



TUGAS AKHIR - KI091391

***RANCANG BANGUN FUZZY ASSOCIATION
RULE MINER UNTUK MENDETEKSI FRAUD
PADA PROSES BISNIS ENTERPRISE RESOURCE
PLANNING (ERP)***

Fernandes P. Sinaga
NRP 5110100190

Dosen Pembimbing
Prof. Drs.Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D.
Abdul Munif, S.Kom, M.Sc.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2014



FINAL PROJECT - KI091391

***FUZZY ASSOCIATION RULE MINER
ARCHITECTURE FOR FRAUD DETECTION IN
BUSINESS PROCESS ENTERPRISE RESOURCE
PLANNING (ERP)***

Fernandes P. Sinaga
NRP 5110100190

Dosen Pembimbing
Prof. Drs.Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D.
Abdul Munif, S.Kom, M.Sc.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2014

Rancang Bangun *Fuzzy Association Rule Miner* untuk Mendeteksi *Fraud* pada Proses Bisnis *Enterprise Resource Planning (ERP)*

Nama Mahasiswa : Fernandes P. Sinaga
NRP : 5110 100 190
Jurusan : Teknik Informatika FTIf-ITS
Dosen Pembimbing 1 : Prof. Drs.Ec. Ir. Riyanarto Sarno, M.Sc.,
Ph.D.
Dosen Pembimbing 2 : Abdul Munif, S.Kom, M.Sc.

ABSTRAKSI

Saat ini banyak perusahaan yang telah menggunakan sistem Enterprise Resource Planning (ERP) untuk mengatur proses bisnis yang dijalankan. Proses bisnis ERP pada suatu perusahaan akan selalu mengalami perubahan secara dinamis. Perubahan yang terjadi dapat menghasilkan variasi-variasi terhadap proses bisnis ERP tersebut. Variasi antara proses bisnis yang berjalan terhadap proses bisnis yang standar dapat diperiksa menggunakan process mining. Pada tugas akhir ini, dibahas mengenai variasi proses bisnis yang mengandung kecurangan. Kecurangan pada variasi proses bisnis dapat dideteksi dengan menggunakan metode process mining dan dengan pendekatan fuzzy association rule learning. Process mining mendeteksi kecurangan pada proses bisnis dengan cara memeriksa ketidaksesuaian antara event logs dari proses bisnis berjalan dengan proses bisnis yang sesuai standar perusahaan. Hasil pemeriksaan ketidaksesuaian tersebut berupa kumpulan pelanggaran yang dilakukan terhadap proses bisnis. Kumpulan pelanggaran ini kemudian diolah dengan metode fuzzy association rule learning untuk menghasilkan aturan asosiasi antara perilaku kecurangan yang dilakukan serta mengukur tingkat keparahan yang disebabkan oleh kecurangan yang dilakukan. Akan tetapi, aturan asosiasi yang dihasilkan tidak sepenuhnya dapat mendeteksi kecurangan karena terdapat kemungkinan adanya proses fraud dengan bobot yang rendah dan proses normal dengan bobot yang tinggi. Sehingga dalam penelitian ini ditambahkan aturan tambahan untuk mengatasi masalah

tersebut. Dari percobaan yang dilakukan terhadap proses bisnis aplikasi kredit bank dan proses procurement pada ERP, diperoleh akurasi yang tinggi dalam pendeteksian kecurangan yang dilakukan. Hal ini menunjukkan bahwa penggabungan metode process mining dan fuzzy association rule learning dapat digunakan untuk mendeteksi kecurangan pada proses bisnis dengan efektif dan akurat.

Kata kunci: *Penggalian Proses, Algoritma Fuzzy Association Rule Mining, Fuzzy Multi Attribute Decision Making, Plugin ProM, Pemeriksaan Kesesuaian, Deteksi Kecurangan.*

Fuzzy Association Rule Miner Architecture for Fraud Detection in Business Process Enterprise Resource Planning (ERP)

Student Name : Fernandes P. Sinaga
NRP : 5110 100 190
Major : Teknik Informatika FTIf-ITS
Advisor 1 : Prof. Drs.Ec. Ir. Riyanarto Sarno, M.Sc.,
Ph.D.
Advisor 2 : Abdul Munif, S.Kom, M.Sc.

ABSTRACTION

Currently many companies are already using Enterprise Resource Planning (ERP) system to manage running business process. ERP business process in a company will always be changing dynamically. Changes that occur can result in variations to the ERP business process. Variation between business processes, running on standard business process can be checked using process mining. In this final project, we discussed about the variations of business process that contain fraud. Fraud in business process variations can be detected by using process mining and fuzzy association rule learning approach. Process mining can detect fraud on business process by examining the discrepancy between event logs of running business processes with company's business process standard. The results of compliance checking are collection of offenses committed against business process. This set of violations is then processed by fuzzy association rule learning method to generate association rules between the behaviors of committed fraud and measure the severity caused by the committed fraud. However, the resulting association rules are not fully able to detect fraud because there is a possibility of fraud with the low weight and normal process with a high weight. Thus, in this study we add some association rules to resolve the issue. From the experiments conducted on a bank credit application business process and the ERP procurement process, obtained high accuracy in the detection of fraud committed. This suggest that the

combination of process mining method with fuzzy association rule learning can be used to detect fraud in business process effectively and accurately.

Keyword: *Process mining, Algoritma Fuzzy Association Rule Mining, Fuzzy Multi Attribute Decision Making, Plugin ProM, Conformance Checking, Fraud Detection.*

LEMBAR PENGESAHAN

LEMBAR PENGESAHAN

Rancang Bangun *Fuzzy Association Rule Miner* untuk Mendeteksi *Fraud* pada Proses Bisnis *Enterprise Resource Planning (ERP)*

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Rekayasa Perangkat Lunak
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :
FERNANDES P. SINAGA
NRP : 5110 100 190

Disetujui oleh Dosen Pembimbing Tugas Akhir :

Prof. Drs.Ec. Ir. Rivanarto Sarno, M.Sc.
NIP: 195908031986011004

Abdul Munif, S.Kom, M.Sc.
NIP: 051100114



(Pembimbing 1)

(Pembimbing 2)

SURABAYA
JUNI 2014

vii

vii

BAB I

PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar Tugas Akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan Tugas Akhir, dan sistematika penulisan.

1.1. Latar Belakang

Perusahaan-perusahaan di dunia telah menggunakan sistem *Enterprise Resource Planning* (ERP) untuk mengatur proses bisnis yang dimiliki. Proses bisnis dapat mengalami perubahan secara dinamis [1]. Perubahan ini disebabkan oleh bertambahnya kebutuhan dan informasi, berubahnya kondisi pasar, atau perubahan kebijakan terhadap proses bisnis dalam perusahaan [2]. Perubahan ini menghasilkan variasi terhadap proses bisnis. Dalam kenyataannya, variasi proses bisnis ada yang telah dijalankan dengan benar dan ada juga yang mengandung kecurangan di dalamnya. Kecurangan tersebut dapat dikatakan sebagai *fraud* pada proses bisnis. Kecurangan yang dilakukan dengan banyak cara mengakibatkan kerugian yang signifikan dalam perusahaan. Penelitian dalam ranah penggalian data dan penggalian proses memberikan berbagai solusi untuk pendeteksian kecurangan [3]. Penipuan adalah suatu masalah yang mengancam di dunia. Ada 1.388 penipuan yang menyebabkan kerugian 1,4 miliar dolar AS di 96 negara [4]. Rata-rata, organisasi kehilangan 7 persen dari pendapatan kotor setiap tahun. Ada 20 persen orang di perusahaan tidak pernah mencuri, 60 persen dari mereka bergantung pada kasus dan kesempatan, dan 20 persen terakhir benar-benar tidak jujur [5].

Pada penelitian sebelumnya, penggalian proses telah memberi perhatian untuk mengurangi penipuan internal proses bisnis [6]. Penelitian tersebut menggunakan beberapa perangkat penggalian proses di ProM seperti *conformance checker*, *dotted-chart analysis*, *social network miner*, *originator by task matrix*, dan

lainnya untuk melakukan investigasi terhadap *event log* yang dihasilkan [7].

Penelitian lainnya yang mendukung pendeteksian penipuan adalah menggunakan *Association Rule Learning* (ARL) untuk mengekstrak aturan asosiasi yang terdapat dalam sejumlah data transaksi dalam perusahaan grosir [8].

Dari sejumlah penelitian di atas, penipuan dapat dideteksi dari proses yang berjalan. Akan tetapi, kecurangan yang dapat dideteksi adalah kecurangan yang bersifat pasti atau bernilai 1. Sedangkan untuk kecurangan yang intensitasnya rendah tidak dapat dideteksi.

Dalam tugas akhir ini dijelaskan mengenai penelitian dalam pendeteksian *fraud* pada studi kasus proses bisnis aplikasi kredit pada bank dan proses *procurement* pada sistem ERP dengan menggunakan metode *Fuzzy Association Rule* yang dibantu dengan metode penggalan proses. Tujuannya adalah untuk menghasilkan data aturan asosiasi perilaku pelanggaran beserta nilai *confidence*-nya terhadap *fraud* yang digunakan untuk mendeteksi *fraud* pada *event logs* proses bisnis. Selain itu, pada tugas akhir ini juga dijelaskan mengenai perhitungan bobot *fraud* dengan menggunakan teori *Fuzzy Set Multi Attribute Decision Making*.

1.2. Tujuan

Tujuan dari Tugas Akhir ini adalah sebagai berikut:

1. Membuat *plugin* ProM dengan *input* atau masukan berupa *event log* dan model proses bisnis yang telah dimodifikasi sesuai dengan Standar Operasional Prosedur (SOP).
2. Memodifikasi *plugin Check Compliance using Conformance Checking* dan membuat *plugin* pada ProM 6 untuk menghasilkan data pelanggaran.
3. Membuat *plugin* ProM yang mampu menghitung bobot *fraud* dan aturan asosiasi dari data pelanggaran.

4. Membuat *plugin* ProM yang mampu mendeteksi *fraud* dari bobot *fraud* dan aturan asosiasi pelanggaran yang dihasilkan.
5. Membuat *plugin* ProM yang mampu mengkategorikan hasil pendeteksian *fraud* ke dalam kelas *fraud* yang sesuai.

1.3. Rumusan Permasalahan

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana mendapatkan data pelanggaran yang terdapat pada *event log* dari proses bisnis yang dijalankan.
2. Bagaimana cara menghitung bobot *fraud* dengan menggunakan algoritma *Fuzzy Sets Multi Attribute Decision Making*.
3. Bagaimana cara menggabungkan algoritma *Fuzzy* dan *Association Rule Learning (ARL)* untuk menentukan aturan asosiasi data pelanggaran.
4. Bagaimana mendeteksi *fraud* dari suatu *case* dengan menggunakan bobot *fraud* dan aturan asosiasi pelanggaran yang dihasilkan.
5. Bagaimana melakukan kategorisasi hasil pendeteksian *fraud* ke dalam kelas *fraud* yang sesuai.
6. Bagaimana membuat *plugin* ProM yang menjalankan setiap perhitungan dan pendeteksian *fraud*.

1.4. Batasan Permasalahan

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan, di antaranya sebagai berikut:

1. Bahasa pemrograman menggunakan bahasa Java.

2. Data uji yang digunakan berupa *event log* dan model proses bisnis berekstensi *pnml* yang telah dimodifikasi sesuai dengan Standar Operasional Prosedur (SOP).
3. Data uji yang digunakan tidak riil, melainkan simulasi untuk proses bisnis aplikasi kredit pada bank dan proses bisnis *procurement* ERP.
4. Penilaian kepentingan kriteria pelanggaran menggunakan data dari pakar.
5. Nilai *Threshold* pada algoritma *Fuzzy Association Rule Learning* ditentukan oleh pengguna *plugin*.
6. *Plugin* yang dihasilkan khusus untuk *software* ProM 6.0.

1.5. Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan Tugas Akhir ini yaitu:

1. Studi literatur

Pada tahap ini dilakukan pengumpulan informasi mengenai *plugin* ProM beserta algoritmanya. Mengumpulkan dan menggali informasi dan literatur yang diperlukan dalam proses perancangan dan implementasi sistem yang dibangun. Literatur yang digunakan adalah sebagai berikut:

- a. Pendeteksian Kecurangan.
- b. *Conformance Checking*.
- c. *Fuzzy Sets Multiple Attribute Decision Making*.
- d. *Association Rule Learning*.
- e. *Fuzzy Association Rule Learning*.
- f. *ProM*.
- g. *Plugin* ProM.
- h. *Event Log*
- i. *Petri Net*.

2. Analisis dan Perancangan Sistem

Proses analisis dari perangkat lunak yang akan dibuat ini adalah dengan pemecahan masalah-masalah yang telah dirumuskan dalam bab rumusan masalah. Seperti bagaimana cara mendapatkan data pelanggaran dari *event log* dari proses bisnis yang berjalan, menghitung bobot *fraud* dan mencari aturan asosiasi dari data pelanggaran yang digunakan untuk mendeteksi kecurangan. Tentu sebelum masuk ke dalam tahap implementasi harus merumuskan dan menganalisis terlebih dahulu.

Proses desain dari perangkat lunak yang akan dibuat ini adalah dengan mengolah hasil dari analisis yang telah dilakukan sebelumnya. Setelah masalah mampu dianalisis dan dipecahkan, maka dengan prosedur atau algoritma yang ada, akan digunakan di dalam pengimplementasian sehingga mendapat hasil akhir yang diinginkan dalam tugas akhir ini yaitu mendeteksi kecurangan atau *fraud* pada sebuah *case (event log)* dari proses bisnis yang berjalan.

3. Implementasi

Pengembangan aplikasi untuk tugas akhir ini berupa *plugin* untuk ProM 6.0. *Input* berupa *event log* dan model proses bisnis yang telah dimodifikasi sesuai dengan SOP dan *output* yang dihasilkan berupa data *fraud* hasil pendeteksian *fraud*. Beberapa hal yang diperlukan dalam implementasi ini adalah:

- a. IDE Eclipse Kepler.
- b. *Project* dasar ProM 6.0.
- c. Bahasa Pemrograman Java.
- d. Java Development Kit 7 (JDK).
- e. Java Runtime Environment (JRE).
- f. *Library* ProM.

4. Pengujian dan evaluasi

Pengujian aplikasi ini akan menggunakan *black box testing*, dimana *input* berupa beberapa *event log* dan model proses bisnis yang telah dimodifikasi sesuai dengan SOP, dan akan dianalisis apakah hasil atau *output* yang dihasilkan memiliki hasil yang sesuai. Dan juga mendeteksi apakah program dapat berjalan dengan seharusnya.

Tahapan-tahapan dari pengujian adalah sebagai berikut:

- a. pencocokan hasil uji *plugin* dengan perhitungan manual;
- b. pengujian dari fitur-fitur tambahan; dan
- c. pengujian kegunaan pada pengguna ProM;

5. Penyusunan buku Tugas Akhir

Pada tahap ini dilakukan pendokumentasian dan pelaporan dari seluruh konsep, dasar teori, implementasi, proses yang telah dilakukan, dan hasil-hasil yang telah didapatkan selama pengerjaan Tugas Akhir.

1.6. Sistematika Penulisan

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini.

Bab I Pendahuluan

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan Tugas Akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan Tugas Akhir.

Bab II Dasar Teori

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan mendasari pembuatan Tugas Akhir ini.

Bab III Metode Pemecahan Masalah

Bab ini membahas mengenai Metode yang digunakan untuk memecahkan masalah yang dipaparkan pada rumusan permasalahan.

Bab IV Analisis dan Perancangan Sistem

Bab ini membahas mengenai perancangan perangkat lunak. Perancangan perangkat lunak meliputi perancangan data, arsitektur, proses dan perancangan antarmuka pada *plugin*.

Bab V Implementasi

Bab ini berisi implementasi dari perancangan perangkat lunak *plugin* dan implementasi fitur-fitur penunjang *plugin*.

Bab VI Pengujian dan Evaluasi

Bab ini membahas pengujian dengan metode pengujian subjektif untuk mengetahui penilaian aspek kegunaan (*usability*) dari perangkat lunak dan pengujian fungsionalitas yang dibuat dengan memperhatikan keluaran yang dihasilkan serta evaluasi terhadap fitur-fitur *plugin*.

Bab VII Kesimpulan

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan. Bab ini membahas saran-saran untuk pengembangan sistem lebih lanjut.

Daftar Pustaka

Merupakan daftar referensi yang digunakan untuk mengembangkan Tugas Akhir.

Lampiran

Merupakan bab tambahan yang berisi daftar istilah yang penting pada aplikasi ini.

[Halaman ini sengaja dikosongkan]

BAB II

DASAR TEORI

Pada bab ini akan dibahas mengenai teori-teori yang menjadi dasar dari pembuatan Tugas Akhir. Teori-teori tersebut meliputi Pendeteksian Kecurangan, *Conformance Checking*, *Fuzzy Sets Multiple Attribute Decision Making*, *Association Rule Learning*, *Fuzzy Association Rule Learning*, *ProM*, *Plugin ProM*, *Event Log*, dan *Petri Net*.

2.1. Pendeteksian Kecurangan (*Fraud Detection*)

2.1.1. *Fraud*

Fraud (kecurangan/penipuan) merupakan ketidakjujuran yang direncanakan menyebabkan cedera lain dengan membuat pernyataan palsu, menyembunyikan, atau mengabaikan fakta material [3]. *Fraud* juga dapat diartikan sebagai penyalahgunaan sistem organisasi tanpa memimpin dasarnya konsekuensi hukum [9]. Proses penipuan bisnis adalah penipuan yang disebabkan oleh salah mengartikan urutan proses.

Dalam sistem perbankan Indonesia, *fraud* merupakan tindakan penyimpangan atau pembiaran yang sengaja dilakukan untuk mengelabui, menipu, atau memanipulasi bank, nasabah atau pihak lain, yang terjadi di lingkungan bank dan/atau menggunakan sarana bank sehingga mengakibatkan bank, nasabah, atau pihak lain menderita kerugian dan/atau pelaku *fraud* memperoleh keuntungan keuangan baik secara langsung maupun tidak langsung [10].

Berdasarkan sebab terjadinya *fraud*, konsep segitiga *fraud* menjelaskan bahwa *fraud* dapat terjadi karena tiga hal yaitu:

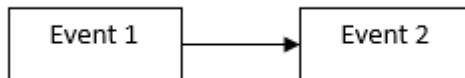
1. Penekanan/pemaksaan.
2. Melihat kesempatan.
3. Pelaku *fraud* pasti berlaku dengan rasional atau sama dengan normal *flow*.

Dalam upaya untuk mendeteksi adanya kecurangan di dalam proses, dapat dilakukan kontrol internal sebagai *counter measure*

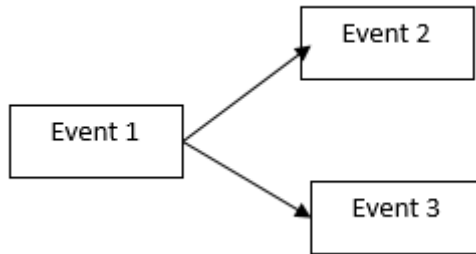
terhadap kecurangan yang mungkin terjadi [11]. Contohnya adalah dengan mendefinisikan Standar Operasional Prosedur (SOP) untuk proses bisnis yang terdiri dari proses bisnis standar dan catatan waktu, *resource*, *organization/role*, dan penentuan keputusan yang dapat digunakan sebagai kontrol terhadap proses berjalan. *Fraud* dapat terjadi karena pelanggaran pada proses bisnis / standar operasional prosedur dan adanya manipulasi data. Untuk menganalisis manipulasi pada data tersebut dapat menggunakan teknik penggalian data seperti algoritma *neural network*, *support vector machine*, dan lain-lain. Sedangkan untuk menganalisis pelanggaran pada proses bisnis dapat menggunakan teknik penggalian proses seperti *fuzzy miner* dan *heuristic* [12]. *Process-based Fraud* (PBF) merupakan *fraud* yang terjadi pada proses bisnis [12]. Berdasarkan penelitian yang telah dilakukan mengenai PBF, telah diidentifikasi atribut dan pola untuk menggambarkan PBF [13]. Hasil identifikasi PBF tersebut menemukan atribut kecurangan dalam proses bisnis yang dapat didefinisikan menjadi enam tipe berdasarkan cara kecurangan tersebut dilakukan yaitu:

1. *Skipped Activity*

Jenis *fraud* di mana terdapat aktivitas yang di dalam SOP seharusnya dikerjakan, tetapi kenyataannya tidak dikerjakan atau dilewati [3]. *Skip* aktivitas ini dibagi ke dalam dua bagian berdasarkan jenis aktivitas yang dilewati, yaitu *Skip Sequence* untuk aktivitas yang termasuk *sequence* biasa yang dapat dilihat pada Gambar 2.1 dan *Skip Decision* untuk aktivitas yang merupakan *decision* di mana terdapat pengambilan keputusan atau terdapat percabangan *event* [13] yang dapat dilihat pada Gambar 2.2.

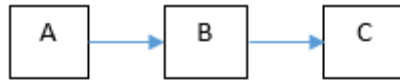


Gambar 2.1. Event Sequence

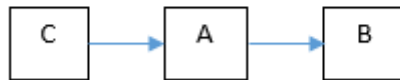


Gambar 2.2. Event Decision

2. *Wrong Throughput Time*
 Jenis *fraud* di mana terjadi pengerjaan aktivitas yang lebih cepat atau lebih lama dari batasan waktu yang ditetapkan dalam SOP. Jenis *fraud* ini dibagi menjadi dua bagian yaitu *Throughput Time Min* (pelanggaran waktu di bawah waktu normal) dan *Throughput Time Max* (pelanggaran waktu di atas waktu normal).
3. *Wrong Resource*
 Jenis *fraud* di mana suatu aktivitas tidak dikerjakan oleh orang yang mempunyai *role* yang sesuai dengan SOP.
4. *Wrong Duty*
 Jenis *fraud* di mana terdapat seorang pegawai yang melakukan dua atau lebih aktivitas yang berbeda dalam satu proses yang berjalan. Jenis *fraud* ini dibagi menjadi tiga bagian yaitu *Wrong Duty Sequence* (terjadi pada aktivitas *sequence*), *Wrong Duty Decision* (terjadi pada aktivitas *decision*), dan *Wrong Duty Combine* (terjadi pada aktivitas *sequence* dan *decision*).
5. *Wrong Pattern*
 Jenis *fraud* di mana terjadi kesalahan urutan aktivitas yang tidak sesuai dengan urutan aktivitas yang ditetapkan pada model proses bisnis standar di dalam SOP. Contohnya dapat dilihat pada Gambar 2.3.



Pola model proses di SOP



Case yang melanggar pola

Gambar 2.3. Contoh *Wrong Pattern*

6. *Wrong Decision.*

Jenis *fraud* di mana terjadi kesalahan penarikan keputusan yang tidak sesuai dengan standar yang ditetapkan di dalam SOP.

Kumpulan atribut PBF di atas dapat dilihat pada Gambar 2.4. Kecurangan dapat dideteksi dengan menggunakan dua pendekatan yaitu pendekatan *Data Mining* dan *Process Mining*.

Skip		Throughput Time		Wrong Resource	Wrong duty			Wrong Pattern	Wrong Decision
sequence	decision	Min	Max		Sequence	Decision	Combine		

Gambar 2.4. Atribut PBF

2.1.2. *Data Mining for Fraud Detection*

Data mining merupakan metode klasik dalam bidang ilmu komputer yang digunakan untuk menganalisis data yang besar. Metode yang terdapat pada *data mining* yaitu *decision tree*, *machine learning*, *sequential pattern mining*, *association rule learning*, *neural network*, dan lain-lain.

Dalam konteks pendeteksian kecurangan, metode *data mining* dapat digunakan untuk memeriksa data-data sebagai *input* dan menghasilkan model atau pola sebagai *output* yang dapat digunakan untuk mendeteksi adanya kecurangan dalam suatu

proses [14]. Beberapa metode *data mining* seperti *Decision Tree*, *Neural* dan *Bayesian Network*, serta *Support Vector Machine* telah digunakan di dalam penelitian sebelumnya untuk menangkap kecurangan dalam proses [3]. Akan tetapi, metode-metode tersebut memiliki kelemahan dalam mendeteksi kecurangan karena tidak mampu menganalisis perilaku aliran kontrol proses. Hal ini dikarenakan metode-metode tersebut hanya memeriksa pola di dalam *data set*. Padahal terdapat kemungkinan di mana penipu melakukan kecurangan pada tugas yang tidak sesuai dengan SOP. Kecurangan dalam hal ini dapat dideteksi dengan melakukan penggalian proses (*Process Mining*).

2.1.3. Process Mining for Fraud Detection

Dalam kasus deteksi penipuan, *Process Mining* memberikan kontribusi beberapa keuntungan. Pemeriksaan kesesuaian (*Conformance Checking*) dapat digunakan untuk membandingkan data aktual untuk model standar. Manfaat lain yang dihasilkan dari penyelidikan menggunakan penggalian proses adalah analisis kontrol aliran. Hal ini bisa mendeteksi aktivitas yang dilewati, aktivitas yang dimasukkan, dan urutan yang salah. Dalam konteks deteksi penipuan, bagian menyimpang dipandang sebagai tipuan yang mencurigakan [3].

Penggalian proses dapat membantu dalam mendeteksi kecurangan dengan melakukan empat analisis, yaitu analisis *control flow*, analisis *role resource*, analisis *throughput time*, dan analisis *decision point* [3].

1. Analisis Control Flow

Analisis *control flow* dapat dilakukan dengan analisis manual atau dengan bantuan dari beberapa *plugin*. Analisis ini penting untuk dapat mendeteksi kecurangan dalam bentuk *skip activity* dan *wrong pattern*. Secara manual, analisis dilakukan dengan menemukan *event log* menggunakan algoritma pencarian proses. Algoritma *Fuzzy Miner* merupakan algoritma yang dianjurkan untuk

proses pencarian ini karena membandingkan model *fuzzy* dengan SOP. Akan tetapi, algoritma ini terdapat keterbatasan yaitu tergantung pada penentuan nilai *threshold* [3]. Selain cara manual, analisis dapat dilakukan dengan menggunakan *plugin Conformance Checking* yang terdapat pada ProM untuk menghasilkan nilai *fitness*, *precision*, dan nilai struktur antara *event log* dan model proses. *Fitness* menunjukkan kesamaan perilaku *event log* dan model proses. Struktur mengevaluasi jika model proses menjelaskan struktur proses yang cocok. Dan *precision* mengevaluasi seberapa tepatnya model menjelaskan proses yang diamati. Nilai-nilai tersebut dapat digunakan untuk mengukur kesamaan antara proses yang berjalan dengan SOP.

Tujuan dari analisis *control flow* adalah mengukur kesamaan dan perbedaan antara *event log* dan model proses standar. Bagian-bagian yang berbeda dari model proses dapat dicurigai sebagai kecurangan dalam proses tersebut. Hasilnya adalah nilai *fitness* menunjukkan penyimpangan perilaku proses berjalan.

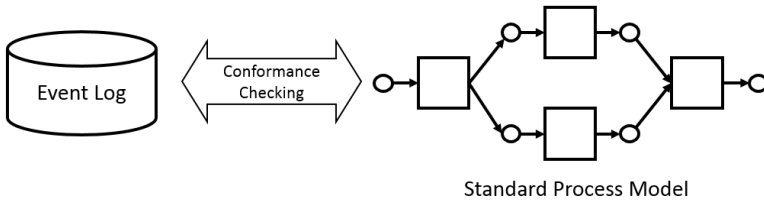
2. Analisis *Role Resource*

Analisis ini dilakukan dengan menggunakan *social network miner* yang terdapat di dalam ProM untuk mendapatkan *role-role* yang terdapat di dalam model proses. Kemudian *role-role* untuk tiap *event* yang dijalankan oleh suatu proses dibandingkan dengan *role-role* yang terdapat di dalam SOP untuk dapat menemukan kemungkinan terjadinya kecurangan dalam sisi *resource/originator* nya [3]. Selain itu, analisis ini juga dapat dilakukan dengan melakukan *filtering* pada *event log* yaitu dengan mengambil atribut *resource* dan *role* dalam *event log*, kemudian dicocokkan dengan atribut *resource/role* dalam model SOP.

3. *Analisis Throughput Time*
Analisis ini dilakukan dengan menghitung interval waktu antar aktivitas. Interval ini dihitung dari *start time stamp* dan *completed time stamp*. Kedua data tersebut dapat diperoleh dari *event log* yang direkam dari proses. Kemudian, waktu pengerjaan aktivitas tersebut dibandingkan dengan SOP proses bisnis tersebut dari sisi waktu pengerjaannya. Contohnya untuk tiap aktivitas dibatasi waktu pengerjaannya adalah 10 menit. Maka tiap aktivitas dari suatu *case* yang melanggar SOP tersebut dapat dicurigai sebagai sebuah kecurangan.
4. *Analisis Decision Point*
Analisis ini dilakukan untuk dapat mengetahui adanya *case* yang spesifik sebagai akibat dilakukannya suatu *decision* atau keputusan dalam proses. Untuk mendeteksi adanya kecurangan, dilakukan dengan membuat basis data relasional untuk dapat melakukan kueri terhadap *case* yang spesifik tersebut. Sehingga dapat dibandingkan dengan aturan yang terdapat di dalam SOP untuk penentuan ditariknya suatu keputusan dalam proses.

2.2. Conformance Checking

Salah satu analisis spesifik pada *process mining* adalah *conformance checking* [15]. Asumsi untuk mampu melakukan pengecekan kesesuaian adalah adanya model proses yang mewakili proses yang berjalan [16]. Dalam hal audit proses, pengecekan kesesuaian harus dilakukan setelah menemukan model proses yang sebenarnya. Terdapat beberapa teknik untuk mengukur kesesuaian model seperti algoritma A*, *Cost-Based Fitness Analysis*, dll. Tujuannya adalah untuk mengukur penyimpangan perilaku antara *event log* dengan model proses sehingga kesamaan dan perbedaan bisa diselidiki. Gambar 2.5 menunjukkan ilustrasi dari tujuan utama pengecekan kesesuaian pada *process mining*.



Gambar 2.5. Conformance Checking in Process Mining

Terdapat tiga istilah yang umum pada *conformance checking* yaitu: *Play-In*, *Play-Out*, dan *Replay*. *Play-In* berarti perilaku proses yang mempengaruhi model proses. *Play-Out* merupakan kebalikan dari *Play-In* yaitu model proses yang mempengaruhi perilaku proses. Dan *Replay* merupakan kombinasinya di mana menggunakan *event log* dan model proses sebagai masukan. *Replay* mampu mendeteksi adanya masalah dalam kesesuaian proses karena mempertimbangkan dua aspek yaitu model dan kontrol. Hal ini dapat meningkatkan kesesuaian [6]. Kasus yang dapat dilakukan *replay* disebut *conform/sesuai*, sedangkan kasus yang tidak disebut menyimpang. Terdapat tiga metode analisis untuk menghitung metric kesesuaian yaitu *Log Replay*, *State Space Analysis*, dan *Structural Analysis*.

Dalam aplikasi ProM terdapat plugin *Conformance Checker* yang dapat menghitung dan menganalisis kesesuaian suatu proses berjalan dan model prosesnya. Hasil analisis terbagi menjadi tiga yaitu *Fitness Analysis*, *Behavioral Appropriateness*, dan *Structural Appropriateness*. *Fitness Analysis* akan menghasilkan informasi-informasi seperti *Token Counter* yang menampilkan token hilang atau yang tersisa saat *log replay* untuk setiap *place*, *Failed Tasks* yang menampilkan transisi yang tidak berhasil dijalankan dalam *log replay*, *Remaining Tasks* yang menampilkan transisi yang tetap diaktifkan setelah *log replay* yang menunjukkan penyelesaian yang tidak tepat dari proses karena tugas tersebut harus telah dikerjakan, *Path Coverage* yang menampilkan transisi yang dikerjakan

sepanjang *log replay*, serta *Passed Edges* yang menunjukkan seberapa sering tiap *edge* diikuti selama *log replay*. *Behavioural Appropriateness* menghasilkan informasi besarnya presisi suatu model proses yang dapat menjalankan beberapa perilaku yang tercatat di dalam *log*. *Structural Appropriateness* menghasilkan informasi seperti *Redundant Invisible Tasks* yang menampilkan tugas-tugas duplikat yang tidak pernah digunakan bersama-sama di salah satu jalur model yang mungkin, serta *Alternative Duplicate Task* yang menampilkan tugas-tugas tak terlihat yang tidak mempengaruhi perilaku proses sehingga tugas-tugas ini dapat dihapus dari model.

2.3. *Fuzzy Sets Multiple Attribute Decision Making*

Teori himpunan *fuzzy* dikenalkan oleh Zadeh pada tahun 1965 sebagai teori matematika yang digunakan untuk membaurkan konsep [17]. Himpunan *fuzzy* dapat didefinisikan sebagai sebuah kelas dengan batas-batas *fuzzy*. Jika X didefinisikan sebagai alam semesta anggota $X = \{x_1, x_2, \dots, x_n\}$, dan Y himpunan *fuzzy* dari X , maka:

$$Y = \left\{ \left(x_1, f_y(x_1) \right), \left(x_2, f_y(x_2) \right), \left(x_n, f_y(x_n) \right) \right\} \quad (2.1)$$

Pada Persamaan 2.1 f_y merupakan derajat keanggotaan dari x_1 di Y . Himpunan *fuzzy* merupakan himpunan umum yang memungkinkan untuk adanya tingkat keanggotaan untuk elemen-elemennya [18]. Biasanya interval satuan riil $[0; 1]$ dipilih sebagai struktur derajat keanggotaan [18]. Diberikan X sebagai himpunan global. μ_A merupakan fungsi keanggotaan yang mendefinisikan himpunan A [19].

Pengambilan keputusan adalah kegiatan manusia. Pada dasarnya melibatkan pemilihan alternatif yang paling disukai dari sebuah himpunan alternatif berhingga untuk mencapai tujuan tertentu yang telah ditetapkan [20]. Seorang pembuat keputusan sering menghadapi masalah dalam mengidentifikasi solusi dari

sebuah himpunan alternatif berhingga. *Multiple attribute decision making* (MADM) dapat digunakan untuk memilih sebuah alternatif dari sekumpulan alternatif yang ditandai dengan beberapa atribut.

Akan tetapi, metode MADM ini memiliki kelemahan dalam mengambil keputusan dengan informasi yang tidak tepat dan linguistik. Sehingga diperlukan metode *Fuzzy multiple attribute decision making* (FMADM). FMADM pada dasarnya melibatkan dua tahap sebelum pengambilan keputusan yaitu agregasi dan eksploitasi [21]. Tahap agregasi menggabungkan peringkat kinerja untuk semua atribut yang berkaitan dengan setiap alternatif. Tahap eksploitasi memberi peringkat kepada alternatif sehubungan dengan peringkat kinerja agregat yang global. Dalam lingkungan *fuzzy*, dalam pencarian keputusan dengan beberapa atribut, mempertimbangkan beberapa elemen berikut: $A = \{ A_1, A_2, \dots, A_m \}$ meliputi himpunan alternatif berhingga, dan himpunan atribut berhingga $C = \{ C_1, C_2, \dots, C_k \}$, di mana atribut-atribut ini diklasifikasikan menjadi atribut subjektif $\{ C_1, C_2, \dots, C_s \}$ dan atribut objektif $\{ C_{s+1}, C_{s+2}, \dots, C_k \}$. Sebuah prosedur umum untuk pengambilan keputusan dengan analisis beberapa atribut *fuzzy* memerlukan tiga tahapan utama [22], yaitu:

1. Pendapat atau evaluasi dari para ahli harus disatukan/dikumpulkan.
2. Kumpulan pendapat tersebut diagregasikan untuk membentuk opini kolektif untuk setiap alternatif. Opini ini biasanya berupa angka *fuzzy* atau label linguistik yang digunakan untuk alternatif.
3. Memilih alternatif yang lebih baik berdasarkan urutan peringkat mereka.

Dalam tahap kedua, untuk membentuk agregasi dari opini berupa angka *fuzzy*, perlu dilakukan pengukuran peringkat atau bobot untuk tiap alternatif dengan menggunakan Persamaan 2.2:

$$S = \frac{\sum_{i=1}^n x_i \mu(x_i)}{\sum_{i=1}^n \mu(x_i)} \quad (2.2)$$

Di mana, x_i menunjukkan peringkat preferensi atribut ke- i dan $\mu(x_i)$ adalah fungsi keanggotaan yang sesuai.

2.4. Association Rule Learning

Association Rule (aturan asosiasi) merupakan salah satu metode di dalam *Data Mining* yang menggambarkan keterkaitan dari sebuah kumpulan dari hubungan-hubungan nilai atribut, yang biasa dipanggil *items*. *Association Rule Learning* (ARL) merupakan asosiasi antara dua atau lebih *item*/elemen/tugas di dalam sebuah *database*. *Market Basket Analysis* (MBA) merupakan aplikasi terbesar untuk algoritma yang menemukan aturan asosiasi ini. Hal ini merupakan teknik pemodelan yang berdasar pada teori di mana jika seorang membeli kumpulan item tertentu, maka dia juga akan cenderung membeli kumpulan item lainnya [19]. Contoh aturan asosiasi adalah sebagai berikut :

Bread, Milk => Butter | 90%

Item yang berada pada sebelah kiri dari aturan asosiasi di atas disebut *antecedent* (yang mendahului) dan *item* yang berada pada sebelah kanan disebut *consequents*. Sebuah aturan asosiasi dapat memiliki banyak *antecedent* dan *consequent*. Bilangan 90% di atas mengindikasikan bahwa 90% pembeli yang membeli roti dan susu juga membeli mentega. Persentase ini mengindikasikan kepastian (*confidence*) dari aturan asosiasi ini.

Kepastian/*Confidence* merupakan satu dari unsur pengukuran tingkat ketertarikan dari sebuah aturan asosiasi. Pengukuran lainnya adalah *support*. *Support* mengindikasikan kegunaan dari sebuah aturan asosiasi. Sebagai contoh, jika aturan di atas memiliki nilai *support* 5%, hal itu berarti 5% dari semua transaksi yang telah dianalisis menunjukkan bahwa roti, susu dan mentega dibeli secara bersamaan. Ketika aturan asosiasi ini diaplikasikan terhadap *event log*, akan diambil asosiasi dan pola yang sering ada di antara berbagai *event* di dalam *event log*.

Aturan asosiasi secara formal didefinisikan sebagai pernyataan $X \Rightarrow Y$, di mana X dan Y adalah *itemset*, dan Y bukan merupakan *itemset* kosong. X dan Y adalah kumpulan item dari

data transaksi. Aturan asosiasi menyarankan sebuah relasi kejadian yang kuat antara item pada *antecedent* dan *consequent* pada aturannya.

Aturan asosiasi memerlukan kumpulan *item* yang sering muncul di *database* atau catatan transaksi. Hal ini menandakan bahwa hanya *item* dengan nilai *support* yang tinggi yang digunakan. Nilai *support* merupakan jumlah atau banyaknya suatu *item* muncul di dalam catatan transaksi. Di dalam aturan asosiasi, terdapat penentuan nilai ambang batas minimum untuk nilai *support*. Jika sebuah kumpulan *item* melebihi ambang batas minimum *support* tersebut, maka kumpulan *item* tersebut dapat diarahkan sebagai frekuensi *itemset*. Nilai *support* dapat dihitung dengan menggunakan Persamaan 2.3:

$$Support(X,Y) = \left(\frac{\text{Jumlah Transaksi Mengandung } X \text{ dan } Y}{\text{Jumlah Transaksi}} \right) \quad (2.3)$$

Confidence dari sebuah aturan asosiasi $X \Rightarrow Y$ adalah probabilitas untuk menemukan Y pada kumpulan transaksi. Dengan kata lain, *confidence* mengindikasikan sebagaimana sering munculnya *item* di Y pada transaksi yang mengandung X . Nilai *confidence* dapat dihitung dengan menggunakan Persamaan 2.4:

$$Confidence(X,Y) = \left(\frac{\text{Jumlah Transaksi Mengandung } X \text{ dan } Y}{\text{Jumlah Transaksi Mengandung } X} \right) \quad (2.4)$$

Selain itu terdapat istilah *certainty factor* (faktor kepastian) dari sebuah *rule* yang dihasilkan yang dapat dihitung dengan menggunakan nilai *support* dan *confidence*. Cara menghitung *certainty factor* (CF) adalah sebagai berikut:

- Jika $Conf(A \Rightarrow C) > supp(C)$ maka dapat menggunakan Persamaan 2.5.

$$CF(A \Rightarrow C) = \frac{Conf(A \Rightarrow C) - supp(C)}{1 - supp(C)} \quad (2.5)$$

- Jika $Conf(A \Rightarrow C) \leq supp(C)$ maka dapat menggunakan Persamaan 2.6.

$$CF(A \Rightarrow C) = \frac{\text{Conf}(A \Rightarrow C) - \text{supp}(C)}{\text{supp}(C)} \quad (2.6)$$

- Jika $\text{supp}(C) = 1$, maka nilai $CF(A \Rightarrow C) = 1$
- Jika $\text{supp}(C) = 0$, maka nilai $CF(A \Rightarrow C) = -1$

Suatu aturan asosiasi dikatakan kuat jika nilai faktor kepastian dan nilai *support* lebih besar dari nilai *threshold* minCF dan minSupp yang dimasukkan pengguna.

Terdapat beberapa algoritma untuk pencarian aturan asosiasi seperti *Apriori*, *Apriori-Tid*, *PredictiveApriori*, *Tertius*, dan sebagainya [19]. Semua algoritma tersebut meliputi dua tahap :

1. Cari semua frekuensi *itemset*.
Merupakan *itemset* yang paling tidak muncul sebanyak batas *support* minimum yang telah ditentukan.
2. Hasilkan aturan asosiasi yang kuat dari frekuensi *itemset*.
Pada tahap ini, aturan asosiasi yang kuat berasal dari frekuensi *itemset* yang dihasilkan pada tahap pertama. Aturan yang kuat adalah aturan yang memenuhi batas ambang minimum *support* dan *confidence*.

