**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**INSTITUT TEKNLOGI SEPULUH NOPEMBER**

**USULAN TUGAS AKHIR**

# IDENTITAS PENGUSUL

**NAMA : MOHAMMAD FARID NAUFAL**

**NRP : 5110100094**

**DOSEN WALI : Anny Yuniarti, S.Kom, M.Comp.Sc**

**DOSEN PEMBIMBING : 1. Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc. Ph.D.**

# JUDUL TUGAS AKHIR

“Penerapan Metode Proses Mining Pada Ontologi Event Log Untuk Mendeteksi Fraud pada Proses Bisnis ERP”

# LATAR BELAKANG

Saat ini hampir semua perusahaan menjalankan proses bisnisnya dengan menerapkan sistem *Enterprise Resource Planning* (ERP). ERP memiliki banyak proses bisnis yang saling terhubung satu dengan yang lainnya. Perusahaan yang memiliki proses bisnis yang terdefinisi dengan baik pasti akan menjalankan proses bisnisnya dengan efektif dan efisien. Namun pada praktiknya proses bisnis yang dieksekusi tidak selalu sesuai dengan proses bisnis yang telah didefiniskan. Hal ini dapat diketahui dengan menganalisa sebuah *Event Logs* yang dihasilkan dari proses bisnis. Ketidaksesuaian antara proses bisnis yang dijalankan dengan proses bisnis yang didefiniskan ini disebut dengan *fraud*.

Event Log yang dihasilkan dari sebuah proses bisnis dapat dilakukan analisa yang lebih mendalam. Proses Mining memiliki potensi secara efektif untuk melakukan audit dengan mengekstrak pengetahuan yang didapat dari *event logs* yang terekam dalam sistem informasi bisnis [[1](#Mie11)]. Namun analisa yang dilakukan masih terbatas dikarenakan hanya berdasarkan tabel yang terdapat di dalam *event log* tanpa memperhatikan relasi antar tabel. Analisa lanjut terhadap *Ontology Event Log* yang memperhatikan relasi antar *instance* dan memiliki sifat *triples* yaitu terdiri dari subjek, predikat, dan objek dalam menggambarkan konsep memungkinkan untuk lebih mempertajam dan mempermudah analisa proses mining untuk mendeteksi adanya fraud.

Pada tugas akhir ini akan diimplementasikan analisa proses mining terhadap *Ontology Event Logs*. Analisa event log yang dilakukan adalah pada *control flow*, *role* *resource*, *throughput time*, dan *decision point*. Hasil yang diharapkan adalah terdeteksinya *fraud* pada *case* tertentu dengan tepat.

# RUMUSAN MASALAH

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

1. Bagaimana mengubah *event log* menjadi *ontology event log*.
2. Bagaimana menganalisa *event log* dan *ontology event log* menggunakan metode proses mining.
3. Bagaimana cara mendeteksi *fraud* pada case tertentu dengan tepat.
4. Bagaimana cara memberikan tingkatan nilai *fraud* pada case tertentu.

# BATASAN MASALAH

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

1. Bahasa yang digunakan adalah bahasa Java.
2. Data uji yang digunakan adalah berupa *event log* proses bisnis ERP dengan *case id* yang telah ditandai.
3. Bahasa query untuk penggalian data menggunakan SPARQL.
4. Tugas akhir ini tidak bertujuan untuk penggalian data atau analisa algoritma.
5. Metode yang digunakan hanya untuk mendeteksi fraud berdasarkan pada proses bisnis.
6. Metode yang digunakan fokus pada pendeteksian fraud bukan penanggulangan fraud.
7. Metode yang digunakan untuk pendeteksian fraud tidak secara *real time processing*.

# TUJUAN PEMBUATAN TUGAS AKHIR

Tujuan dari Tugas Akhir ini adalah sebagai berikut:

1. Untuk menambah tingkat respon sebuah sistem dalam mendeteksi fraud.
2. Mengimplementasikan proses mining yang berguna untuk penggalian data pada proses bisnis ERP.
3. Mengkombinasikan ontologi event log dan proses mining dalam pendeteksian fraud.

