Class | Occurrences (absolute) | Occurrences (relative) |
|-------|------------------------|------------------------|
| NULL | 1069 | 97.551% |
| 3 | 32 | 2.621% |
| 5 | 32 | 2.621% |
| 2 | 29 | 2.375% |
| 4 | 29 | 2.375% |
| 1 | 20 | 1.628% |
| 6 | 10 | 0.814% |

Gambar 7. Log Summary Products_id

Terlihat pada Gambar 7. *log summary* juga dapat menampilkan produk yang paling banyak dilihat oleh *user*, pada *products_id* 3 dilihat oleh 32 *user*, diikuti dengan *products_id* 2 dilihat oleh 29 *user*, dan produk yang paling sedikit yaitu dengan *products_id* 6 dilihat oleh 10 *user*. Hal ini berfungsi untuk melihat minat dari *user* terhadap produk yang terdapat pada aplikasi.

3.3. Penerapan Algoritma Alpha

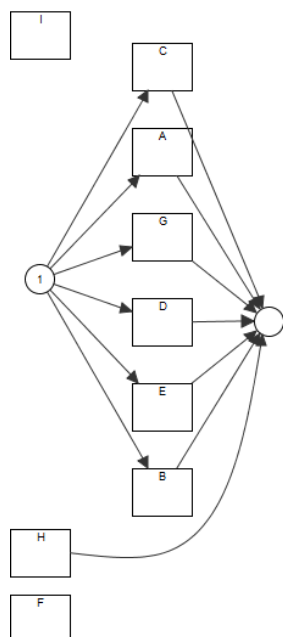
Proses *discovery* dilakukan untuk menghasilkan model proses dengan menggunakan *tools* ProM, pada *tool* tersebut telah terdapat *plugin* algoritma alpha miner. Seperti yang ditunjukkan pada Gambar 8.



Gambar 8. Plugin Alpha Miner

Setelah dilakukan *discovery* menggunakan algoritma alpha akan terbentuk model proses pada Gambar 9. Terdapat sebuah *shortloop* pada aktivitas I dan F. Dikarenakan aktivitas yang ada pada data terdapat

banyak aktivitas berulang yang mengakibatkan tidak terhubung satu sama lain. Di mana contoh *length-one-loop* seperti pada aktivitas (**BBADDD**, **BBACB**). Dan *length-two-loop* terdapat pada aktivitas berikut (**BABADFEH**, **CGEAEA**).



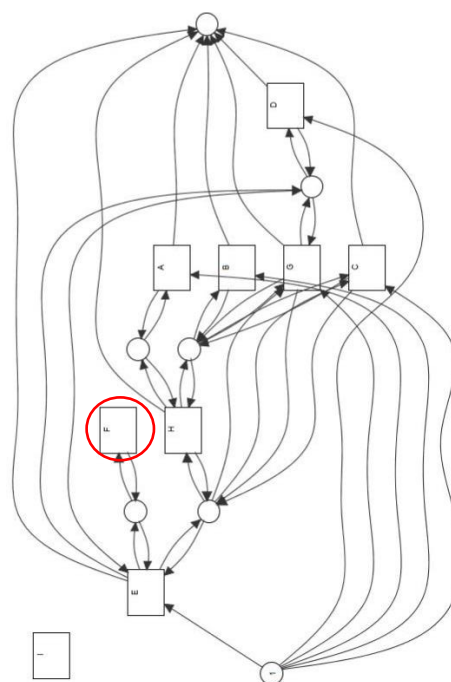
Gambar 9. Petri net Algoritma Alpha

Dari Gambar 9. disimpulkan alpha tidak bisa mengatasi sebuah *shortloop* [18]. Selain itu, alpha juga memiliki kekurangan (1) *length one loop*, merupakan sebuah aktivitas yang dieksekusi beberapa kali secara berurutan. (2) *length two loop*, dua buah aktivitas yang berurutan dieksekusi beberapa kali, (3) aktivitas tak terlihat, sebuah aktivitas yang tidak memiliki jejak catatan kejadian, di mana aktivitas tersebut tidak termasuk ke dalam *transition* serta tidak mampu menampilkan ke dalam net yang dihasilkan algoritma alpha, (4) aktivitas ganda, aktivitas dalam alur kerja yang sama terjadi lebih dari sekali, (5) Implisit *place*, jika sebuah lokasi dalam *petri net* dikatakan implisit maka keberadaannya tidak memberikan pengaruh catatan kejadian sebuah aliran kerja, dan (6) pilihan tidak bebas, di mana algoritma alpha tidak bisa menambang informasi dengan baik karena jaringan tersebut tidak bisa mendapatkan log yang benar [7].