2.4.1. Algoritma *Apriori*

Algoritma *apriori* menggunakan pencarian berdasarkan level untuk menghasilkan frekuensi *itemset* yang melintas dari frekuensi 1-*itemset* hingga ke ukuran maksimum dari frekuensi *itemset*. Pencarian *iterative* ini berlanjut hingga tidak terdapat frekuensi *itemset* baru yang dihasilkan. Langkah-langkahnya adalah sebagai berikut:

1. Untuk batas ambang nilai *support* s , *item-item* yang lolos nilai minimum *support* dinotasikan dengan $L1$.
2. Pasangan dari $L1$ adalah kandidat untuk *itemset* ukuran dua. Kandidat ini dinotasikan dengan $C2$. Pasangan frekuensi *itemset* yang memenuhi nilai *support* s menjadi frekuensi *itemset* ukuran dua dan dinotasikan dengan $L2$.

3. Lakukan langkah kedua berikut sampai tidak ada frekuensi *itemset* yang dihasilkan.
4. Setiap kali frekuensi *itemset* diperoleh, untuk tiap frekuensi *itemset* 1, hasilkan semua subset yang tidak kosong dari 1.
5. Untuk setiap subset s dari 1 yang tidak kosong, didapatkan aturan " $s \Rightarrow (1-s)$ " jika perbandingan dari $support(1)$ ke $support(s)$ lebih besar dari atau sama dengan nilai ambang batas minimum *confidence*.

2.5. Fuzzy Association Rule Learning

Metode ini sama halnya dengan metode ARL yaitu untuk mencari aturan asosiasi dari kumpulan data. Yang menjadi perbedaan adalah jenis data yang diolah. Dalam ARL, data yang digunakan merupakan data *crisp* atau data biner yaitu 1 dan 0. 1 menandakan suatu *item* terdapat di dalam data dan 0 menandakan *item* tersebut tidak terdapat di dalam data. Sedangkan *Fuzzy* ARL menggunakan data *fuzzy* yang nilainya antara 0 sampai 1. Nilai yang digunakan merupakan derajat keanggotaan suatu data dalam suatu kelas yang menggunakan konsep keanggotaan pada teori himpunan *fuzzy*. Sama seperti ARL, data yang diolah dalam *Fuzzy* ARL ini berupa transaksi *fuzzy* yang merupakan sub himpunan *fuzzy* yang tidak kosong. Dan juga terdapat notasi yang menjelaskan derajat keanggotaan dari tiap *item* yang terdapat di dalam transaksi. Selain itu pada *fuzzy* ARL terdapat derajat inklusi dari *itemset*. Nilai derajat inklusi ini yang membedakan ARL dan *fuzzy* ARL [23].

Data-data transaksi tersebut direpresentasikan dalam sebuah tabel. Tabel tersebut terdiri dari baris dan kolom yang mewakili *item* dan transaksi. Tabel tersebut diisi dengan nilai derajat keanggotaan dari masing-masing *item* yang berupa nilai antara 0 sampai 1. Dalam *fuzzy* ARL terdapat perbedaan dalam perhitungan nilai *support*. Contohnya *support* untuk $A \Rightarrow C$ dalam himpunan transaksi *fuzzy* adalah $supp(A \cup C)$ [23]. Misalkan nilai $A = 0,5$ dan nilai $C = 0,7$, maka nilai *support* nya adalah $\min(0,5; 0,7) = 0,5$.

Sedangkan untuk perhitungan nilai *confidence* sama dengan konsep ARL.

Terdapat tiga pendekatan yang dapat digunakan untuk melakukan *fuzzy association rule* ini yaitu:

- Pendekatan kuantitatif
- Struktur Taksonomi *Fuzzy*
- Pendekatan *Item Set* perkiraan

Untuk menemukan aturan asosiasi pada *fuzzy ARL*, terdapat beberapa algoritma untuk menyelesaikan. Berikut merupakan beberapa contoh algoritma yang dapat digunakan untuk menghasilkan aturan asosiasi:

1. Algoritma untuk menggali *fuzzy association rule* pada basis data kuantitatif telah diusulkan yaitu F-APACS, yang mengerjakan istilah linguistik untuk menggambarkan keteraturan dan pengecualian tersembunyi daripada memisahkan atribut kuantitatif ke dalam himpunan *fuzzy* [24].
2. Algoritma lainnya seperti *Apriori* juga dapat digunakan sebagai dasar tetapi menggabungkan himpunan *fuzzy* untuk menggali nilai kuantitatif dari sebuah *database*. Algoritma ini pertama mengubah tiap atribut kuantitatif menjadi himpunan *fuzzy* dan memetakan *item-item* tersebut melalui fungsi keanggotaan. Kemudian algoritma *apriori* menghasilkan aturan asosiasi menggunakan jumlah *fuzzy* yang dikumpulkan sebelumnya [25].
3. Algoritma lain dengan pendekatan *Apriori* yang membagi proses penggalian menjadi 2 tahap yaitu pencarian frekuensi *itemset* dan pembentukan aturan asosiasi dari basis data kuantitatif [26].

2.6. ProM

ProM merupakan kerangka kerja umum yang *open-source* untuk mengimplementasikan penggalian proses. Keunggulan dari ProM adalah memiliki pendekatan analisis yang sangat luas, karena memiliki lebih dari 250 *plugins*, mulai dari *Process Discovery*, *Social Network*, *Conformance Checking*. Selain itu,

ProM juga mampu untuk melakukan konversi berbagai macam model, mulai dari model *Petri Net*, *EPC*, *BPMN*, *BPEL*, *YAWL*. Aplikasi ini juga mampu untuk membaca *XML* atau *MXML* berupa *log* format. ProM dikembangkan oleh sejumlah dosen dan profesor dari Universitas Tue Netherland [27].

2.7. Plugin ProM

Sebuah *plugin* (atau *plugin* ekstensi) adalah komponen perangkat lunak yang menambahkan fitur khusus untuk aplikasi perangkat lunak yang ada. Bila aplikasi yang mendukung *plugin*, memungkinkan kustomisasi. Dalam tugas akhir ini, *plugin* yang dibuat khusus untuk *software process mining* bernama ProM [27]. *Plugin* ProM ini muncul dalam pilihan ketika melakukan *process mining* menggunakan ProM.

2.8. Event log

Untuk dapat menerapkan teknik penggalian proses, hal yang penting untuk dilakukan yaitu mengekstrak *event log* dari sumber data seperti *database*, *log* transaksi, jejak audit, dll. Dalam aplikasi ProM, format *event log* yang dapat didukung yaitu *MXML* (*Mining eXtensible Markup Language*) dan *XES* (*eXtensible Event Stream*). Dan format *XES* telah dipilih sebagai format standar untuk pencatatan *event log* oleh *IEEE Task Force on Process Mining*. *XES* memiliki definisi standar yang dibuat oleh Christian W. Günther and Eric Verbeek dari Universitas Tue Netherland.

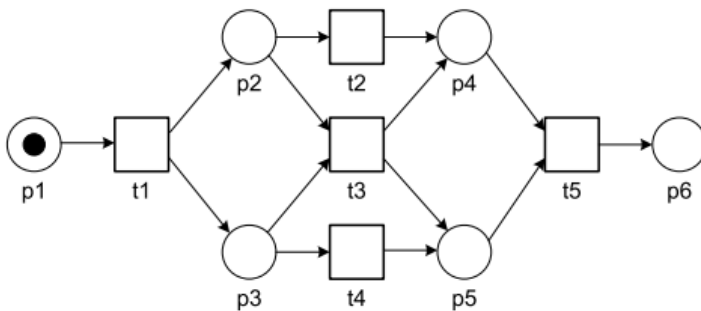
Dalam tugas akhir ini, terdapat dua macam *event log* yang digunakan untuk menggambarkan proses yang berjalan yaitu *event log training* yang digunakan sebagai data *training* dan *event log testing* yang digunakan sebagai data *testing*.

2.9. Petri Net

Seperti yang dijelaskan pada [28], ditunjukkan terminologi dan notasi yang digunakan untuk tugas akhir ini. Seperti yang

terlihat pada Gambar 2.6, *Petri Net* klasik ialah suatu *triple* (P, T, F) dimana P adalah set bilangan terbatas dari *place*. T adalah set bilangan terbatas dari transisi ($P \cap T = \emptyset$) dan $F \subseteq (P \times T) \cup (T \times P)$ adalah set dari *arcs* (relasi *flow*).

Dalam tugas akhir ini, konsep *petri net* akan digunakan untuk menggambarkan SOP dalam bentuk model proses dalam ekstensi PNML (Petri Net Markup Language). Di dalam *petri net* SOP itu terdapat *arc*, *place*, dan transisi sebagai aktivitas, serta terdapat informasi tambahan seperti standar waktu, *role*, dan *resource* yang menjalankan satu aktivitas. Selain itu, juga terdapat informasi tambahan untuk mendefinisikan aturan *decision* yang terdapat pada model.

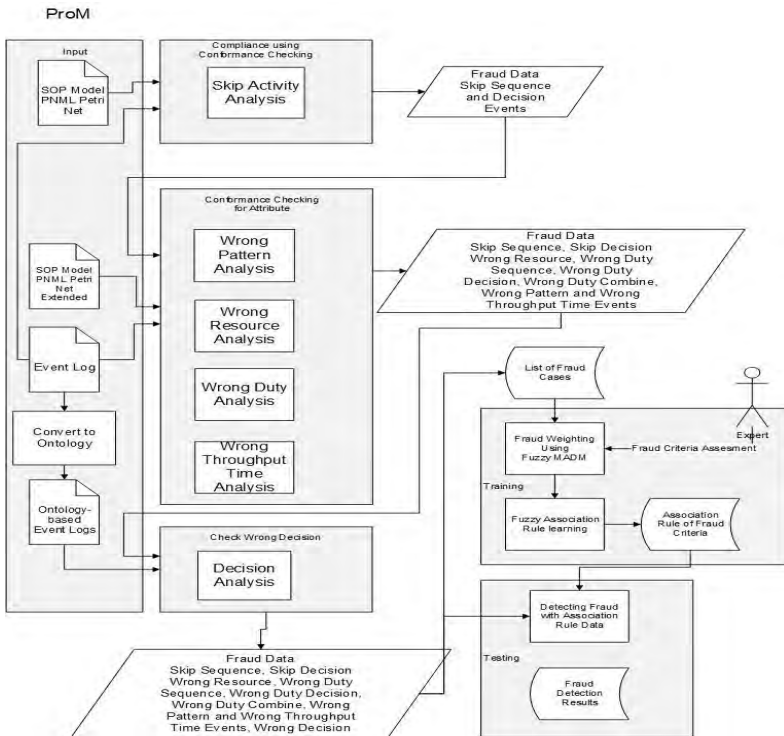


Gambar 2.6. *Petri Net* dasar dengan *places* (p1,p2,p3,p4,p5,p6) dan *transisi* (t1,t2,t3,t4,t5).

[Halaman ini sengaja dikosongkan]

BAB III METODOLOGI PEMECAHAN MASALAH

Pada bab ini menjelaskan mengenai teknik pengecekan kesesuaian dengan menggunakan ProM untuk menghasilkan nilai / jumlah pelanggaran untuk tiap atribut *fraud* untuk data *training* maupun *testing*, mencari aturan asosiasi dengan menggunakan pendekatan *fuzzy ARL*, mendeteksi *fraud* suatu proses dengan aturan asosiasi yang terbentuk, sampai melakukan analisis bobot *fraud* yang terjadi dan mengkategorikannya ke dalam kelas *fraud* yang sesuai agar dapat mengatasi permasalahan yang dijelaskan pada bab I. Metodologi yang digunakan dapat dilihat pada Gambar 3.1.



Gambar 3.1. Metodologi

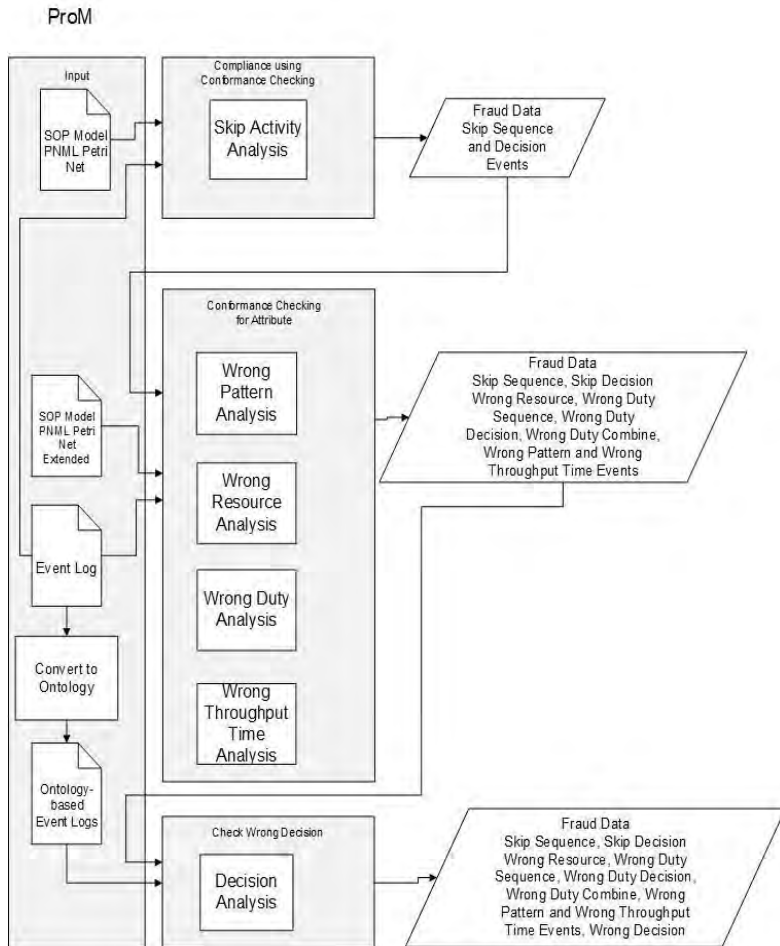
Gambar 3.1 menunjukkan dalam metodologi yang dilakukan memerlukan beberapa data masukan berupa *event logs* dan model proses bisnis standar sesuai dengan SOP yang dibentuk. *Event logs* yang digunakan terdiri dari dua macam yaitu *event logs* standar (berekstensi *xes*) dan *event logs ontology* (berekstensi *owl*). Selanjutnya, *event logs* dan model proses bisnis standar yang dimasukkan akan dilakukan beberapa tahapan proses mulai dari proses pengecekan kesesuaian yang terdiri dari tiga pengecekan yaitu *check compliance using conformance checking*, *conformance checking for atribut*, *check wrong decision*. Kemudian dilanjutkan dengan proses pembobotan *fraud* menggunakan metode *fuzzy MADM*, *training data fraud* dengan menggunakan metode *fuzzy ARL*, dan *testing* dengan menggunakan data hasil *training*.

Dalam tugas akhir ini digunakan dua studi kasus yaitu aplikasi kredit bank dan proses *procurement* pada sistem ERP. Untuk tiap studi kasus memiliki data uji yang terdiri dari data *training* dan data *testing*. Untuk studi kasus aplikasi kredit disediakan 130 *case data training* yang terdiri dari 100 *case fraud* dan 30 *case normal* serta 100 *case data testing* yang terdiri dari 30 *case normal* dan 70 *case fraud*. Untuk studi kasus *procurement* ERP disediakan 100 *case data training* yang terdiri dari 70 *case fraud* dan 30 *case normal* serta 100 *case data testing* yang terdiri dari 60 *case normal* dan 40 *case fraud*. Dari dua studi kasus tersebut dihitung akurasi dari metode yang digunakan untuk pendeteksian *fraud* pada tugas akhir ini.

3.1. Conformance Checking

Conformance Checking atau pengecekan kesesuaian merupakan salah satu analisis spesifik dalam *process mining*. Tujuannya adalah untuk mengukur penyimpangan perilaku antara *event log* dengan model proses bisnis standar yang telah didefinisikan di dalam SOP sehingga kesamaan dan perbedaan bisa diselidiki. Hasil dari pengecekan kesesuaian ini berupa jumlah pelanggaran sesuai dengan jenis pelanggaran yang dilakukan dan dimasukkan sebagai nilai untuk atribut dalam tabel PBF.

Dalam tugas akhir ini, terdapat tiga macam pengecekan kesesuaian yang dapat dilakukan yaitu *Check Compliance using Conformance Checking*, *Conformance Checking for Attribute*, dan *Check Wrong Decision*. Untuk alur proses pengecekan kesesuaian ini dapat dilihat pada Gambar 3.2.



Gambar 3.2. Alur Proses Pengecekan Kesesuaian

3.1.1. *Check Compliance using Conformance Checking*

Pengecekan kesesuaian ini merupakan pengecekan kesesuaian antara proses yang berjalan yang dicatat dalam suatu *event log* dan model proses bisnis standar dalam SOP. Pengecekan ini menggunakan konsep *Replay* yang menggunakan *event log* dan model proses sebagai masukan. *Replay* mampu mendeteksi adanya masalah dalam kesesuaian proses karena mempertimbangkan dua aspek yaitu model dan kontrol. Hal ini dapat meningkatkan kesesuaian [6]. Kasus yang dapat dilakukan *replay* disebut *conform*/sesuai, sedangkan kasus yang tidak disebut menyimpang. Pengecekan kesesuaian ini dapat dilakukan dengan menjalankan *plugin Compliance Check Using Conformance Cheking* yang telah terdapat di dalam ProM. Pengecekan ini dilakukan untuk menangkap pelanggaran berupa *skipped activity* sehingga dari pengecekan ini dihasilkan data *fraud* berupa jumlah pelanggaran untuk *skip sequence* dan *skip decision*. Nilai *skip sequence* diisi dengan jumlah kejadian *skip* pada *event sequence* dan nilai *skip decision* diisi dengan jumlah kejadian *skip* pada *event decision*. Hal ini dikarenakan dalam pengecekan ini, dapat diidentifikasi aktivitas-aktivitas yang pada model proses harus dijalankan, akan tetapi dalam *event log* tidak dijalankan atau dilewati. Hal ini disebut sebagai penyimpangan dalam proses. Pelanggaran berupa *skipped activity* ini juga akan mempengaruhi jumlah pelanggaran pada atribut *wrong pattern* dikarenakan tiap terjadi *skip* pada suatu aktivitas akan mengakibatkan terjadinya kesalahan pola proses yang tidak sesuai dengan SOP.

3.1.2. *Conformance Checking for Attribute*

Pengecekan kesesuaian ini dilakukan untuk memeriksa kesesuaian antara atribut yang terdapat pada *event log* dan atribut yang terdapat dalam model proses SOP. Pengecekan ini memerlukan dua masukan berupa *event log* dan model proses SOP dalam bentuk *PNML Extended* yang telah dimodifikasi untuk

mengambil informasi mengenai atribut yang terdapat pada tiap aktivitas. Atribut tersebut dapat berupa waktu pengerjaan suatu aktivitas, *role* dan *resource* yang menjalankan aktivitas, serta urutan aktivitas dan hubungan antar aktivitas. Pengecekan ini menghasilkan jumlah pelanggaran yang terjadi untuk atribut / jenis pelanggaran *throughput time min/max*, *wrong resource*, *wrong duty sequence/decision/combine*, dan *wrong pattern* yang terdapat dalam tabel PBF. Nilai untuk *throughput time min* diisi dengan jumlah *event* yang waktu pelaksanaannya lebih kecil dari waktu yang ditetapkan dalam SOP. Nilai untuk *throughput time max* diisi dengan jumlah *event* yang waktu pelaksanaannya lebih besar dari waktu yang ditetapkan dalam SOP. Nilai untuk *wrong resource* diisi dengan jumlah *event* yang dieksekusi oleh orang yang tidak terotorisasi. Nilai *wrong duty sequence* diisi dengan jumlah *event* yang melanggar aturan pemisahan tugas yang terjadi pada *event sequence*. Nilai *wrong duty decision* diisi dengan jumlah *event* yang melanggar aturan pemisahan tugas yang terjadi pada *event decision*. Nilai *wrong duty combine* diisi dengan jumlah *event* yang melanggar aturan pemisahan tugas yang terjadi pada *event sequence* dan *event decision*. Dan *wrong pattern* diisi dengan jumlah *event* yang melanggar pola atau alur proses.

3.1.3. Check Wrong Decision

Pengecekan ini dilakukan untuk memeriksa kesesuaian pengambilan keputusan dalam proses dengan aturan pengambilan keputusan yang ditetapkan dalam SOP. Pengecekan ini dilakukan antara *event log* yang terlebih dahulu dimodelkan dalam bentuk *ontology* dan model proses SOP dalam bentuk *PNML Extended* yang dimodifikasi untuk menyimpan atribut yang diperlukan untuk pengecekan ini.

Event log dimodelkan ke dalam bentuk *ontology* untuk dapat mendefinisikan hubungan yang terjadi antara *event-event* dalam *event log* tersebut. Hubungan tersebut nantinya akan digunakan untuk melakukan proses kueri dengan menggunakan

konsep kueri *ontology* yaitu kueri SPARQL untuk mendapatkan kesesuaian antara pengambilan keputusan yang dilakukan dalam proses dan aturan pengambilan keputusan pada SOP.

Pengecekan ini menghasilkan nilai untuk atribut *wrong decision* pada tabel PBF. Nilai *wrong decision* diisi dengan jumlah kesalahan pengambilan keputusan yang dilakukan pada *event decision*.

3.2. *Fuzzy Set Multi Attribute Decision Making*

Metode ini digunakan untuk menentukan bobot *fraud* dari kumpulan pelanggaran yang terjadi dalam suatu proses. Penentuan bobot *fraud* dilakukan dengan menggunakan penggabungan dua konsep yaitu konsep *fuzzy* dan konsep *multiple attribute decision making* (MADM). MADM dapat digunakan untuk memilih sebuah alternatif dari sekumpulan alternatif yang ditandai dengan beberapa atribut. Akan tetapi, MADM masih memiliki kelemahan dalam menangani informasi yang tidak tepat atau linguistik. Sehingga ditambahkan konsep *fuzzy* untuk membantu menangani informasi yang linguistik.

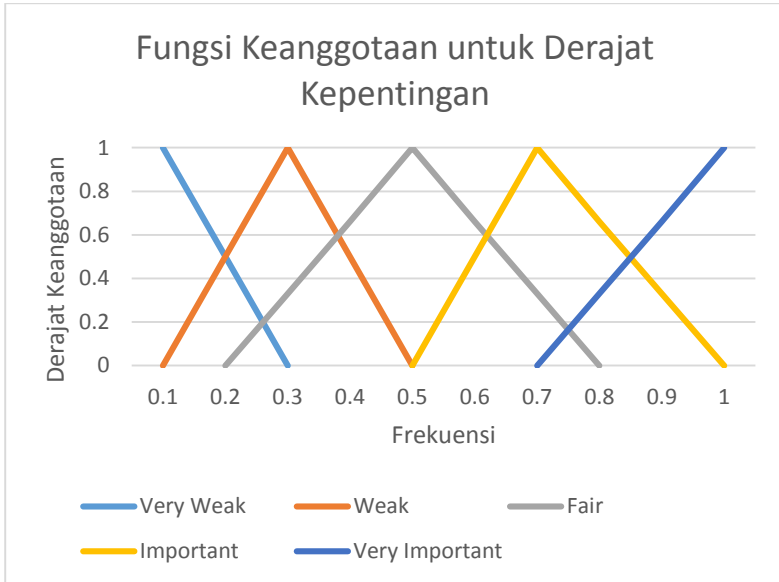
Data yang diperlukan dalam menentukan bobot *fraud* ini ada dua yaitu data penilaian kepentingan atribut-atribut PBF dari para pakar dan data pelanggaran proses yang berasal dari hasil *conformance checking*. Kedua data tersebut kemudian dimodelkan ke bentuk bilangan *fuzzy* berdasarkan tabel kepentingan dan derajat keanggotaannya.

Untuk data penilaian kepentingan dari pakar, tabel kepentingannya dapat dilihat pada Tabel 3.1 dan fungsi keanggotaan kepentingan dapat dilihat pada Gambar 3.3.

Tabel 3.1. Tabel Kepentingan

Tingkat Kepentingan	Vector				Rating
	a	b	c	d	
Very Important	0,7	1	1	1	100% - 70%
Important	0,5	0,7	0,7	1	100% - 50%

Fair	0,2	0,5	0,5	0,8	80% - 20%
Weak	0	0,3	0,3	0,5	0% - 50%
Very Weak	0	0	0	0,3	0% - 30%



Gambar 3.3. Fungsi Keanggotaan kepentingan

Dalam tugas akhir ini, data penilaian kepentingan dari pakar diambil dari penelitian yang dilakukan oleh [13] kepada para pakar audit suatu bank. Data penelitian kepentingan dapat dilihat pada Gambar 3.4.

Penilaian expert 1											
Skip		Throughput Time		Wrong	Wrong duty			Wrong	Wrong	Confor	
sequence	decision	Min	Max	resource	Sequence	Decision	Combine	pattern	Decision	mance	
W	VI	F	F	W	VW	W	W	W	VI	W	
Penilaian expert 2											
Skip		Throughput Time		Wrong	Wrong duty			Wrong	Wrong	Confor	
sequence	decision	Min	Max	resource	Sequence	Decision	Combine	pattern	Decision	mance	
W	VI	I	F	W	VW	W	W	W	VI	W	
Penilaian expert 3											
Skip		Throughput Time		Wrong	Wrong duty			Wrong	Wrong	Confor	
sequence	decision	Min	Max	resource	Sequence	Decision	Combine	pattern	Decision	mance	
W	VI	I	I	W	VW	W	W	W	VI	W	
Penilaian expert 4											
Skip		Throughput Time		Wrong	Wrong duty			Wrong	Wrong	Confor	
sequence	decision	Min	Max	resource	Sequence	Decision	Combine	pattern	Decision	mance	
W	VI	I	F	W	VW	W	W	W	VI	W	

Gambar 3.4. Data Penelitian Kepentingan

Kemudian dari data tersebut dilakukan perhitungan bobot/*weight* untuk tiap kategori/atribut *fraud*. Nilai bobot dibagi ke dalam empat bagian yaitu *lower bound weight*, *middle weight 1*, *middle weight 2*, dan *upper bound weight*. Untuk menghitung keempat nilai bobot untuk tiap kategori tersebut dapat menggunakan Persamaan 3.1, Persamaan 3.2, Persamaan 3.3, dan Persamaan 3.4.

$$\text{Lower Bound} = \frac{\sum_{k=1}^n a_k}{n} = \frac{0+0+0+0}{4} = 0 \quad (3.1)$$

$$\text{Middle Weight 1} = \frac{\sum_{k=1}^n b_k}{n} = \frac{0,3+0,3+0,3+0,3}{4} = 0,3 \quad (3.2)$$

$$\text{Middle Weight 2} = \frac{\sum_{k=1}^n c_k}{n} = \frac{0,3+0,3+0,3+0,3}{4} = 0,3 \quad (3.3)$$

$$\text{Upper Bound} = \frac{\sum_{k=1}^n d_k}{n} = \frac{0,5+0,5+0,5+0,5}{4} = 0,5 \quad (3.4)$$

Di mana untuk keempat persamaan di atas terdapat notasi n yang merupakan jumlah pakar, nilai a , b , c , dan d merupakan nilai vektor a , vektor b , vektor c , dan vektor d pada Tabel 3.1. Untuk hasil perhitungannya dapat dilihat pada Tabel 3.2.

Tabel 3.2. Hasil Perhitungan Bobot Kepentingan

Category	Lower	Middle1	Middle2	Upper
SKIP SEQUENCE	0	0,3	0,3	0,5
SKIP DECISION	0,7	1	1	1
THROUGHPUT TIME MINIMUM	0,425	0,65	0,65	0,95
THROUGHPUT TIME MAXIMUM	0,275	0,55	0,55	0,85
WRONG RESOURCE	0	0,3	0,3	0,5
WRONG DUTY SEQUENCE	0	0	0	0,3
WRONG DUTY DECISION	0	0,3	0,3	0,5
WRONG DUTY COMBINE	0	0,3	0,3	0,5
WRONG PATTERN	0	0,3	0,3	0,5
WRONG DECISION	0,7	1	1	1

Selanjutnya, untuk data pelanggaran proses, tabel derajat pelanggaran dapat dilihat pada Tabel 3.3 dan fungsi keanggotaanya dapat dilihat pada Gambar 3.5 dengan rumus fungsi keanggotaannya dapat dilihat pada Tabel 3.4.

Tabel 3.3. Tabel Derajat Pelanggaran

Level Pelanggaran	a	b	c	d
Very Weak	0	0	0,1	0,2
Between Very Weak & Weak	0	0,1	0,2	0,3
Weak	0,1	0,2	0,3	0,4
Between Weak & Fair	0,2	0,3	0,4	0,5
Fair	0,3	0,4	0,5	0,6
Between Fair & Strong	0,4	0,5	0,6	0,7
Strong	0,5	0,6	0,7	0,8
Between Strong & Very Strong	0,7	0,8	0,9	1
Very Strong	0,8	0,9	1	1



Gambar 3.5. Fungsi Keanggotaan Pelanggaran

Tabel 3.4. Rumus Fungsi Keanggotaan Pelanggaran

Fungsi Keanggotaan VW		Fungsi Keanggotaan BVW&W, W, BW&F, F, BF&S, S, BS&VS		Fungsi Keanggotaan of VS	
Derajat	Kondisi	Derajat	Kondisi	Derajat	Kondisi
1	$a \leq x \leq c$	0	$x \leq a$	0	$x \leq a$
$(d-x)/(d-c)$	$c < x < d$	$(x-a)/(b-a)$	$a < x < b$	$(x-a)/(b-a)$	$a < x < b$
0	$x \geq d$ $x < a$	1	$b \leq x \leq c$	1	$x \geq b$
		$(d-x)/(d-c)$	$c < x < d$		
		0	$x \geq d$		

Pada rumus fungsi keanggotaan yang terlihat pada Tabel 3.4, terdapat nilai x merupakan nilai probabilitas pelanggaran yang diperoleh dari perhitungan dengan Persamaan 3.5.

$$x = \frac{\text{Jumlah Pelanggaran}}{\text{Jumlah Event Maksimal yang dapat dilanggar}} \tag{3.5}$$

Perhitungan probabilitas pelanggaran ini dilakukan untuk setiap atribut pelanggaran. Hasil dari perhitungan ini akan digunakan untuk tahap fuzzifikasi ke kelas keanggotaan pada Tabel 3.3.

Langkah selanjutnya adalah menghitung nilai bobot/*weight* untuk data pelanggaran. Data pelanggaran diambil dari hasil *conformance checking* yang telah dilakukan. Untuk menghitung bobot pelanggaran, digunakan nilai hasil *conformance* tersebut dan nilai kepentingan dari pakar yang disesuaikan dengan nilai *conformance* yang dihasilkan. Sehingga untuk perhitungan bobot pelanggaran menggunakan atribut-atribut yang terdapat pada Tabel 3.5.

Tabel 3.5. Tabel Perhitungan bobot Pelanggaran

Atribut	Pakar 1	Pakar 2	Pakar 3	Pakar 4	CONFORMANCE
SKIP SEQUENCE					
SKIP DECISION					
THROUGHPUT TIME MINIMUM					
THROUGHPUT TIME MAXIMUM					
WRONG RESOURCE					
WRONG DUTY SEQUENCE					
WRONG DUTY DECISION					
WRONG DUTY COMBINE					

WRONG PATTERN					
WRONG DECISION					

Untuk tiap penilaian pakar diisi dengan derajat pelanggaran yang disesuaikan dengan nilai dari pakar tersebut dan nilai *conformance*. Penyesuaiannya adalah dengan mengikuti Persamaan 3.6.

$$P = \frac{\left(\frac{C_a+C_b+C_c+C_d}{4}\right) + \left(\frac{D_a+D_b+D_c+D_d}{4}\right)}{2} \quad (3.6)$$

Pada Persamaan 3.6, notasi $C_a, C_b, C_c,$ dan C_d merupakan nilai vektor a,b,c,dan d pada Tabel 3.3 sesuai dengan nilai fuzzifikasi pelanggaran (kolom *Conformance* pada Tabel 3.5). Selain itu notasi $D_a, D_b, D_c,$ dan D_d merupakan nilai vektor a,b,c,dan d pada Tabel 3.1 sesuai dengan nilai kepentingan yang diberikan pakar. Nilai P tersebut kemudian difuzzifikasi ke dalam kelas pelanggaran sesuai dengan fungsi keanggotaan derajat pelanggaran. Sebagai contoh, hasil *conformance* untuk kategori *Skip Sequence* adalah *Weak*, dan penilaian pakar pertama untuk kategori yang sama adalah *Very Important*. Maka nilai penyesuaiannya adalah sebagai berikut:

$$P = \frac{\left(\frac{(0,1 + 0,2 + 0,3 + 0,4)}{4}\right) + \left(\frac{(0,7 + 1 + 1 + 1)}{4}\right)}{2} = 0,5875$$

Selanjutnya nilai $P = 0,5875$ difuzzifikasi ke dalam kelas pelanggaran sesuai dengan fungsi keanggotaannya menjadi *Between Fair & Strong*.

Setelah nilai pelanggaran telah disesuaikan dengan nilai kepentingan pakar, langkah selanjutnya adalah menghitung bobot/*weight* untuk pelanggaran dengan menggunakan persamaan yang sama pada saat menghitung bobot kepentingan yaitu dengan menggunakan Persamaan 3.1, Persamaan 3.2, Persamaan 3.3, dan Persamaan 3.4 dengan nilai notasi $n = \text{jumlah pakar} + 1$.

Selanjutnya dari hasil perhitungan bobot kepentingan dan pelanggaran itu, dilakukan perhitungan *final rating* untuk

menghitung bobot *lower bound*, *middle1*, *middle 2*, dan *upper bound* dengan menggunakan Persamaan 3.7.

$$Final_{Rating} = \frac{1}{k} x [(S_{C1} x W_{C1}) + \dots + (S_{Cn} x W_{Cn})] \quad (3.7)$$

Di mana, k merupakan jumlah kategori, S merupakan bobot pelanggaran, W merupakan bobot kepentingan, dan C_n merupakan kategori ke- n . Setelah keempat bobot *final rating* telah dihitung, bobot *fraud* suatu *case* merupakan penjumlahan dari keempat bobot *final rating* tersebut. Untuk lebih jelasnya dapat dilihat pada Persamaan 3.8.

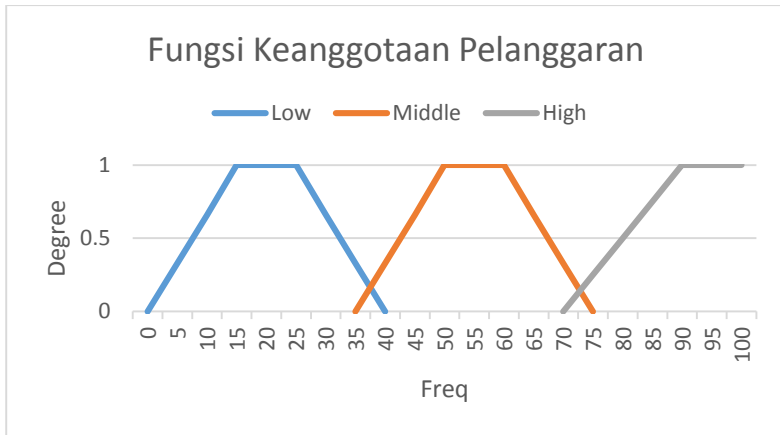
$$Bobot_{Fraud} = Fr(LB) + Fr(M1) + Fr(M2) + Fr(UB) \quad (3.8)$$

Perhitungan bobot *fraud* ini dilakukan untuk tiap proses yang akan diperiksa. Nilai bobot *fraud* ini kemudian akan digunakan untuk proses pencarian aturan asosiasi dengan menggunakan *Fuzzy ARL*.

3.3. *Fuzzy Association Rule Learning*

Fuzzy Association Rule Learning merupakan metode yang digunakan untuk mencari aturan asosiasi antara pelanggaran yang terjadi. Data yang diolah merupakan data pelanggaran-pelanggaran yang telah disertai dengan bobot *fraud* untuk tiap proses. Data pelanggaran yang telah diperoleh dari pengecekan kesesuaian tersebut dihitung probabilitasnya menggunakan Persamaan 3.5. Nilai probabilitas untuk tiap atribut pelanggaran kemudian digunakan untuk menghitung derajat keanggotaannya.

Dalam pencarian aturan asosiasi ini, tiap atribut pelanggaran dibagi ke dalam tiga tipe berdasarkan jumlah pelanggaran yang dilakukan yaitu *Low*, *Middle* dan *High*. Sehingga fungsi keanggotaan yang digunakan untuk proses pencarian aturan asosiasi ini membagi pelanggaran ke dalam tiga tipe tersebut. Contohnya membagi atribut *Skip Sequence* menjadi *Skip Sequence Low*, *Skip Sequence Middle*, dan *Skip Sequence High*. Fungsi keanggotaannya dapat dilihat pada Gambar 3.6 dengan rumus untuk fungsi keanggotaannya dapat dilihat pada Gambar 3.7.



Gambar 3.6. Fungsi Keanggotaan Pelanggaran

Fungsi membership Low			Fungsi membership Middle			Fungsi membership High		
Nilai	Fungsi	Status	Nilai	Fungsi	Status	Nilai	Fungsi	Status
a = 0	0	$x < a$	a = 35	0	$x < a$	a = 70	0	$x < a$
b = 15	$(x-a)/(b-a)$	$a \leq x < b$	b = 50	$(x-a)/(b-a)$	$a \leq x < b$	b = 90	$(x-a)/(b-a)$	$a \leq x < b$
c = 25	1	$b \leq x \leq c$	c = 60	1	$b \leq x \leq c$		1	$x \geq b$
d = 40	$(d-x)/(d-c)$	$c < x < d$	d = 75	$(d-x)/(d-c)$	$c < x < d$			
	0	$x \geq d$		0	$x \geq d$			

Gambar 3.7. Rumus Fungsi Keanggotaan Pelanggaran

Langkah selanjutnya adalah menghitung derajat keanggotaan tiap atribut pelanggaran terhadap tiga tipe/kelas keanggotaan *low*, *middle* dan *high*. Nilai derajat keanggotaan untuk tiap atribut pelanggaran yang telah dibagi menjadi tiga tipe ini digunakan untuk mencari derajat asosiasi dengan metode *fuzzy* ARL.

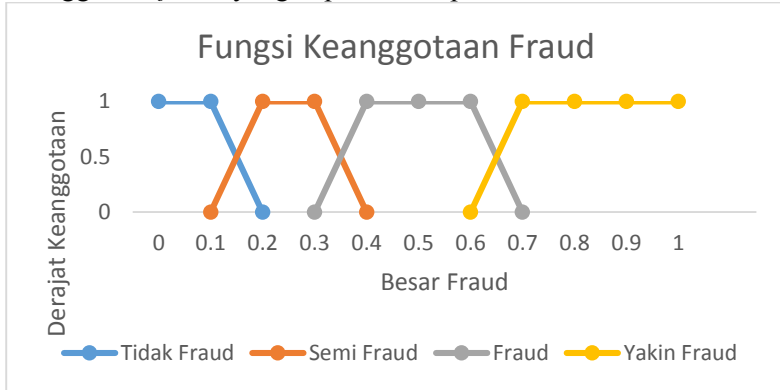
Pencarian aturan asosiasi pada tugas akhir ini menggunakan algoritma *apriori* dengan menggunakan data *fuzzy*. Untuk langkah-langkah algoritma *apriori* telah dijelaskan pada sub-bab 2.4.1. Inti dari algoritma *apriori* adalah menghitung nilai *support* untuk tiap *item*. Kemudian untuk tiap *n-itemset* tentukan nilai ambang batas (*threshold*) yang diperlukan untuk menyeleksi kandidat *item* untuk *itemset* berikutnya. Nilai *threshold* untuk tiap

itemset ditentukan oleh pengguna dengan mempertimbangkan nilai *support* untuk masing-masing *item*. *Item* yang terseleksi adalah *item* yang memiliki nilai *support* yang lebih besar atau sama dengan nilai *threshold* yang ditentukan. *Item* yang lolos seleksi kemudian dikombinasikan dengan *item* lainnya untuk membentuk *itemset* yang baru. Hal ini dilakukan sampai tidak dapat terbentuknya lagi kombinasi antara *item* atau dapat juga dibatasi pembentukan *itemset* sampai *k-itemset*. Pada tugas akhir ini, ditentukan proses pencarian aturan asosiasi sampai pada *5-itemset*. Hal ini dikarenakan dalam tugas akhir ini terdapat 10 atribut pelanggaran, sehingga diambil setengah dari jumlah atribut pelanggaran sebagai batas kombinasi pembentukan *itemset*. *5-itemset* ini berarti untuk satu *item* terdiri dari kombinasi lima *item* dasar.

Aturan asosiasi yang diambil sebagai hasil untuk tugas akhir ini adalah atribut-atribut pelanggaran yang berkombinasi dengan atribut *fraud* untuk tiap *n-itemset*. Sebagai contoh untuk *2-itemset* terdapat aturan asosiasi *SkipSL-Fraud* yang menandakan hubungan pelanggaran *Skip Sequence Low* terhadap *Fraud*. Nilai untuk tiap aturan yang terbentuk merupakan nilai *confidence* untuk aturan tersebut yang dapat dihitung menggunakan Persamaan 2.4. Nilai *confidence* tersebut menunjukkan seberapa besar pengaruh sebuah pelanggaran terhadap *fraud* yang terjadi. Contohnya untuk aturan asosiasi *SkipSL-Fraud*, nilai *confidence*-nya menunjukkan seberapa besar pengaruh pelanggaran *Skip Sequence* dengan intensitas yang *low* (kecil) terhadap *fraud* yang terjadi.