# MANFAAT TUGAS AKHIR

Manfaat dari Tugas Akhir ini adalah sebagai berikut:

1. Mempermudah pengubahan *event log* menjadi *ontology event log*.
2. Mempermudah analisa *ontology event log* menggunakan metode proses mining.
3. Mempermudah mendeteksi *fraud* pada case tertentu dengan tepat.
4. Mempermudah cara memberikan tingkatan nilai *fraud* pada *case* tertentu.

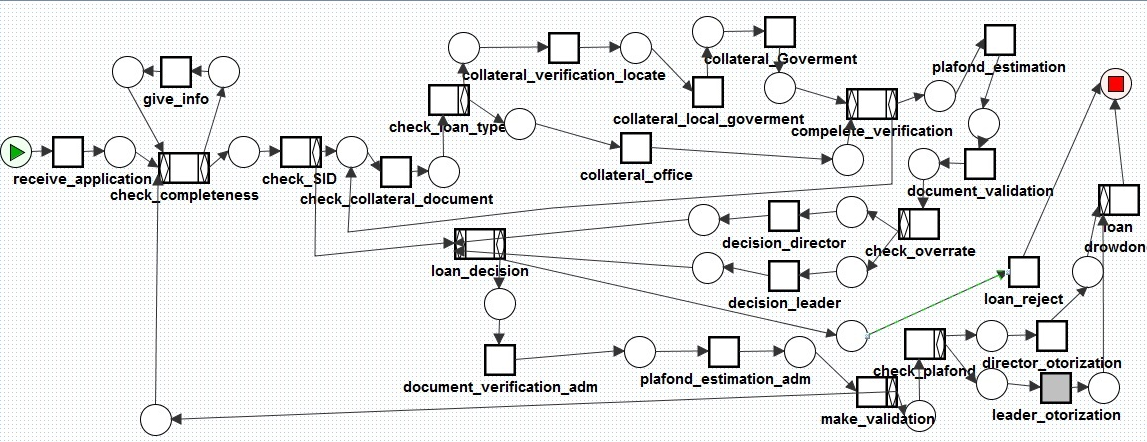
# TINJAUAN PUSTAKA

## Fraud

Penelitian sebelumnya yang berkaitan dengan *fraud* telah mendefiniskan arti *fraud* sebagai sebuah kecurangan yang sengaja dilakukan oleh pihak tertentu untuk tujuan dan maksud tertentu. Selain itu, [[2](#ELu02)] memiliki definisi fraud sebagai penyalahgunaan sistem organisasi tanpa berdasarkan aturan yang legal. Penelitian [[3](#HKv03)] mendefiniskan fraud sebagai penipuan kriminal, yang menggunakan keterangan palsu untuk meningkatkan keuntungan yang tidak adil. Dalam studi kasus pada tugas akhir ini terdapat beberapa macam tipe fraud yang mungkin terjadi dalam sebuah proses bisnis yaitu skipped activity, wrong roles, wrong troughput time, dan false decision.

### Kasus Fraud pada Proses Bisnis

Kemungkinan terjadinya *fraud* pada proses bisnis dalam kehidupan nyata sangat besar. Dalam tugas akhir ini kita akan memberikan contoh munculnya *fraud* pada proses bisnis *credit application* pada bank.



*Gambar 1. Proses Bisnis Credit Application*

Gambar 1 merupakan sebuah proses bisnis pengajuan kredit pada suatu bank. Aktifitas 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.

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

Aktifitas *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. Aktifitas ini penting untuk memeriksa keaslian dokumen yang telah diajukan.

Aktifitas *plafond estimation* digunakan untuk memperkirakan jumlah kredit yang akan dicairkan. Hal ini tergantung dari pengecekan agunan yang telah dicek sebelumnya.

Aktifitas *check overrate* digunakan untuk menentukan aktifitas selanjutnya yang akan dieksekusi. Yaitu apakah menuju *decision director* atau *decision leader*. Hal ini tergantung pada jumlah pengajuan kredit.