3.4. Penerapan Algoritma Alpha++

Dalam algoritma alpha++ ini merupakan perbaikan dari kelemahan yang ada pada alpha. Pada Gambar 10. terdapat *place* yang digunakan sebagai pre-condition dan post-condition pada transisi [16]. *Place* juga ditambahkan melalui tahap *discovery*. Kemudian pada *short loop* dan *place* ditampilkan ke dalam *petri nets*. Algoritma alpha++ mendeteksi *non-free choice* dengan cara menemukan *short loop*. Pada alpha++ ini terdapat *non-free choice* pada aktivitas F dan H. Dalam model proses yang dihasilkan terdapat aktivitas I yang masih

belum terhubung oleh aktivitas lainnya sedangkan untuk aktivitas F sudah masuk ke dalam aktivitas lain.



Gambar 10. Petri net Algoritma Alpha++

Kasus seperti ini disebabkan adanya *noise* pada log [19] dan *incompleteness* [14]. Penjelasan *noise* pada log merupakan suatu *event log* yang berisi aktivitas yang langka dan tidak sama dengan aktivitas lain. Sedangkan untuk *incompleteness* yaitu aktivitas yang terlalu sedikit dan mengakibatkan sedikitnya varian yang sama. Berikut contoh aktivitas pada salah satu pengunjung yang mengakibatkan aktivitas I tidak terhubung pada aktivitas lain yaitu

(AADFEIEHAEADDFEAEIEIEEADAAAAADDFEIEIEADACAAAAACGAAAABADDFEIE).

4. Kesimpulan

Dari hasil penelitian yang telah dilakukan dapat diketahui bahwa aplikasi *e-commerce* yang telah dibuat dapat menangkap sebuah *event log* dari pengunjung (*user behavior*) yang melakukan aktivitas terhadap aplikasi dan aktivitas tersebut tersimpan ke dalam *database*.

Terkait dengan evaluasi dua algoritma alpha dan alpha++ untuk menganalisa sebuah aktivitas pengunjung aplikasi *e-commerce*, pada hasil yang diperoleh pada algoritma alpha yaitu *length one loop*, merupakan sebuah aktivitas yang dieksekusi beberapa kali secara berurutan berdasarkan event log dari aplikasi *e-commerce* yang dirancang yaitu (**BBADDD**, **BBACB**) tidak dapat menghasilkan model proses. Selain itu pada *length two loop*, merupakan dua buah aktivitas yang berurutan dieksekusi beberapa kali yaitu pada aktivitas (**BABADFEH**, **CGEAEA**). Sedangkan untuk kelebihan

pada algoritma alpha++ dapat mendeteksi *non-free choice* pada aktivitas tetapi untuk aktivitas I tidak dapat membentuk model proses yaitu pada aktivitas (AADFEIEHAEADDDFEAEIEIEEADAAAAADDFEIEIEADACAAAACGAAAABADDFEIE).

Adapun saran yang dapat dipertimbangkan pada penelitian selanjutnya adalah dapat menambahkan *IN* dan *OUT* dengan tujuan agar *user* dapat terdeteksi pada saat masuk atau keluar aplikasi, selain itu dapat menggunakan algoritma yang bisa memperbaiki kekurangan yang ada pada algoritma alpha dan alpha++.