Hasil dari metode ini adalah kumpulan aturan asosiasi yang menggambarkan hubungan antara atribut-atribut pelanggaran dengan *fraud* yang digunakan sebagai data untuk mendeteksi *fraud* pada proses lainnya. Sehingga untuk mendeteksi proses lainnya, cukup dengan mencocokkan atribut pelanggaran yang dilakukan dengan aturan asosiasi yang tersedia. Besarnya pelanggaran merupakan nilai *confidence* dari aturan asosiasi yang sesuai. Apabila suatu proses tidak memiliki kesamaan antara pelanggaran yang terjadi dengan aturan asosiasi, maka dilakukan perhitungan

bobot *fraud* sehingga menghasilkan bobot *fraud* yang digunakan sebagai besar pelanggaran untuk proses tersebut. Besar pelanggaran tersebut kemudian dilakukan fuzzifikasi ke dalam empat macam *fraud* yaitu “Yakin *Fraud*”, “*Fraud*”, “Semi *Fraud*”, dan “Tidak *Fraud*”. Fuzzifikasi yang dilakukan berdasarkan fungsi keanggotaan *fraud* yang dapat dilihat pada Gambar 3.8.



Gambar 3.8. Fungsi Keanggotaan Fraud

Dan untuk rumus fungsi keanggotaan untuk masing-masing kelas dapat dilihat pada Gambar 3.9.

Fungsi membership Tidak Fraud			Fungsi membership Semi Fraud		
Nilai	Fungsi	Status	Nilai	Fungsi	Status
a = 0.1	1	$x \leq a$	a = 0.1	0	$x < a$
b = 0.2	$(b-x)/(b-a)$	$a < x < b$	b = 0.2	$(x-a)/(b-a)$	$a \leq x < b$
	0	$x \geq b$	c = 0.3	1	$b \leq x \leq c$
			d = 0.4	$(d-x)/(d-c)$	$c < x < d$
				0	$x \geq d$
Fungsi membership Fraud			Fungsi membership High		
Nilai	Fungsi	Status	Nilai	Fungsi	Status
a = 0.3	0	$x < a$	a = 0.6	0	$x < a$
b = 0.4	$(x-a)/(b-a)$	$a \leq x < b$	b = 0.7	$(x-a)/(b-a)$	$a \leq x < b$
c = 0.6	1	$b \leq x \leq c$		1	$x \geq b$
d = 0.7	$(d-x)/(d-c)$	$c < x < d$			
	0	$x \geq d$			

Gambar 3.9. Rumus Fungsi Keanggotaan Fraud

BAB IV

ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas tahap analisis permasalahan dan perancangan Tugas Akhir. Analisis permasalahan membahas permasalahan yang diangkat dalam pengerjaan Tugas Akhir. Solusi yang ditawarkan oleh penulis juga dicantumkan pada tahap permasalahan analisis ini. Analisis kebutuhan mencantumkan kebutuhan-kebutuhan yang diperlukan perangkat lunak. Selanjutnya dibahas mengenai perancangan sistem yang dibuat. Pendekatan yang dibuat dalam perancangan ini adalah pendekatan berorientasi objek. Perancangan direpresentasikan dengan diagram UML (*Unified Modelling Language*).

4.1. Analisis

Tahap analisis dibagi menjadi beberapa bagian antara lain cakupan permasalahan, deskripsi umum sistem, kasus penggunaan sistem, dan kebutuhan perangkat lunak.

4.1.1. Cakupan Permasalahan

Permasalahan utama yang diangkat dalam pembuatan Tugas Akhir ini adalah bagaimana cara mendeteksi kecurangan (*fraud*) pada proses bisnis yang berjalan dan mengelompokkannya ke dalam kelas *fraud* yang sesuai berdasarkan tingkat resiko yang dihasilkan. Permasalahan kedua adalah bagaimana mendapatkan data pelanggaran pada proses bisnis yang tercatat melalui *event log*. Permasalahan ketiga adalah bagaimana menghitung bobot *fraud* dengan menggunakan algoritma *Fuzzy Sets Multi Attribute Decision Making*. Permasalahan keempat adalah bagaimana mencari aturan asosiasi dari data pelanggaran dengan menggunakan algoritma *Fuzzy Association Rule Learning*. Permasalahan kelima adalah bagaimana mendeteksi *fraud* suatu proses dengan menggunakan bobot *fraud* dan aturan asosiasi yang dihasilkan. Dan permasalahan yang terakhir adalah bagaimana

memadukan hasil perhitungan dan pendeteksian *fraud* sehingga dapat menjadi sebuah *plugin* pada *ProM*.

4.1.2. Deskripsi Umum Sistem

Plugin ProM ini dibangun untuk mendeteksi kecurangan/*fraud* pada proses bisnis. Proses pendeteksian kecurangan pada proses bisnis ini membutuhkan data proses bisnis yang sudah ada dan berjalan yang dicatat pada *event log* dan standar operasional prosedur dari proses bisnis yang direpresentasikan dalam bentuk model proses berbentuk *Petri Net*. *Event log* yang digunakan dalam sistem ini dibagi ke dalam dua macam yaitu *event log* standar (berekstensi *xes*) dan *event log* yang diubah ke bentuk *ontology*. *Event log* standar digunakan untuk pengecekan kesesuaian untuk mendeteksi pelanggaran pada atribut-atribut pelanggaran kecuali atribut *Wrong Decision*. Dan *event log ontology* digunakan untuk mendeteksi pelanggaran *Wrong Decision* dikarenakan dalam pengecekan ini menggunakan konsep *ontology*. Dan untuk model proses SOP dibagi ke dalam 2 macam yaitu model proses standar dalam bentuk *Petri Net* (berekstensi *pnml*) yang berisi informasi mengenai *transition*, *arc*, dan *place* yang mendefinisikan alur proses bisnis dan dalam bentuk *PNML Extended* yang dibuat untuk menyimpan informasi berupa atribut *resource*, *role*, *time*, dan aturan pengambilan keputusan pada proses bisnis. Kemudian kedua data tersebut diolah sesuai dengan alur pemrosesan dari aplikasi yang akan dibangun yang dapat dilihat pada Gambar 3.1. Alur sistem yang dibangun terdiri dari dua tahapan yaitu tahap *training* data dan tahap *testing* data.

Tahap *training* merupakan tahap pengolahan data *training* proses bisnis untuk menghasilkan bobot *fraud* dan aturan asosiasi antara atribut pelanggaran. Tahap ini dimulai dengan proses pengecekan kesesuaian antara *event log* dengan model proses SOP yang dikenal dengan *Conformance Checking*. Pengecekan kesesuaian terdiri dari tiga macam pengecekan yaitu *Check Compliance Using Conformance Checking*, *Conformance Checking for Attribute*, dan *Check Wrong Decision*. Alur proses

conformance checking dapat dilihat pada Gambar 3.2. Hasil dari pengecekan kesesuaian ini adalah daftar jumlah pelanggaran untuk tiap atribut pelanggaran yang dilakukan oleh suatu proses. Kemudian, tahap selanjutnya adalah penghitungan bobot *fraud* untuk tiap *case* (proses). Untuk menghitung bobot *fraud* dibutuhkan data pelanggaran hasil pengecekan kesesuaian dan data penilaian kepentingan dari para pakar yang diinputkan oleh pengguna dalam sistem ini. Hal ini dikarenakan untuk tiap proses bisnis yang berbeda, penilaian kepentingan atribut pelanggaran dapat dilakukan oleh pakar yang berbeda sesuai dengan bidangnya. Dan proses terakhir pada tahap *training* ini adalah pembentukan aturan asosiasi antara atribut pelanggaran dengan menggunakan algoritma *Fuzzy ARL*. Dalam proses pencarian aturan asosiasi ini dibutuhkan nilai *threshold* untuk menyeleksi tiap *item* pada semua *itemset*. Nilai *threshold* diinputkan oleh pengguna *plugin* ini. Hasilnya berupa kumpulan aturan asosiasi beserta nilai *support* dan *confidencenya*.

Tahap *testing* merupakan tahap pendeteksian *fraud* dengan menggunakan data hasil *training* berupa aturan asosiasi. Tiap proses bisnis pada data *testing* dilakukan pengecekan kesesuaian untuk mendapatkan jumlah pelanggaran untuk tiap atribut pelanggaran. Proses pengecekan kesesuaian sama seperti pada tahap *training*. Setelah data pelanggaran untuk tiap proses bisnis diperoleh, selanjutnya dilakukan pengecekan kecurangan (*fraud*) proses dengan memeriksa atribut-atribut pelanggaran yang dilanggar dengan aturan asosiasi yang tersedia. Apabila kombinasi atribut pelanggaran yang dilanggar sama dengan aturan asosiasi yang tersedia, maka dalam proses bisnis tersebut terjadi *fraud* dengan besar risikonya berdasarkan nilai *confidence* aturan asosiasinya. Apabila tidak terdapat aturan asosiasi yang sama dengan kombinasi atribut pelanggaran yang terjadi, maka dilakukan perhitungan bobot *fraud* seperti pada tahap *training*. Untuk data penilaian kepentingan dari pakar menggunakan data yang sama dengan data pada tahap *training*. Sehingga besar resiko kecurangan untuk proses bisnis tersebut menggunakan bobot *fraud*.

Setelah mendeteksi *fraud* beserta besar resiko yang dihasilkan, tiap proses bisnis pada data *testing* akan dikelompokkan ke dalam empat kelas *fraud* berdasarkan besar resiko yang dihasilkan. Untuk pembagiannya menggunakan konsep fungsi keanggotaan *fuzzy* dengan menggunakan fungsi keanggotaan seperti pada Gambar 3.9 dan rumus perhitungan keanggotaannya dapat dilihat pada Gambar 3.10. Sehingga hasil dari aplikasi yang dibangun ini adalah hasil pendeteksian kecurangan (*fraud*) yang disertai dengan besar resiko yang dihasilkan, serta pengelompokkan proses bisnis ke dalam kelas *fraud* berdasarkan besar resikonya.

4.1.3. Spesifikasi Kebutuhan Perangkat Lunak

Bagian ini berisi semua kebutuhan perangkat lunak yang diuraikan secara rinci dalam bentuk diagram kasus, diagram urutan, dan diagram aktivitas. Masing-masing diagram menjelaskan perilaku atau sifat dari sistem ini. Kebutuhan perangkat lunak dalam sistem ini mencakup kebutuhan fungsional saja. Pada bab ini juga dijelaskan tentang spesifikasi terperinci pada masing-masing kebutuhan fungsional. Rincian spesifikasi dari kasus penggunaan disajikan dalam bentuk tabel.

4.1.3.1. Kebutuhan Fungsional

Kebutuhan fungsional berisi kebutuhan utama yang harus dipenuhi oleh sistem agar dapat bekerja dengan baik. Kebutuhan fungsional mendefinisikan layanan yang harus disediakan oleh sistem, bagaimana reaksi terhadap masukan, dan apa yang harus dilakukan sistem pada situasi khusus. Daftar kebutuhan fungsional dapat dilihat pada Tabel 4.1. Kode kebutuhan yang ditandai dengan tanda * menandakan bahwa kebutuhan fungsional tersebut dikerjakan oleh orang lain (bukan penulis).

Tabel 4.1. Daftar Kebutuhan Fungsional Perangkat Lunak

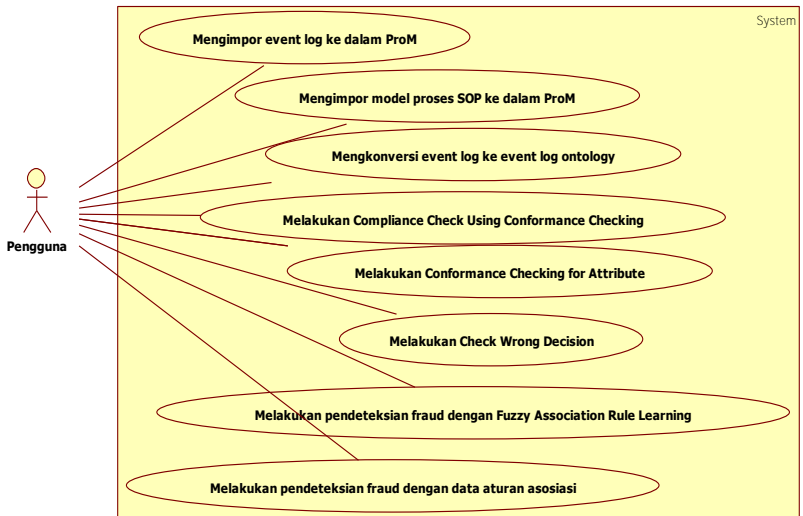
Kode Kebutuhan	Kebutuhan Fungsional	Deskripsi
TA-F0001	Mengimpor <i>event log</i>	Pengguna dapat mengimpor <i>event log</i> ke dalam aplikasi ProM
TA-F0002	Mengimpor model proses SOP	Pengguna dapat mengimpor model proses SOP dalam bentuk <i>Petri Net</i> dan <i>PNML Extended</i> ke dalam aplikasi ProM
TA-F0003*	Mengkonversi <i>event log</i> ke <i>event log ontology</i>	Pengguna dapat mengkonversi <i>event log</i> ke bentuk <i>event log ontology</i>
TA-F0004	Melakukan <i>Compliance Check Using Conformance Checking</i>	Pengguna dapat melakukan pengecekan kesesuaian untuk mendeteksi pelanggaran pada atribut <i>Skip Sequence</i> dan <i>Skip Decision</i> .
TA-F0005	Melakukan <i>Conformance Checking for Attribute</i>	Pengguna dapat melakukan pengecekan kesesuaian antara atribut pada <i>event log</i> dan model proses SOP untuk mendeteksi pelanggaran pada atribut <i>Throughput Time Min</i> , <i>Throughput Time Max</i> , <i>Wrong Resource</i> , <i>Wrong Duty Sequence</i> , <i>Wrong Duty Decision</i> , <i>Wrong Duty Combine</i> , dan <i>Wrong Pattern</i> .
TA-F0006*	Melakukan <i>Check Wrong Decision</i>	Pengguna dapat melakukan pengecekan kesesuaian untuk aturan penarikan keputusan pada proses bisnis yang disesuaikan dengan model proses SOP untuk mendeteksi pelanggaran <i>Wrong Decision</i> .
TA-F0007	Melakukan pendeteksian <i>fraud</i> dengan <i>Fuzzy Association Rule Learning</i>	Pengguna dapat melakukan perhitungan bobot <i>fraud</i> dan pembentukan aturan asosiasi untuk atribut pelanggaran yang disertai dengan nilai <i>confidence</i>
TA-F0008	Melakukan pendeteksian <i>fraud</i> dengan data aturan asosiasi	Pengguna dapat melakukan pendeteksian <i>fraud</i> dengan menggunakan data aturan asosiasi yang dihasilkan dan mengelompokkan ke dalam kelas <i>fraud</i> yang sesuai.

4.1.4. Aktor

Aktor mendefinisikan entitas-entitas yang terlibat dan berinteraksi langsung dengan sistem. Entitas ini bisa berupa manusia maupun sistem atau perangkat lunak yang lain. Penulis mendefinisikan aktor untuk sistem ini yaitu perancang perangkat lunak, dan pengembang perangkat lunak yang menggunakan ProM. ProM sebagai aplikasi ditambahkan sebagai aktor yang berupa sistem. Hal ini dikarenakan *plugin* ini memiliki fitur yang memerlukan pengiriman dan penerimaan data dengan ProM. Data ini akan diproses dan diolah oleh ProM dan *plugin* dengan cara masing-masing.

4.1.5. Kasus Penggunaan

Kasus-kasus penggunaan dalam sistem ini akan dijelaskan secara rinci pada subbab ini. Kasus penggunaan secara umum akan digambarkan oleh salah satu model UML, yaitu diagram kasus penggunaan. Rincian kasus penggunaan berisi spesifikasi kasus penggunaan, diagram aktivitas, dan diagram urutan untuk masing-masing kasus penggunaan. Diagram kasus penggunaan dapat dilihat pada Gambar 4.1. Daftar kode diagram kasus penggunaan sistem dapat dilihat pada Tabel 4.2.



Gambar 4.1. Diagram Kasus Penggunaan Sistem

Tabel 4.2 Daftar Kode Diagram Kasus Penggunaan

Kode Kasus Penggunaan	Nama
TA-UC0001	Mengimpor <i>event log</i> ke dalam ProM
TA-UC0002	Mengimpor model proses SOP ke dalam ProM
TA-UC0003*	Mengkonversi <i>event log</i> ke <i>event log ontology</i>
TA-UC0004	Melakukan <i>Compliance Check Using Conformance Checking</i>
TA-UC0005	Melakukan <i>Conformance Checking for Attribute</i>
TA-UC0006*	Melakukan <i>Check Wrong Decision</i>
TA-UC0007	Melakukan pendeteksian <i>fraud</i> dengan <i>Fuzzy Association Rule Learning</i>
TA-UC0008	Melakukan pendeteksian <i>fraud</i> dengan data aturan asosiasi

Keterangan: * menandakan kasus penggunaan menggunakan *plugin* yang dikembangkan oleh orang lain.

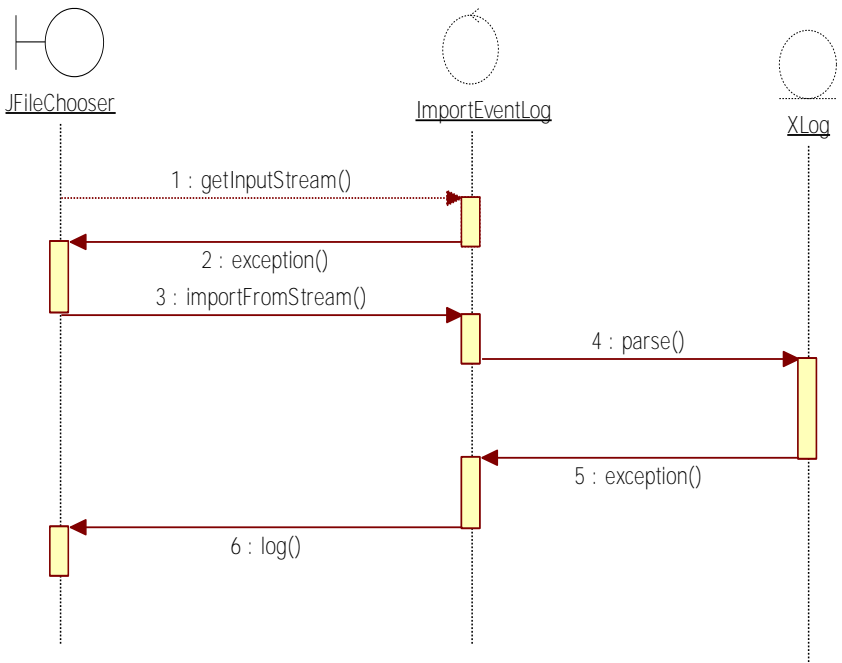
4.1.5.1. Mengimpor *Event Log* ke dalam ProM

Pada kasus penggunaan ini, sistem akan menerima perintah dari pengguna untuk mengimpor *event log* ke dalam aplikasi ProM, sehingga *event log* tersebut dapat tampil untuk kemudian dapat digunakan dalam proses lainnya. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 4.3. Diagram urutan dan diagram aktivitas dari kasus penggunaan ini bisa dilihat pada Gambar 4.2 dan Gambar 4.3.

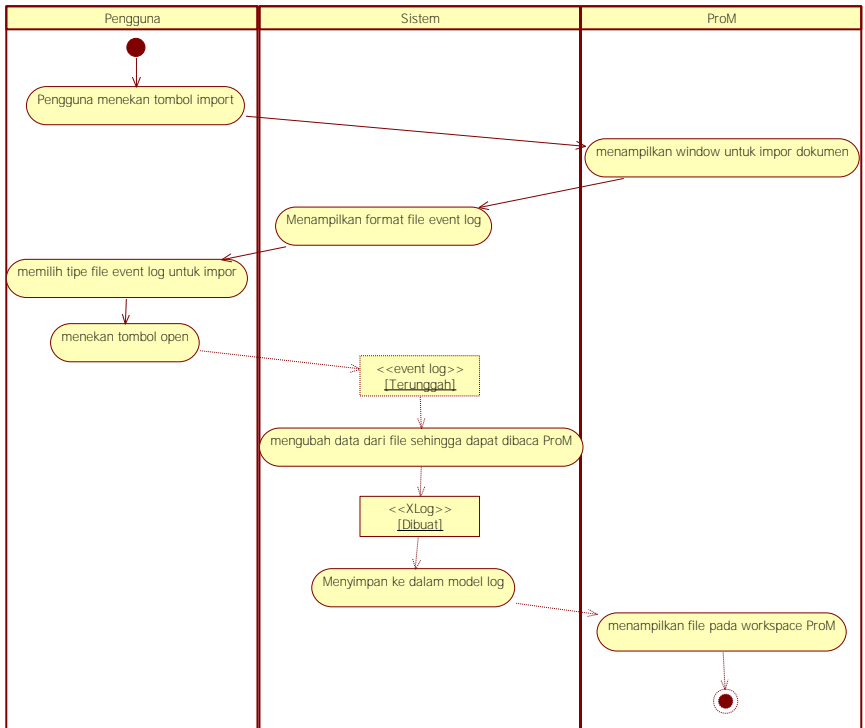
Tabel 4.3 Spesifikasi Kasus Penggunaan Mengimpor *event log* ke dalam ProM

Nama	Mengimpor <i>event log</i> ke dalam ProM
Kode	TA-UC0001
Deskripsi	Mengimpor <i>event log</i> yang telah dipersiapkan sebelumnya ke dalam ProM agar dapat diolah kembali
Tipe	Fungsional
Pemicu	Pengguna menekan tombol perintah mengimpor <i>event log</i> pada panel aksi pada system
Aktor	Pengguna
Kondisi Awal	Belum terdapat <i>event log</i> pada <i>workspace</i> ProM
Aliran:	
- Kejadian Normal	<ol style="list-style-type: none"> 1. Pengguna menekan tombol <i>import</i> yang terdapat pada <i>ProM</i> 2. Sistem menampilkan <i>window</i> untuk melakukan impor dokumen 3. Pengguna memilih tipe <i>file event log</i> sebagai tipe <i>file</i> yang akan diimpor 4. Pengguna memilih <i>file</i> yang akan diimpor 5. Pengguna menekan tombol <i>open</i> 6. Sistem mengubah data dari <i>file</i> yang akan diimpor sehingga dapat dibaca oleh <i>plugin</i> 7. Sistem menyimpan data yang telah diubah ke dalam model <i>log</i> 8. Sistem menampilkan <i>file</i> yang telah dipilih ke dalam <i>workspace</i>.

- Kejadian Alternatif	Tidak ada
Kondisi Akhir	Sistem menampilkan <i>file</i> yang telah diimpor ke dalam <i>workspace</i>
Kebutuhan Khusus	Tidak ada



Gambar 4.2. Diagram Urutan Mengimpor Event Log ke dalam ProM.



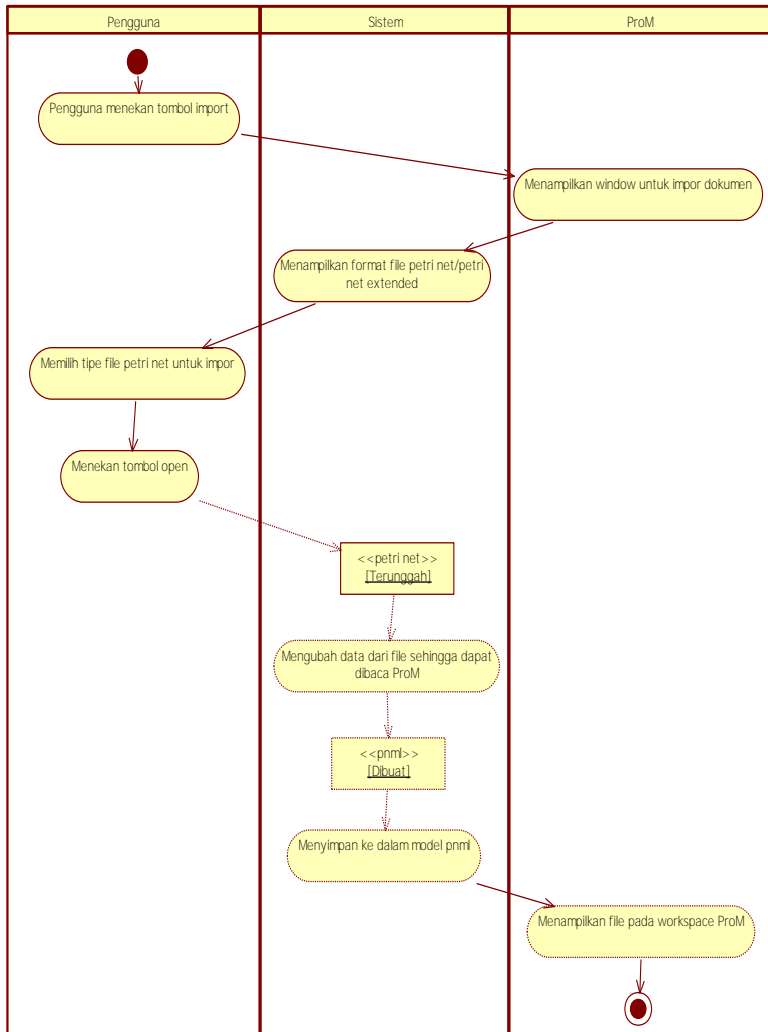
Gambar 4.3 Diagram Aktivitas Mengimpor *Event Log* ke dalam ProM

4.1.5.2. Mengimpor model proses SOP ke dalam ProM

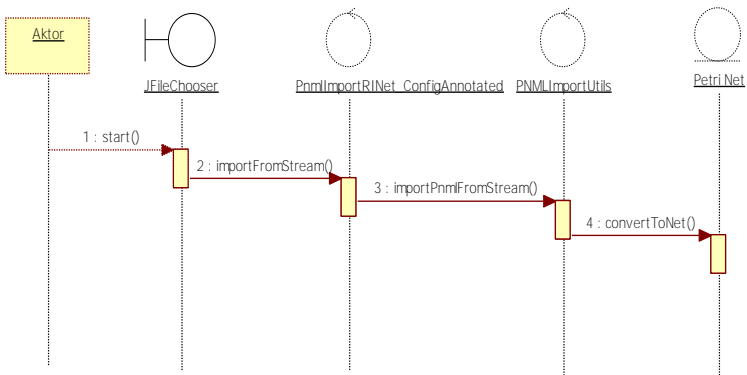
Pada kasus penggunaan ini, sistem akan menerima perintah dari pengguna untuk mengimpor model proses dalam bentuk *petri net* maupun *PNML extended* ke dalam aplikasi ProM, sehingga *petri net* tersebut dapat tampil untuk kemudian dapat digunakan dalam proses lainnya. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 4.4. Diagram aktivitas dan diagram urutan dari kasus penggunaan ini bisa dilihat pada Gambar 4.4 dan Gambar 4.5

Tabel 4.4 Spesifikasi Kasus Penggunaan Mengimpor model proses SOP ke Dalam ProM

Nama	Mengimpor model proses SOP ke dalam ProM
Kode	TA-UC0002
Deskripsi	Mengimpor model proses SOP yang telah dipersiapkan sebelumnya ke dalam ProM agar dapat diolah kembali.
Tipe	Fungsional
Pemicu	Pengguna menekan tombol perintah mengimpor <i>petri net</i> pada panel aksi pada system
Aktor	Pengguna
Kondisi Awal	Belum terdapat <i>petri net</i> model proses yang diinginkan pada <i>workspace</i> ProM
Aliran:	
- Kejadian Normal	<ol style="list-style-type: none"> 1. Pengguna menekan tombol <i>import</i> yang terdapat pada ProM 2. Sistem menampilkan <i>window</i> untuk melakukan impor dokumen 3. Pengguna memilih tipe <i>file petri net/petri net extended</i> sebagai tipe <i>file</i> yang akan diimpor 4. Pengguna memilih <i>file</i> yang akan diimpor 5. Pengguna menekan tombol <i>open</i> 6. Sistem mengubah data dari <i>file</i> yang akan diimpor sehingga dapat dibaca oleh <i>plugin</i> 7. Sistem menyimpan data yang telah diubah ke dalam model <i>petri net/ReadPNML</i> 8. Sistem menampilkan <i>file petri net/ReadPNML</i> yang telah dipilih ke dalam <i>workspace</i>
- Kejadian Alternatif	Tidak ada
Kondisi Akhir	Sistem menampilkan <i>file</i> yang telah diimpor ke dalam <i>workspace</i>
Kebutuhan Khusus	Tidak ada



Gambar 4.4 Diagram Aktivitas Mengimpor model proses SOP ke Dalam ProM



Gambar 4.5. Diagram Urutan Mengimpor model proses SOP ke Dalam ProM

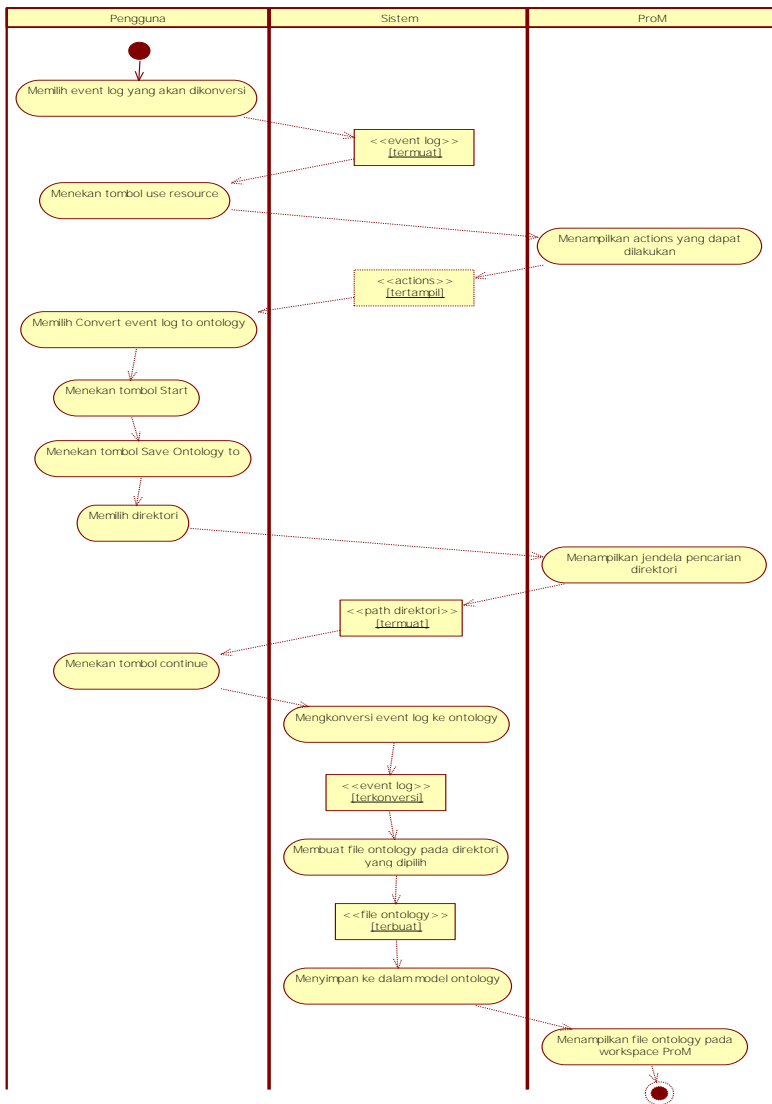
4.1.5.3. Mengkonversi *event log* ke *event log ontology*

Sistem dapat mengkonversi *event log* yang sebelumnya telah diimpor dan terdapat pada *workspace* ProM ke bentuk *event log ontology*. Kasus penggunaan ini menggunakan *plugin* yang dikembangkan oleh orang lain. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 4.5. Diagram aktivitas dan diagram urutan dari kasus penggunaan ini bisa dilihat pada Gambar 4.6 dan Gambar 4.7.

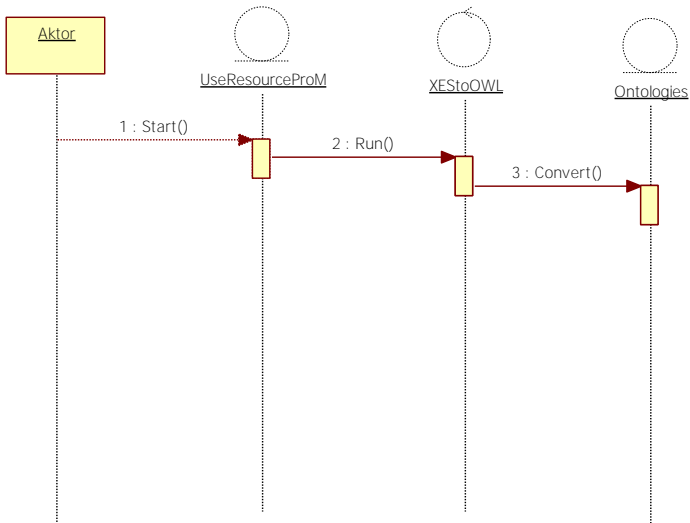
Tabel 4.5 Spesifikasi Kasus Penggunaan Mengkonversi *event log* ke *event log ontology*

Nama	Mengkonversi <i>event log</i> ke <i>event log ontology</i>
Kode	TA-UC0003
Deskripsi	Sistem dapat mengkonversi <i>event log</i> ke <i>event log ontology</i>
Tipe	Fungsional
Pemicu	Pengguna menekan tombol <i>use resource</i> dan memilih <i>Convert Event Log to Ontology</i>

Aktor	Pengguna
Kondisi Awal	Sistem telah memuat <i>event log</i> yang akan dikonversi
Aliran:	
- Kejadian Normal	<ol style="list-style-type: none"> 1. Pengguna memilih <i>event log</i> yang akan dikonversi 2. Pengguna menekan tombol <i>use resource</i> 3. Sistem menampilkan <i>actions</i> yang dapat dilakukan terhadap <i>event log</i> 4. Pengguna memilih <i>Convert Event Log to Ontology</i> 5. Pengguna menekan tombol <i>start</i> 6. Pengguna menekan tombol <i>Save Ontology to</i> 7. Pengguna memilih direktori tempat menyimpan file <i>event log ontology</i>. 8. Pengguna menekan tombol <i>Continue</i> 9. Sistem mengkonversi <i>event log</i> ke <i>event log ontology</i> dan membuat file <i>ontology</i> baru pada direktori yang dipilih. 10. Sistem menyimpan data yang telah diubah ke dalam model <i>ontology</i>. 11. Sistem menampilkan file <i>ontology</i> yang dihasilkan pada <i>workspace ProM</i>
- Kejadian Alternatif	
Kondisi Akhir	Sistem menampilkan file <i>event log ontology</i> yang dihasilkan pada <i>workspace ProM</i>
Kebutuhan Khusus	Tidak ada



Gambar 4.6 Diagram Aktivitas Mengkonversi event log ke event log ontology



Gambar 4.7 Diagram Urutan Mengkonversi *event log* ke *event log ontology*

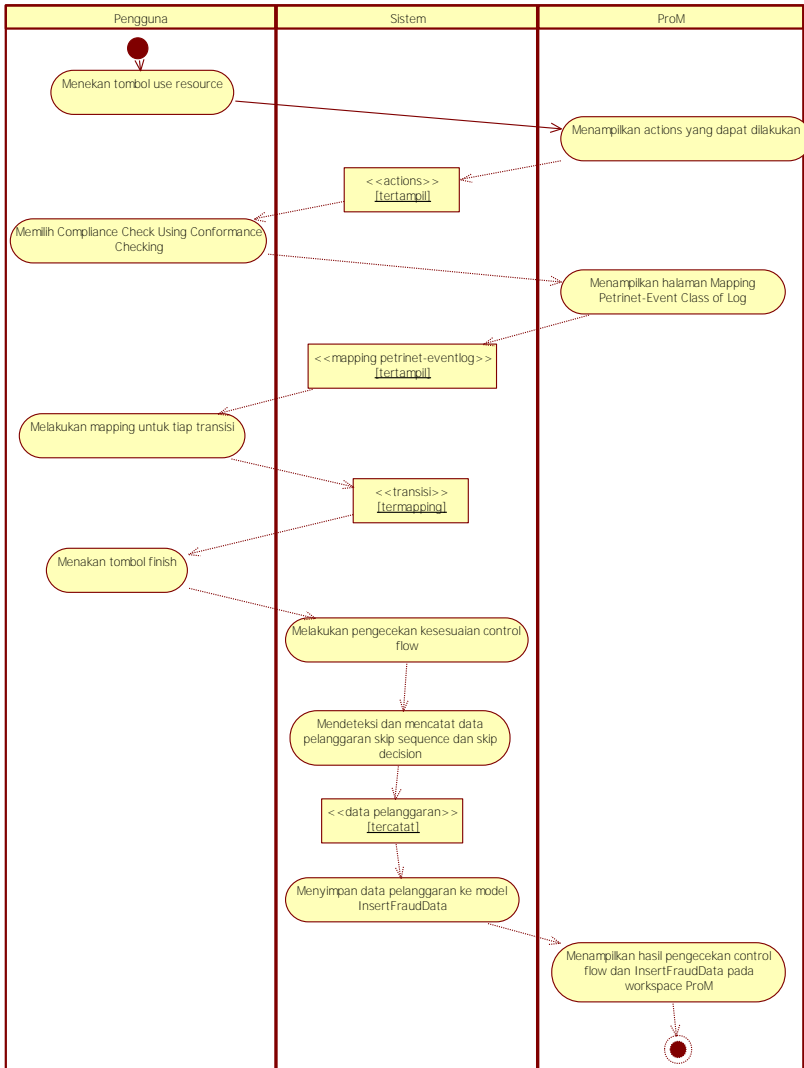
4.1.5.4. Melakukan *Compliance Check Using Conformance Checking*

Sistem dapat melakukan pengecekan kesesuaian *control flow* antara *event log* dan model proses SOP untuk menghasilkan data pelanggaran *Skip Sequence* dan *Skip Decision*. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 4.6. Diagram aktivitas dan diagram urutan dari kasus penggunaan ini bisa dilihat pada Gambar 4.8 dan Gambar 4.9.

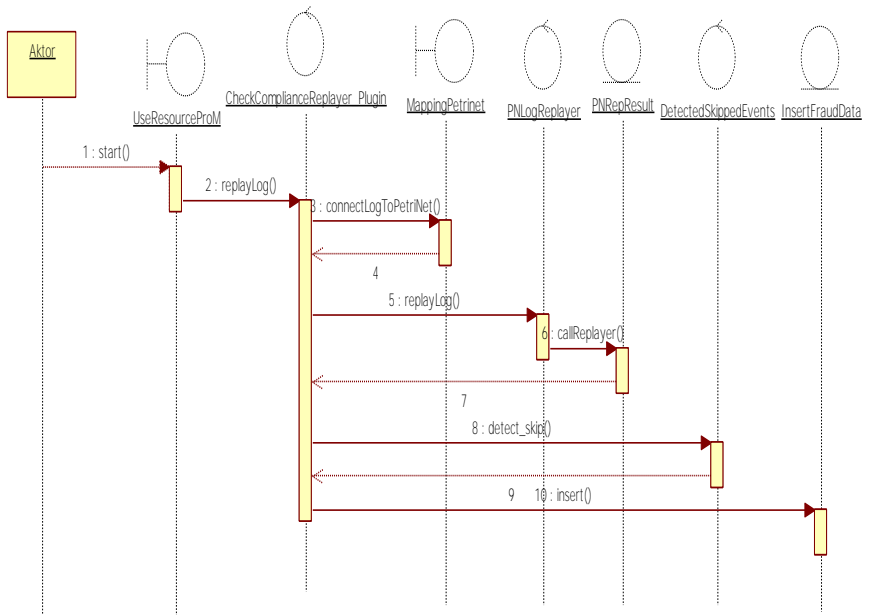
Tabel 4.6 Spesifikasi Kasus Penggunaan Melakukan *Compliance Check Using Conformance Checking*

Nama	Melakukan <i>Compliance Check Using Conformance Checking</i>
Kode	TA-UC0004

Deskripsi	Sistem dapat melakukan pengecekan kesesuaian <i>control flow</i> antara <i>event log</i> dan model proses SOP untuk menghasilkan data pelanggaran <i>Skip Sequence</i> dan <i>Skip Decision</i>
Tipe	Fungsional
Pemicu	Pengguna menekan tombol <i>use resource</i> dan memilih <i>Check Compliance Using Conformance Checking</i>
Aktor	Pengguna
Kondisi Awal	Sistem telah memuat <i>event log</i> dan <i>petri net</i> ke dalam <i>workspace</i>
Aliran:	
- Kejadian Normal	<ol style="list-style-type: none"> 1. Pengguna menekan tombol <i>use resource</i> 2. Sistem menampilkan <i>Actions</i> yang dapat dilakukan 3. Pengguna memilih <i>Check Compliance Using Conformance Checking</i> pada <i>Actions</i> 4. Sistem menampilkan halaman <i>Mapping Petrinet-Event Class of Log</i>. 5. Pengguna melakukan <i>mapping</i> untuk tiap transisi pada model petri net 6. Pengguna menekan tombol finish 7. Sistem melakukan pengecekan kesesuaian <i>control flow</i> antara <i>event log</i> dan <i>petri net</i>. 8. Sistem melakukan pendeteksian dan pencatatan pelanggaran skip sequence dan skip decision. 9. Sistem menyimpan data pelanggaran yang dihasilkan ke dalam model <i>InsertFraudData</i> 10. Sistem menampilkan hasil pengecekan <i>control flow</i> dan <i>InsertFraudData</i> pada <i>workspace</i> ProM.
- Kejadian Alternatif	Tidak ada
Kondisi Akhir	Sistem menampilkan hasil pengecekan <i>control flow</i> dan <i>InsertFraudData</i> pada <i>workspace</i> ProM.
Kebutuhan Khusus	Tidak ada



Gambar 4.8 Diagram Aktivitas Melakukan *Compliance Check Using Conformance Checking*



Gambar 4.9. Diagram Urutan Melakukan Compliance Check Using Conformance Checking

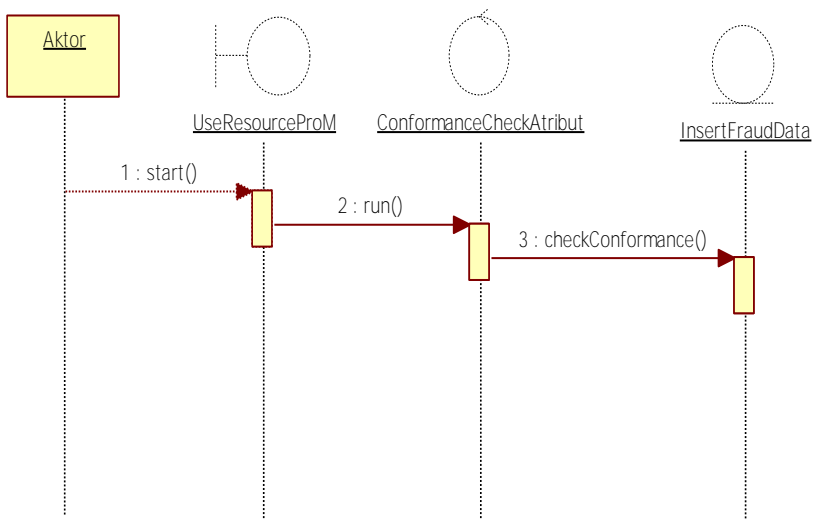
4.1.5.5. Melakukan Conformance Checking for Attribute

Sistem dapat melakukan pengecekan kesesuaian untuk atribut waktu, *resource*, *role*, dan pola antara *event log* dan model proses SOP untuk mendeteksi pelanggaran *Throughput Time*, *Wrong Resource*, *Wrong Duty*, dan *Wrong Pattern*. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 4.7. Diagram urutan dan diagram aktivitas dari kasus penggunaan ini bisa dilihat pada Gambar 4.10 dan Gambar 4.11.