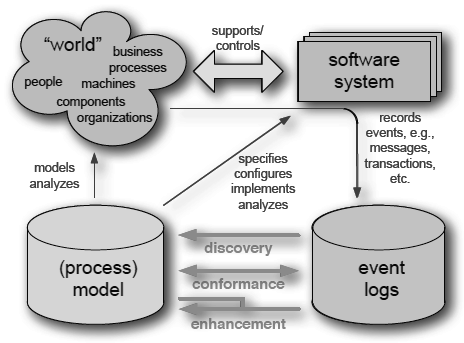
Aktifitas *document\_verification\_adm* dan *plafond\_verification\_adm* digunakan untuk mengecek kembali dokumen dan plafond yang telah. Aktifitas selanjutnya yang dilakukan setelah dua aktifitas ini adalah *make validation* yaitu untuk memvalidasi kelengkapan. Apabila validasi telah valid maka akan 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 aktifitas yang terlewat atau tidak dilakukan dengan benar. Hal ini menandakan adanya kecurangan pada aktifitas 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 aktifitas yang mengeluarkan output percabangan lebih dari satu aktifitas seperti pada *check overrate* di mana seharusnya pengajuan kredit yang lebih dari lima puluh juta rupiah seharusnya menuju pada aktifitas decision director namun pada kenyataannya menuju pada decision leader.

## Proses Mining

Proses mining merupakan studi yang sedang berkembang saat ini. Yang bertujuan untuk menggali informasi dan pengetahuan yang didapatkan dari *event logs*. Proses mining menyediakan jembatan penting antara data mining dan pemodelan proses. Proses mining sangat berguna berdasarkan dua alasan. Pertama proses mining dapat digunakan sebagai alat untuk mengetahui bagaimana aktor atau *resource* dan prosedur bekerja dalam proses bisnis. Kedua proses mining dapat digunakan sebagai *Delta analysis*, yaitu membandingkan proses aktual yang terjadi di dunia nyata dengan proses yang telah ditetapkan sebelumnya [[4](#WMP)].



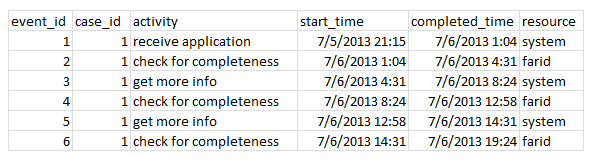
*Gambar 2 Arsitektur Process Mining*

ProM (*www.processmining.org*) merupakan sebuah tool yang berguna untuk proses penggalian data. Tools ini bersifat *open source* dan memiliki banyak *plugin* yang dapat dikembangkan. *Plugin* yang terdapat pada ProM mendukung proses penggalian data yang berhubungan dengan pendeteksian *fraud*, diantaranya adalah *Conformance Checking, Originator Task by Matrix,* dan *Dotted Chart Analysis*. Input dari ProM yang digunakan pada kasus ini adalah berupa *Event Log*.

### Event Logs

*Event logs* adalah sebuah rekaman kejadian yang dieksekusi pada saat proses bisnis dijalankan atau juga seringkali disebut *history, audit trail,* atau *transaction log*. *Event logs* ini dijadikan bahan baku untuk menggali informasi dari proses bisnis yang telah dijalankan. *Event Logs* berisi informasi tentang aktifitas apa yang dijalankan oleh siapa, kapan, dan data apa yang diterima. Misalnya dalam studi kasus pengajuan kredit terdapat aktifitas *make decision* yang dilakukan oleh seorang manager pada waktu tertentu, dalam menentukan keputusan manager harus mempertimbangkan kelengkapan persyaratan pengaju kredit. Kelengkapan pengaju kredit adalah berupa data atribut yang tercatat pada *event logs*.