Daftar Rujukan

- [1] A. N. Anastasia and I. Handriani, "Aplikasi Sistem Order Jasa Graphic Designer Berbasis Web Pada PT. Decorner," *J. Ilm. FIFO*, vol. 10, no. 1, p. 87, 2018, doi: 10.22441/fifo.v10i1.2943.
- [2] S. Kosasi, "Perancangan Sistem *E-commerce* Untuk Memperluas Pasar Produk Oleh-Oleh Khas Pontianak," *Snastia*, vol. 2015, no. Oktober, pp. 110–119, 2015.
- [3] B. Jokonowo, R. Sarno, S. Rochimah, and B. Priambodo, "Process mining: Measuring key performance indicator container dwell time," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 16, no. 1, pp. 401–411, 2019, doi: 10.11591/ijeecs.v16.i1.pp401-411.
- [4] W. M. P. Van Der Aalst, "Process-aware information systems: Lessons to be learned from process mining," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5460 LNCS, no. January 2009, pp. 1–26, 2009, doi: 10.1007/978-3-642-00899-3_1.
- [5] D. T. Nguyen and J. E. Jung, "Real-time event detection for online behavioral analysis of big social data," *Futur. Gener. Comput. Syst.*, vol. 66, pp. 137–145, 2017, doi: 10.1016/j.future.2016.04.012.
- [6] R. Adhim, M. A. Shiddiq, F. F. Ghizbunaza, and M. A. Yaqin, "Process Discovery pada *Event log* Permainan Hay Day menggunakan Algoritma Inductive Miner," *Semin. Nas. Inov. dan Apl. Teknol. di Ind. 2019*, no. February, pp. 66–73, 2019.
- [7] D. Reißner, A. Armas-Cervantes, R. Conforti, M. Dumas, D. Fahland, and M. La Rosa, "Scalable alignment of process models and event logs: An approach based on automata and S-components," *Inf. Syst.*, vol. 94, p. 101561, 2020, doi: 10.1016/j.is.2020.101561.
- [8] A. Hermawan, "Business Process Context Analysis Based on 'Event log,'" *J. Penelit. dan Pengemb. Komun. dan Inform.*, vol. 4, no. 3, p. 122699, 2014.
- [9] H. Rizqifaluthi and M. A. Yaqin, "Process mining Akademik Sekolah menggunakan RapidMiner," *Matics*, vol. 10, no. 2, p. 47, 2019, doi: 10.18860/mat.v10i2.5158.
- [10] P. Nafasa, I. Waspada, N. Bahtiar, and A. Wibowo, "Implementation of Alpha Miner Algorithm in Process mining Application Development for Online Learning Activities Based on MOODLE Event log Data," *ICICOS 2019 - 3rd Int. Conf. Informatics Comput. Sci. Accel. Informatics Comput. Res. Smarter Soc. Era Ind. 4.0, Proc.*, no. October, 2019, doi: 10.1109/ICICoS48119.2019.8982384.
- [11] R. Sarno, K. R. Sungkono, M. Taufiqulsa'di, H. Darmawan, A. Fahmi, and K. Triyana, "Improving efficiency for discovering business processes containing invisible tasks in non-free choice," *J. Big Data*, vol. 8, no. 1, 2021, doi: 10.1186/s40537-021-00487-x.
- [12] A. K. A. De Medeiros, W. M. P. Van Der Aalst, and A. J. M. M. Weijters, "Workflow Mining: Current Status and Future Directions," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 2888, no. i, pp. 389–406, 2003, doi: 10.1007/978-3-540-39964-3_25.
- [13] Y. A. Effendi and R. Sarno, "Parallel process discovery using a new Time-Based Alpha++ Miner," *IJUM Eng. J.*, vol. 21, no. 1, pp. 126–141, 2020, doi: 10.31436/iiumej.v21i1.1173.
- [14] R. Accorsi, M. Ullrich, and W. M. P. Van Der Aalst, *Process mining Data Science In Action Second Edition*, vol. 35, no. 5, 2012.
- [15] 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.*, 2004.
- [16] L. Wen, W. M. P. Van Der Aalst, J. Wang, and J. Sun, "Mining process models with non-free-choice constructs," *Data Min. Knowl. Discov.*, vol. 15, no. 2, pp. 145–180, 2007, doi: 10.1007/s10618-007-0065-y.
- [17] Afina Lina Nurlaili and Agung Mustika Rizki, "Analisis Kualitas Model Proses dalam Implementasi Process mining : Literature Review," *J. Comput. Electron. Telecommun.*, vol. 1, no. 2, pp. 1–10, 2021, doi: 10.52435/complete.v1i2.74.
- [18] H. W. Sun, W. Liu, L. Qi, Y. Y. Du, X. Ren, and X. Y. Liu, "A process mining algorithm to mixed multiple-concurrency short-loop structures," *Inf. Sci. (Ny)*, vol. 542, pp. 453–475, 2021, doi: 10.1016/j.ins.2020.07.003.
- [19] D. Yue, X. Wu, H. Wang, and J. Bai, "A review of process mining algorithms," *BMEI 2011 - Proc. 2011 Int. Conf. Bus. Manag. Electron. Inf.*, vol. 5, no. 70872082, pp. 181–185, 2011, doi: 10.1109/ICBMEI.2011.5914454.