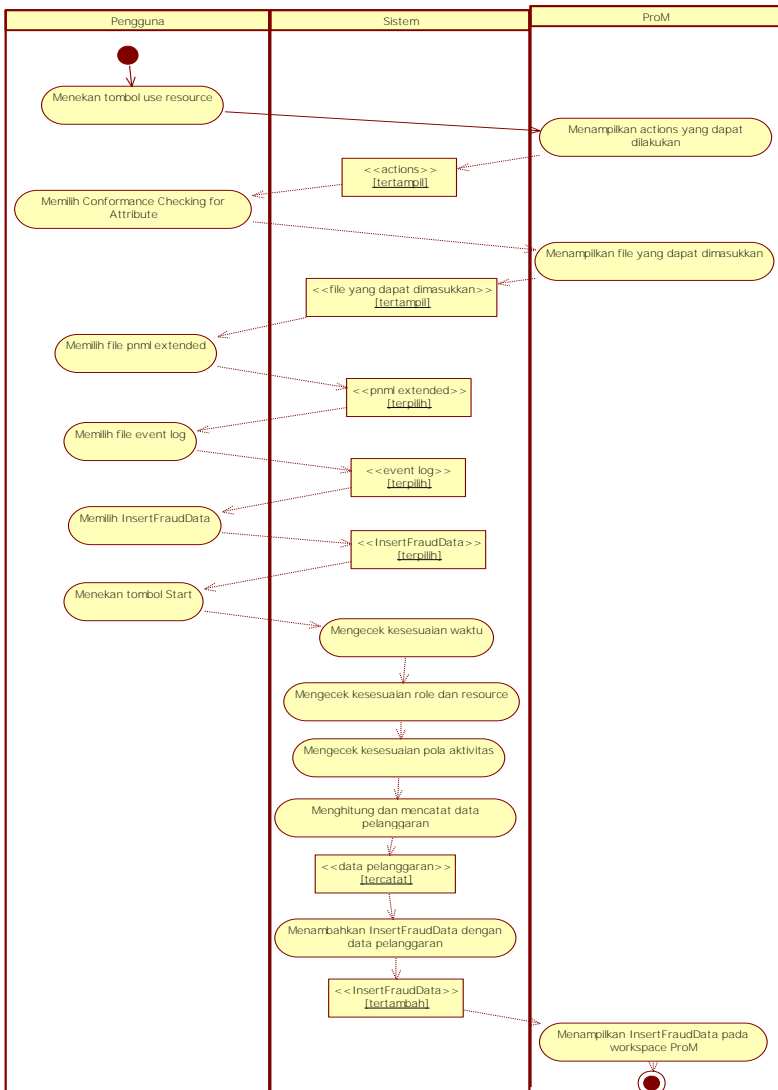
Tabel 4.7 Spesifikasi Kasus Penggunaan Melakukan *Conformance Checking for Attribute*

Nama	Melakukan <i>Conformance Checking for Attribute</i>
Kode	TA-UC0005
Deskripsi	Sistem dapat melakukan pengecekan kesesuaian untuk atribut waktu, <i>resource</i> , <i>role</i> , dan pola antara <i>event log</i> dan model proses SOP untuk mendeteksi pelanggaran <i>Throughput Time</i> , <i>Wrong Resource</i> , <i>Wrong Duty</i> , dan <i>Wrong Pattern</i> .
Tipe	Fungsional
Pemicu	Pengguna menekan tombol <i>use resource</i> dan memilih <i>Conformance Checking for Attribute</i>
Aktor	Pengguna
Kondisi Awal	Sistem telah memuat <i>event log</i> , <i>ReadPNML</i> , dan <i>InsertFraudData</i> .
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Pengguna menekan tombol <i>use resource</i> 2. Sistem menampilkan <i>Actions</i> yang dapat dilakukan 3. Pengguna memilih <i>Conformance Checking for Attribute</i> 4. Sistem menampilkan <i>file</i> yang dapat dimasukkan 5. Pengguna memilih <i>input</i> yang pertama berupa <i>ReadPNML</i> yang telah ada pada <i>workspace</i> 6. Pengguna memilih <i>input</i> yang kedua berupa <i>event log</i> yang telah ada pada <i>workspace</i> 7. Pengguna memilih <i>input</i> yang ketiga berupa <i>InsertFraudData</i> yang telah ada pada <i>workspace</i> 8. Pengguna menekan tombol <i>Start</i> 9. Sistem mengecek kesesuaian waktu transisi <i>event log</i> dan model <i>petri net</i> 10. Sistem mengecek kesesuaian <i>role</i> dan <i>resource</i> transisi <i>event log</i> dan model <i>petri net</i> 11. Sistem mengecek kesesuaian pola aktivitas <i>event log</i> dan model <i>petri net</i> 12. Sistem menghitung dan mencatat data jumlah pelanggaran <i>Throughput Time Min</i>, <i>Throughput</i>

	<p><i>Time Max, Wrong Resource, Wrong Duty Sequence, Wrong Duty Decision, Wrong Duty Combine, dan Wrong Pattern.</i></p> <p>13. Sistem menambahkan data pelanggaran ke dalam <i>InsertFraudData</i></p> <p>14. Sistem menampilkan <i>InsertFraudData</i> pada <i>workspace</i> ProM</p>
- Kejadian Alternatif	Tidak ada
Kondisi Akhir	Sistem menampilkan <i>InsertFraudData</i> pada <i>workspace</i> ProM
Kebutuhan Khusus	Tidak ada



Gambar 4.10. Diagram Urutan *Conformance Checking for Attribute*



Gambar 4.11 Diagram Aktivitas *Conformance Checking for Attribute*

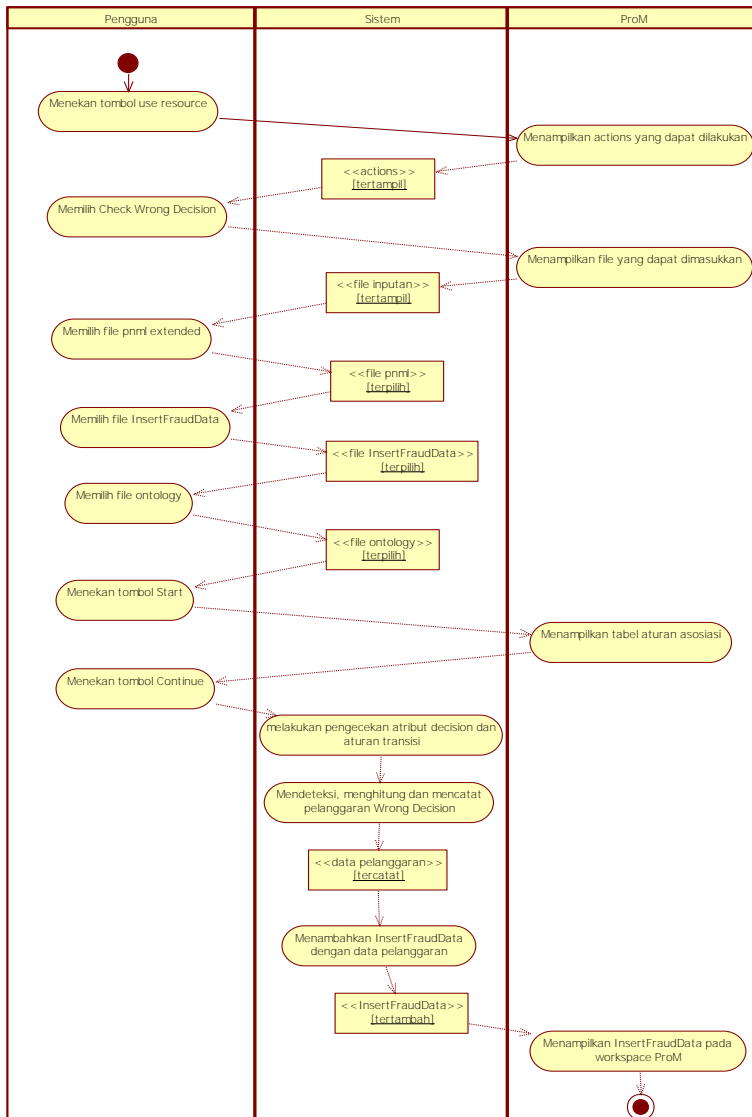
4.1.5.6. Melakukan *Check Wrong Decision*

Sistem dapat melakukan pengecekan kesesuaian aturan pengambilan keputusan antara *event log* dan model proses SOP untuk mendeteksi dan menghasilkan data pelanggaran *Wrong Decision*. Kasus penggunaan ini menggunakan *plugin* yang dikembangkan oleh orang lain. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 4.8. Diagram aktivitas dan diagram urutan dari kasus penggunaan ini bisa dilihat pada Gambar 4.12 dan Gambar 4.13.

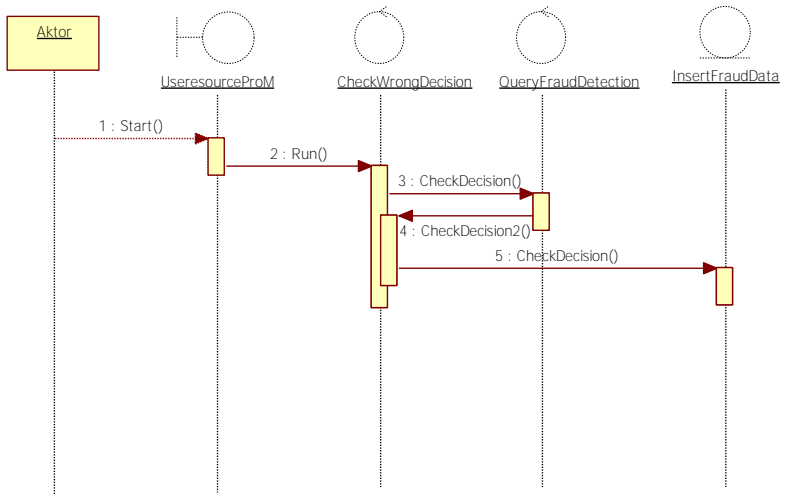
Tabel 4.8. Spesifikasi Kasus Penggunaan Melakukan *Check Wrong Decision*

Nama	Melakukan <i>Check Wrong Decision</i>
Kode	TA-UC0006
Deskripsi	Sistem dapat melakukan pengecekan kesesuaian aturan pengambilan keputusan antara <i>event log</i> dan model proses SOP untuk mendeteksi dan menghasilkan data pelanggaran <i>Wrong Decision</i>
Tipe	Fungsional
Pemicu	Pengguna menekan tombol <i>use resource</i> dan memilih <i>Check Wrong Decision</i>
Aktor	Pengguna
Kondisi Awal	Sistem telah memuat <i>event log ontology</i> , <i>ReadPNML</i> , dan <i>InsertFraudData</i>
Aliran:	
- Kejadian Normal	<ol style="list-style-type: none"> 1. Pengguna menekan tombol <i>use resource</i> 2. Sistem menampilkan <i>Actions</i> yang dapat dilakukan 3. Pengguna memilih <i>Check Wrong Decision</i> 4. Sistem menampilkan <i>file</i> yang dapat dimasukkan

	<ol style="list-style-type: none"> 5. Pengguna memilih <i>input</i> yang pertama berupa <i>ReadPNML</i> yang telah ada pada <i>workspace</i> 6. Pengguna memilih <i>input</i> yang kedua berupa <i>InsertFraudData</i> yang telah ada pada <i>workspace</i> 7. Pengguna memilih <i>input</i> yang ketiga berupa <i>event log ontology</i> yang telah ada pada <i>workspace</i> 8. Pengguna menekan tombol <i>Start</i> 9. Sistem menampilkan tabel aturan transisi. 10. Pengguna menekan tombol <i>Continue</i>. 11. Sistem melakukan pengecekan kesesuaian antara atribut <i>decision</i> dan aturan transisi pada <i>event log ontology</i> dan model SOP. 12. Sistem mendeteksi, menghitung, dan mencatat jumlah pelanggaran <i>Wrong Decision</i>. 13. Sistem menambahkan data pelanggaran ke dalam <i>InsertFraudData</i> 14. Sistem menampilkan <i>InsertFraudData</i> pada <i>workspace</i> ProM
- Kejadian Alternatif	Tidak ada
Kondisi Akhir	Sistem menampilkan <i>InsertFraudData</i> pada <i>workspace</i> ProM.
Kebutuhan Khusus	Tidak ada



Gambar 4.12. Diagram Aktivitas Melakukan *Check Wrong Decision*



Gambar 4.13 Diagram Urutan Melakukan Check Wrong Decision

4.1.5.7. Melakukan pendeteksian *Fraud* dengan *Fuzzy Association Rule Learning*

Sistem dapat melakukan perhitungan bobot *fraud* dengan menggunakan *fraud data* yang telah dihasilkan dan kemudian dapat mendeteksi *fraud* dengan menghasilkan aturan asosiasi beserta nilai *confidence*-nya dari atribut pelanggaran yang dilakukan. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 4.9. Diagram aktivitas dan diagram urutan dari kasus penggunaan ini bisa dilihat pada Gambar 4.14 dan Gambar 4.15.

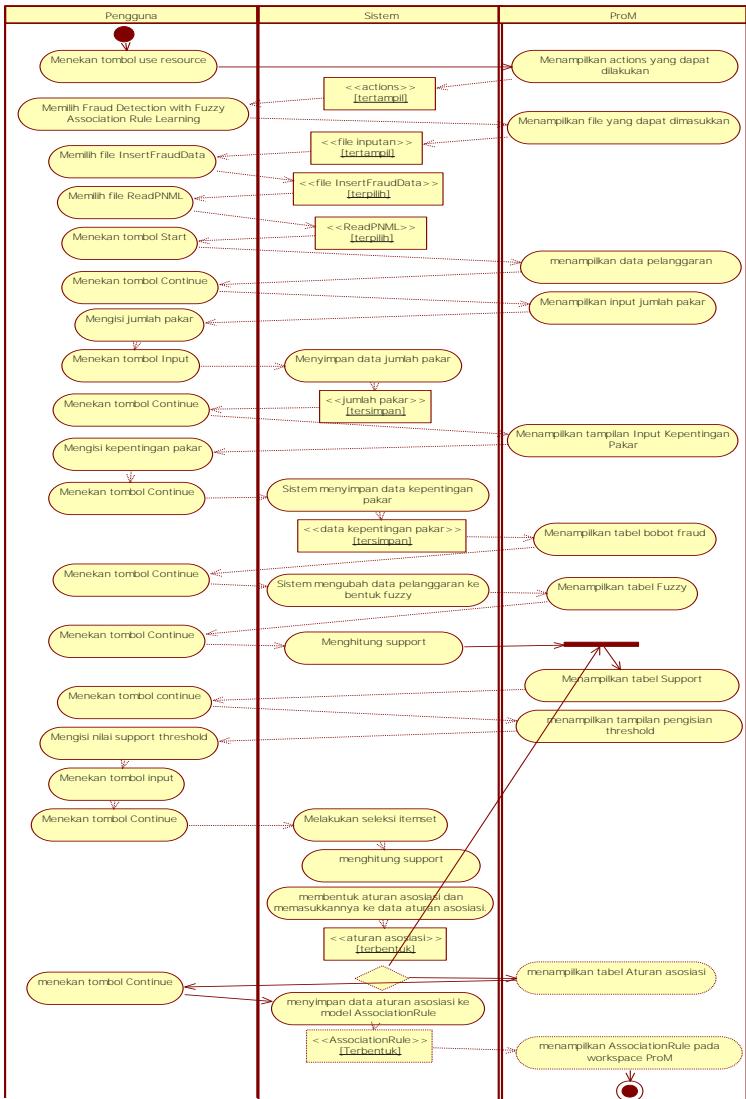
Tabel 4.9. Spesifikasi Kasus Penggunaan Melakukan pendeteksian *Fraud* dengan *Fuzzy Association Rule Learning*

Nama	Melakukan pendeteksian <i>Fraud</i> dengan <i>Fuzzy Association Rule Learning</i>
Kode	TA-UC0007

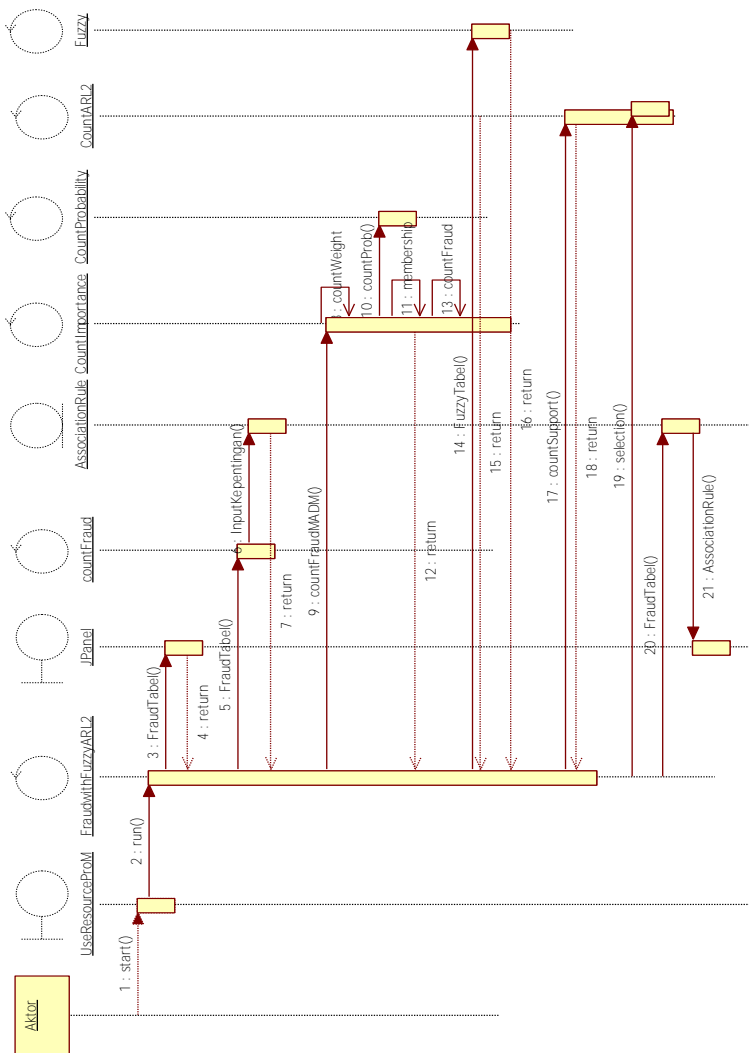
Deskripsi	Sistem dapat melakukan perhitungan bobot <i>fraud</i> dengan menggunakan <i>fraud data</i> yang telah dihasilkan dan kemudian dapat mendeteksi <i>fraud</i> dengan menghasilkan aturan asosiasi beserta nilai <i>confidencenya</i> dari atribut pelanggaran yang dilakukan
Tipe	Fungsional
Pemicu	Pengguna menekan tombol <i>use resource</i> dan memilih <i>Fraud Detection with Fuzzy Association Rule Learning</i>
Aktor	Pengguna
Kondisi Awal	Sistem telah memuat <i>InsertFraudData</i> dan <i>ReadPNML</i>
Aliran:	
- Kejadian Normal	<ol style="list-style-type: none"> 1. Pengguna menekan tombol <i>use resource</i> 2. Sistem menampilkan <i>Actions</i> yang dapat dilakukan 3. Pengguna memilih <i>Fraud Detection with Fuzzy Association Rule Learning</i> 4. Sistem menampilkan <i>file</i> yang dapat dimasukkan 5. Pengguna memilih <i>input</i> yang pertama berupa <i>InsertFraudData</i> yang telah ada pada <i>workspace</i> 6. Pengguna memilih <i>input</i> yang kedua berupa <i>ReadPNML</i> yang telah ada pada <i>workspace</i> 7. Pengguna menekan tombol <i>Start</i> 8. Sistem menampilkan tabel Data Pelanggaran 9. Pengguna menekan tombol <i>Continue</i>. 10. Sistem menampilkan tampilan <i>Input Jumlah Pakar</i> 11. Pengguna memasukkan jumlah pakar pada <i>textbox</i> yang disediakan 12. Pengguna menekan tombol <i>Input</i> 13. Sistem menyimpan data jumlah pakar 14. Pengguna menekan tombol <i>Continue</i>. 15. Sistem menampilkan tampilan <i>Input Kepentingan Pakar ke-i</i>.

	<ol style="list-style-type: none">16. Pengguna mengisi kepentingan dengan memilih <i>combo box</i> untuk tiap atribut pelanggaran17. Pengguna menekan tombol <i>Continue</i>.18. Sistem menyimpan data kepentingan pakar.19. Sistem menampilkan tabel Bobot <i>Fraud</i>20. Pengguna menekan tombol <i>Continue</i>21. Sistem mengubah data pelanggaran ke bentuk <i>fuzzy</i>.22. Sistem menampilkan Tabel <i>Fuzzy</i>.23. Pengguna menekan tombol <i>Continue</i>24. Sistem menghitung nilai <i>support</i> untuk <i>1-itemset</i>25. Sistem menampilkan tabel <i>Support 1-itemset</i>26. Pengguna menekan tombol <i>Continue</i>.27. Sistem menampilkan tampilan pengisian <i>threshold</i> untuk <i>1-itemset</i>28. Pengguna mengisi kotak isian <i>support threshold</i>29. Pengguna menekan tombol <i>input</i>.30. Pengguna menekan tombol <i>Continue</i>31. Sistem melakukan seleksi <i>itemset</i> dan menghitung <i>support</i> untuk <i>2-itemset</i>32. Sistem menampilkan tabel <i>Support 2-itemset</i>33. Pengguna menekan tombol <i>Continue</i>.34. Sistem menampilkan tampilan pengisian <i>threshold</i> untuk <i>2-itemset</i>35. Pengguna mengisi kotak isian <i>support threshold</i>36. Pengguna menekan tombol <i>input</i>.37. Pengguna menekan tombol <i>Continue</i>38. Sistem membentuk aturan asosiasi dan memasukkannya ke data aturan asosiasi.39. Sistem melakukan seleksi <i>itemset</i> dan menghitung <i>support</i> untuk <i>3-itemset</i>40. Sistem menampilkan tabel <i>Support 3-itemset</i>41. Pengguna menekan tombol <i>Continue</i>.42. Sistem menampilkan tampilan pengisian <i>threshold</i> untuk <i>3-itemset</i>43. Pengguna mengisi kotak isian <i>support threshold</i>44. Pengguna menekan tombol <i>input</i>.45. Pengguna menekan tombol <i>Continue</i>
--	--

	<p>46. Sistem membentuk aturan asosiasi dan memasukkannya ke data aturan asosiasi.</p> <p>47. Sistem melakukan seleksi <i>itemset</i> dan menghitung <i>support</i> untuk <i>4-itemset</i></p> <p>48. Sistem menampilkan tabel <i>Support 4-itemset</i></p> <p>49. Pengguna menekan tombol Continue.</p> <p>50. Sistem menampilkan tampilan pengisian <i>threshold</i> untuk <i>4-itemset</i></p> <p>51. Pengguna mengisi kotak isian <i>support threshold</i></p> <p>52. Pengguna menekan tombol <i>input</i>.</p> <p>53. Pengguna menekan tombol <i>Continue</i></p> <p>54. Sistem membentuk aturan asosiasi dan memasukkannya ke data aturan asosiasi.</p> <p>55. Sistem melakukan seleksi <i>itemset</i> dan menghitung <i>support</i> untuk <i>5-itemset</i></p> <p>56. Sistem menampilkan tabel <i>Support 5-itemset</i></p> <p>57. Pengguna menekan tombol Continue.</p> <p>58. Sistem menampilkan tampilan pengisian <i>threshold</i> untuk <i>5-itemset</i></p> <p>59. Pengguna mengisi kotak isian <i>support threshold</i></p> <p>60. Pengguna menekan tombol <i>input</i>.</p> <p>61. Pengguna menekan tombol <i>Continue</i></p> <p>62. Sistem menampilkan tabel Aturan asosiasi</p> <p>63. Pengguna menekan tombol Continue</p> <p>64. Sistem menyimpan data aturan asosiasi ke model <i>AssociationRule</i></p> <p>65. Sistem menampilkan <i>AssociationRule</i> pada <i>workspace</i> ProM</p>
<p>- Kejadian Alternatif</p>	<p>A1. Jika jumlah pakar lebih dari 1.</p> <ul style="list-style-type: none"> - Ulangi langkah 15-18 sejumlah pakar.
<p>Kondisi Akhir</p>	<p>Sistem menampilkan <i>AssociationRule</i> pada <i>workspace</i> ProM</p>
<p>Kebutuhan Khusus</p>	<p>Tidak ada</p>



Gambar 4.14. Diagram Aktivitas Melakukan Pendeteksian Fraud dengan Fuzzy Association Rule Learning



Gambar 4.15. Diagram Urutan Melakukan Pendeteksian *Fraud* dengan *Fuzzy Association Rule Learning*

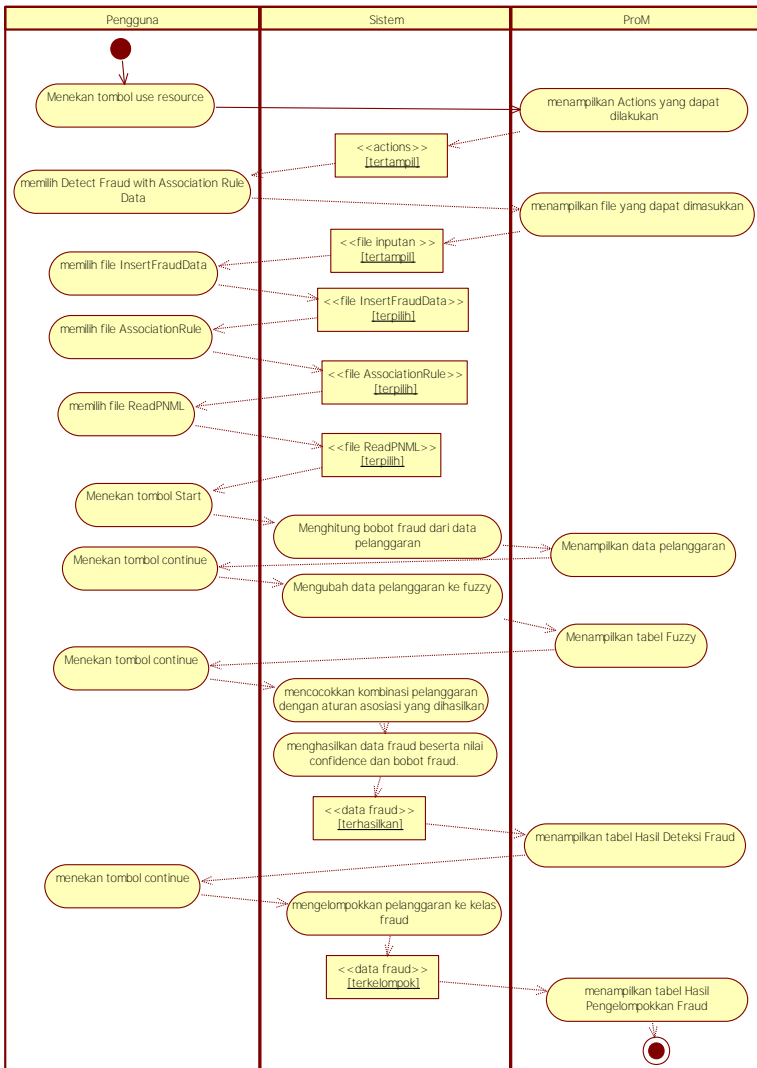
4.1.5.8. Melakukan pendeteksian *Fraud* dengan data aturan asosiasi

Sistem dapat melakukan pendeteksian *fraud* pada proses bisnis dengan menggunakan data aturan asosiasi yang dihasilkan dan mengkategorikan ke dalam kelas *fraud* yang sesuai dengan besar resiko yang dialami. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 4.10. Diagram aktivitas dan diagram urutan dari kasus penggunaan ini bisa dilihat pada Gambar 4.16 dan Gambar 4.17.

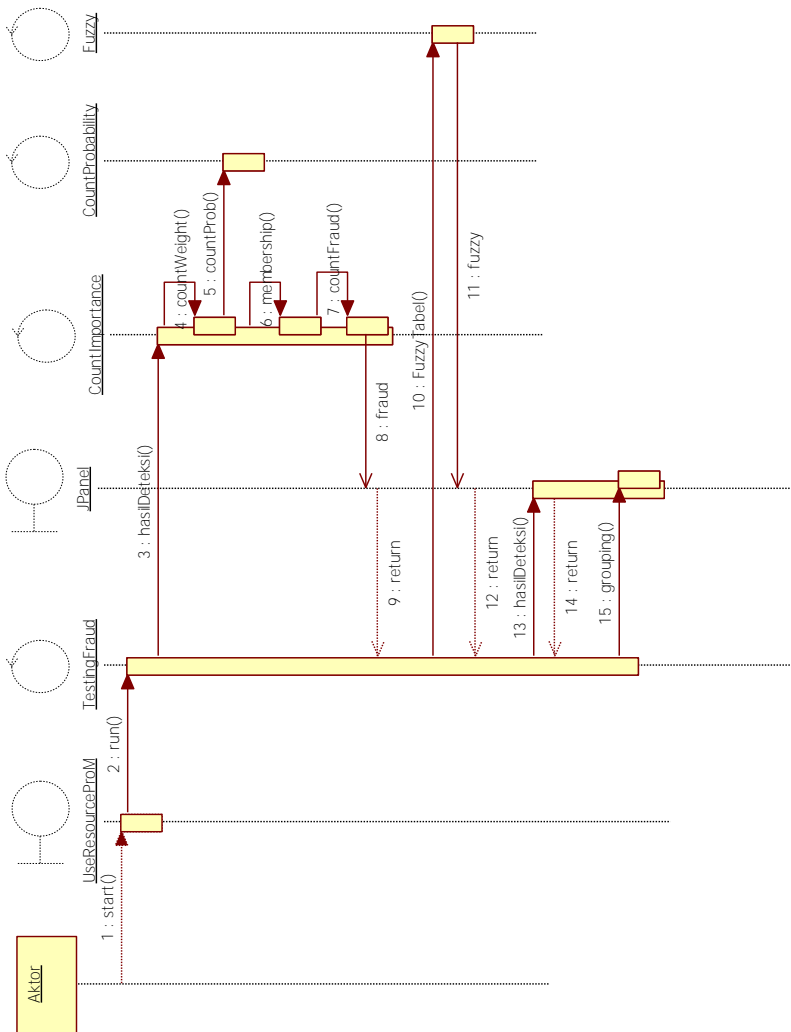
Tabel 4.10. Spesifikasi Kasus Penggunaan Melakukan Pendeteksian *Fraud* dengan data aturan asosiasi

Nama	Melakukan pendeteksian <i>Fraud</i> dengan data aturan asosiasi
Kode	TA-UC0007
Deskripsi	Sistem dapat melakukan pendeteksian <i>fraud</i> pada proses bisnis dengan menggunakan data aturan asosiasi yang dihasilkan dan mengkategorikan ke dalam kelas <i>fraud</i> yang sesuai dengan besar resiko yang dialami
Tipe	Fungsional
Pemicu	Pengguna menekan tombol <i>use resource</i> dan memilih <i>Detect Fraud with Association Rule Data</i>
Aktor	Pengguna
Kondisi Awal	Sistem telah memuat <i>file InsertFraudData, AssociationRule</i> dan <i>ReadPNML</i>
Aliran:	
- Kejadian Normal	<ol style="list-style-type: none"> 1. Pengguna menekan tombol <i>use resource</i> 2. Sistem menampilkan <i>Actions</i> yang dapat dilakukan 3. Pengguna memilih <i>Detect Fraud with Association Rule Data</i> 4. Sistem menampilkan <i>file</i> yang dapat dimasukkan

	<ol style="list-style-type: none"> 5. Pengguna memilih <i>input</i> yang pertama berupa <i>InsertFraudData</i> yang telah ada pada <i>workspace</i> 6. Pengguna memilih <i>input</i> yang kedua berupa <i>AssociationRule</i> yang telah ada pada <i>workspace</i> 7. Pengguna memilih <i>input</i> yang ketiga berupa <i>ReadPNML</i> yang telah ada pada <i>workspace</i> 8. Pengguna menekan tombol <i>Start</i> 9. Sistem melakukan perhitungan bobot dengan penilaian pakar yang terdapat pada data <i>AssociationRule</i> 10. Sistem menampilkan tabel Pelanggaran 11. Pengguna menekan tombol <i>Continue</i> 12. Sistem mengubah data pelanggaran ke bentuk <i>fuzzy</i> 13. Sistem menampilkan tabel <i>Fuzzy</i> 14. Pengguna menekan tombol <i>continue</i> 15. Sistem mencocokkan kombinasi pelanggaran dengan aturan asosiasi yang dihasilkan 16. Sistem menghasilkan data <i>fraud</i> beserta nilai <i>confidence</i> dan bobot <i>fraud</i>. 17. Sistem menampilkan tabel Hasil Deteksi <i>Fraud</i> 18. Pengguna menekan tombol <i>continue</i> 19. Sistem mengelompokkan pelanggaran ke kelas <i>fraud</i> 20. Sistem menampilkan tabel Hasil Pengelompokkan Fraud
- Kejadian Alternatif	Tidak ada
Kondisi Akhir	Sistem menampilkan tabel Hasil Pengelompokkan <i>Fraud</i>
Kebutuhan Khusus	Tidak ada



Gambar 4.16. Diagram Aktivitas Melakukan Pendeteksian Fraud dengan data aturan asosiasi



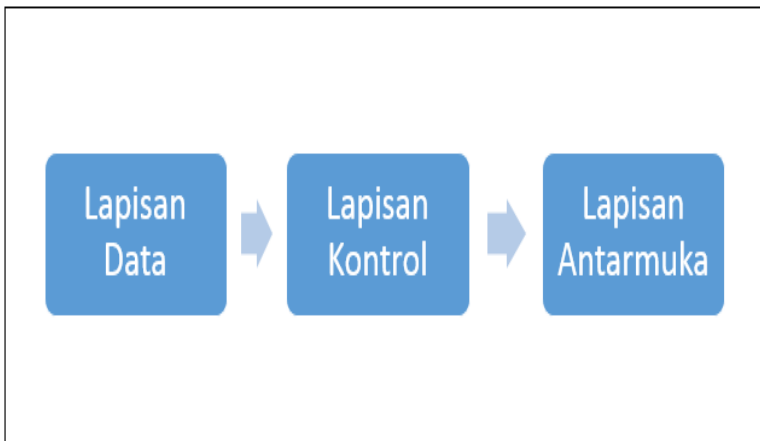
Gambar 4.17. Diagram Urutan Melakukan Pendeteksian *Fraud* dengan data aturan asosiasi

4.2. Perancangan Sistem

Penjelasan tahap perancangan perangkat lunak dibagi menjadi beberapa bagian yaitu perancangan diagram kelas, perancangan proses analisis, dan perancangan antarmuka.

4.2.1. Perancangan Diagram Kelas

Perancangan diagram kelas berisi rancangan dari kelas-kelas yang digunakan untuk membangun sistem. Pada subbab ini, hubungan dan perilaku antar kelas dan antar digambarkan dengan lebih jelas. Model yang digunakan dalam arsitektur ini ialah model tiga-lapis. Tiga lapisan pada arsitektur ini terdiri dari lapisan antarmuka, kontrol, dan pengaturan data. Hubungan antara 3 lapisan ini dapat dilihat pada Gambar 4.18. Lapisan kontrol merupakan penghubung antara lapisan antarmuka dengan lapisan data. Subbab ini dibagi menjadi tiga bagian, yaitu diagram kelas untuk lapisan antarmuka, kontrol, dan data. Hal ini sesuai dengan perancangan arsitektur sistem yang telah dibahas pada subbab 3.2.1.



Gambar 4.18. Rancangan Model Tiga-Lapis

4.2.1.1. Diagram Kelas Lapisan Antarmuka

Diagram kelas Lapisan Antarmuka dapat dilihat pada lampiran Diagram A.1. Terdapat enam kelas penyusun lapisan antarmuka ini. Keenam kelas tersebut adalah *CheckWrongDecision*, *FraudwithFuzzyARL2*, *countFraud*, *Fuzzy*, *ARLParameter2*, dan *TestingFraud*. Kegunaan *CheckWrongDecision* adalah untuk menampilkan tabel informasi mengenai transisi yang terdapat pada model proses SOP yang digunakan untuk melakukan pengecekan kesesuaian pengambilan keputusan antara *event log* dan model SOP. Kelas *FraudwithFuzzyARL2* adalah untuk menampilkan tiap hasil dan proses yang dilakukan dalam penghitungan bobot *fraud* dan pencarian aturan asosiasi. Kelas *countFraud* digunakan untuk menampilkan *form* untuk masukan jumlah pakar dan kepentingan pakar yang digunakan dalam perhitungan bobot *fraud*. Kelas *Fuzzy* digunakan untuk menampilkan tabel hasil fuzzifikasi dari tabel pelanggaran proses bisnis. Kelas *ARLParameter2* digunakan untuk menampilkan *form* masukan untuk nilai *threshold* yang digunakan dalam proses seleksi pada pencarian aturan asosiasi. Kelas *TestingFraud* berguna untuk menampilkan hasil pendeteksian *fraud* dengan menggunakan data aturan asosiasi yang dilakukan pengguna.

4.2.1.2. Diagram Kelas Lapisan Kontrol

Lampiran Diagram A.2 menunjukkan kelas-kelas penyusun lapisan kontrol. Beberapa kelas dari lapisan ini memiliki hubungan langsung dengan lapisan data. Sedangkan beberapa kelas lainnya memiliki hubungan langsung dengan lapisan antarmuka. Kelas-kelas penyusun lapisan ini, yaitu *ImportEventLog*, *ImportOWL*, *ImportPNML*, *ImportPNML2*, *DetectedSkippedEvents*, *ConformanceCheckAtribut*, *CheckWrongDecision*, *CountARL2*, dan *CountImportance*.