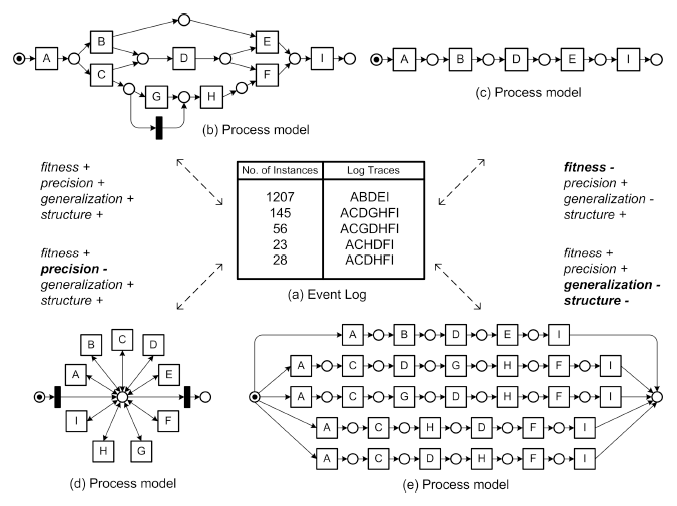
*Event logs* harus terdiri dari *case id*, *activity*, *start time*, *complete* *time, resource*. *Case id* menunjukkan id satu proses instan yang berjalan dari awal sampai akhir. *Activity* menunjukkan aktifitas apa yang dieksekusi dalam satu *event*. *Start time* menunjukkan waktu dimulainya sebuah aktifitas dieksekusi. *Complete time* menunjukkan waktu berakhirnya sebuah aktifitas dieksekusi. *Resource* menunjukkan aktor yang mengeksekusi aktifitas. Gambar 3 menunjukkan contoh *event logs*.



*Gambar 3. Contoh Event Log*

### Proses Discover

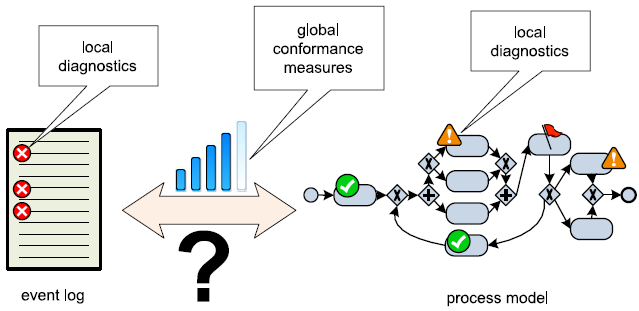
Proses discover adalah sebuah aktifitas penting dalam proses mining. Proses ini membangun model aktual dari proses yang terekam dalam *event logs.* Tujuan lebih lanjut dari proses discover adalah untuk mendapat pengetahuan dari proses yang ada di *event logs*. Beberapa paper yang membahas tentang proses mining mengusulkan algoritma untuk menemukan proses model yang aktual. Algoritma alpha adalah algoritma pertama yang diperkenalkan untuk mendiscover proses model dari *event log* [[5](#WMP04)]. Algoritma ini memperkenalkan hubungan kausal antar aktifitas. Bagaimanapun juga algoritma alpha tidak dapat menjangkau permasalahan tertentu (e.g., *length-one and two loop, noise, implicit dependency, non-free choice, and invisible task*). Algoritma yang lain diusulkan untuk mengatasi masalah ini seperti Alpha++ dan Heuristic Miner.



*Gambar 4. Process* Discover [[6](#ARo07)]

### Conformance Checking

Dalam proses audit, conformance checking seharusnya dilakukan sesudah discover proses model. Terdapat beberapa teknik conformance seperti A\* algorithm, *Cost-Based Fitness Analysis*. Tujuan utamanya adalah menghitung perbedaan perilaku antara proses model dan *event logs*. Sehingga tingkat kesamaan dan perbedaannya bisa dihitung dan diselidiki.



*Gambar 5 . Conformance Checking pada Proses Mining* [[7](#Wil11)]

ProM memilki plugin yang dapat digunakan untuk melakukan *conformance checking* yang membutuhkan *event logs* dan model proses sebagai input. Keluaran yang dihasilkan adalah berupa persen kemiripan setiap *case* yang ada di *event logs* dengan proses model.

### Analisa Control Flow

Pada bagian ini akan dilakukan pengecekan flow atau urutan terjadinya aktifitas pada sebuah case yang tersimpan dengan *event logs* dan dibandingkan dengan SOP model. *Control Flow* merupakan salah satu kelebihan yang dimiliki oleh process mining dibandingkan dengan data mining. *Control Flow* menginvestigasi pola yang ada dalam data set, di mana terdapat beberapa kemungkinan fraud pada proses.