Kelas *ImportEventLogs* adalah untuk melakukan impor *event log* yang sebelumnya telah dipersiapkan oleh pengguna kemudian setelah selesai melakukan impor, akan ditampilkan pada

workspace ProM. Kelas *ImportOWL* adalah untuk melakukan impor *file ontology* yang telah dibuat sebelumnya yang akan ditampilkan pada *workspace* ProM. Kelas *Import PNML* adalah untuk melakukan impor model proses bisnis standar dalam bentuk *petri net* yang sebelumnya telah dipersiapkan oleh pengguna kemudian setelah selesai melakukan impor, akan ditampilkan pada *workspace* ProM. Kelas *ImportPNML2* adalah untuk melakukan model proses dalam bentuk *petri net extended* yang sebelumnya telah dipersiapkan oleh pengguna kemudian setelah selesai melakukan impor, akan ditampilkan pada *workspace* ProM. Kelas *DetectedSkippedEvents* digunakan untuk melakukan pengecekan dan perhitungan jumlah aktivitas yang dilewati untuk mengisi atribut pelanggaran *Skip Sequence* dan *Skip Decision*. Kelas *ConformanceCheckAtribut* digunakan untuk melakukan pengecekan kesesuaian terhadap atribut pada proses bisnis dan untuk menghitung jumlah pelanggaran untuk atribut pelanggaran *Throughput Time Min/Max*, *Wrong Resource*, *Wrong Duty Sequence/Decision/Combine*, dan *Wrong Pattern*. Kelas *CheckWrongDecision* adalah kelas yang berguna untuk melakukan pengecekan kesesuaian untuk pengambilan keputusan dan perhitungan jumlah pelanggaran atribut *Wrong Decision*. Kelas *CountARL2* digunakan untuk menghitung *support itemset*, seleksi *item*, dan pencarian aturan asosiasi dalam proses metode *FuzzyARL*. Kelas *CountImportance* digunakan untuk melakukan perhitungan bobot *fraud* untuk tiap pelanggaran proses bisnis

4.2.1.3. Diagram Kelas Lapisan Data

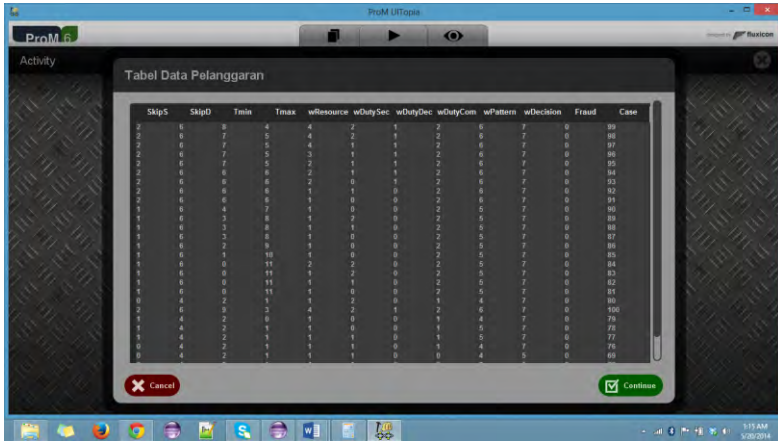
Diagram kelas pada lapisan data ditunjukkan oleh Diagram A.3. Lapisan data ini hanya berhubungan dengan modul kontrol saja. Lapisan ini hanya bisa diakses oleh lapisan kontrol. Lapisan antarmuka tidak dapat berkomunikasi secara langsung dengan lapisan ini. Diagram kelas ini berisi beberapa kelas yaitu *Fraud*, *ReadPNML*, *Transition*, *Decision*, *Ontologies*, dan *AssociationRule*.

Kelas *Fraud* merupakan kelas lapisan data yang menampung data-data tentang atribut pelanggaran yang dihasilkan dari proses pengecekan kesesuaian. Atributnya terdiri dari *Case*, *SkipSeq*, *SkipDec*, *Tmin*, *Tmax*, *WResource*, *WDutySeq*, *WDutyDec*, *WDutyCom*, *WPattern*, *WDecision*, *Fraud*, *maxSkipSeq*, *maxSkipDec*, *maxTmin*, *maxTmax*, *maxWResource*, *maxWDutySeq*, *maxWDutyDec*, *maxWDutyCom*, *maxWPattern*, dan *max WDecision*. Kelas *ReadPNML* merupakan kelas lapisan data yang menampung data-data tentang *petri net* model proses bisnis. Atributnya terdiri dari *places*, *transitions*, *arcs*, dan *decisions*. Kelas *Transition* merupakan kelas lapisan data yang menampung data-data tentang *ID*, *Name*, *Role*, *Resource*, *time*, dan *Decision*. Kelas *Decision* merupakan kelas lapisan data yang menampung data-data tentang *firstTransition*, *nextTransition*, *attribute*, *typeAttribute*, *ruleDecision*, *value*, *predicate*, dan *nextAttribute* yang digunakan untuk pengecekan kesesuaian pengambilan keputusan. Kelas *Ontologies* merupakan kelas lapisan data yang menampung data-data tentang *event log ontology* yang berisi *path* dari *file ontology* yang telah disediakan oleh pengguna. Kelas *AssociationRule* merupakan kelas lapisan data yang menampung data-data tentang *jumlahPakar*, *jumlahRules*, dan *tabelARL* yang digunakan untuk menampung aturan asosiasi yang dihasilkan.

4.2.2. Perancangan Antarmuka Pengguna

Bagian ini membahas mengenai perancangan antarmuka pada sistem. Hanya terdapat satu halaman utama antarmuka pada sistem ini, yaitu halaman tampilan kode sumber. Terdapat tiga jendela pendukung fitur fungsional sistem. Jendela tersebut yaitu jendela masukan pencarian, jendela informasi, dan jendela *metrics* berkas. Satu tambahan antarmuka dari Eclipse yaitu tampilan hasil pencarian pada bagian tampilan Eclipse.

4.2.2.1. Halaman Tampilan Tabel Data Pelanggaran



Gambar 4.19 Rancangan Halaman Tampilan Tabel Data Pelanggaran

Halaman ini merupakan tampilan atau melihat data pelanggaran yang dihasilkan dari tahap *conformance checking*. Rancangan halaman tabel data pelanggaran ini dapat dilihat pada Gambar 4.19. Penjelasan mengenai atribut-atribut yang terdapat pada halaman ini bisa dilihat pada Tabel 4.11.

Tabel 4.11 Spesifikasi Atribut Antarmuka Tabel Data Pelanggaran

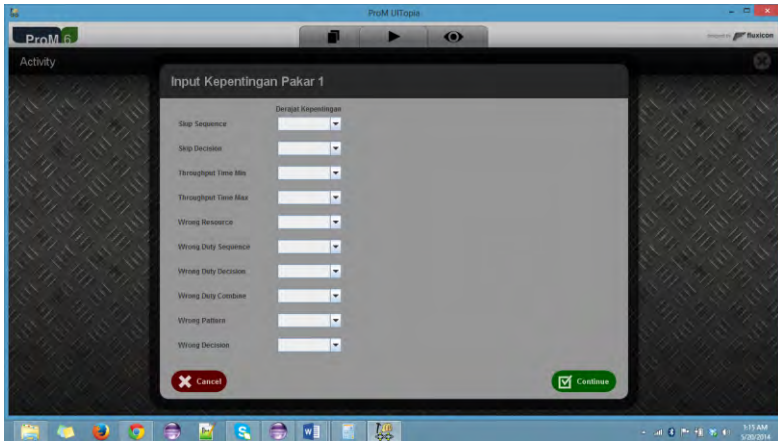
No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
1	Tabel Data Pelanggaran	<i>PromTable</i>	Menampilkan Data Pelanggaran	Tabel
2	<i>SkipS</i>	<i>Text</i>	Merepresentasikan data pelanggaran <i>SkipS</i>	<i>Integer</i>
3	<i>SkipD</i>	<i>Text</i>	Merepresentasikan data pelanggaran <i>SkipD</i>	<i>Integer</i>

4	<i>Tmin</i>	<i>Text</i>	Merepresentasikan data pelanggaran <i>Tmin</i>	<i>Integer</i>
5	<i>Tmax</i>	<i>Text</i>	Merepresentasikan data pelanggaran <i>Tmax</i>	<i>Integer</i>
6	<i>wResource</i>	<i>Text</i>	Merepresentasikan data pelanggaran <i>wResource</i>	<i>Integer</i>
7	<i>wDutySeq</i>	<i>Text</i>	Merepresentasikan data pelanggaran <i>wDutySeq</i>	<i>Integer</i>
8	<i>wDutyDec</i>	<i>Text</i>	Merepresentasikan data pelanggaran <i>wDutyDec</i>	<i>Integer</i>
9	<i>wDutyCom</i>	<i>Text</i>	Merepresentasikan data pelanggaran <i>wDutyCom</i>	<i>Integer</i>
10	<i>wPattern</i>	<i>Text</i>	Merepresentasikan data pelanggaran <i>wPattern</i>	<i>Integer</i>
11	<i>wDecision</i>	<i>Text</i>	Merepresentasikan data pelanggaran <i>wDecision</i>	<i>Integer</i>
12	<i>Fraud</i>	<i>Text</i>	Merepresentasikan nilai <i>Fraud</i>	<i>Integer</i>
13	<i>Case</i>	<i>Text</i>	Merepresentasikan nomor <i>case</i> proses bisnis	<i>Integer</i>

4.2.2.2. Halaman *Input Kepentingan Pakar*

Halaman ini muncul ketika pengguna mengeksekusi atau memanggil *Fraud Detection with Fuzzy Association Rule Learning Plugin*. Sebelum menjalankan perhitungan bobot *fraud*, terlebih dahulu pengguna memasukkan beberapa nilai *inputan* kepentingan atribut pelanggaran dari pakar sebagai parameter untuk menjalankan perhitungan tersebut. Rancangan halaman *Input Kepentingan Pakar* ini dapat dilihat pada Gambar 4.20. Penjelasan

mengenai atribut-atribut yang terdapat pada halaman ini bisa dilihat pada Tabel 4.12.



Gambar 4.20 Rancangan Antarmuka *Input Kepentingan Pakar*

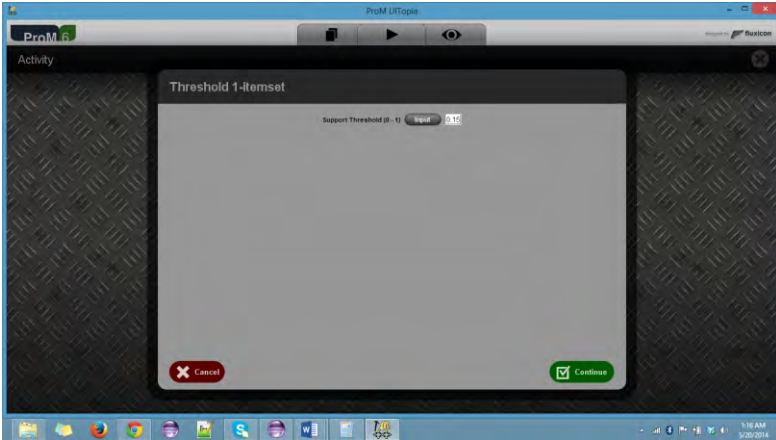
Tabel 4.12 Spesifikasi Atribut Antarmuka *Input Kepentingan Pakar*

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
1	<i>Skip Sequence</i>	<i>Combo Box</i>	Memasukkan nilai kepentingan atribut <i>Skip Sequence</i>	<i>String</i>
2	<i>Skip Decision</i>	<i>Combo Box</i>	Memasukkan nilai kepentingan atribut <i>Skip Decision</i>	<i>String</i>
3	<i>Throughput Time Min</i>	<i>Combo Box</i>	Memasukkan nilai kepentingan atribut <i>Throughput Time Min</i>	<i>String</i>
4	<i>Throughput Time Max</i>	<i>Combo Box</i>	Memasukkan nilai kepentingan atribut <i>Throughput Time Max</i>	<i>String</i>

5	<i>Wrong Resource</i>	<i>Combo Box</i>	Memasukkan nilai kepentingan atribut <i>Wrong Resource</i>	<i>String</i>
6	<i>Wrong Duty Sequence</i>	<i>Combo Box</i>	Memasukkan nilai kepentingan atribut <i>Wrong Duty Sequence</i>	<i>String</i>
7	<i>Wrong Duty Decision</i>	<i>Combo Box</i>	Memasukkan nilai kepentingan atribut <i>Wrong Duty Decision</i>	<i>String</i>
8	<i>Wrong Duty Combine</i>	<i>Combo Box</i>	Memasukkan nilai kepentingan atribut <i>Wrong Duty Combine</i>	<i>String</i>
9	<i>Wrong Pattern</i>	<i>Combo Box</i>	Memasukkan nilai kepentingan atribut <i>Wrong Pattern</i>	<i>String</i>
10	<i>Wrong Decision</i>	<i>Combo Box</i>	Memasukkan nilai kepentingan atribut <i>Wrong Decision</i>	<i>String</i>
11	<i>Continue</i>	<i>Button</i>	Menyimpan data kepentingan pada sistem.	<i>Action</i>

4.2.2.3. Halaman *Input Threshold*

Halaman ini muncul ketika pengguna mengeksekusi atau memanggil *Fraud Detection with Fuzzy Association Rule Learning Plugin*. Sebelum menjalankan proses seleksi *itemset*, terlebih dahulu pengguna memasukkan nilai *threshold* sebagai parameter untuk menjalankan proses seleksi *itemset* tersebut. Rancangan *Input Threshold* ini dapat dilihat pada Gambar 4.21. Penjelasan mengenai atribut-atribut yang terdapat pada halaman ini bisa dilihat pada Tabel 4.13.



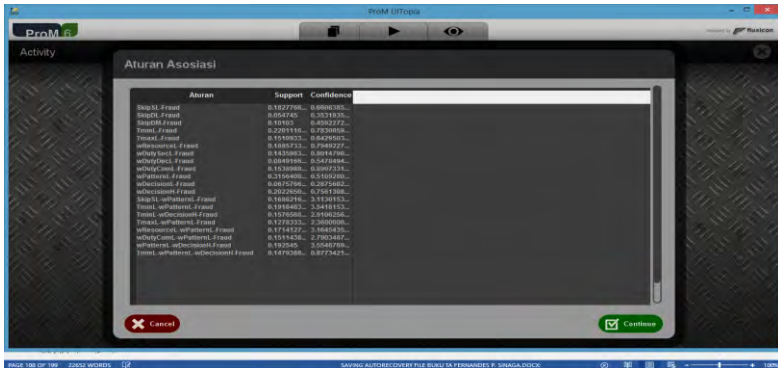
Gambar 4.21 Rancangan Antarmuka Halaman *Input Threshold*

Tabel 4.13 Spesifikasi Atribut Antarmuka Halaman *Input Threshold*

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
1	<i>Support Threshold</i>	<i>Text</i>	Memberikan nilai <i>threshold</i> untuk seleksi <i>support item</i>	<i>Double</i>
2	<i>Input</i>	<i>Button</i>	Menyimpan nilai <i>threshold</i> ke dalam sistem.	<i>Action</i>

4.2.2.4. Halaman Aturan Asosiasi

Halaman ini muncul ketika pengguna mengeksekusi atau memanggil *Fraud Detection with Fuzzy Association Rule Learning Plugin*. Halaman ini menampilkan hasil pencarian aturan asosiasi yang dilakukan dalam *plugin* yang dieksekusi. Rancangan halaman Rancangan Antarmuka Aturan Asosiasi ini dapat dilihat pada Gambar 4.22. Penjelasan mengenai atribut-atribut yang terdapat pada halaman ini bisa dilihat pada Tabel 4.14.



Gambar 4.22. Rancangan Antarmuka Aturan Asosiasi

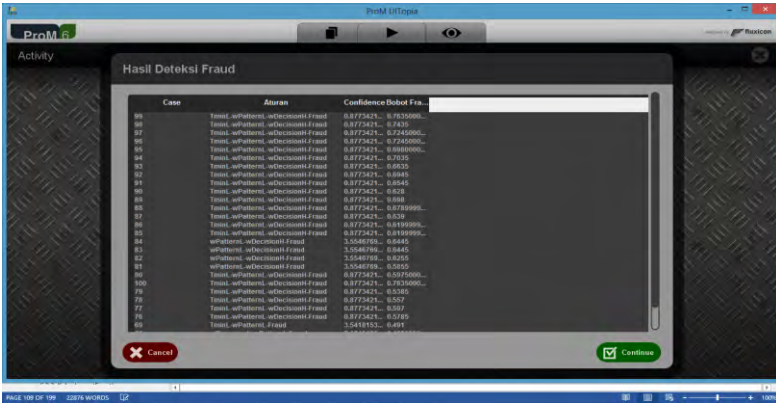
Tabel 4.14. Spesifikasi Atribut Antarmuka Rancangan Antarmuka Aturan Asosiasi

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
1	Aturan Asosiasi	PromTable	Menampilkan aturan asosiasi	String
2	Aturan	String	Menampilkan aturan yang terbentuk	String
3	Support	Double	Menampilkan nilai support untuk aturan asosiasi	Double
4	Confidence	Double	Menampilkan nilai confidence untuk aturan asosiasi	Double

4.2.2.5. Halaman Hasil Deteksi Fraud

Halaman ini muncul ketika pengguna mengeksekusi atau memanggil *Detect Fraud with Association Rule Data*. Halaman ini menampilkan hasil pendeteksian *fraud* pada proses bisnis dengan menggunakan data aturan asosiasi. Rancangan halaman Rancangan

Antarmuka Hasil Deteksi *Fraud* ini dapat dilihat pada Gambar 4.23. Penjelasan mengenai atribut-atribut yang terdapat pada halaman ini bisa dilihat pada Tabel 4.15.



Gambar 4.23. Rancangan Antarmuka Hasil Deteksi Fraud

Tabel 4.15 Spesifikasi Atribut Antarmuka Rancangan Antarmuka Hasil Deteksi Fraud

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
1	<i>Case</i>	<i>String</i>	Menampilkan aturan asosiasi	<i>String</i>
2	<i>Aturan</i>	<i>String</i>	Menampilkan aturan terdeteksi untuk suatu <i>case</i>	<i>String</i>
3	<i>Confidence</i>	<i>Double</i>	Menampilkan nilai <i>confidence</i> untuk <i>case</i> yang dideteksi	<i>Double</i>
4	<i>Bobot Fraud</i>	<i>Double</i>	Menampilkan nilai bobot <i>fraud</i> untuk <i>case</i> yang terdeteksi	<i>Double</i>

BAB V IMPLEMENTASI

Bab ini membahas tentang implementasi dari perancangan sistem. Bab ini berisi proses implementasi dari setiap kelas pada semua modul dan berkas nyata *plugin.xml*. Bahasa pemrograman yang digunakan adalah bahasa pemrograman Java.

5.1. Implementasi Lapisan Antarmuka

Lapisan antarmuka merupakan lapisan yang bertanggung jawab dengan tampilan sistem kepada pengguna. Pada bagian ini akan dijelaskan secara terperinci mengenai implementasi kelas-kelas yang berada dalam lapisan ini. Urutan penjelasan kelas dikelompokkan berdasarkan modul-modul.

5.1.1. Kelas CheckWrongDecision

Kegunaan `CheckWrongDecision` adalah untuk menampilkan tabel informasi mengenai transisi yang terdapat pada model proses SOP yang digunakan untuk melakukan pengecekan kesesuaian pengambilan keputusan antara *event log* dan model proses bisnis standar. Di dalam kelas ini terdapat satu fungsi utama yang merealisasikan tampilan sehingga muncul pada aplikasi ProM. Fungsi ini menerima masukan berupa *PluginContext*, *ReadPNML*, *InsertFraudData*, dan *Ontologies*. Fungsi ini mengambil data aturan *decision* pada transisi dari model *petri net* dan dimasukkan ke dalam tabel. Di dalam kelas tersebut membangun *PromTable* dan menyesuaikannya dengan *JPanel* agar dapat tampil pada ProM. Rincian fungsi tersebut dapat dilihat pada Kode Sumber 5.1

```
public InsertFraudData checkDecision(final UIPluginContext
context, ReadPNML pnml, InsertFraudData fraud1, Ontologies
owlFiles)
{
    JPanel panel = new JPanel();
    tableTransition = new
Object[pnml.decisions.size()][columnName.length];
```



```

    QueryFraudDetection wrongDec = new
    QueryFraudDetection();
    List<String> caseFrauds = new ArrayList<String>();
    tableModelTransition2 = new
    DefaultTableModel(table, columnname);

    for(int i=0; i<tableTransition.length; i++)
    {
        for(int j=0; j<columnname.length; j++)
        {
            tableModelTransition2.setValueAt(tableTransition[i][j],
            i, j);
        }
    }
    ProMTable Ptable = new ProMTable(tableModelTransition2);
    Ptable.setPreferredSize(new Dimension(1000, 500));
    Ptable.setAutoResizeMode(0);
    panel.add(Ptable);

    context.showConfiguration("Tabel Transisi", panel);

    return fraud1;
}

```

Kode Sumber 5.1 Fungsi `checkDecision()` untuk menampilkan data aturan transisi pada aplikasi ProM

5.1.2. Kelas `FraudwithFuzzyARL2`

Kelas ini digunakan untuk menampilkan tiap hasil dan proses yang dilakukan dalam penghitungan bobot *fraud* dan pencarian aturan asosiasi. Proses dan hasil aturan asosiasi dibentuk dalam *PromTable* dan disesuaikan dengan *JPanel* untuk dapat ditampilkan pada ProM.

5.1.3. Kelas `countFraud`

Kelas ini digunakan untuk menampilkan *form* untuk masukan jumlah pakar dan kepentingan pakar yang digunakan dalam perhitungan bobot *fraud*. Di dalam kelas ini terdapat dua fungsi yang merealisasikan tampilan *form* masukan jumlah pakar dan kepentingan pakar sehingga muncul pada aplikasi ProM. Fungsi ini mengambil data dari *textbox* dan *combo box* yang

diinputkan pakar untuk diolah dalam perhitungan bobot *fraud*. Rincian fungsi tersebut dapat dilihat pada Kode Sumber 5.2

```

public JPanel InputJumlahPakar(final Integer[]
jumlahPakar)
{
    JPanel panel = new JPanel();
    JLabel jLabel1 =
SlickerFactory.instance().createLabel("Jumlah Pakar: ");
    JButton jButton1 =
SlickerFactory.instance().createButton("Input");

    GroupLayout layout = new GroupLayout(panel);
    panel.setLayout(layout);
    layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignme
nt.LEADING)
        .addGroup(layout.createSequentialGroup())
            .addGap(38, 38, 38)
            .addComponent(jLabel1)
            .addComponent(jButton1)
            .addGap(32, 32, 32)
            .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE, 118,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(142, Short.MAX_VALUE)
        );
    layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignme
nt.LEADING)
        .addGroup(layout.createSequentialGroup())
            .addGap(37, 37, 37)
            .addGroup(layout.createParallelGroup(javax.s
wing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel1)
                .addComponent(jButton1)
                .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(30, Short.MAX_VALUE)
            );

        jButton1.addActionListener(new ActionListener() {

            public void actionPerformed(ActionEvent
arg0) {

```

```

        // TODO Auto-generated method stub
        jumlahPakar[0] =
Integer.parseInt(jTextField1.getText());
    }
});

panel.setLayout(layout);
return panel;
}

```

Kode Sumber 5.2. Fungsi InputJumlahPakar () untuk menampilkan *textbox* untuk masukan jumlah pakar pada aplikasi ProM

5.1.4. Kelas Fuzzy

Kelas Fuzzy digunakan untuk menampilkan tabel hasil fuzzifikasi dari tabel pelanggaran proses bisnis. Tampilannya berupa tabel dengan beberapa atribut seperti *SkipSL, SkipSM, SkipSH, SkipDL, SkipDM, SkipDH, TminL, TminM, TminH, TmaxL, TmaxM, TmaxH, wResourceL, wResourceM, wResourceH, wDutySecL, wDutySecM, wDutySecH, wDutyDecL, wDutyDecM, wDutyDecH, wDutyComL, wDutyComM, wDutyComH, wPatternL, wPatternM, wPatternH, wDecisionL, wDecisionM, wDecisionH, Fraud*. Di dalam Fungsi ini dapat dilihat pada Kode Sumber 5.3.

```

public JPanel FuzzyTabel(Object[][] tableFraud, String[]
columnName, Object[][] tableFuzzy)
{
    JPanel panel3 = new JPanel();

    //isi tabel dengan nilai fuzzy
    PercenTabel(tableFraud, columnName);

    //tableFuzzy = new
Object[tableDummy.length][columnName2.length];

    for(int i=0;i<tableDummy.length;i++)
    {
        int k=0;
        for(int j=0;j<columnName2.length;j++)
        {

            if(j==columnName2.length-1)

```

```

        {
            tableFuzzy[i][j] = tableDummy[i][k];
        }
        else if(j%3==0)
        {
            tableFuzzy[i][j] =
fuzzyMin((Double)tableDummy[i][k]);
        }
        else if (j%3==1)
        {
            tableFuzzy[i][j] =
fuzzyMid((Double)tableDummy[i][k]);
        }
        else if(j%3==2)
        {
            tableFuzzy[i][j] =
fuzzyHigh((Double)tableDummy[i][k]);
            k++;
        }
    }
}

Object[][] tabel2 = new Object[tableFuzzy.length][];
Object[][] tabel3 = new Object[tableFuzzy.length][];
//public DefaultTableModel tableModelPercent = new
DefaultTableModel(tabel2, ffa.columnsName);
DefaultTableModel tableModelFuzzy = new
DefaultTableModel(tabel2, columnsName2);
//isi tabel fuzzy ke tabel model
for(int i=0; i<tableFuzzy.length; i++)
{
    for(int j=0; j<tableModelFuzzy.getColumnCount(); j++)
    {
        tableModelFuzzy.setValueAt(tableFuzzy[i][j], i,
j);
    }
}
PromTable Ptabel1 = new PromTable(tableModelFuzzy);
Ptabel1.setPreferredSize(new Dimension(1000, 500));
Ptabel1.setAutoResizeMode(0);
panel3.add(Ptabel1);
//context.showConfiguration("Tabel Fraud", panel2);

return panel3;
}

```

Kode Sumber 5.3 Fungsi FuzzyTabel () pada Kelas Fuzzy

5.1.5. Kelas `ARLParameter2`

Kelas `ARLParameter2` ini digunakan untuk menampilkan *form* masukan untuk nilai *threshold* yang digunakan dalam proses seleksi pada pencarian aturan asosiasi.

5.1.6. Kelas `TestingFraud`

Kelas `TestingFraud` ini digunakan untuk menampilkan hasil pendeteksian *fraud* dengan menggunakan data aturan asosiasi yang dilakukan pengguna. Tampilannya berupa tabel dengan beberapa atribut seperti *Case*, *Aturan*, *Confidence*, *Bobot* dan *Kelas*. Di dalam kelas tersebut membangun *PromTable* dan menyesuaikannya dengan *JPanel* agar dapat tampil pada *ProM*.

5.2. Implementasi Lapisan Kontrol

Lapisan kontrol merupakan lapisan yang bertanggung jawab dengan tingkah laku sistem. Lapisan ini bertugas menghubungkan lapisan data dengan lapisan antarmuka. Pada bagian ini akan dijelaskan secara terperinci mengenai implementasi kelas-kelas yang berada dalam lapisan ini. Urutan penjelasan kelas dikelompokkan berdasarkan modul-modul.

5.2.1. Kelas `ImportEventLog`

Kelas `ImportEventLog` adalah untuk melakukan impor *event log* yang sebelumnya telah dipersiapkan oleh pengguna kemudian setelah selesai melakukan impor, akan ditampilkan pada *workspace* *ProM*. Fungsi `importFromStream` dapat dilihat pada Kode Sumber 5.4.

```
protected Object importFromStream(PluginContext context,
    InputStream input, String filename, long fileSizeInBytes,
    XFactory factory)
    throws Exception {
    context.getFutureResult(0).setLabel(filename);
```

```

        XParser parser;
        if (filename.toLowerCase().endsWith(".xes")
|| filename.toLowerCase().endsWith(".xez")
|| filename.toLowerCase().endsWith(".xes.gz")) {
            parser = new XesXmlParser(factory);
        } else {
            parser = new XMxmlParser(factory);
        }
        Collection<XLog> logs = null;
        try {
            logs = parser.parse(new
XContextMonitoredInputStream(input, fileSizeInBytes,
context.getProgress()));
        } catch (Exception e) {
            logs = null;
        }
        if (logs == null) {
            // try any other parser
            for (XParser p :
XParserRegistry.instance().getAvailable()) {
                if (p == parser) {
                    continue;
                }
                try {
                    logs = p.parse(new
XContextMonitoredInputStream(input, fileSizeInBytes,
context.getProgress()));
                    if (logs.size() > 0)
{
                        break;
                    }
                } catch (Exception e1) {
                    logs = null;
                }
            }
        }

        if (logs == null || logs.size() == 0) {
            throw new Exception("No processes
contained in log!");
        }

        XLog log = logs.iterator().next();
        if
(XConceptExtension.instance().extractName(log) == null) {

```

```

        XConceptExtension.instance().assignName(log,
"Anonymous log imported from " + filename);
    }

    if (log.isEmpty()) {
        throw new Exception("No process
instances contained in log!");
    }

    if (context != null) {
        context.getFutureResult(0).setLabel(XConceptExtensi
on.instance().extractName(log));
    }

    return log;
}

```

Kode Sumber 5.4. Fungsi *importFromStream()* pada Kelas *ImportEventLog*

5.2.2. Kelas *ImportOWL*

Kelas *ImportOWL* adalah untuk melakukan impor *file ontology* yang telah dibuat sebelumnya yang akan ditampilkan pada *workspace* ProM. Fungsi utama dari kelas *ImportOWL* adalah *importFromStream* yang dapat dilihat pada Kode Sumber 5.5.

```

protected Ontologies importFromStream(PluginContext
context, InputStream input, String filename, long
fileSizeInBytes)
{
    System.out.println("Path Asli: " +
this.getFile().getPath());
    Ontologies ont = new
Ontologies(this.getFile().getPath());
    return ont;
}

```

Kode Sumber 5.5 Fungsi *importFromStream()* pada kelas *ImportOWL*

5.2.3. Kelas `ImportPNML`

Kelas `ImportPNML` adalah untuk melakukan impor model proses bisnis standar dalam bentuk *petri net* yang sebelumnya telah dipersiapkan oleh pengguna kemudian setelah selesai melakukan impor, akan ditampilkan pada *workspace* ProM. Fungsi `importFromStream` dapat dilihat pada Kode Sumber 5.6.

```

protected Object importFromStream(PluginContext context,
InputStream input, String filename, long fileSizeInBytes)
    throws Exception {
    // TODO Auto-generated method stub
    PNMLImportUtils utils = new
PNMLImportUtils();
    Pnml pnml =
utils.importPnmlFromStream(context, input, filename,
fileSizeInBytes);
    //Pnml pnml =
utils.importPnmlFromStream(context, input, filename,
fileSizeInBytes);
    if (pnml == null) {
        /*
        * No PNML found in file. Fail.
        */
        return null;
    }

    PetrinetGraph net =
PetrinetFactory.newPetrinet(pnml.getLabel() + " (imported
from " + filename + ")");
    return utils.connectNet(context, pnml, net,
this.getFile().getAbsolutePath());
}

```

Kode Sumber 5.6 Fungsi `importFromStream` pada Kelas `ImportPNML`

5.2.4. Kelas `ImportPNML2`

Kelas `ImportPNML2` adalah untuk melakukan model proses dalam bentuk *petri net extended* yang sebelumnya telah dipersiapkan oleh pengguna kemudian setelah selesai melakukan impor, akan ditampilkan pada *workspace* ProM.

5.2.5. Kelas DetectedSkippedEvents

Kelas DetectedSkippedEvents adalah untuk melakukan pengecekan dan perhitungan jumlah aktivitas yang dilewati untuk mengisi atribut pelanggaran *Skip Sequence* dan *Skip Decision*.

5.2.6. Kelas ConformanceCheckAtribut

Kelas ConformanceCheckAtribut adalah kelas yang berguna untuk melakukan pengecekan kesesuaian terhadap atribut pada proses bisnis dan untuk menghitung jumlah pelanggaran untuk atribut pelanggaran *Throughput Time Min/Max*, *Wrong Resource*, *Wrong Duty Sequence/Decision/Combine*, dan *Wrong Pattern*. Rincian dari fungsi *checkConformance* dapat dilihat pada Kode Sumber 5.7.

```

public void checkConformance(InsertFraudData fraud1)
{
    int counter=0;
    boolean flag=false;
    String temp="";
    int tmin=0;
    int tmax=0;
    String role="";
    int wresource=0;
    int dutyS=0;
    int dutyD=0;
    int wDutyS=0;
    int wDutyD=0;
    int wDutyC=0;
    int wPattern=0;
    String CaseID = "";
    String next = "";
    List<String> checkedEvents = new ArrayList<String>();

    System.out.println("Seq: "+SeqTransitions.size()+" --
Dec: "+DecTransitions.size());
    for(int j=0;j<tableLog.length-1;j++)
    {
        //fungsi mendapatkan role
        boolean cek = false;

        for(int i=0;i<c+1;i++)
        {

```

```

        if(tableLog[j][3].equals(tableRole[i][0]))
        {
            role=(String)tableRole[i][1];
            cek=true;
        }
    }

    if(cek==false)
    {
        role="Unidentified";
    }
    dutyD=0;
    dutyS=0;
    ////////////////////////////////////////////fungsi
menghitung duty

    String simpan="";
    if(!checkedEvents.contains(tableLog[j][3]))
    {
for(int i = 0; i < checkedEvents.size(); i++)
    {
        }
        boolean dec =
checkDecision((String)tableLog[j][1]);
        int count = 0;
        //fungsi menghitung wrong duty
        for(int p=j;p<tableLog.length-1;p++)
        {
            count++;
            if(simpan=="")
            {
                simpan=(String)tableLog[p][0];
            }

            if(tableLog[p][0].equals(simpan))
            {

                if(tableLog[j][3].equals(tableLog[p][3])&&!(tableLog[j][
1].equals(tableLog[p][1])))
                    }
                boolean dec2 =
checkDecision((String)tableLog[p][1]);

                if(dec2==true &&
dec==true)
                {
                    wDutyD++;
                }
            }
        }
    }

```

```

else if((dec2==true
&& dec==false) || (dec2==false && dec==true))
{
    wDutyC++;
}
else if(dec2==false
&& dec==false)
{
    wDutyS++;
}
}
else
{
    break;
}
}
checkedEvents.add((String) tableLog[j][3]);
if(count == 2)
{
    //System.out.println("Ganti Case");
    checkedEvents.clear();
}
}
else System.out.println("Wes Onok");
//////////fungsi
menghitung duty
if(temp=="")
{
    temp=(String) tableLog[j][0];

    for(int k=0;k<tableTransition.length;k++)
    {

        if(tableLog[j][1].equals(tableTransition[k][0]))
        {
            //fungsi menghitung tmin

            if((Long)tableLog[j][2]<((Integer)tableTransition[k][3]-
2))
            {
                tmin++;
            }
            //fungsi menghitung tmax
            else
            if((Long)tableLog[j][2]>((Integer)tableTransition[k][3]+2)
)

```

```

        {
            tmax++;
        }
        //fungsi menghitung wresource
    if (role.equals(tableTransition[k][1]))
        {
            continue;
        }
        else
        {
            wresource++;
        }
    }
}
else
{
    if (j < tableLog.length - 2)
        {
            boolean cekPattern = false;
            boolean cekPattern2 = false;
            boolean cekPattern3 = false;
            String[] belum = new
String[5];
            String[] sudah = new
String[5];
            int ss = 0;
            int sb = 0;
            //hitung wrong pattern
            for (int x = 0; x < c2; x++)
            {
                if (tableLog[j][1].equals(tableTransition3[x][0]))
                    {
                        belum[sb] = (String) tableTransition3[x][4];
                        if (tableLog[j - 1][1].equals(tableTransition3[x][4]))
                            cekPattern = true;
                    }
                    }
            }
            for (int x = 0; x < c3; x++)
            {
                if (tableLog[j][1].equals(tableTransition4[x][0]))
                    {

```

```

belum[ss]=(String) tableTransition4[x][4];

if(tableLog[j+1][1].equals(tableTransition4[x][4]))
    {
cekPattern2=true;
                                                                    break;
    }
    }
    }

    if(cekPattern==false)
    {
        wPattern++;
    }
}
for(int
k=0;k<tableTransition.length;k++)
{

    if(tableLog[j][1].equals(tableTransition[k][0]))
    {
                                                                    //fungsi menghitung
tmin
        if((Long)tableLog[j][2]<((Integer)tableTransition[k][3]-
2))
            {
                tmin++;
            }
                                                                    //fungsi menghitung
tmax
            else
if((Long)tableLog[j][2]>((Integer)tableTransition[k][3]+2)
)
            {
                tmax++;
            }

wresource
        if(role.equals(tableTransition[k][1]))
        {
            continue;
        }
        else
        {
            wresource++;
        }
    }
}

```

```

    }
    if(!temp.equals(tableLog[j+1][0]))
    {
        //fungsi menggabungkan data fraud
        for(int
z=0;z<fraud1.frauds.size();z++)
        {
            if(tableLog[j-
1][0].equals(fraud1.frauds.get(z).getCase()))
            {
                Fraud      =      new
fraud(fraud1.frauds.get(z).getCase(),
fraud1.frauds.get(z).getSkipSeq(),
fraud1.frauds.get(z).getSkipDec(), tmin, tmax, wresource,
wDutyS, wDutyD, wDutyC, wPattern, 0, 0);
            }
            frauds.add(Fraud);

            temp="";
            tmin=0;
            tmax=0;
            wresource=0;
            wDutyC=0;
            wDutyD=0;
            wDutyS=0;
            counter++;
            wPattern=0;
            checkedEvents.clear();
        }
    }
    CaseID = (String) tableLog[j][0];
}
//fungsi menggabungkan data fraud
frauds.add(Fraud);
temp="";
tmin=0;
tmax=0;
wresource=0;
wDutyC=0;
wDutyD=0;
wDutyS=0;
wPattern=0;
}

```

**Kode Sumber 5.7. Fungsi checkConformance pada kelas
CheckConformanceAtribut**

5.2.7. Kelas `CheckWrongDecision`

Kelas `CheckWrongDecision` adalah kelas yang berguna untuk melakukan pengecekan kesesuaian untuk pengambilan keputusan dan perhitungan jumlah pelanggaran atribut *Wrong Decision*.

5.2.8. Kelas `CountARL2`

Kelas `CountARL2` digunakan untuk menghitung *support itemset*, seleksi *item*, dan pencarian aturan asosiasi dalam proses metode *FuzzyARL*. Dalam kelas ini terdapat beberapa fungsi yang digunakan seperti fungsi *countSupport*, *selection*, *select*, dan sebagainya.

5.2.9. Kelas `CountImportance`

Kelas `CountImportance` adalah kelas yang digunakan untuk melakukan perhitungan bobot *fraud* untuk tiap pelanggaran proses bisnis. Dalam kelas ini terdapat beberapa fungsi yang digunakan seperti fungsi *countWeight*, *countRating*, *countFraud*, dan sebagainya.

5.3. Implementasi Lapisan Data

Lapisan ini hanya berhubungan dengan kelas-kelas pada lapisan kontrol. Lapisan ini berfungsi sebagai penyimpan data.

5.3.1. Kelas `Fraud`

Kelas `Fraud` merupakan kelas lapisan data yang menampung data-data tentang atribut pelanggaran yang dihasilkan dari proses pengecekan kesesuaian. Atributnya terdiri dari *Case*, *SkipSeq*, *SkipDec*, *Tmin*, *Tmax*, *WResource*, *WDutySeq*, *WDutyDec*, *WDutyCom*, *WPattern*, *WDecision*, *Fraud*, *maxSkipSeq*, *maxSkipDec*, *maxTmin*, *maxTmax*, *maxWResource*, *maxWDutySeq*, *maxWDutyDec*, *maxWDutyCom*, *maxWPattern*, dan *maxWDecision*.

5.3.2. Kelas ReadPNML

Kelas `ReadPNML` merupakan kelas lapisan data yang menampung data-data tentang *petri net* model proses bisnis. Atributnya terdiri dari *places*, *transitions*, *arcs*, dan *decisions*.

5.3.3. Kelas Transition

Kelas `Transition` merupakan kelas lapisan data yang menampung data-data tentang *ID*, *Name*, *Role*, *Resource*, *time*, dan *Decision*.

5.3.4. Kelas Decision

Kelas `Decision` merupakan kelas lapisan data yang menampung data-data tentang *firstTransition*, *nextTransition*, *attribute*, *typeAttribute*, *ruleDecision*, *value*, *predicate*, dan *nextAttribute* yang digunakan untuk pengecekan kesesuaian pengambilan keputusan.

5.3.5. Kelas Ontologies

Kelas `Ontologies` merupakan kelas lapisan data yang menampung data-data tentang *event log ontology* yang berisi *path* dari *file ontology* yang telah disediakan oleh pengguna.

5.3.6. Kelas AssociationRule.

Kelas `AssociationRule` merupakan kelas lapisan data yang menampung data-data tentang *jumlahPakar*, *jumlahRules*, dan *tabelARL* yang digunakan untuk menampung aturan asosiasi yang dihasilkan.

5.4. Implementasi Antarmuka Pengguna

Implementasi tampilan antarmuka pengguna pada Eclipse PDE dilakukan dengan menggunakan komponen pada SWT Widgets. Berikut ini akan dijelaskan mengenai implementasi tampilan antarmuka pengguna yang terdapat pada *plugin*.

5.4.1. Halaman Tampilan Tabel Data Pelanggaran

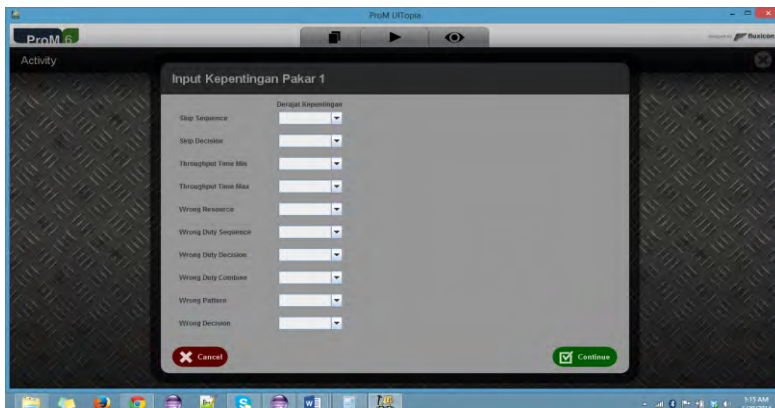
Halaman ini merupakan tampilan atau melihat data pelanggaran yang dihasilkan dari tahap *conformance checking*. Di dalam halaman ini berguna untuk melihat atribut yang dilanggar beserta jumlah pelanggaran yang dilakukan. Rancangan tampilan tabel data pelanggaran ini dapat dilihat pada Gambar 5.1.