Dalam mendeteksi fraud SOP model dalam file berbentuk pnml akan dilakukan matching dengan semua pola yang ada dalam event log. Apabila terdapat case yang memiliki pola berbeda dengan SOP maka kemungkinan terdapat fraud pada case tersebut

### Analisa Role Resource

Pada bagian ini akan dilakukan analisa *role* atau *resource* pada *event log*. Dari setiap event yang telah dieksekusi tersimpan *resource* yang mengeksekusi event tersebut. Dalam mendeteksi fraud akan dilakukan pemeriksaan dari setiap event apakah telah sesuai dengan *role* SOP atau tidak.

### Analisa Throughput Time

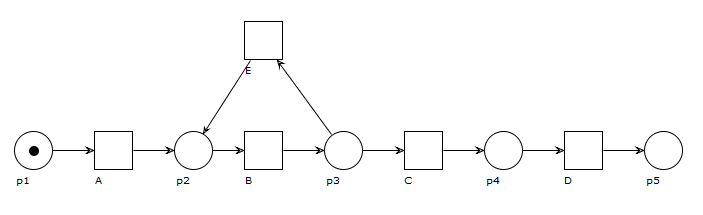
Pada bagian ini akan dianalisa eksekusi waktu pada setiap event yang dieksekusi. SOP akan menyimpan *rule* *Troughput Time*. Kemungkinan fraud yang terjadi pada setiap event adalah event tersebut dieksekusi terlalu cepat atau lambat dari SOP *Throughput Time*.

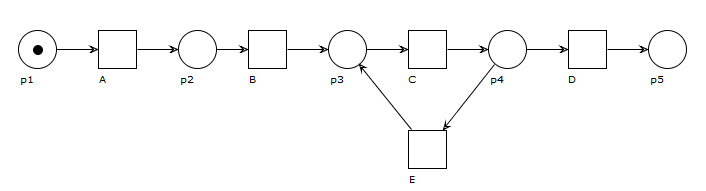
### Analisa Decision Point

Pada bagian ini akan dianalisa pada event yang terdapat percabangan. Event ini biasanya terkait dengan data atribut atau content. Tujuannya adalah untuk memverifikasi lebih lanjut data yang terkait dalam sebuah case.

### Structural Similarity

Pada bagian ini akan dibandingkan bentuk atau model *petri net* yang ada di SOP dengan setiap *case* yang ada di *event logs.* Structural similarity membandingkan pada struktur *petri net* yang terdiri dari *place, transition,* dan *edge*. Pada *place* dan *transition* akan dibandingkan identifier yang dimiliki, sedangkan *edge* akan dibandingkan *source* dan *output*.





Gambar 6 Perbandingan *petri net*

Untuk menghitung kesamaan antara dua *petri net* adalah sebagai berikut:

Pada Gambar 6 didapatkan similarity sebesar 0.818

### Behavioural Similarity

Pada bagian ini akan dibandingkan behaviour antara 2 *petri net* yaitu dengan membandingkan kemungkinan urutan eksekusi dengan memperhatikan relasi atau disebut juga *Transition Adjacent Relation* antara dua *petri net*. Untuk menghitung kemiripannya adalah sebagai berikut:

Pada Gambar 6 *Petri Net* pertama memiliki 5 relasi eksekusi yaitu AB, BC, CD, BE, EB. Kemudian, untuk *Petri Net* kedua memiliki 5 relasi juga AB, BC, CD, CE, EC. Sehingga jumlah irisan yang sama adalah 3 dan masing-masing proses memiliki 5 TARS. Jadi didapatkan nilai kesamaan 9/25 = 0,36.

## Ontologi

Model data yang digunakan dalam Tugas Akhir ini adalah ontologi. Ontologi menjembatani komunikasi dalam level semantik. Tujuan dari penggunaan ontologi adalah pertukaran data tidak hanya pada sintaksis biasa, tetapi juga pada level semantik. Komponen utama dari ontologi adalah konsep, relasi, instan, dan aksiom. Konsep merupakan sekumpulan kelas atau entitas dalam sebuah domain. Hubungan antar konsep dapat juga disebut hirarki.