SkipS	SkipD	Tmin	Tmax	wResource	wDutySec	wDutyDec	wDutyCom	wPattern	wDecision	Fraud	Case
2	0	3	4	4	2	1	2	0	7	0	99
2	0	7	5	4	2	1	2	0	7	0	99
2	0	7	5	4	1	1	2	0	7	0	99
2	0	7	5	3	1	1	2	0	7	0	99
3	0	7	5	2	1	1	2	0	7	0	95
2	0	6	6	2	1	1	2	0	6	0	94
2	0	6	6	2	0	1	2	0	6	0	92
2	0	6	6	2	0	1	2	0	6	0	92
2	0	6	6	1	0	2	0	0	7	0	91
2	0	6	7	1	0	2	0	0	7	0	90
1	0	3	0	1	2	0	2	5	7	0	88
1	0	3	0	1	1	0	2	5	7	0	86
1	0	3	0	1	0	0	2	5	7	0	87
1	0	2	0	1	0	0	2	5	7	0	86
1	0	1	10	1	0	0	2	5	7	0	85
1	0	0	11	2	2	0	2	5	7	0	84
1	0	0	11	4	2	0	2	5	7	0	83
1	0	0	11	1	1	0	2	5	7	0	82
1	0	0	11	0	0	0	2	5	7	0	81
0	2	2	1	1	2	0	1	4	7	0	80
2	0	2	3	4	2	0	2	6	7	0	100
1	4	2	0	0	0	0	1	4	7	0	76
1	4	2	1	1	0	0	1	5	7	0	75
1	4	2	1	1	1	0	1	5	7	0	72
0	4	2	1	1	1	0	1	4	7	0	76
0	4	2	1	1	1	0	0	4	6	0	69

Gambar 5.1 Tampilan Tabel Data Pelanggaran

5.4.2. Halaman Input Kepentingan Pakar

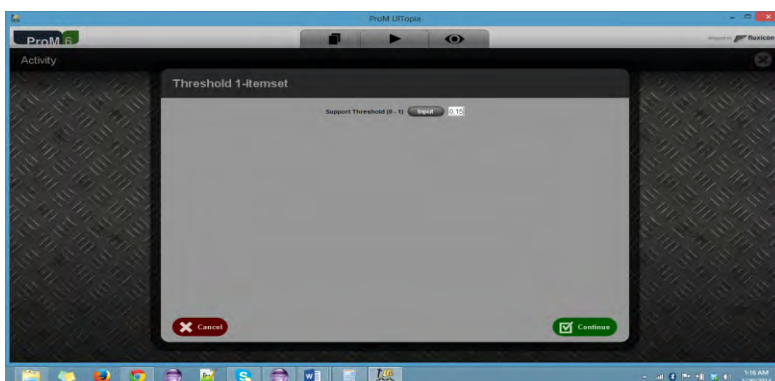
Halaman ini muncul ketika pengguna mengeksekusi atau memanggil *Fraud Detection with Fuzzy Association Rule Learning Plugin*. Sebelum menjalankan perhitungan bobot *fraud*, terlebih dahulu pengguna memasukkan beberapa nilai *inputan* kepentingan atribut pelanggaran dari pakar sebagai parameter untuk menjalankan perhitungan tersebut. Rancangan halaman *Input Kepentingan Pakar* ini dapat dilihat pada Gambar 5.2.



Gambar 5.2 Tampilan *Input Kepentingan Pakar*

5.4.3. Halaman *Input Threshold*

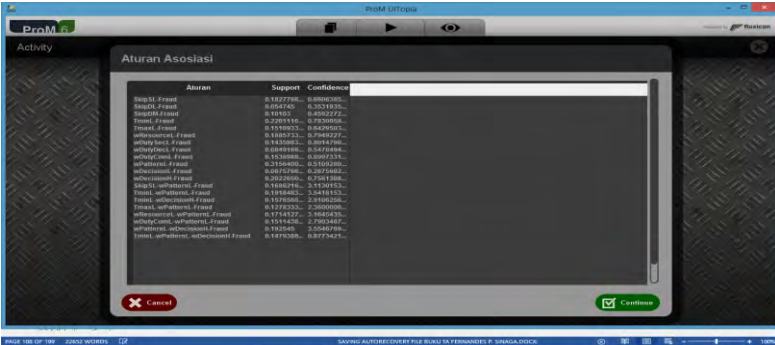
Halaman ini muncul ketika pengguna mengeksekusi atau memanggil *Fraud Detection with Fuzzy Association Rule Learning Plugin*. Sebelum menjalankan proses seleksi *itemset*, terlebih dahulu pengguna memasukkan nilai *threshold* sebagai parameter untuk menjalankan proses seleksi *itemset* tersebut. Tampilan *Input Threshold* dapat dilihat pada Gambar 5.3.



Gambar 5.3 Tampilan *Input Threshold*

5.4.4. Halaman Aturan Asosiasi

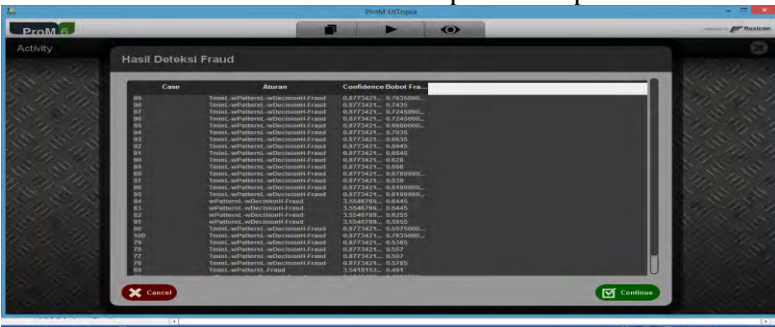
Halaman ini muncul ketika pengguna mengeksekusi atau memanggil *Fraud Detection with Fuzzy Association Rule Learning Plugin*. Halaman ini menampilkan hasil pencarian aturan asosiasi yang dilakukan dalam *plugin* yang dieksekusi. Rancangan halaman Rancangan Antarmuka Aturan Asosiasi ini dapat dilihat pada Gambar 5.4.



Gambar 5.4. Tampilan Halaman Aturan Asosiasi

5.4.5. Halaman Hasil Deteksi *Fraud*

Halaman ini muncul ketika pengguna mengeksekusi atau memanggil *Detect Fraud with Association Rule Data*. Halaman ini menampilkan hasil pendeteksian *fraud* pada proses bisnis dengan menggunakan data aturan asosiasi. Rancangan halaman Rancangan Antarmuka Hasil Deteksi *Fraud* ini dapat dilihat pada Gambar 5.5.



Gambar 5.5. Tampilan Halaman Hasil Deteksi *Fraud*

BAB VI

PENGUJIAN DAN EVALUASI

Bab ini membahas pengujian dan evaluasi pada *plugin* yang dikembangkan. Pengujian yang dilakukan adalah pengujian terhadap kebutuhan fungsionalitas sistem dan kegunaan sistem. Pengujian fungsionalitas mengacu pada kasus penggunaan pada bab empat. Pengujian kegunaan program dilakukan dengan mengetahui tanggapan dari pengguna terhadap sistem. Hasil evaluasi menjabarkan tentang rangkuman hasil pengujian pada bagian akhir bab ini.

6.1. Lingkungan Pengujian

Lingkungan pengujian sistem pada pengerjaan Tugas Akhir ini dilakukan pada lingkungan dan alat kakas sebagai berikut:

Prosesor	: Intel Core i7-3610QM CPU @ 2.30GHz
Memori	: 8.00 GB
Jenis Device	: Laptop
Sistem Operasi	: Microsoft Windows 8 Enterprise 64 bit
IDE Eclipse	: Eclipse Kepler

6.2. Skenario Pengujian

Pada bagian ini akan dijelaskan tentang skenario pengujian yang dilakukan. Pengujian dilakukan dalam tiga tahap yaitu pengujian kebutuhan fungsionalitas, pengujian kegunaan sistem, dan pengujian perbandingan *metrics*. Pengujian kebutuhan fungsionalitas menggunakan metode kotak hitam (*black box*). Metode ini menekankan pada hasil keluaran sistem. Pengujian kegunaan dilakukan dengan menggunakan metode kuesioner dari responden (pengguna sistem).

Studi kasus yang digunakan ialah studi kasus proses pengadaan/*procurement Enterprise Resource Planning (ERP)* Untuk proses bisnisnya dapat dilihat pada lampiran Diagram A.4. Di dalam studi kasus tersebut, terdapat dua bagian penting yang

menjalankan proses bisnis, yaitu bagian *Purchase Requisition* (PR), yang mengurus permohonan persetujuan untuk pembelian barang. Kemudian ada bagian *Purchase Order*, yang mengurus proses pembelian barang berdasarkan PR yang telah disetujui dan termasuk dalam proses pemilihan *supplier* sampai persetujuan kontrak dengan pihak *supplier*.

Di dalamnya terdapat 25 aktivitas yang dikerjakan oleh masing-masing bagian. Aktivitas tersebut adalah *receiving purchase request, receive supplier data, fill supplier rating form, review supplier standards conditions, choose as eligible supplier, update supplier data, send supplier evaluation report, define material needed, define purchasing modal, define quantity, check contract availability, define order lead time, define vendor lead time, filter vendor based on lead time, filter vendor based on purchasing price, define vendor, define purchasing price, verify order, making contract, sending contract to supplier, negotiating with supplier, signing contract, sending document and other report to management and supplier, making purchase order, dan making failure report*. Proses dari aktivitas *receiving purchase request* sampai *verify order* merupakan proses yang dilakukan bagian *purchase requisition*. Dan proses yang dimulai dari aktivitas *making contract* sampai *making purchase order* atau *making failure report* merupakan proses yang dilakukan bagian *purchase order*.

6.2.1. Pengujian Fungsionalitas

Pengujian fungsionalitas sistem dilakukan secara mandiri dengan menyiapkan sejumlah skenario sebagai tolak ukur keberhasilan pengujian. Pengujian fungsionalitas dilakukan dengan mengacu pada kasus penggunaan yang telah dijelaskan pada subbab 4.1.5. Pengujian pada kebutuhan fungsionalitas dapat dijabarkan pada subbab berikut.

6.2.1.1. Pengujian Fitur Mengimpor *Event Log*

Pengujian fitur mengimpor *event log* dilakukan dengan beberapa *event log* yang telah disediakan. *Event log* tersebut memiliki ekstensi *.xes , *.xml dan *.mxml yang merupakan ekstensi standar untuk *event log*. Selain itu juga terdapat *file* dengan ekstensi lain yang tidak berhubungan dengan *event log*, tujuannya adalah untuk menguji apakah *file* tersebut dapat diimpor atau tidak. Rincian pengujian fitur ini dapat dilihat pada Tabel 6.1.

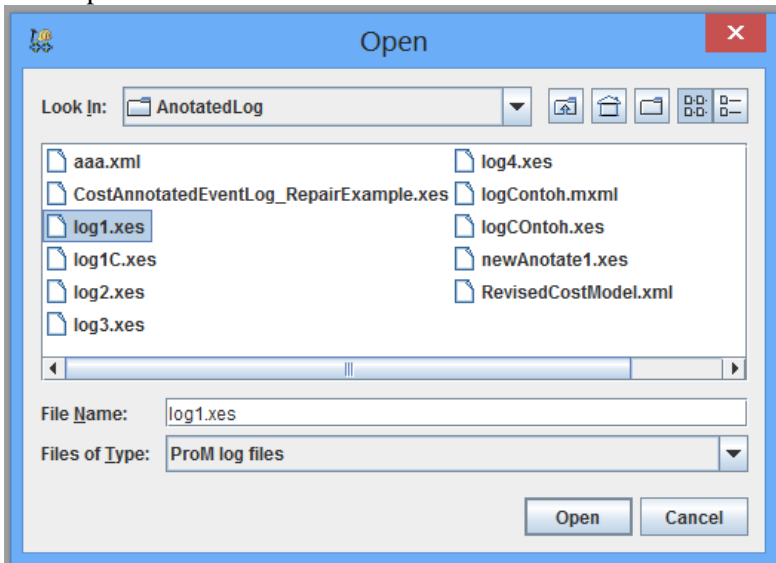
Tabel 6.1 Pengujian Fitur Mengimpor Event Log

ID	TA-UJ.UC0001
Referensi Kasus Penggunaan	TA-UC0001
Nama	Pengujian fitur mengimpor <i>event log</i>
Tujuan Pengujian	Menguji fitur untuk mengimpor <i>event log</i> .
Skenario 1	Mengimpor langsung File event log menggunakan event log yang telah disediakan
Kondisi Awal	<i>ProM</i> sudah dijalankan dan <i>file event log</i> sudah disiapkan dalam satu folder.
Data Uji	Data uji merupakan file event log, menggunakan event log yang telah disediakan
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih menu <i>import</i>. 2. Pengguna memilih tipe dokumen <i>event log</i>. 3. Pengguna memilih <i>folder</i> yang telah disiapkan. 4. Pengguna memilih dokumen <i>event log</i>
Hasil Yang Diharapkan	<i>Event log</i> dapat diimpor dan nama <i>event log</i> yang diimpor akan muncul pada <i>workspace ProM</i> .
Hasil Yang Didapat	<i>Event log</i> berhasil diimpor dan nama <i>event log</i> yang diimpor muncul pada <i>workspace ProM</i> .
Hasil Pengujian	Berhasil.
Kondisi Akhir	Tampilan dokumen-dokumen yang berhasil diimpor pada <i>workspace ProM</i> .

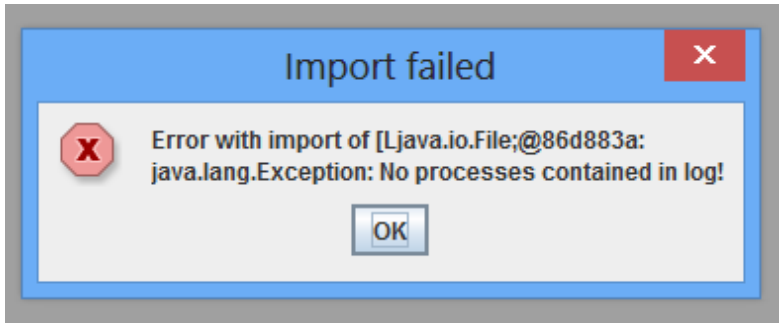
Skenario 2	Mengimpor <i>file</i> dengan ekstensi yang tidak berhubungan dengan <i>event log</i> ke dalam ProM.
Kondisi Awal	<i>ProM</i> sudah dijalankan dan <i>file</i> dengan ekstensi <i>docx</i> sudah disiapkan dalam satu folder.
Data Uji	Data uji merupakan <i>file</i> dengan ekstensi <i>*.docx</i>
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih menu <i>import</i>. 2. Pengguna memilih tipe dokumen <i>event log</i>. 3. Pengguna memilih <i>folder</i> yang telah disiapkan.
Hasil Yang Diharapkan	<i>File docx</i> tidak muncul pada saat akan dipilih
Hasil Yang Didapat	<i>File doc</i> tidak terlihat sehingga tidak dapat dipilih untuk diimpor.
Hasil Pengujian	Berhasil
Kondisi Akhir	Tidak dapat mengimpor <i>file docx</i> .
Skenario 3	Mengimpor <i>file</i> dengan ekstensi yang dikenali sebagai <i>event log</i> namun didalamnya tidak memiliki atribut tentang <i>event log</i> ke dalam ProM.
Kondisi Awal	<i>ProM</i> sudah dijalankan dan <i>file</i> yang memiliki ekstensi <i>xes</i> namun tidak memiliki atribut tentang <i>event log</i> sudah disiapkan dalam satu <i>folder</i> .
Data Uji	Data uji merupakan <i>file xes</i> yang berisi tentang <i>cost model</i> bukan <i>event log</i>, menggunakan <i>event log</i> yang terdapat pada <i>paper</i> Wiebe Nauta
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih menu <i>import</i>. 2. Pengguna memilih tipe dokumen <i>event log</i>. 3. Pengguna memilih <i>folder</i> yang telah disiapkan. 4. Pengguna memilih <i>file yang telah disiapkan</i>.
Hasil Yang Diharapkan	Muncul <i>message box</i> yang menampilkan pesan <i>error</i> menunjukkan bahwa <i>file</i> tersebut tidak terdapat atribut tentang <i>event log</i>
Hasil Yang Didapat	Muncul <i>message box</i> yang menampilkan pesan <i>error</i> menunjukkan bahwa <i>file</i> tersebut tidak terdapat atribut tentang <i>event log</i> seperti yang terlihat pada <small>Gambar 6.3.</small>

Hasil Pengujian	Berhasil
Kondisi Akhir	<i>Message Box</i> yang menampilkan pesan <i>error</i> dilihat pada Gambar 6.3.

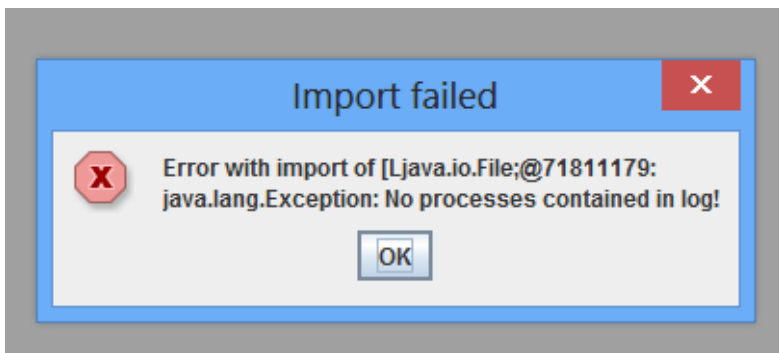
Untuk hasil uji data pertama oleh *plugin* dapat dilihat pada Gambar 6.1. Untuk hasil uji data kedua oleh *plugin* dapat dilihat pada Gambar 6.2. Untuk hasil uji data ketiga oleh *plugin* dapat dilihat pada Gambar 6.3.



Gambar 6.1 Hasil Program pada Uji Pertama



Gambar 6.2 Hasil Program pada Uji Kedua



Gambar 6.3 Hasil Program pada Uji Ketiga

6.2.1.2. Pengujian Fitur Mengimpor model proses SOP ke dalam ProM

Pengujian fitur mengimpor model proses dilakukan dengan beberapa model proses yang telah disediakan. Model proses tersebut memiliki ekstensi *.pnml yang merupakan ekstensi standar untuk model proses. Selain itu juga terdapat *file* dengan ekstensi lain yang tidak berhubungan dengan model proses. Tujuannya untuk menguji apakah *file* tersebut dapat diimpor

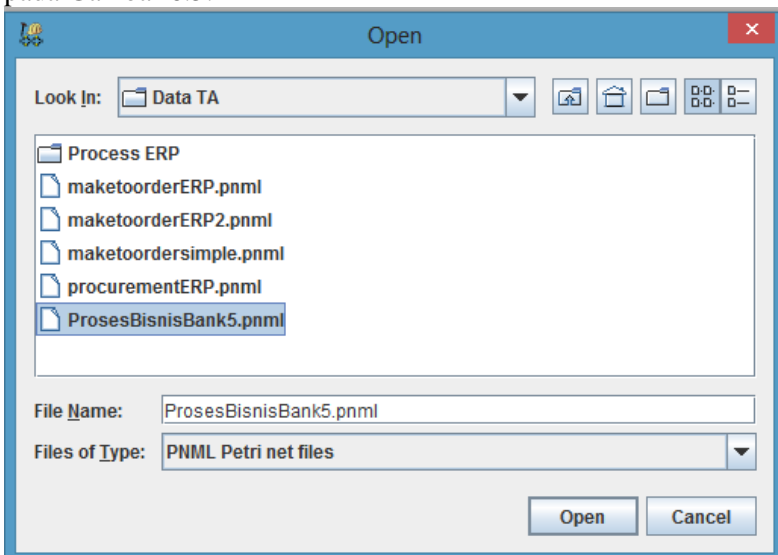
ataukah tidak. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 6.2.

Tabel 6.2 Pengujian Fitur Mengimpor model proses SOP

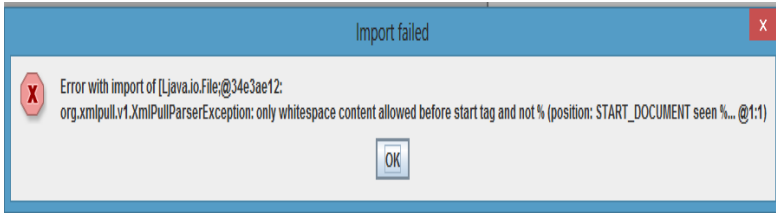
ID	TA-UJ.UC0002
Referensi Kasus Penggunaan	TA-UC0002
Nama	Pengujian fitur untuk mengimpor model proses SOP.
Tujuan Pengujian	Menguji fitur untuk mengimpor model proses SOP.
Skenario 1	Mengimpor langsung <i>file</i> model proses menggunakan model proses yang telah disediakan.
Kondisi Awal	<i>ProM</i> sudah dijalankan dan <i>file</i> model proses sudah disiapkan.
Data Uji	Data uji merupakan <i>file</i> model proses, menggunakan model proses yang telah disediakan
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih menu <i>import</i>. 2. Pengguna memilih tipe dokumen model proses. 3. Pengguna memilih folder yang telah disiapkan. 4. Pengguna memilih dokumen model proses.
Hasil Yang Diharapkan	Model proses dapat diimpor dan nama model proses yang diimpor akan muncul pada <i>workspace</i> ProM.
Hasil Yang Didapat	Model proses berhasil diimpor dan nama model proses yang diimpor muncul pada <i>workspace</i> ProM. Hasil dapat dilihat pada Gambar 6.6
Hasil Pengujian	Berhasil
Kondisi Akhir	Tampilan dokumen-dokumen yang berhasil diimpor pada <i>workspace</i> ProM.
Skenario 2	Mengimpor <i>file</i> dengan ekstensi yang tidak berhubungan dengan model proses ke dalam ProM.
Kondisi Awal	<i>ProM</i> sudah dijalankan dan <i>file</i> dengan ekstensi docx sudah disiapkan dalam satu <i>folder</i> .
Data Uji	Data uji merupakan file dengan ekstensi *.docx

Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih menu <i>import</i>. 2. Pengguna memilih tipe dokumen model proses 3. Pengguna memilih folder yang telah disiapkan.
Hasil Yang Diharapkan	<i>File docx</i> tidak muncul pada saat akan dipilih.
Hasil Yang Didapat	<i>File docx</i> tidak terlihat sehingga tidak dapat dipilih untuk diimpor.
Hasil Pengujian	Berhasil
Kondisi Akhir	Tidak dapat mengimpor <i>file docx</i>.

Untuk hasil uji data pertama oleh *plugin* dapat dilihat pada Gambar 6.4. Untuk hasil uji data kedua oleh *plugin* dapat dilihat pada Gambar 6.5.



Gambar 6.4 Tampilan Hasil Uji Fitur mengimpor model proses.



Gambar 6.5 Tampilan Hasil Uji Fitur mengimpor model proses

6.2.1.3. Pengujian Fitur Mengkonversi *event log* ke *event log ontology*

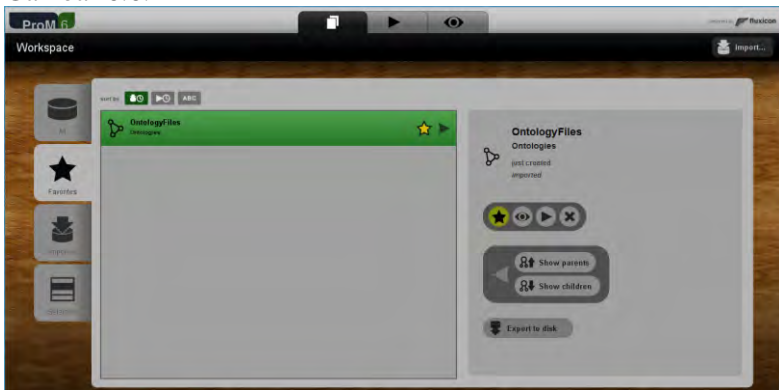
Pengujian fitur mengkonversi *event log* ke *event log ontology* merupakan pengujian terhadap kemampuan *plugin* untuk melakukan konversi *event log* yang telah diimpor ke bentuk *event log ontology*. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 6.3. Tabel tersebut bertujuan untuk menguji fungsionalitas sistem dalam mengkonversi *event log*.

Tabel 6.3 Pengujian Fitur Mengkonversi *event log* ke *event log ontology*

ID	TA-UJ.UC0003
Referensi Kasus Penggunaan	TA-UC0003
Nama	Pengujian fitur Mengkonversi <i>event log</i> ke <i>event log ontology</i>
Tujuan Pengujian	Menguji fungsionalitas sistem dalam mengkonversi <i>event log</i> yang diimpor ke bentuk <i>event log ontology</i>
Skenario 1	Menampilkan <i>event log ontology</i> yang telah dikonversi.
Kondisi Awal	<i>Event log</i> sudah diimpor dan terdapat pada <i>workspace ProM</i> .
Data Uji	Data uji berupa <i>event log</i> yang sudah diimpor ke ProM

Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih <i>event log</i> yang ingin dikonversi dari <i>workspace ProM</i> 2. Pengguna menekan tombol <i>use resource</i> dan memilih <i>Convert Event Log to Ontology</i> pada pilihan plugin 3. Memilih <i>path folder</i> tempat menyimpan <i>file event log ontology</i> hasil konversi
Hasil Yang Diharapkan	Dapat mengkonversi dan menampilkan <i>event log ontology</i> hasil konversi ke <i>workspace ProM</i>
Hasil Yang Didapat	Dapat menampilkan <i>event log ontology</i> hasil konversi ke <i>workspace ProM</i>
Hasil Pengujian	Berhasil
Kondisi Akhir	<i>Event log ontology</i> tampil pada <i>workspace ProM</i>

Untuk hasil uji data pertama oleh *plugin* dapat dilihat pada Gambar 6.6.



Gambar 6.6. Hasil Pengujian Mengkonversi Event Log ke Event Log Ontology

6.2.1.4. Pengujian Fitur Melakukan *Compliance Check Using Conformance Checking*

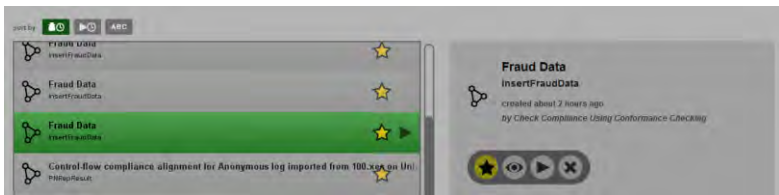
Pengujian fitur melakukan *Compliance Check Using Conformance Checking* merupakan pengujian terhadap

kemampuan *plugin* untuk melakukan *Compliance check using conformance checking* antara *event log* dan model proses yang sebelumnya telah diimpor ke dalam *workspace*. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 6.4. Untuk hasil pengujian pada skenario 1 dapat dilihat pada Gambar 6.7.

Tabel 6.4 Pengujian Fitur Melakukan *Compliance Check Using Conformance Checking*

ID	TA-UJ.UC0004
Referensi Kasus Penggunaan	TA-UC0004
Nama	Pengujian fitur melakukan <i>Compliance Check Using Conformance Checking</i>
Tujuan Pengujian	Menguji fungsionalitas sistem dalam Melakukan <i>Compliance Check Using Conformance Checking</i>
Skenario 1	Melakukan <i>Compliance Check Using Conformance Checking</i> terhadap <i>event log</i> dan model proses yang telah diimpor ke ProM
Kondisi Awal	<i>Event log</i> dan model proses sudah terdapat pada <i>workspace</i> ProM.
Data Uji	Data uji berupa <i>event log</i> dan model proses yang telah disediakan.
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih menu <i>use resource</i>. 2. Pengguna memilih <i>event log</i> dan model proses yang akan dianotasikan dari <i>workspace</i> 3. Pengguna memilih <i>plugin Compliance Check Using Conformance Checking</i>.
Hasil Yang Diharapkan	Tidak muncul pesan <i>error</i> dan akan muncul <i>file InsertFraudData</i> pada <i>workspace</i> .

Hasil Yang Didapat	Tidak muncul pesan <i>error</i> dan akan muncul file <i>InsertFraudData</i> pada <i>workspace</i> . Hasil dapat dilihat pada Gambar 6.7.
Hasil Pengujian	Berhasil
Kondisi Akhir	Terdapat file <i>InsertFraudData</i> pada <i>workspace</i> ProM.



Gambar 6.7. Hasil Pengujian Melakukan Compliance Check Using Conformance Checking

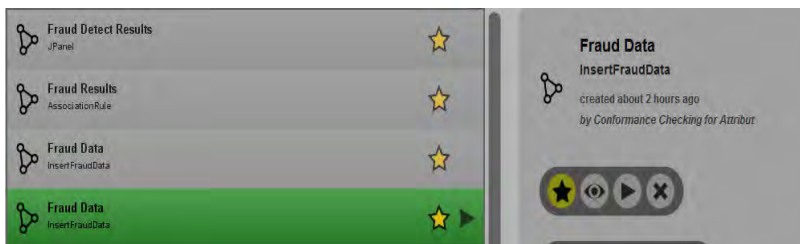
6.2.1.5. Pengujian Fitur Melakukan *Conformance Checking for Attribute*

Pengujian fitur melakukan *Conformance Checking for Attribute* merupakan pengujian terhadap kemampuan *plugin* dalam melakukan *Conformance Checking for Attribute* untuk mengecek kesesuaian atribut antara *event log* dan model proses yang telah diimpor sebelumnya ke dalam ProM. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 6.5. Hasil pengujian pada skenario 1 dapat dilihat pada Gambar 6.8.

Tabel 6.5 Pengujian Fitur Melakukan *Conformance Checking for Attribute*

ID	TA-UJ.UC0005
Referensi Kasus Penggunaan	TA-UC0005
Nama	Pengujian fitur Melakukan <i>Conformance Checking for Attribute</i>

Tujuan Pengujian	Menguji fungsionalitas sistem dalam melakukan <i>Conformance Checking for Attribute</i>
Skenario 1	Melakukan <i>Conformance Checking for Attribute</i>
Kondisi Awal	Terdapat <i>event log</i> , model proses, dan <i>InsertFraudData</i> yang dihasilkan sebelumnya pada <i>workspace</i> .
Data Uji	Data uji berupa <i>event log</i> dan model proses yang telah disediakan
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih <i>event log</i>, model proses, dan <i>InsertFraudData</i> yang akan digunakan dalam pengecekan kesesuaian. 2. Pengguna memilih <i>use resource</i> 3. Pengguna memilih <i>plugin Conformance Checking for Attribute</i>
Hasil Yang Diharapkan	Tidak muncul pesan <i>error</i> dan akan muncul <i>file InsertFraudData</i> pada <i>workspace</i> .
Hasil Yang Didapat	Tidak muncul pesan <i>error</i> dan akan muncul file <i>InsertFraudData</i> pada <i>workspace</i>. Hasil pencarian dapat dilihat pada Gambar 6.8.
Hasil Pengujian	Berhasil
Kondisi Akhir	Jendela yang menunjukkan <i>InsertFraudData</i> hasil <i>Conformance Checking for Attribute</i>



Gambar 6.8 Hasil Pengujian Melakukan *Conformance Checking for Attribute*

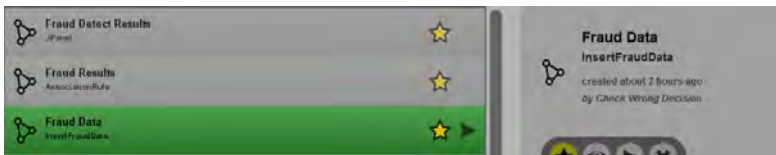
6.2.1.6. Pengujian Fitur Melakukan *Check Wrong Decision*

Pengujian fitur melakukan *Check Wrong Decision* merupakan pengujian terhadap kemampuan *plugin* dalam melakukan *Check Wrong Decision* untuk mengecek kesesuaian pengambilan keputusan antara *event log* dan model proses yang telah diimpor sebelumnya ke dalam ProM. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 6.6. Hasil pengujian pada skenario 1 dapat dilihat pada Gambar 6.9.

Tabel 6.6. Pengujian Fitur Melakukan *Check Wrong Decision*

ID	TA-UJ.UC0006
Referensi Kasus Penggunaan	TA-UC0006
Nama	Pengujian fitur Melakukan <i>Check Wrong Decision</i>
Tujuan Pengujian	Menguji fungsionalitas sistem dalam melakukan <i>Check Wrong Decision</i>
Skenario 1	Melakukan <i>Check Wrong Decision</i>
Kondisi Awal	Terdapat <i>event log</i> , model proses, dan <i>InsertFraudData</i> yang dihasilkan sebelumnya pada <i>workspace</i> .
Data Uji	Data uji berupa <i>event log</i> dan model proses yang telah disediakan
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih <i>event log</i>, model proses, dan <i>InsertFraudData</i> yang akan digunakan dalam pengecekan kesesuaian. 2. Pengguna memilih <i>use resource</i> 3. Pengguna memilih <i>plugin Check Wrong Decision</i>
Hasil Yang Diharapkan	Tidak muncul pesan <i>error</i> dan akan muncul <i>file InsertFraudData</i> pada <i>workspace</i> .

Hasil Yang Didapat	Tidak muncul pesan <i>error</i> dan akan muncul file <i>InsertFraudData</i> pada <i>workspace</i> . Hasil pencarian dapat dilihat pada Gambar 6.9.
Hasil Pengujian	Berhasil
Kondisi Akhir	Jendela yang menunjukkan <i>InsertFraudData</i> hasil <i>Check Wrong Decision</i>



Gambar 6.9. Hasil Pengujian Melakukan Check Wrong Decision

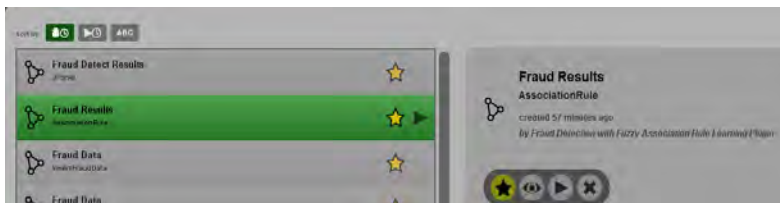
6.2.1.7. Pengujian Fitur Melakukan pendeteksian *Fraud* dengan *Fuzzy Association Rule Learning*

Pengujian fitur melakukan pendeteksian *Fraud* dengan *Fuzzy Association Rule Learning* merupakan pengujian terhadap kemampuan *plugin* dalam melakukan pendeteksian *Fraud* dengan *Fuzzy Association Rule Learning* untuk mendeteksi *fraud* dan menghasilkan aturan asosiasi dari pelanggaran yang dilakukan. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 6.7. Hasil pengujian pada skenario 1 dapat dilihat pada Gambar 6.10.

Tabel 6.7 Pengujian Fitur Melakukan pendeteksian *Fraud* dengan *Fuzzy Association Rule Learning*

ID	TA-UJ.UC0007
Referensi Kasus Penggunaan	TA-UC0007
Nama	Pengujian fitur melakukan pendeteksian <i>Fraud</i> dengan <i>Fuzzy Association Rule Learning</i>

Tujuan Pengujian	Menguji fungsionalitas sistem dalam melakukan pendeteksian <i>Fraud</i> dengan <i>Fuzzy Association Rule Learning</i>
Skenario 1	Melakukan pendeteksian <i>Fraud</i> dengan <i>Fuzzy Association Rule Learning</i>
Kondisi Awal	Terdapat <i>InsertFraudData</i> yang dihasilkan dari 3 pengecekan kesesuaian yang dilakukan dan model proses yang diimpor sebelumnya pada <i>workspace</i> .
Data Uji	Data uji berupa <i>InsertFraudData</i> dan model proses SOP yang telah disediakan
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih model proses, dan <i>InsertFraudData</i> yang akan digunakan dalam pendeteksian <i>Fraud</i>. 2. Pengguna memilih <i>use resource</i> 3. Pengguna memilih <i>plugin Fraud Detection with Fuzzy Association Rule Learning</i>
Hasil Yang Diharapkan	Tidak muncul pesan <i>error</i> dan akan muncul <i>file AssociationRule</i> pada <i>workspace</i> .
Hasil Yang Didapat	Tidak muncul pesan <i>error</i> dan akan muncul <i>file AssociationRule</i> pada <i>workspace</i>. Hasil pencarian dapat dilihat pada Gambar 6.10.
Hasil Pengujian	Berhasil
Kondisi Akhir	Jendela yang menunjukkan <i>InsertFraudData</i> hasil <i>Conformance Checking for Attribute</i>



Gambar 6.10 Hasil Pengujian Melakukan pendeteksian *Fraud* dengan *Fuzzy Association Rule Learning*

6.2.1.8. Pengujian Fitur Melakukan pendeteksian *Fraud* dengan data aturan asosiasi

Pengujian fitur melakukan pendeteksian *Fraud* dengan data aturan asosiasi merupakan pengujian terhadap kemampuan *plugin* dalam melakukan pendeteksian *Fraud* dengan data aturan asosiasi untuk melakukan pendeteksian *Fraud* dengan menggunakan aturan asosiasi yang telah dihasilkan. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 6.8. Hasil pengujian pada skenario 1 dapat dilihat pada Gambar 6.11.

Tabel 6.8 Pengujian Fitur Melakukan pendeteksian *Fraud* dengan data aturan asosiasi

ID	TA-UJ.UC0008
Referensi Kasus Penggunaan	TA-UC0008
Nama	Pengujian fitur Melakukan pendeteksian <i>Fraud</i> dengan data aturan asosiasi
Tujuan Pengujian	Menguji fungsionalitas sistem dalam melakukan pendeteksian <i>Fraud</i> dengan data aturan asosiasi
Skenario 1	Melakukan pendeteksian <i>Fraud</i> dengan data aturan asosiasi
Kondisi Awal	Terdapat model proses, <i>InsertFraudData</i> dan <i>AssociationRule</i> yang dihasilkan sebelumnya pada <i>workspace</i> .
Data Uji	Data uji berupa <i>AssociationRule</i> yang dihasilkan, <i>InsertFraudData</i> hasil pengecekan kesesuaian dan model proses yang telah disediakan
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih model proses, <i>InsertFraudData</i> dan <i>Association Rule</i> yang akan digunakan dalam pengecekan kesesuaian. 2. Pengguna memilih <i>use resource</i> 3. Pengguna memilih <i>plugin Detect Fraud with Association Rule Data</i>

Hasil Yang Diharapkan	Tidak muncul pesan <i>error</i> dan akan muncul tabel hasil pendeteksian <i>fraud</i> pada jendela ProM.
Hasil Yang Didapat	Tidak muncul pesan <i>error</i> dan akan muncul tabel hasil pendeteksian <i>fraud</i> pada jendela ProM. Hasil pencarian dapat dilihat pada Gambar 6.11.
Hasil Pengujian	Berhasil
Kondisi Akhir	Jendela yang menunjukkan tabel hasil pendeteksian <i>Fraud</i> pada proses bisnis.

Hasil Deteksi Fraud

Case	Aturan	Confide...	Bobot Fra...	Kelas Fraud
99	SkipSL-wDutyDecl-wPatternL-wDecisionL-Fraud	2.302619...	0.6589	Yakin Fraud
98	SkipSL-wDutyDecl-wPatternL-wDecisionL-Fraud	2.302619...	0.69049900...	Yakin Fraud
97	SkipSL-wDutyDecl-wPatternL-wDecisionL-Fraud	2.302619...	0.68090000...	Yakin Fraud
96	SkipSL-wResourceL-wPatternL-Fraud	4.16017	0.2127	Yakin Fraud
95	SkipSL-wResourceL-wPatternL-Fraud	4.16017	0.2087	Yakin Fraud
94	SkipSL-wDutyDecl-wPatternL-wDecisionL-Fraud	2.302619...	0.3082	Yakin Fraud
93	SkipSL-wDutyDecl-wPatternL-wDecisionL-Fraud	2.302619...	0.3082	Yakin Fraud
92	SkipSL-wDutyDecl-wPatternL-Fraud	5.319840...	0.0955990...	Yakin Fraud
91	SkipSL-wResourceL-wPatternL-Fraud	4.16017	0.1584490...	Yakin Fraud
90	SkipSL-TmaxL-wPatternL-Fraud	3.845090...	0.2066	Yakin Fraud
89	wResourceL-wDutyDecl-Fraud	1.462814...	0.087	Yakin Fraud
88	wResourceL-Fraud	0.523431...	0.0809990...	Fraud
87	TminL-TmaxL-Fraud	0.986244...	0.2115	Yakin Fraud
86	TminL-TmaxL-Fraud	0.986244...	0.2211	Yakin Fraud
85	wResourceL-wDutyDecl-wDecisionL-Fraud	4.3395	0.36955	Yakin Fraud
84	wResourceL-wDutyDecl-wDecisionL-Fraud	4.3395	0.34895	Yakin Fraud
83	wDutyDecl-wDutyDecl-Fraud	0.906860...	0.037	Yakin Fraud
82	TmaxL-wResourceL-wDecisionL-Fraud	1.637290...	0.35405	Yakin Fraud
81	TminL-TmaxL-Fraud	0.986244...	0.2402500...	Yakin Fraud
80	wResourceL-wPatternL-wDecisionL-Fraud	4.620759...	0.2996	Yakin Fraud
100	SkipSL-wDutyDecl-wPatternL-wDecisionL-Fraud	2.302619...	0.7168990...	Yakin Fraud
79	SkipDL-TmaxL-wPatternL-wDecisionL-Fraud	1.999999...	0.6494	Yakin Fraud
78	wDecisionL-Fraud	0.436985...	0.2301990...	Tidak Fraud
77	wResourceL-Fraud	0.523431...	0.0482000...	Fraud
76	Tidak Ada	0.0	0.1087500...	Tidak Fraud
69	SkipSL-wDutyDecl-wPatternL-Fraud	5.319840...	0.0896	Yakin Fraud

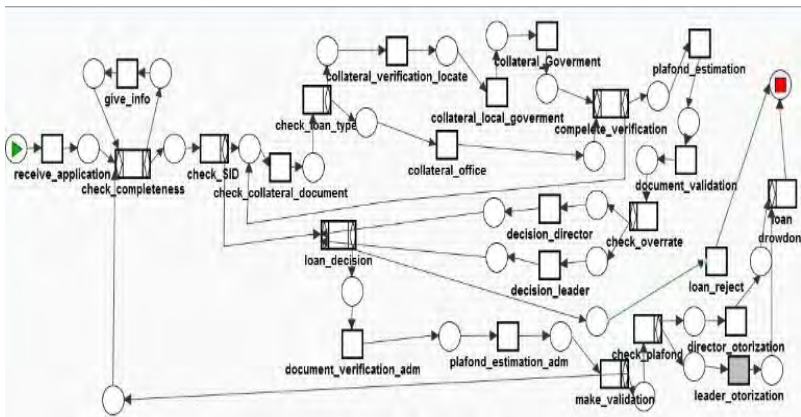
Buttons:

Gambar 6.11 Hasil Pengujian Melakukan pendeteksian *Fraud* dengan data aturan asosiasi

6.2.2. Pengujian Pada Studi Kasus

6.2.2.1. Studi Kasus Aplikasi Kredit Bank

Studi kasus aplikasi kredit bank adalah studi kasus yang digunakan untuk memeriksa kecurangan pada proses aplikasi kredit pada suatu bank dengan proses bisnis seperti pada Gambar 6.12.



Gambar 6.12. Proses bisnis aplikasi kredit bank

Aktivitas pertama yang dieksekusi adalah *receive application* yaitu menerima sebuah pengajuan kredit. Data yang diterima seperti KTP, KSK, dan dokumen persyaratan lainnya yang dibutuhkan dalam pengajuan kredit.

Aktivitas *check completeness* yaitu memeriksa kelengkapan dokumen persyaratan apabila belum lengkap maka akan dilakukan aktivitas *give info* untuk mendapatkan dokumen yang lebih lengkap lagi. Aktivitas ini bisa terjadi terus berulang apabila kelengkapan dokumen belum terpenuhi.

Aktivitas *check SID* dilakukan oleh sistem untuk mengecek histori pengajuan kredit yang dilakukan oleh pengaju kredit. Apabila seorang pengaju kredit ternyata sudah pernah melakukan

pengajuan kredit, aktifitas selanjutnya adalah menuju pada *loan decision* sebaliknya adalah menuju pada *check loan type*.

Aktifitas *check loan type*, *collateral verification locate*, *collateral local government*, *collateral office*, dan *complete verification* dilakukan untuk pengecekan agunan yang dimiliki pengaju kredit sesuai dengan tipe kredit yang diajukan. Aktivitas ini penting untuk memeriksa keaslian dokumen yang telah diajukan.

Aktivitas *plafond estimation* digunakan untuk memperkirakan jumlah kredit yang akan dicairkan. Hal ini tergantung dari pengecekan agunan yang telah dicek sebelumnya. Aktivitas *check overrate* digunakan untuk menentukan aktivitas selanjutnya yang akan dieksekusi. Yaitu apakah menuju *decision director* atau *decision leader*. Hal ini tergantung pada jumlah pengajuan kredit.

Aktivitas *document_verification_adm* dan *plafond_verification_adm* digunakan untuk mengecek kembali dokumen dan plafond. Aktivitas selanjutnya yang dilakukan setelah dua aktivitas ini adalah *make validation* yaitu untuk memvalidasi kelengkapan. Apabila validasi telah valid maka akan menuju pada aktifitas *plafond check* dan apabila tidak akan kembali pada aktifitas *check completeness*.

Aktifitas *director otorization* dan *leader otorization* merupakan pengesahan yang dilakukan oleh *director* dan *leader* untuk mencairkan kredit. Aktifitas terakhir yang dilakukan adalah *loan drowdown* yang merupakan aktifitas pencairan kredit.

Kemungkinan terjadinya *fraud* salah satunya adalah *skipped activity* atau aktivitas yang terlewat atau tidak dilakukan dengan benar. Hal ini menandakan adanya kecurangan pada aktivitas tersebut seperti pada *complete verification* yang seharusnya dilakukan oleh kepala bagian bank namun ternyata dilakukan oleh seorang staff. Contoh lainnya adalah pada aktivitas *document validation* yang seharusnya dilakukan paling lama satu minggu namun ternyata dilakukan lebih dari satu minggu. Contoh lainnya lagi adalah pada aktivitas yang mengeluarkan *output*

percabangan lebih dari satu aktivitas seperti pada *check overrate* di mana seharusnya pengajuan kredit yang lebih dari lima puluh juta rupiah seharusnya menuju pada aktivitas *decision director* namun pada kenyataannya menuju pada *decision leader*.

Dengan menggunakan metode *fuzzy* MADM dihasilkan perhitungan bobot *fraud* terhadap dataset pelanggaran yang dilakukan yang dapat dilihat pada Tabel 6.9. Dengan metode *fuzzy* ARL dihasilkan aturan asosiasi dari data pelanggaran yang dapat dilihat pada Tabel 6.10. Dan hasil pendeteksian *fraud* dengan menggunakan aturan asosiasi yang dihasilkan dapat dilihat pada Tabel 6.11. Percobaan ini dilakukan pada data *event logs* dari studi kasus aplikasi kredit bank ini terdiri dari:

1. Data *training* sebanyak 130 *case* yang terdiri dari 100 *case fraud* dan 30 *case* normal.
2. Data *testing* sebanyak 100 *case* yang terdiri dari 70 *case fraud* dan 30 *case* normal.

Tabel 6.9. Tabel Perhitungan bobot pelanggaran

Case	Skip		Through put Time		Wrong	Wrong duty			Wrong	Wrong	Fraud
	sequence	decision	Min	Max	resource	Sequence	Decision	Combine	Pattern	Decision	
1	2	0	0	0	0	0	0	0	2	1	0,233
4	0	0	0	0	0	2	0	0	0	0	0,009
3	0	0	0	0	3	0	0	0	0	0	0,0328
5	0	0	0	0	0	0	2	0	0	0	0,035
9	1	0	0	0	0	1	0	0	1	0	0,0496
6	0	0	0	0	1	0	0	1	0	0	0,0546

Tabel 6.10. Tabel Aturan asosiasi

ARL	Support	Confidence
SkipSL-Fraud	0,2152105	0,777869277
SkipDL-Fraud	0,093789	0,605090323
SkipDM-Fraud	0,1519425	0,690647727
SkipDH-Fraud	0,055	1
TminL-Fraud	0,254703778	0,90606087
TminM-Fraud	0,010785667	0,277345714
TmaxL-Fraud	0,192004167	0,817039007
TmaxM-Fraud	0,0449905	0,70419913
wResourceL-Fraud	0,205331222	0,865564871

Tabel 6.11. Hasil Deteksi Fraud

Case	Aturan	Confidence	bobot	Kelas
1	Tidak Ada	0	0	Tidak Fraud
2	Tidak Ada	0	0	Tidak Fraud
11	TminL-Fraud	0,85907837	0,08095	Tidak Fraud
12	TminL-Fraud	0,85907837	0,08095	Tidak Fraud
24	wDutyDecL-Fraud	0,61327692	0,035	Tidak Fraud
25	wDutyDecL-Fraud	0,61327692	0,0328	Tidak Fraud
26	wPatternL- wDecisionL-Fraud	1,41164615	0,2208	Yakin Fraud
27	wPatternL- wDecisionL-Fraud	1,41164615	0,2112	Yakin Fraud
28	wPatternL- wDecisionL-Fraud	1,41164615	0,2112	Yakin Fraud
29	wDecisionL-Fraud	0,41682898	0,1784	Fraud
30	wDecisionL-Fraud	0,41682898	0,1784	Fraud
34	SkipDL- wPatternL- wDecisionL-Fraud	0,36747989	0,4298	Semi Fraud

Setelah itu dilakukan percobaan terhadap 100 data *testing*, dilakukan perhitungan akurasi untuk mengukur efektifitas dari metode *fuzzy* ARL terhadap studi kasus aplikasi kredit bank. Perhitungan akurasi dilakukan dengan menggunakan metode *Receiver Operating Characteristic* (ROC) dengan rumus yang dapat dilihat pada Persamaan 6.1.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (6.1)$$

Untuk nilai TP merupakan jumlah *case fraud* yang terdeteksi sebagai *fraud* oleh sistem, TN merupakan jumlah *case* normal yang terdeteksi normal oleh sistem, FP merupakan jumlah *case* normal yang terdeteksi sebagai *fraud* oleh sistem, dan FN merupakan jumlah *case fraud* yang terdeteksi normal oleh sistem.

Perhitungan akurasi pada studi kasus pertama menghasilkan akurasi tertinggi sebesar 0,83 untuk nilai minimum *confidence* 0,1. Pada Tabel 6.12, dapat dilihat bahwa terdapat jumlah yang besar untuk kasus *false positive* dengan probabilitas 17/30 sedangkan jumlah *false negative* lebih sedikit dengan probabilitas 0. Hal ini terjadi karena kasus yang normal belum didefinisikan dengan benar. Sehingga, diperlukan pembentukan aturan bantuan yang mendefinisikan kasus normal untuk mengurangi jumlah *false positive* dan meningkatkan akurasi.

Aturan tambahan mendefinisikan kasus normal. Pada penelitian ini, kasus normal didefinisikan sebagai sebuah kasus di mana hanya terdapat 1 jenis pelanggaran dengan intensitas yang rendah. Contohnya, SkipSL(*Skip Sequence Low*), SkipDL(*Skip Decision Low*), dll.

Tabel 6.12. Akurasi metode terhadap kasus aplikasi kredit bank

Min Confidence	TP	FP	TN	FN	Accuracy
0,1	70	17	13	0	0,83
0,2	67	17	13	3	0,8
0,3	67	17	13	3	0,8
0,4	67	17	13	3	0,8
0,5	65	17	13	5	0,78
0,6	65	17	13	5	0,78
0,7	65	15	15	5	0,8
0,8	62	14	16	8	0,78
0,9	53	10	20	17	0,73

Selanjutnya, uji coba dilakukan terhadap data *testing* dengan aturan tambahan dan akurasi meningkat menjadi 0,96 dengan probabilitas jumlah *false positive* berkurang menjadi 4/30. Hasil dari perhitungan akurasi dengan bantuan aturan tambahan dapat dilihat pada Tabel 6.13.

Tabel 6.13. Hasil Akurasi dengan bantuan *rule*

Min Confidence	TP	FP	TN	FN	Accuracy
0,1	70	5	25	0	0,95
0,2	70	4	26	0	0,96
0,3	70	4	26	0	0,96
0,4	65	3	27	5	0,92
0,5	64	3	27	6	0,91
0,6	61	3	27	9	0,88
0,7	56	3	27	14	0,83
0,8	54	3	27	16	0,81
0,9	49	3	27	21	0,76

6.2.2.2. Studi Kasus proses *procurement* ERP

Studi kasus yang digunakan ialah studi kasus ERP pada bagian *procurement* atau pengadaan barang perusahaan. Model proses bisnisnya dapat dilihat pada Diagram A.4. Di dalam studi kasus tersebut, terdapat dua bagian penting yang menjalankan

proses bisnis, yaitu bagian *Purchase Requisition (PR)*, yang mengurus permohonan persetujuan untuk pembelian barang. Kemudian ada bagian *Purchase Order*, yang mengurus proses pembelian barang berdasarkan PR yang telah disetujui dan termasuk dalam proses pemilihan *supplier* sampai persetujuan kontrak dengan pihak *supplier*. Penjelasan aktivitasnya dapat dilihat pada subbab 6.2.

Kemungkinan terjadinya *fraud* salah satunya adalah *skipped activity* atau aktifitas yang terlewat atau tidak dilakukan dengan benar. Hal ini menandakan adanya kecurangan pada aktifitas tersebut. Selanjutnya *fraud* dapat terjadi jika terdapat aktivitas yang dilakukan oleh orang yang tidak berwenang seperti pada *negotiating with supplier* yang seharusnya dilakukan oleh *Contract Supervisor* namun ternyata dilakukan oleh seorang *Contract Officer*. Bentuk *fraud* lainnya adalah pada aktivitas *sending contract to supplier* yang seharusnya dilakukan paling lama satu minggu namun ternyata dilakukan lebih dari satu minggu. Contoh lainnya lagi adalah pada aktifitas yang mengeluarkan *output* percabangan lebih dari satu aktifitas seperti pada *negotiating with supplier* di mana hasil negosiasi tidak menghasilkan persetujuan, tetapi aktivitas dilanjutkan pada *signing contract* yang seharusnya dilanjutkan pada aktivitas *making failure report*. Contoh *fraud* lainnya adalah terjadinya kesalahan terhadap aturan *segregation of duty* di mana dalam dua aktivitas yang berbeda dalam satu proses dijalankan oleh orang yang sama. Dan contoh *fraud* lainnya adalah terdapat kesalahan pola atau aturan aliran proses yang terjadi karena adanya aktivitas yang dijalankan dalam urutan yang tidak sesuai dengan aliran pada proses bisnis standar. Contohnya seperti *signing contract* harus dilakukan setelah melakukan *negotiating with supplier*. Jika *signing contract* dilakukan terlebih dahulu maka hal ini dapat dikatakan sebagai pelanggaran *wrong pattern*.

Dengan menggunakan metode *fuzzy* MADM dihasilkan perhitungan bobot *fraud* terhadap dataset pelanggaran yang dilakukan yang dapat dilihat pada Tabel 6.14. Dengan metode *fuzzy*

ARL dihasilkan aturan asosiasi dari data pelanggaran yang dapat dilihat pada Tabel 6.15. Dan hasil pendeteksian *fraud* dengan menggunakan aturan asosiasi yang dihasilkan dapat dilihat pada Tabel 6.16. Percobaan ini dilakukan pada data *event logs* dari studi kasus *procurement ERP* ini terdiri dari:

1. Data *training* sebanyak 100 *case* yang terdiri dari 70 *case fraud* dan 30 *case normal*.
2. Data *testing* sebanyak 100 *case* yang terdiri dari 40 *case fraud* dan 60 *case normal*.

Tabel 6.14. Tabel Bobot Pelanggaran kasus *procurement ERP*

Case	Skip		Throghput time		Wrong resource	Wrong Duty			Wrong Pattern	Wrong Decision	Fraud
	sequence	decision	min	max		sequence	decision	combine			
1	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0
11	0	0	1	0	0	0	0	0	0	0	0,081
12	0	0	0	1	0	0	0	0	0	0	0,06
21	0	0	0	0	0	2	0	0	0	0	0,006
22	0	0	0	0	0	0	1	0	0	0	0,035

Tabel 6.15. Aturan asosiasi pelanggaran *procurement ERP*

ARL	Support	Confidence
SkipSL-Fraud	0,055998	0,36921868
SkipDL-Fraud	0,074731	0,62275833
TminL-Fraud	0,047079	0,40938502

TmaxL-Fraud	0,055466	0,40420202
SkipSL-wResourceL-Fraud	0,023112	1,02720247
SkipSL-wDutySecL-Fraud	0,020882	0,92807407
SkipSL-wDutyDecL-Fraud	0,029555	1,31354074
SkipSL-SkipDL-TmaxL-Fraud	0,011111	2
SkipSL-SkipDL-wResourceL-Fraud	0,005556	1
SkipSL-SkipDL-wDutySecL-Fraud	0,016667	3
SkipSL-SkipDL-wDutyDecL-Fraud	0,016667	3
SkipDL-TmaxL-wPatternL-wDecisionL-Fraud	0,022222	2
SkipDL-wDutyDecL-wPatternL-wDecisionL-Fraud	0,025	2,25

Tabel 6.16. Hasil Pendeteksian *Fraud* kasus *procurement* ERP

Case	Aturan	Confidence	bobot	Kelas
1	Tidak Ada	0	0	Tidak Fraud
2	Tidak Ada	0	0	Tidak Fraud
3	Tidak Ada	0	0	Tidak Fraud
4	Tidak Ada	0	0	Tidak Fraud
28	wDutySecL-Fraud	0,281262252	0,009	Tidak Fraud
29	wDutyDecL-Fraud	0,259924528	0,035	Tidak Fraud
30	wDecisionL-Fraud	0,436985882	0,1858	Tidak Fraud
31	SkipSL-wPatternL-Fraud	2,474330864	0,0436	Yakin Fraud
32	SkipDL-wPatternL-wDecisionL-Fraud	9,66796	0,3934	Yakin Fraud

33	SkipSL-wPatternL-wDecisionL-Fraud	5,8772	0,31035	Yakin Fraud
98	SkipSL-wDutyDecl-wPatternL-wDecisionL-Fraud	2,30262	0,6905	Yakin Fraud
99	SkipSL-wDutyDecl-wPatternL-wDecisionL-Fraud	2,30262	0,6589	Yakin Fraud
100	SkipSL-wDutyDecl-wPatternL-wDecisionL-Fraud	2,30262	0,7169	Yakin Fraud

Setelah itu dilakukan percobaan terhadap 100 data *training* dan 100 data *testing*, dilakukan perhitungan akurasi untuk mengukur efektifitas dari metode *fuzzy* ARL terhadap studi kasus *procurement* ERP dengan menggunakan *rule* yang sama seperti pada percobaan kasus aplikasi kredit bank yaitu *rule* untuk kejadian normal / kejadian yang bukan *fraud* yaitu *rule* yang terdiri dari 1 jenis pelanggaran yang dilakukan dalam intensitas yang rendah (low) contohnya seperti *rule* SkipSL-> Fraud. Hal ini dilakukan untuk memastikan bahwa *rule* yang dibentuk dapat digunakan secara umum untuk pendeteksian *fraud* pada proses bisnis manapun. Hasil akurasi dapat dilihat pada Tabel 6.17.

Tabel 6.17. Hasil perhitungan akurasi pendeteksian *fraud* pada proses *procurement* ERP dengan bantuan *rule* yang sama

Min Confidence	TP	FP	TN	FN	Accuracy
0,1	39	2	58	1	0,97
0,2	37	2	58	3	0,95
0,3	36	2	58	4	0,94
0,4	36	2	58	4	0,94
0,5	36	2	58	4	0,94
0,6	31	2	58	9	0,89
0,7	31	2	58	9	0,89
0,8	31	2	58	9	0,89
0,9	31	2	58	9	0,89

6.2.3. Pengujian Kegunaan

Tugas Akhir ini juga membutuhkan pengujian aplikasi berdasarkan pengamatan pengguna lain yang belum pernah menggunakan *plugin* ini. Tujuan dari pengujian kegunaan adalah memberikan penilaian tentang kelayakan *plugin* ini dan nilai kebenaran dari *plugin* ini serta komentar-komentar terkait dari fitur-fitur yang telah diimplementasikan. Tujuan lainnya yaitu untuk mengetahui tingkat kemudahan dalam pengoperasian *plugin*. Pengujian ini dilakukan pengguna dengan metode *blackbox testing*.

Pengujian dilakukan oleh beberapa orang yang dipilih oleh penulis. Penulis memilih orang-orang yang berlatarbelakang pendidikan teknik komputer. Penulis juga memilih orang-orang yang pernah menggunakan aplikasi ProM untuk melakukan *process mining*.

Pengujian dilakukan dengan memberikan kesempatan pada pengguna untuk mencoba sendiri *plugin* yang telah dikembangkan. Uji coba yang dilakukan pengguna meliputi melakukan *conformance checking*, melakukan pendeteksian *fraud* dengan *fuzzy association rule*, dan mendeteksi *fraud* dengan data aturan asosiasi. Rincian isi kuesioner yang diberikan pada pengguna dapat dilihat pada Tabel 6.18.

Tabel 6.18 Daftar Pertanyaan Kuesioner

No.	Pernyataan	Pilihan jawaban			
		Sangat setuju	Setuju	Tidak setuju	Sangat tidak setuju
1	Menurut saya, <i>plugin</i> ini mudah digunakan.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	Menurut saya, cukup mudah untuk mengimpor <i>event log</i> ke dalam aplikasi ProM dengan <i>plugin</i> ini.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	Menurut saya, cukup mudah untuk mengimpor model	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

No.	Pernyataan	Pilihan jawaban			
		Sangat setuju	Setuju	Tidak setuju	Sangat tidak setuju
	proses ke dalam aplikasi ProM dengan <i>plugin</i> ini.				
4	Menurut saya, cukup mudah dalam mengkonversi <i>event log</i> ke <i>event log ontology</i>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	Menurut saya, cukup mudah dalam melakukan setiap tahap <i>conformance checking</i> dalam <i>plugin</i> ini.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	Menurut saya, cukup mudah untuk memasukkan data penilaian pakar dan nilai <i>threshold</i> ke dalam <i>plugin</i> .	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	Menurut saya, cukup mudah untuk melakukan pencarian aturan asosiasi dan pendeteksian <i>fraud</i> .	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8	Menurut saya, tabel hasil aturan asosiasi mudah dibaca dan dipahami.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9	Menurut saya, tabel hasil deteksi <i>fraud</i> mudah dibaca dan dipahami.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10	Menurut saya, tampilan <i>plugin</i> ini menarik dan mudah dipahami.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Tabel 6.19 Daftar Penguji Plugin

No.	Nama	Profesi	Jurusan
1	Bagus Ardiansyah	Mahasiswa	Teknik Informatika
2	Endang Wahyu P.	Mahasiswa	Teknik Informatika
3	Galang Amanda D.P.	Mahasiswa	Teknik Informatika
4	Chairaja Almas Djani	Mahasiswa	Teknik Informatika
5	Ramadhani Tegar	Mahasiswa	Teknik Informatika
6	Azi Prastyo	Mahasiswa	Teknik Informatika
7	Suliadi Marsetya	Mahasiswa	Teknik Informatika
8	M Hanif Budiarto	Mahasiswa	Teknik Informatika

9	Febry Amin Nurhidayah	Mahasiswa	Teknik Informatika
10	Devanda Tamba	Mahasiswa	Teknik Informatika

Tabel 6.20 Hasil Kuesioner

No.	Pernyataan	Pilihan jawaban			
		Sangat setuju	Setuju	Tidak setuju	Sangat tidak setuju
1	Menurut saya, <i>plugin</i> ini mudah digunakan.	50 %	50%		
2	Menurut saya, cukup mudah untuk mengimpor <i>event log</i> ke dalam aplikasi ProM dengan <i>plugin</i> ini.	80%	20%		
3	Menurut saya, cukup mudah untuk mengimpor model proses ke dalam aplikasi ProM dengan <i>plugin</i> ini.	90%	10%		
4	Menurut saya, cukup mudah dalam mengkonversi <i>event log</i> ke <i>event log ontology</i>	80%	20%		
5	Menurut saya, cukup mudah dalam melakukan setiap tahap <i>conformance checking</i> dalam <i>plugin</i> ini.	30%	70%		
6	Menurut saya, cukup mudah untuk memasukkan data penilaian pakar dan nilai <i>threshold</i> ke dalam <i>plugin</i> .	70%	30%		
7	Menurut saya, cukup mudah untuk melakukan pencarian aturan asosiasi dan pendeteksian <i>fraud</i> .	40%	60%		
8	Menurut saya, tabel hasil aturan asosiasi mudah dibaca dan dipahami.	50%	50%		
9	Menurut saya, tabel hasil deteksi <i>fraud</i> mudah dibaca dan dipahami.	70%	30%		

No.	Pernyataan	Pilihan jawaban			
		Sangat setuju	Setuju	Tidak setuju	Sangat tidak setuju
10	Menurut saya, tampilan <i>plugin</i> ini menarik dan mudah dipahami.	80%	20%		

Penguji *plugin* ini adalah orang-orang yang pernah menggunakan aplikasi ProM untuk melakukan *process mining*. Daftar penguji *plugin* ini dapat dilihat pada Tabel 6.19. Hasil rekap kuesioner dapat dilihat pada Tabel 6.20. Hasil rekap kuesioner menunjukkan rangkuman jawaban dari seluruh penguji pada masing-masing pertanyaan.

6.3. Evaluasi Pengujian

Pada subbab ini akan diberikan hasil evaluasi dari pengujian-pengujian yang telah dilakukan. Evaluasi yang diberikan meliputi evaluasi pengujian kebutuhan fungsional dan evaluasi pengujian kegunaan.

6.3.1. Evaluasi Pengujian Fungsionalitas

Rangkuman mengenai hasil pengujian fungsionalitas dapat dilihat pada Tabel 6.21. Berdasarkan data pada tabel tersebut, semua skenario pengujian berhasil dan program berjalan dengan baik. Sehingga bisa ditarik disimpulkan bahwa fungsionalitas dari program telah bisa bekerja sesuai dengan yang diharapkan.

Tabel 6.21 Rangkuman Hasil Pengujian

ID	Nama	Skenario	Hasil
TA-UJ.UC0001	Pengujian fitur mengimpor <i>event log</i>	Skenario 1	Berhasil
		Skenario 2	Berhasil
		Skenario 3	Berhasil
TA-UJ.UC0002	Pengujian fitur untuk mengimpor model proses SOP.	Skenario 1	Berhasil
		Skenario 2	Berhasil

ID	Nama	Skenario	Hasil
TA-UJ.UC0003	Pengujian fitur Mengkonversi <i>event log</i> ke <i>event log ontology</i>	Skenario 1	Berhasil
TA-UJ.UC0004	Pengujian fitur melakukan <i>Compliance Check Using Conformance Checking</i>	Skenario 1	Berhasil
TA-UJ.UC0005	Pengujian fitur Melakukan <i>Conformance Checking for Attribute</i>	Skenario 1	Berhasil
TA-UJ.UC0006	Menguji fungsionalitas sistem dalam melakukan <i>Check Wrong Decision</i>	Skenario 1	Berhasil
TA-UJ.UC0007	Pengujian fitur melakukan pendeteksian <i>Fraud</i> dengan <i>Fuzzy Association Rule Learning</i>	Skenario 1	Berhasil
TA-UJ.UC0008	Pengujian fitur Melakukan pendeteksian <i>Fraud</i> dengan data aturan asosiasi	Skenario 1	Berhasil

6.3.2. Evaluasi Pengujian Studi Kasus

6.3.2.1. Evaluasi Studi Kasus Aplikasi Kredit Bank

Dari Studi kasus ini, didapat bahwa dengan metode *fuzzy association rule learning* mampu mendeteksi *fraud* dengan baik. Hal ini disebabkan dari proses *training* dan *testing* yang dilakukan kepada 230 *dataset*, *fraud* pada kasus aplikasi kredit bank ini dapat ditangkap dengan akurasi sebesar 0,96. Aturan asosiasi yang dihasilkan dapat menangkap setiap pelanggaran yang dilakukan menjadi *fraud*. Dan aturan/*rule* yang ditambahkan untuk contoh *case* yang normal juga dapat menangkap dengan baik *case-case* yang normal (bukan *fraud*).

6.3.2.2. Evaluasi Studi Kasus proses *procurement* ERP

Dari Studi kasus ini, didapat bahwa dengan metode *fuzzy association rule learning* juga mampu mendeteksi *fraud* dengan baik. Hal ini disebabkan dari proses *training* dan *testing* yang dilakukan kepada 200 *dataset*, *fraud* pada kasus proses *procurement* ERP ini dapat ditangkap dengan akurasi sebesar 0,97. Aturan asosiasi yang dihasilkan dapat menangkap setiap pelanggaran yang dilakukan menjadi *fraud*. Dan aturan/*rule* yang ditambahkan untuk contoh *case* yang normal juga dapat menangkap dengan baik *case-case* yang normal (bukan *fraud*). Dalam percobaan terhadap kasus ini dapat dilihat bahwa aturan/*rule* tambahan yang dibentuk dapat digunakan secara umum untuk mendeteksi *fraud* pada proses bisnis manapun.

6.3.3. Evaluasi Pengujian Kegunaan

Berdasarkan hasil kuesioner pada Tabel 6.20, dapat ditarik kesimpulan bahwa *plugin* cukup mudah untuk digunakan, bermanfaat bagi pengguna. Hal ini dapat dilihat dari 100% setuju bahwa *plugin* mudah digunakan dengan komposisi 50% sangat setuju dan 50% setuju, dan 100% mengatakan bahwa mudah untuk memahami dan melakukan proses pendeteksian *fraud* dengan metode *fuzzy association rule learning*.

BAB VII

KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan yang diambil selama pengerjaan Tugas Akhir, saran-saran tentang pengembangan yang dapat dilakukan terhadap Tugas Akhir ini di masa yang akan datang, dan kegunaan lanjutan dari hasil yang didapat dari Tugas Akhir ini.

7.1. Kesimpulan

Dari hasil pengamatan selama proses perancangan, implementasi, dan pengujian perangkat lunak yang dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Data pelanggaran yang terdapat pada *event log* dari proses bisnis yang berjalan dapat diperoleh dengan menggunakan konsep *conformance checking* pada *process mining* yang memeriksa kesesuaian antara *event log* dan model proses bisnis standar yang dibantu dengan konsep *ontology* dalam pendeteksian kecurangan pada pengambilan keputusan yang diimplementasikan ke dalam *plugin* ProM 6.0.
2. Setiap data pelanggaran yang dihasilkan membutuhkan pembobotan untuk dapat mengetahui pengaruh pelanggaran tersebut terhadap *fraud*. Pembobotan dilakukan dengan menggunakan metode *fuzzy set multiple attribute decision making*.
3. Proses perhitungan bobot dengan metode *fuzzy set multiple attribute decision making* menggunakan data kepentingan dari setiap atribut pelanggaran oleh pakar dan data pelanggaran hasil *conformance checking*.
4. Pendeteksian *fraud* membutuhkan data *training* yang merepresentasikan jenis-jenis pelanggaran yang dilakukan sehingga menghasilkan aturan asosiasi antara atribut pelanggaran beserta nilai *confidence*-nya. Aturan asosiasi beserta nilai *confidence* tersebut digunakan untuk

menentukan suatu *case* termasuk ke dalam kategori Yakin *Fraud*, *Fraud*, *Semi Fraud*, atau Tidak *Fraud*.

5. Pendeteksian *fraud* pada proses bisnis dapat dilakukan dengan akurasi sekitar 0,96 untuk studi kasus pertama dan 0,97 untuk studi kasus kedua dengan menggunakan aturan asosiasi yang dihasilkan pada saat *training* dan penambahan aturan bantuan.
6. Hasil perhitungan dan pendeteksian *fraud* menjadi *plugin* ProM dapat dipadukan dengan menggunakan *plugin* dasar bernama *GettingStarted* yang didapat dari *repository* ProM.

7.2. Saran

Berikut merupakan beberapa saran untuk pengembangan sistem di masa yang akan datang. Saran-saran ini didasarkan pada hasil perancangan, implementasi dan pengujian yang telah dilakukan.

1. Penambahan jenis pelanggaran yang mungkin dapat terjadi pada proses bisnis yang berjalan. Hal ini dapat membuat pendeteksian *fraud* pada proses bisnis semakin kompleks.
2. Perlunya pendalaman dalam penentuan nilai *threshold* dalam pembentukan aturan asosiasi dan penentuan batas kombinasi atribut pelanggaran untuk menghasilkan aturan asosiasi yang efektif.

DAFTAR PUSTAKA

- [1] R. Sarno, "Petri Net Model of ERP Business Process Variations for Small and Medium Enterprises," *Journal of Theoretical and Applied Information Technology*, vol. 54 No.1, pp. 31-38., 10th August 2013.
- [2] J. Stoop, "A case study on the theoretical and practical value of using process mining for the detection of fraudulent behavior in the procurement process," in *Process Mining and Fraud Detection*, Netherlands, Twente University, 2012.
- [3] R. D. Dewandono and R. Sarno, *Process Sequence Mining For Fraud Detection Using Complex Event Processing*, Surabaya, 2013.
- [4] "Report to the Nations on Occupational Fraud and Abuse," ACFE, 2012.
- [5] P. Goldman and H. Kaufman, *Anti-Fraud Risk and Control Workbook*, 2009.
- [6] W. v. d. Aalst, "Process Mining: Discovery, Conformance and Enhancement of Business Processes.," *Springer*, 2011.
- [7] W.v.d.Aalst and A.K.A. Medeiros, "Process Mining and Security: Detecting Anomalous Process Executions and Checking Process Conformance," *Computer Science*, vol. 121, pp. 3-21, 2005.
- [8] D. Sanchez, M. Vila, L. Cerda, and J. Serrano, "Association rules applied to credit card fraud detection," *Expert Systems with Applications*, 2009.
- [9] E. Lundin, H. Kvarnstrom, and E. Jonsson, *A Synthetic Fraud Data*, ICICS, 2002.
- [10] "Surat Edaran Kepada Semua Bank Umum di Indonesia no.13/28/DPNP," Bank Indonesia, Jakarta, 2011.

- [11] M. Jans, N. Lybaert, K. Vanhoof, and J. M. van der Werf, "Business Process Mining for Internal Fraud Risk Reduction : Result of a Case Study".
- [12] M. Jans, N. Lybaert, K. Vanhoof, and J. M. van der Werf, "A business process mining application for internal transaction fraud mitigation," *Expert Systems with Applications*, vol. 38, pp. 13351-13359, 2011.
- [13] S. Huda, R. Sarno, T. Ahmad, and H. A. Santoso, "Identification of Process-based Fraud Patterns in Credit Application," 2013.
- [14] F. Ogwueleka, Data Mining Application in Credit Card Fraud Detection System, Nigeria: Department of Computer Science, University of Abuja, 2011.
- [15] A. Adriansyah, N. Sidorova, and B. F. v. Dongen, "Cost-Based Fitness in Conformance Checking," 2011.
- [16] N. Gehrke and M. Werner, "Process Mining," 2013.
- [17] M. P. Barreiros, A. Grillo, V. Cruz-Machado, and M. R. Cabrita, "Applying Fuzzy sets for ERP Systems Selection within the Construction Industry," 2010.
- [18] S. Gottwald, Universes of Fuzzy Sets and Axiomatizations of Fuzzy Set Theory, Springer: Studia Logica, 2006.
- [19] L. Helm, Fuzzy Association Rules An Implementation in R, Vienna: Vienna University of Economics and Business Administration, 2007.
- [20] S. J. Chuu, A Fuzzy Multiple attributes Decision-Making for the Evaluation of Advanced Manufacturing Technology, Chungli, Taiwan: Department of Business Administration Nanya Institute of Technology.
- [21] M. Roubens, "Fuzzy sets and decision analysis," *Fuzzy Sets and Systems*, vol. 90, pp. 199-206, 1997.
- [22] F. Herrera, L. Martinez, and P.J. Sanchez, "Managing non-homogeneous," *European Journal of Operational Research*, vol. 166, pp. 115-132, 2005.

- [23] M. Delgado, "Mining Fuzzy Association Rules: An Overview," in *BISC Conferece*, 2003.
- [24] K.C.C. Chan and W. H. Au , "An Effective Algorithm for Discovering Fuzzy Rules in Relational Database," *IEEE World Congress on Computational Intelligence*, 1998.
- [25] T. P. Hong, C. S. Kuo, and S. C. Chi, "A Fuzzy Data Mining Algorithm for Quantitative Values," *Knowledge-Based Intelligent Information Engineering Systems*, 1999.
- [26] A. Gyenesei, "A Fuzzy Approach for Mining Quantitative Association Rules," *Turku Centre for Computer Science Technical Reports*, 2000.
- [27] W. v. d. Aalst, B. v. Dongen, C. Gunther, R. Mans, A. d. Medeiros, A. Rozinat, V. Rubin, M. Song, A. Weijters and H. Verbeek, "ProM 4.0 : comprehensive support for real process analysis," in *26th International Conference on Applications and Theory of Petri Nets*, 2007.
- [28] R. Sarno, H. Ginardi, and E. W. Pamungkas, "Clustering of ERP Business Process Fragments," Surabaya, Indonesia, 2013.

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Penulis, **Fernandes Sinaga**, lahir di Kupang, 4 February 1992. Penulis menempuh pendidikan dasar mulai kelas 1 sampai 3 di SD Don Bosco IV Kupang. Dan melanjutkan kelas 4 sampai 6 di SD Hati Kudus Rajawali Makassar. Untuk pendidikan menengah, penulis tempuh di SMP Katolik Rajawali Makassar dan selanjutnya di SMA Negeri 2 Makassar. Selanjutnya penulis melanjutkan pendidikan sarjana di Jurusan Teknik

Informatika, Institut Teknologi Sepuluh Nopember Surabaya. Selama kuliah penulis aktif menjadi administrator Laboratorium Pemrograman 2 Teknik Informatika.

Penulis dalam menyelesaikan pendidikan S1 mengambil bidang minat Rekayasa Perangkat Lunak (*Software Engineering*) dan memiliki ketertarikan di bidang *Mobile Programming*, *Game Development*, dan *Desktop Software Development*. Penulis dapat dihubungi melalui email: nandes.02@gmail.com dan fernandes.sinaga10@mhs.if.its.ac.id

LAMPIRAN A. DIAGRAM

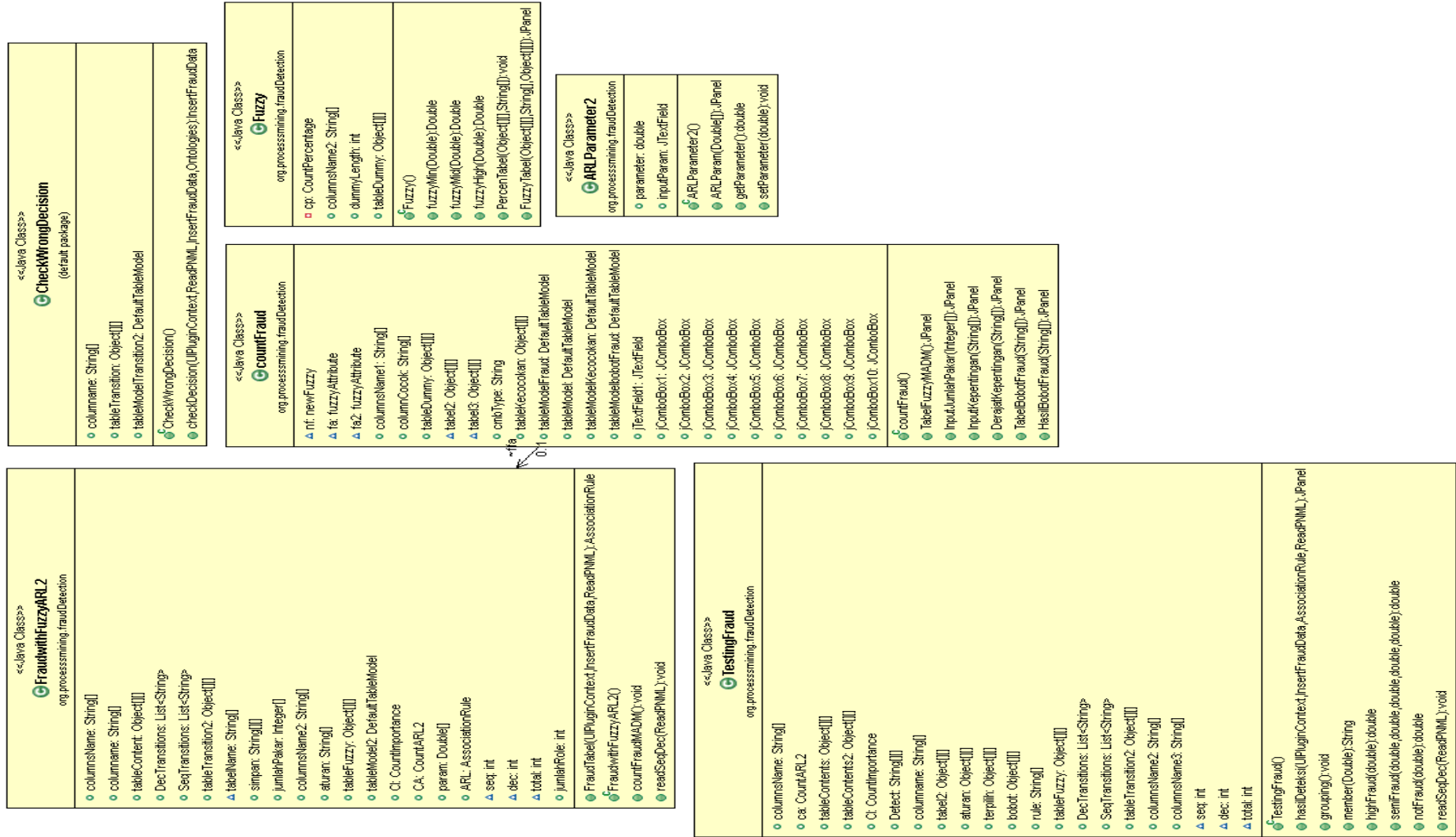


Diagram A.1 Diagram Kelas Antarmuka

[Halaman ini sengaja dikosongkan]

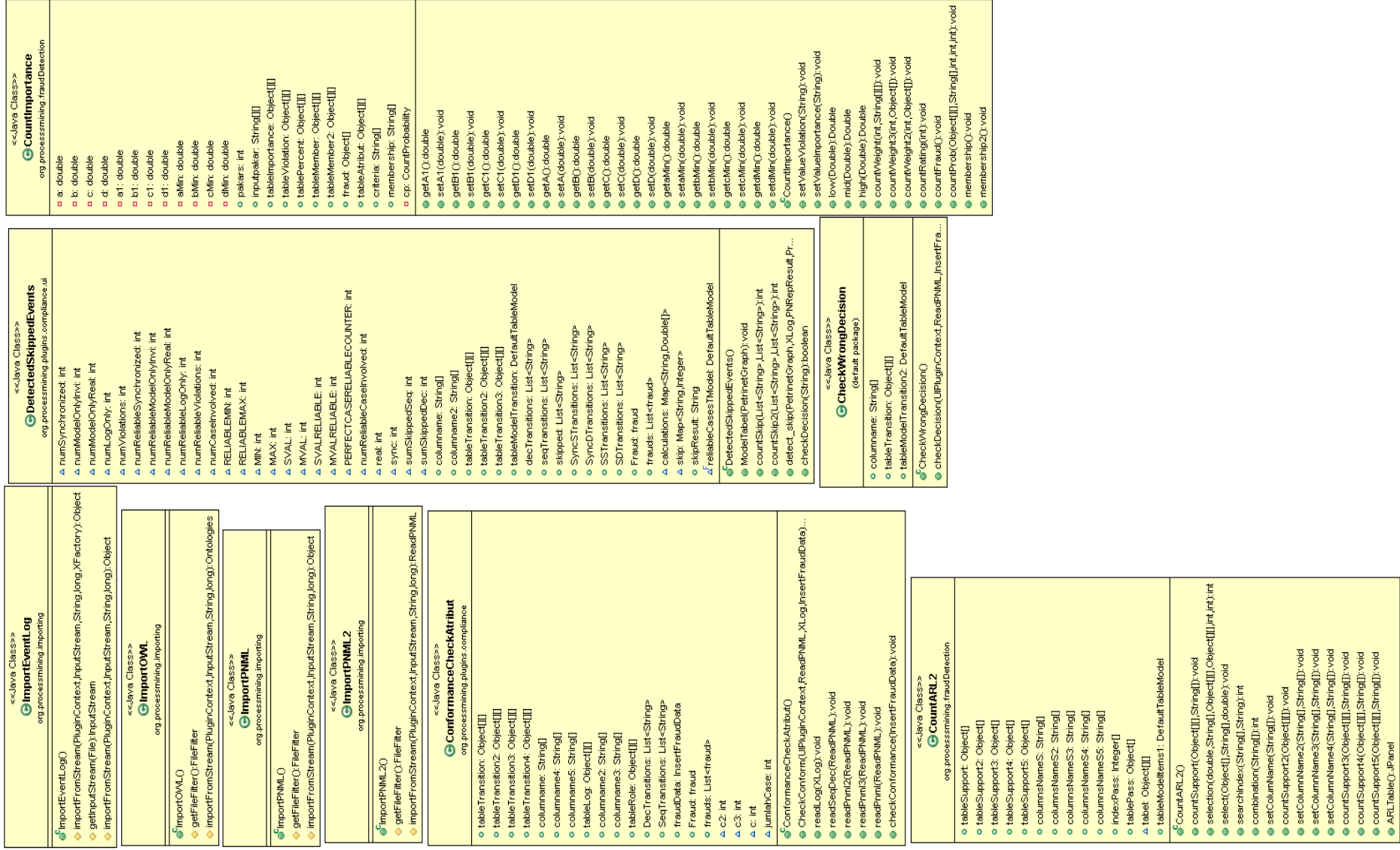


Diagram A.2 Kelas Diagram Kontrol

[Halaman ini sengaja dikosongkan]

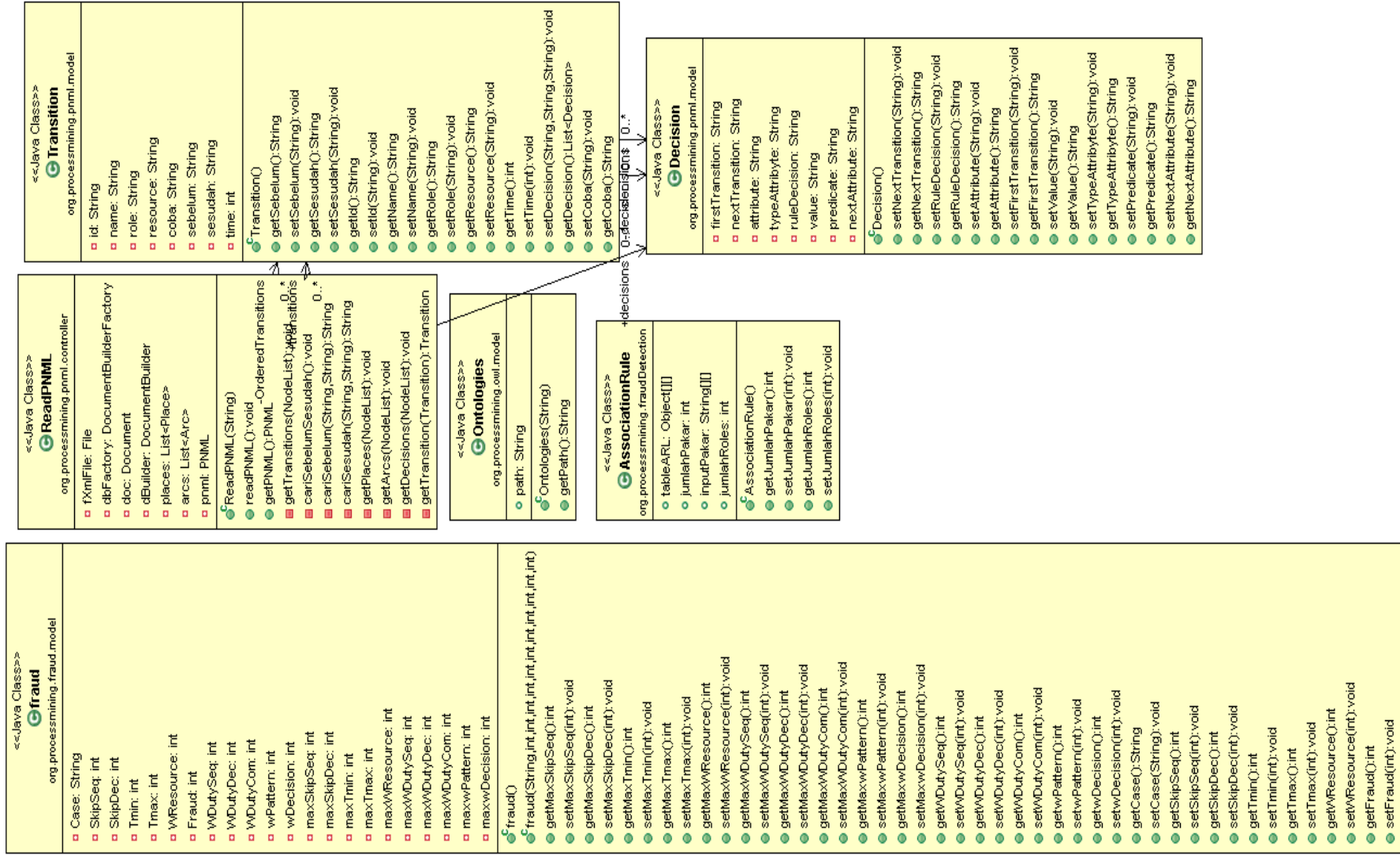


Diagram A.3 Diagram Kelas Data

[Halaman ini sengaja dikosongkan]

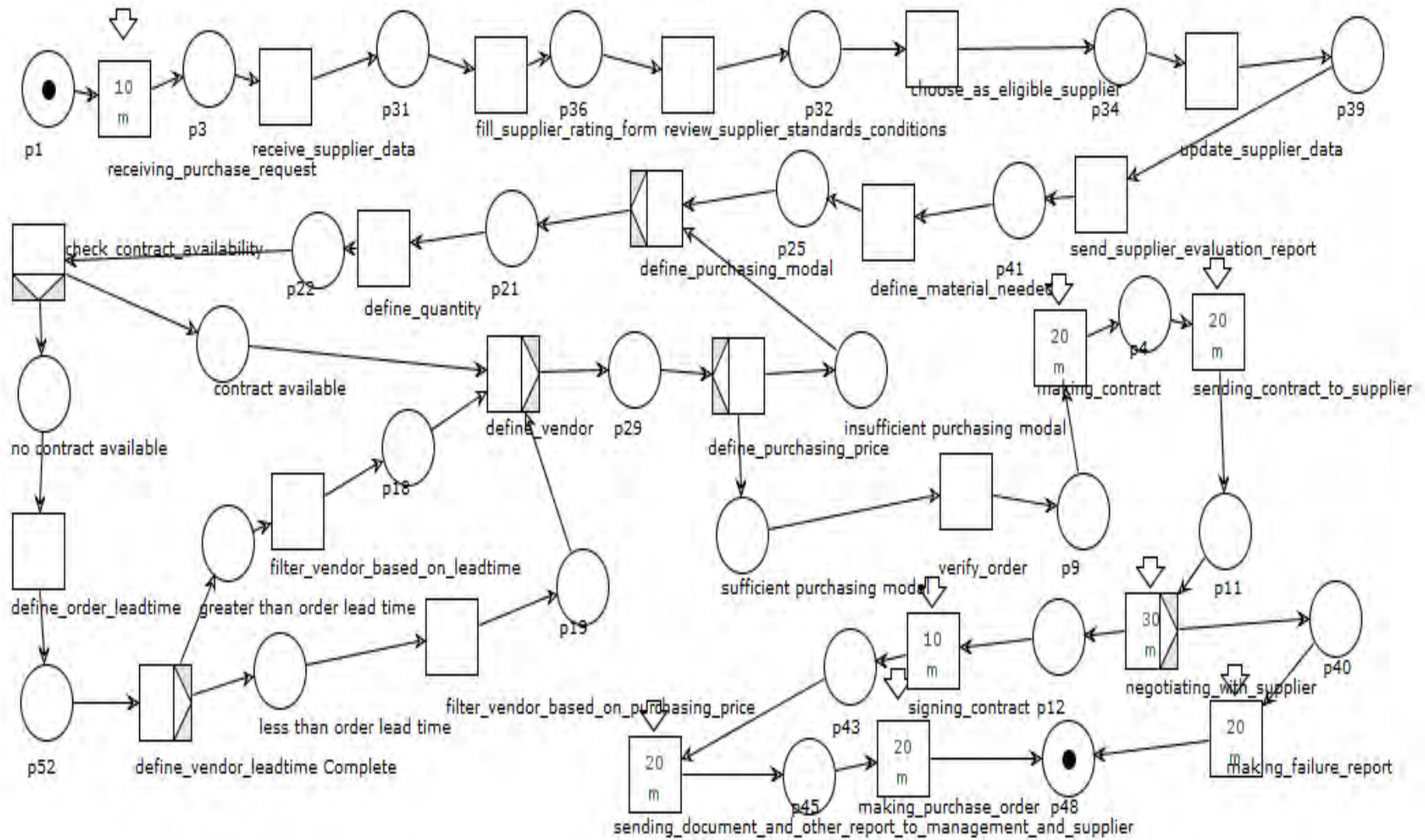


Diagram A.4. Model proses bisnis procurement ERP

[Halaman ini sengaja dikosongkan]

LAMPIRAN B. KUISIONER

Nama : Bagus Ardiansyah

No.	Pernyataan	Pilihan jawaban			
		Sangat setuju	Setuju	Tidak setuju	Sangat tidak setuju
1	Menurut saya, <i>plugin</i> ini mudah digunakan.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	Menurut saya, cukup mudah untuk mengimpor <i>event log</i> ke dalam aplikasi ProM dengan <i>plugin</i> ini.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	Menurut saya, cukup mudah untuk mengimpor model proses ke dalam aplikasi ProM dengan <i>plugin</i> ini.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	Menurut saya, cukup mudah dalam mengkonversi <i>event log</i> ke <i>event log ontology</i>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	Menurut saya, cukup mudah dalam melakukan setiap tahap <i>conformance checking</i> dalam <i>plugin</i> ini.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	Menurut saya, cukup mudah untuk memasukkan data penilaian pakar dan nilai <i>threshold</i> ke dalam <i>plugin</i> .	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	Menurut saya, cukup mudah untuk melakukan pencarian aturan asosiasi dan pendeteksian <i>fraud</i> .	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8	Menurut saya, tabel hasil aturan asosiasi mudah dibaca dan dipahami.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
9	Menurut saya, tabel hasil deteksi <i>fraud</i> mudah dibaca dan dipahami.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
10	Menurut saya, tampilan <i>plugin</i> ini menarik dan mudah dipahami.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Kuisiomer B.1. Bagus Ardiansyah

Nama : Endang Wahyu Pamungkas

No.	Pernyataan	Pilihan jawaban			
		Sangat setuju	Setuju	Tidak setuju	Sangat tidak setuju
1	Menurut saya, <i>plugin</i> ini mudah digunakan.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	Menurut saya, cukup mudah untuk mengimpor <i>event log</i> ke dalam aplikasi ProM dengan <i>plugin</i> ini.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	Menurut saya, cukup mudah untuk mengimpor model proses ke dalam aplikasi ProM dengan <i>plugin</i> ini.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	Menurut saya, cukup mudah dalam mengkonversi <i>event log</i> ke <i>event log ontology</i>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	Menurut saya, cukup mudah dalam melakukan setiap tahap <i>conformance checking</i> dalam <i>plugin</i> ini.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	Menurut saya, cukup mudah untuk memasukkan data penilaian pakar dan nilai <i>threshold</i> ke dalam <i>plugin</i> .	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	Menurut saya, cukup mudah untuk melakukan pencarian aturan asosiasi dan pendeteksian <i>fraud</i> .	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
8	Menurut saya, tabel hasil aturan asosiasi mudah dibaca dan dipahami.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
9	Menurut saya, tabel hasil deteksi <i>fraud</i> mudah dibaca dan dipahami.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10	Menurut saya, tampilan <i>plugin</i> ini menarik dan mudah dipahami.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>

Kuisiner B.2. Endang Wahyu Pamungkas

Nama : *Galang A-D.P.*

No.	Pernyataan	Pilihan jawaban			
		Sangat setuju	Setuju	Tidak setuju	Sangat tidak setuju
1	Menurut saya, <i>plugin</i> ini mudah digunakan.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	Menurut saya, cukup mudah untuk mengimpor <i>event log</i> ke dalam aplikasi ProM dengan <i>plugin</i> ini.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	Menurut saya, cukup mudah untuk mengimpor model proses ke dalam aplikasi ProM dengan <i>plugin</i> ini.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	Menurut saya, cukup mudah dalam mengkonversi <i>event log</i> ke <i>event log ontology</i>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	Menurut saya, cukup mudah dalam melakukan setiap tahap <i>conformance checking</i> dalam <i>plugin</i> ini.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	Menurut saya, cukup mudah untuk memasukkan data penilaian pakar dan nilai <i>threshold</i> ke dalam <i>plugin</i> .	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	Menurut saya, cukup mudah untuk melakukan pencarian aturan asosiasi dan pendeteksian <i>fraud</i> .	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
8	Menurut saya, tabel hasil aturan asosiasi mudah dibaca dan dipahami.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9	Menurut saya, tabel hasil deteksi <i>fraud</i> mudah dibaca dan dipahami.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10	Menurut saya, tampilan <i>plugin</i> ini menarik dan mudah dipahami.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Kuisiner B.3. Galang Amanda

Nama: Chairaja Almas Djeni

No.	Pernyataan	Pilihan jawaban			
		Sangat setuju	Setuju	Tidak setuju	Sangat tidak setuju
1	Menurut saya, <i>plugin</i> ini mudah digunakan.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	Menurut saya, cukup mudah untuk mengimpor <i>event log</i> ke dalam aplikasi ProM dengan <i>plugin</i> ini.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	Menurut saya, cukup mudah untuk mengimpor model proses ke dalam aplikasi ProM dengan <i>plugin</i> ini.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	Menurut saya, cukup mudah dalam mengkonversi <i>event log</i> ke <i>event log ontology</i>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	Menurut saya, cukup mudah dalam melakukan setiap tahap <i>conformance checking</i> dalam <i>plugin</i> ini.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	Menurut saya, cukup mudah untuk memasukkan data penilaian pakar dan nilai <i>threshold</i> ke dalam <i>plugin</i> .	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	Menurut saya, cukup mudah untuk melakukan pencarian aturan asosiasi dan pendeteksian <i>fraud</i> .	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
8	Menurut saya, tabel hasil aturan asosiasi mudah dibaca dan dipahami.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
9	Menurut saya, tabel hasil deteksi <i>fraud</i> mudah dibaca dan dipahami.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
10	Menurut saya, tampilan <i>plugin</i> ini menarik dan mudah dipahami.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Kuisisioner B.4. Chairaja Almas Djeni

Nama : *Ramadhani Tegar P.*

No.	Pernyataan	Pilihan jawaban			
		Sangat setuju	Setuju	Tidak setuju	Sangat tidak setuju
1	Menurut saya, <i>plugin</i> ini mudah digunakan.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	Menurut saya, cukup mudah untuk mengimpor <i>event log</i> ke dalam aplikasi ProM dengan <i>plugin</i> ini.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	Menurut saya, cukup mudah untuk mengimpor model proses ke dalam aplikasi ProM dengan <i>plugin</i> ini.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	Menurut saya, cukup mudah dalam mengkonversi <i>event log</i> ke <i>event log ontology</i>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	Menurut saya, cukup mudah dalam melakukan setiap tahap <i>conformance checking</i> dalam <i>plugin</i> ini.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	Menurut saya, cukup mudah untuk memasukkan data penilaian pakar dan nilai <i>threshold</i> ke dalam <i>plugin</i> .	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	Menurut saya, cukup mudah untuk melakukan pencarian aturan asosiasi dan pendeteksian <i>fraud</i> .	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
8	Menurut saya, tabel hasil aturan asosiasi mudah dibaca dan dipahami.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9	Menurut saya, tabel hasil deteksi <i>fraud</i> mudah dibaca dan dipahami.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10	Menurut saya, tampilan <i>plugin</i> ini menarik dan mudah dipahami.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Kuisisioner B.5. Ramadhani Tegar P.

Nama : AZI PRASTYO

No.	Pernyataan	Pilihan jawaban			
		Sangat setuju	Setuju	Tidak setuju	Sangat tidak setuju
1	Menurut saya, <i>plugin</i> ini mudah digunakan.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	Menurut saya, cukup mudah untuk mengimpor <i>event log</i> ke dalam aplikasi ProM dengan <i>plugin</i> ini.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	Menurut saya, cukup mudah untuk mengimpor model proses ke dalam aplikasi ProM dengan <i>plugin</i> ini.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	Menurut saya, cukup mudah dalam mengkonversi <i>event log</i> ke <i>event log ontology</i>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	Menurut saya, cukup mudah dalam melakukan setiap tahap <i>conformance checking</i> dalam <i>plugin</i> ini.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	Menurut saya, cukup mudah untuk memasukkan data penilaian pakar dan nilai <i>threshold</i> ke dalam <i>plugin</i> .	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	Menurut saya, cukup mudah untuk melakukan pencarian aturan asosiasi dan pendeteksian <i>fraud</i> .	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8	Menurut saya, tabel hasil aturan asosiasi mudah dibaca dan dipahami.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9	Menurut saya, tabel hasil deteksi <i>fraud</i> mudah dibaca dan dipahami.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10	Menurut saya, tampilan <i>plugin</i> ini menarik dan mudah dipahami.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>

Kuisiomer B.6. Azi Prastyo

Nama : Suliadi Marsetya

No.	Pernyataan	Pilihan jawaban			
		Sangat setuju	Setuju	Tidak setuju	Sangat tidak setuju
1	Menurut saya, <i>plugin</i> ini mudah digunakan.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	Menurut saya, cukup mudah untuk mengimpor <i>event log</i> ke dalam aplikasi ProM dengan <i>plugin</i> ini.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	Menurut saya, cukup mudah untuk mengimpor model proses ke dalam aplikasi ProM dengan <i>plugin</i> ini.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	Menurut saya, cukup mudah dalam mengkonversi <i>event log</i> ke <i>event log ontology</i>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	Menurut saya, cukup mudah dalam melakukan setiap tahap <i>conformance checking</i> dalam <i>plugin</i> ini.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	Menurut saya, cukup mudah untuk memasukkan data penilaian pakar dan nilai <i>threshold</i> ke dalam <i>plugin</i> .	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	Menurut saya, cukup mudah untuk melakukan pencarian aturan asosiasi dan pendeteksian <i>fraud</i> .	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8	Menurut saya, tabel hasil aturan asosiasi mudah dibaca dan dipahami.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9	Menurut saya, tabel hasil deteksi <i>fraud</i> mudah dibaca dan dipahami.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10	Menurut saya, tampilan <i>plugin</i> ini menarik dan mudah dipahami.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Kuisisioner B.7. Suliadi Marsetya

Nama : Muhammad Hanif Budiarto

No.	Pernyataan	Pilihan jawaban			
		Sangat setuju	Setuju	Tidak setuju	Sangat tidak setuju
1	Menurut saya, <i>plugin</i> ini mudah digunakan.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	Menurut saya, cukup mudah untuk mengimpor <i>event log</i> ke dalam aplikasi ProM dengan <i>plugin</i> ini.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	Menurut saya, cukup mudah untuk mengimpor model proses ke dalam aplikasi ProM dengan <i>plugin</i> ini.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	Menurut saya, cukup mudah dalam mengkonversi <i>event log</i> ke <i>event log ontology</i>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	Menurut saya, cukup mudah dalam melakukan setiap tahap <i>conformance checking</i> dalam <i>plugin</i> ini.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	Menurut saya, cukup mudah untuk memasukkan data penilaian pakar dan nilai <i>threshold</i> ke dalam <i>plugin</i> .	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	Menurut saya, cukup mudah untuk melakukan pencarian aturan asosiasi dan pendeteksian <i>fraud</i> .	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
8	Menurut saya, tabel hasil aturan asosiasi mudah dibaca dan dipahami.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
9	Menurut saya, tabel hasil deteksi <i>fraud</i> mudah dibaca dan dipahami.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10	Menurut saya, tampilan <i>plugin</i> ini menarik dan mudah dipahami.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Kuisiner B.8. Muhammad Hanif Budiarto

Nama : Febry Amin Nurhidayah

No.	Pernyataan	Pilihan jawaban			
		Sangat setuju	Setuju	Tidak setuju	Sangat tidak setuju
1	Menurut saya, <i>plugin</i> ini mudah digunakan.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	Menurut saya, cukup mudah untuk mengimpor <i>event log</i> ke dalam aplikasi ProM dengan <i>plugin</i> ini.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	Menurut saya, cukup mudah untuk mengimpor model proses ke dalam aplikasi ProM dengan <i>plugin</i> ini.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	Menurut saya, cukup mudah dalam mengkonversi <i>event log</i> ke <i>event log ontology</i>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	Menurut saya, cukup mudah dalam melakukan setiap tahap <i>conformance checking</i> dalam <i>plugin</i> ini.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	Menurut saya, cukup mudah untuk memasukkan data penilaian pakar dan nilai <i>threshold</i> ke dalam <i>plugin</i> .	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	Menurut saya, cukup mudah untuk melakukan pencarian aturan asosiasi dan pendeteksian <i>fraud</i> .	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
8	Menurut saya, tabel hasil aturan asosiasi mudah dibaca dan dipahami.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9	Menurut saya, tabel hasil deteksi <i>fraud</i> mudah dibaca dan dipahami.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10	Menurut saya, tampilan <i>plugin</i> ini menarik dan mudah dipahami.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Kuisiner B.9. Febry Amin Nurhidayah

Nama : Devanda Tamba

No.	Pernyataan	Pilihan jawaban			
		Sangat setuju	Setuju	Tidak setuju	Sangat tidak setuju
1	Menurut saya, <i>plugin</i> ini mudah digunakan.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	Menurut saya, cukup mudah untuk mengimpor <i>event log</i> ke dalam aplikasi ProM dengan <i>plugin</i> ini.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	Menurut saya, cukup mudah untuk mengimpor model proses ke dalam aplikasi ProM dengan <i>plugin</i> ini.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	Menurut saya, cukup mudah dalam mengkonversi <i>event log</i> ke <i>event log ontology</i>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	Menurut saya, cukup mudah dalam melakukan setiap tahap <i>conformance checking</i> dalam <i>plugin</i> ini.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	Menurut saya, cukup mudah untuk memasukkan data penilaian pakar dan nilai <i>threshold</i> ke dalam <i>plugin</i> .	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	Menurut saya, cukup mudah untuk melakukan pencarian aturan asosiasi dan pendeteksian <i>fraud</i> .	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8	Menurut saya, tabel hasil aturan asosiasi mudah dibaca dan dipahami.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
9	Menurut saya, tabel hasil deteksi <i>fraud</i> mudah dibaca dan dipahami.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
10	Menurut saya, tampilan <i>plugin</i> ini menarik dan mudah dipahami.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Kuisiner B.10. Devanda Tamba

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul:

“Rancang Bangun Fuzzy Association Rule Miner untuk Mendeteksi Fraud pada Proses Bisnis Enterprise Resource Planning (ERP)”

Melalui lembar ini, penulis hanya ingin menyampaikan ucapan terima kasih dan penghormatan yang sebesar-besarnya kepada:

1. Bapak, Ibu, adik, kakak dan keluarga yang selalu memberikan dukungan penuh untuk menyelesaikan Tugas Akhir ini.
2. Bapak Riyanarto dan Bapak Munif selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberikan petunjuk selama proses pengerjaan Tugas Akhir ini.
3. Bapak Solichul Huda yang membantu dalam mempelajari kasus *fraud* dan membantu memberikan data-data penelitian.
4. Bapak, Ibu dosen Jurusan Teknik Informatika ITS yang telah banyak memberikan ilmu dan bimbingan yang tak ternilai harganya bagi penulis.
5. Seluruh staf dan karyawan FTif ITS yang banyak memberikan kelancaran administrasi akademik kepada penulis.
6. Teman-teman administrator Laboratorium Pemrograman 2 yaitu Bagus, Agus Tri, Suliadi, Wawang, Anno, Dhea, Deva, Febri yang selalu mencairkan kejenuhan ketika penulis menyusun Tugas Akhir ini.

7. Teman-teman seperjuangan anak didik Tugas Akhir pak Riyanarto yaitu Galang, Bagus, Almas, Firman, Dadang, Farid yang selalu berjuang bersama dan saling menyemangati.
8. Teman-teman angkatan 2010 jurusan Teknik Informatika ITS yang telah menemani perjuangan selama 4 tahun ini atas saran, masukan, dan dukungan terhadap pengerjaan Tugas Akhir ini.
9. Serta pihak-pihak lain yang namanya tidak dapat penulis sebutkan satu -persatu.

Bagaimanapun juga penulis telah berusaha sebaik-baiknya dalam menyusun Tugas Akhir ini, namun penulis mohon maaf apabila terdapat kekurangan, kesalahan maupun kelalaian yang telah penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan selanjutnya.

Surabaya, Juni 2014

Fernandes P. Sinaga

DAFTAR ISI

LEMBAR PENGESAHAN.....	vii
ABSTRAKSI.....	ix
ABSTRACTION.....	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR.....	xix
DAFTAR TABEL.....	xxiii
DAFTAR KODE SUMBER.....	xxv
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Tujuan.....	2
1.3. Rumusan Permasalahan.....	3
1.4. Batasan Permasalahan.....	3
1.5. Metodologi.....	4
1.6. Sistematika Penulisan.....	6
BAB II DASAR TEORI.....	9
2.1. Pendeteksian Kecurangan (<i>Fraud Detection</i>).....	9
2.1.1. <i>Fraud</i>	9
2.1.2. <i>Data Mining for Fraud Detection</i>	12
2.1.3. <i>Process Mining for Fraud Detection</i>	13
2.2. <i>Conformance Checking</i>	15
2.3. <i>Fuzzy Sets Multiple Attribute Decision Making</i>	17
2.4. <i>Association Rule Learning</i>	19
2.4.1. Algoritma Apriori.....	21
2.5. <i>Fuzzy Association Rule Learning</i>	22
2.6. ProM.....	23
2.7. Plugin ProM.....	24
2.8. <i>Event log</i>	24
2.9. Petri Net.....	24
BAB III METODOLOGI PEMECAHAN MASALAH.....	27

3.1.	Conformance Checking	28
3.1.1.	<i>Check Compliance using Conformance Checking</i> 30	
3.1.2.	<i>Conformance Checking for Attribute</i>	30
3.1.3.	<i>Check Wrong Decision</i>	31
3.2.	<i>Fuzzy Set Multi Attribute Decision Making</i>	32
3.3.	<i>Fuzzy Association Rule Learning</i>	39
BAB IV ANALISIS DAN PERANCANGAN SISTEM		43
4.1.	Analisis	43
4.1.1.	Cakupan Permasalahan.....	43
4.1.2.	Deskripsi Umum Sistem.....	44
4.1.3.	Spesifikasi Kebutuhan Perangkat Lunak.....	46
4.1.4.	Aktor.....	48
4.1.5.	Kasus Penggunaan.....	48
4.2.	Perancangan Sistem.....	78
4.2.1.	Perancangan Diagram Kelas.....	78
4.2.2.	Perancangan Antarmuka Pengguna	81
BAB V IMPLEMENTASI		89
5.1.	Implementasi Lapisan Antarmuka.....	89
5.1.1.	Kelas <code>CheckWrongDecision</code>	89
5.1.2.	Kelas <code>FraudwithFuzzyARL2</code>	90
5.1.3.	Kelas <code>countFraud</code>	90
5.1.4.	Kelas <code>Fuzzy</code>	92
5.1.5.	Kelas <code>ARLParameter2</code>	94
5.1.6.	Kelas <code>TestingFraud</code>	94
5.2.	Implementasi Lapisan Kontrol	94
5.2.1.	Kelas <code>ImportEventLog</code>	94
5.2.2.	Kelas <code>ImportOWL</code>	96

5.2.3.	Kelas ImportPNML.....	97
5.2.4.	Kelas ImportPNML2.....	97
5.2.5.	Kelas DetectedSkippedEvents.....	98
5.2.6.	Kelas ConformanceCheckAtribut.....	98
5.2.7.	Kelas CheckWrongDecision.....	104
5.2.8.	Kelas CountARL2.....	104
5.2.9.	Kelas CountImportance.....	104
5.3.	Implementasi Lapisan Data.....	104
5.3.1.	Kelas Fraud.....	104
5.3.2.	Kelas ReadPNML.....	105
5.3.3.	Kelas Transition.....	105
5.3.4.	Kelas Decision.....	105
5.3.5.	Kelas Ontologies.....	105
5.3.6.	Kelas AssociationRule.....	105
5.4.	Implementasi Antarmuka Pengguna.....	105
5.4.1.	Halaman Tampilan Tabel Data Pelanggaran.....	106
5.4.2.	Halaman <i>Input</i> Kepentingan Pakar.....	106
5.4.3.	Halaman <i>Input Threshold</i>	107
5.4.4.	Halaman Aturan Asosiasi.....	108
5.4.5.	Halaman Hasil Deteksi <i>Fraud</i>	108
BAB VI	PENGUJIAN DAN EVALUASI.....	109
6.1.	Lingkungan Pengujian.....	109
6.2.	Skenario Pengujian.....	109
6.2.1.	Pengujian Fungsionalitas.....	110
6.2.2.	Pengujian Pada Studi Kasus.....	127
6.2.3.	Pengujian Kegunaan.....	137

6.3.	Evaluasi Pengujian	140
6.3.1.	Evaluasi Pengujian Fungsionalitas	140
6.3.2.	Evaluasi Pengujian Studi Kasus	141
6.3.3.	Evaluasi Pengujian Kegunaan	142
BAB VII KESIMPULAN DAN SARAN		143
7.1.	Kesimpulan.....	143
7.2.	Saran.....	144
DAFTAR PUSTAKA.....		145
Lampiran A. Diagram.....		149
Lampiran B. Kuisisioner		157
BIODATA PENULIS.....		167

DAFTAR GAMBAR

Gambar 2.1. <i>Event Sequence</i>	10
Gambar 2.2. <i>Event Decision</i>	11
Gambar 2.3. Contoh <i>Wrong Pattern</i>	12
Gambar 2.4. Atribut PBF	12
Gambar 2.5. <i>Conformance Checking in Process Mining</i>	16
Gambar 2.6. <i>Petri Net</i> dasar dengan <i>places</i> (p1,p2,p3,p4,p5,p6) dan transisi (t1,t2,t3,t4,t5).	25
Gambar 3.1. Metodologi	27
Gambar 3.2. Alur Proses Pengecekan Kesesuaian	29
Gambar 3.3. Fungsi Keanggotaan kepentingan.....	33
Gambar 3.4. Data Penelitian Kepentingan	34
Gambar 3.5. Fungsi Keanggotaan Pelanggaran	36
Gambar 3.6. Fungsi Keanggotaan Pelanggaran	40
Gambar 3.7. Rumus Fungsi Keanggotaan Pelanggaran	40
Gambar 3.8. Fungsi Keanggotaan Fraud.....	42
Gambar 3.9. Rumus Fungsi Keanggotaan Fraud.....	42
Gambar 4.1. Diagram Kasus Penggunaan Sistem	49
Gambar 4.2. Diagram Urutan Mengimpor Event Log ke dalam ProM.....	51
Gambar 4.3 Diagram Aktivitas Mengimpor <i>Event Log</i> ke dalam ProM.....	52
Gambar 4.4 Diagram Aktivitas Mengimpor model proses SOP ke Dalam ProM	54
Gambar 4.5. Diagram Urutan Mengimpor model proses SOP ke Dalam ProM.....	55
Gambar 4.6 Diagram Aktivitas Mengkonversi <i>event log</i> ke <i>event log ontology</i>	57
Gambar 4.7 Diagram Urutan Mengkonversi <i>event log</i> ke <i>event log ontology</i>	58
Gambar 4.8 Diagram Aktivitas Melakukan <i>Compliance Check Using Conformance Checking</i>	60

Gambar 4.9. Diagram Urutan Melakukan <i>Compliance Check Using Conformance Checking</i>	61
Gambar 4.10. Diagram Urutan <i>Conformance Checking for Attribute</i>	63
Gambar 4.11 Diagram Aktivitas <i>Conformance Checking for Attribute</i>	64
Gambar 4.12. Diagram Aktivitas Melakukan <i>Check Wrong Decision</i>	67
Gambar 4.13 Diagram Urutan Melakukan Check Wrong Decision	68
Gambar 4.14. Diagram Aktivitas Melakukan pendeteksian <i>Fraud</i> dengan <i>Fuzzy Association Rule Learning</i>	72
Gambar 4.15. Diagram Urutan Melakukan pendeteksian <i>Fraud</i> dengan <i>Fuzzy Association Rule Learning</i>	73
Gambar 4.16. Diagram Aktivitas Melakukan pendeteksian <i>Fraud</i> dengan data aturan asosiasi	76
Gambar 4.17. Diagram Urutan Melakukan pendeteksian <i>Fraud</i> dengan data aturan asosiasi	77
Gambar 4.18. Rancangan Model Tiga-Lapis	78
Gambar 4.19 Rancangan Halaman Tampilan Tabel Data Pelanggaran	82
Gambar 4.20 Rancangan Antarmuka <i>Input</i> Kepentingan Pakar..	84
Gambar 4.21 Rancangan Antarmuka Halaman <i>Input Threshold</i>	86
Gambar 4.22. Rancangan Antarmuka Aturan Asosiasi.....	87
Gambar 4.23. Rancangan Antarmuka Hasil Deteksi <i>Fraud</i>	88
Gambar 5.1 Tampilan Tabel Data Pelanggaran	106
Gambar 5.2 Tampilan <i>Input</i> Kepentingan Pakar	107
Gambar 5.3 Tampilan <i>Input Threshold</i>	107
Gambar 5.4. Tampilan Halaman Aturan Asosiasi.....	108
Gambar 5.5. Tampilan Halaman Hasil Deteksi <i>Fraud</i>	108
Gambar 6.1 Hasil Program pada Uji Pertama	113
Gambar 6.2 Hasil Program pada Uji Kedua.....	114
Gambar 6.3 Hasil Program pada Uji Ketiga.....	114
Gambar 6.4 Tampilan Hasil Uji Fitur mengimpor model proses.	116

Gambar 6.5 Tampilan Hasil Uji Fitur mengimpor model proses	117
Gambar 6.6. Hasil Pengujian Mengkonversi Event Log ke Event Log Ontology	118
Gambar 6.7. Hasil Pengujian Melakukan Compliance Check Using Conformance Checking.....	120
Gambar 6.8 Hasil Pengujian Melakukan <i>Conformance Checking for Attribute</i>	121
Gambar 6.9. Hasil Pengujian Melakukan Check Wrong Decision	123
Gambar 6.10 Hasil Pengujian Melakukan pendeteksian <i>Fraud</i> dengan <i>Fuzzy Association Rule Learning</i>	124
Gambar 6.11 Hasil Pengujian Melakukan pendeteksian <i>Fraud</i> dengan data aturan asosiasi	126
Gambar 6.12. Proses bisnis aplikasi kredit bank.....	127

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 5.1 Fungsi <code>checkDecision()</code> untuk menampilkan data aturan transisi pada aplikasi ProM.....	90
Kode Sumber 5.2. Fungsi <code>InputJumlahPakar()</code> untuk menampilkan <i>textbox</i> untuk masukan jumlah pakar pada aplikasi ProM.....	92
Kode Sumber 5.3 Fungsi <code>FuzzyTabel()</code> pada Kelas Fuzzy	93
Kode Sumber 5.4. Fungsi <code>importFromStream()</code> pada Kelas <i>ImportEventLog</i>	96
Kode Sumber 5.5 Fungsi <code>importFromStream()</code> pada kelas <i>ImportOWL</i>	96
Kode Sumber 5.6 Fungsi <code>importFromStream</code> pada Kelas <i>ImportPNML</i>	97
Kode Sumber 5.7. Fungsi <code>checkConformance</code> pada kelas <i>CheckConformanceAtribut</i>	103

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 3.1. Tabel Kepentingan	32
Tabel 3.2. Hasil Perhitungan Bobot Kepentingan	35
Tabel 3.3. Tabel Derajat Pelanggaran	35
Tabel 3.4. Rumus Fungsi Keanggotaan Pelanggaran.....	36
Tabel 3.5. Tabel Perhitungan bobot Pelanggaran.....	37
Tabel 4.1. Daftar Kebutuhan Fungsional Perangkat Lunak	47
Tabel 4.2 Daftar Kode Diagram Kasus Penggunaan.....	49
Tabel 4.3 Spesifikasi Kasus Penggunaan Mengimpor <i>event log</i> ke dalam ProM.....	50
Tabel 4.4 Spesifikasi Kasus Penggunaan Mengimpor model proses SOP ke Dalam ProM.....	53
Tabel 4.5 Spesifikasi Kasus Penggunaan Mengkonversi <i>event log</i> ke <i>event log ontology</i>	55
Tabel 4.6 Spesifikasi Kasus Penggunaan Melakukan <i>Compliance Check Using Conformance Checking</i>	58
Tabel 4.7 Spesifikasi Kasus Penggunaan Melakukan <i>Conformance Checking for Attribute</i>	62
Tabel 4.8. Spesifikasi Kasus Penggunaan Melakukan <i>Check Wrong Decision</i>	65
Tabel 4.9. Spesifikasi Kasus Penggunaan Melakukan pendeteksian <i>Fraud</i> dengan <i>Fuzzy Association Rule Learning</i>	68
Tabel 4.10. Spesifikasi Kasus Penggunaan Melakukan pendeteksian <i>Fraud</i> dengan data aturan asosiasi.....	74
Tabel 4.11 Spesifikasi Atribut Antarmuka Tabel Data Pelanggaran	82
Tabel 4.12 Spesifikasi Atribut Antarmuka <i>Input</i> Kepentingan Pakar.....	84
Tabel 4.13 Spesifikasi Atribut Antarmuka Halaman <i>Input Threshold</i>	86
Tabel 4.14. Spesifikasi Atribut Antarmuka Rancangan Antarmuka Aturan Asosiasi	87
Tabel 4.15 Spesifikasi Atribut Antarmuka Rancangan Antarmuka Hasil Deteksi Fraud.....	88

Tabel 6.1 Pengujian Fitur Mengimpor Event Log.....	111
Tabel 6.2 Pengujian Fitur Mengimpor model proses SOP.....	115
Tabel 6.3 Pengujian Fitur Mengkonversi <i>event log</i> ke <i>event log ontology</i>	117
Tabel 6.4 Pengujian Fitur Melakukan <i>Compliance Check Using Conformance Checking</i>	119
Tabel 6.5 Pengujian Fitur Melakukan <i>Conformance Checking for Attribute</i>	120
Tabel 6.6. Pengujian Fitur Melakukan Check Wrong Decision	122
Tabel 6.7 Pengujian Fitur Melakukan pendeteksian <i>Fraud</i> dengan <i>Fuzzy Association Rule Learning</i>	123
Tabel 6.8 Pengujian Fitur Melakukan pendeteksian <i>Fraud</i> dengan data aturan asosiasi.....	125
Tabel 6.9. Tabel Perhitungan bobot pelanggaran.....	129
Tabel 6.10. Tabel Aturan asosiasi	130
Tabel 6.11. Hasil Deteksi <i>Fraud</i>	130
Tabel 6.12. Akurasi metode terhadap kasus aplikasi kredit bank	132
Tabel 6.13. Hasil Akurasi dengan bantuan <i>rule</i>	132
Tabel 6.14. Tabel Bobot Pelanggaran kasus <i>procurement</i> ERP	134
Tabel 6.15. Aturan asosiasi pelanggaran <i>procurement</i> ERP	134
Tabel 6.16. Hasil Pendeteksian <i>Fraud</i> kasus <i>procurement</i> ERP	135
Tabel 6.17. Hasil perhitungan akurasi pendeteksian <i>fraud</i> pada proses <i>procurement</i> ERP dengan bantuan rule yang sama	136
Tabel 6.18 Daftar Pertanyaan Kuesioner.....	137
Tabel 6.19 Daftar Penguji Plugin	138
Tabel 6.20 Hasil Kuesioner	139
Tabel 6.21 Rangkuman Hasil Pengujian	140