Relasi mereperesentasikan hubungan atau interaksi antar konsep atau properti dari sebuah konsep. Relasi dibagi menjadi dua jenis yaitu taksonomi dan asosiatif. Taksonomi menyatakan hubungan antar konsep dengan super dan sub konsep tersebut. Sedangkan asosiatif menyatakan hubungan antar konsep. Kumpulan dari sebuah relasi juga dapat dikategorikan sebagai hirarki. Relasi juga memiliki properti yang menjelaskan karakteristik dari relasi tersebut seperti kardinalitas atau ketergantungan konsep yang direlasikan.

Instan merupakan bentuk reperesentasi dari sebuah konsep. Dalam sebuah ontologi instan tidak harus ada dikarenakan ontologi diharapkan hanya menggambarkan konseptualitas dari sebuah domain. Namun, penentuan sebuah konsep dan instan adalah tergantung pada kasus yang direpresentasikan dalam ontologi. Contoh pada kasus perkuliahan “Mata Kuliah” adalah sebuah konsep dan “Bahasa” adalah instan dari konsep “Mata Kuliah” namun “Bahasa” dapat juga menjadi konsep apabila terdapat “Indonesia” yang merupakan instan dari “Bahasa”.

Aksiom merupakan sebuah aturan pada kelas dan instan. Dalam hal ini, properti dari sebuah relasi merupakan aksiom. Namun, aksiom juga memiliki aturan secara keseluruhan seperti setiap mata kuliah setidaknya diajar oleh seorang dosen.

Untuk mempermudah penjelasan, konsep dilambangkan sebagai c dan kumpulan konsep dilambangkan sebagai C (c. Relasi dilambangkan dengan r dan kumpulan relasi dilambangkan dengan R (r. Instan dilambangkan dengan i dan kumpulan instan sebagai I (i sedangkan aksiom dengan A0 [[8](#Ani11)]. Gambar 6 menunjukkan ontologi log.



*Gambar 6. Contoh Ontologi Log Credit Card Application*

### Bahasa Ontologi

Untuk dapat digunakan ontologi harus diekspresikan dalam bentuk yang nyata. Sebuah bahasa ontologi terdiri dari beberapa komponen yang menjadi sebuah struktur ontologi antara lain XML, RDF/S, dan OWL.

XML *(Extensible Markup Language)* merupakan sebuah sintaksis untuk sebuah dokumen terstruktur, namun dokumen XML belum menggunakan batasan-batasan semantik. RDF *(Resource Description Framework)* adalah model data untuk objek dan relasi yang menyediakan sintaksis semantik sederhana yang dapat disajikan dalam sintaksis XML.

RDF *Schema* adalah sebuah kosakata properti dan kelas dari sumber RDF. OWL *(Ontology Web Language)* menambahkan beberapa kosakata untuk menjelaskan properti dan kelas, antara lain relasi antar kelas (misal *disjointness*). Keunggunlan OWL dibandingkan RDF adalah sifat kardinalitasnya. Tabel 1 menunjukkan sifat triples yang dimiliki Ontologi RDF.

|  |  |  |
| --- | --- | --- |
| **Subject (S)** | **Property (P)** | **Object (O)** |
| Event 1 | ConductedBy | Person1 |
| Event 2 | ConductedBy | Person1 |
| Event 3 | ConductedBy | Person1 |
| Event 4 | ConductedBy | Person2 |
| Event 5 | ConductedBy | Person2 |
| Decision 1 | DecisionOf | Event 1 |
| Loan 1 | LoanOf | Event 2 |
| Loan 2 | LoanOf | Event 3 |
| Loan 3 | LoanOf | Event 5 |

Tabel 1. RDF Triples

### SPARQL *(Simple Protocol and RDF Query Language)*

SPARQL *(Simple Protocol and RDF Query Language)* merupakan salah satu bahasa yang digunakan untuk melakukan query pada file ontologi rdf. Sama halnya query pada database SQL, SPARQL digunakan untuk mendapatkan data-data yang diinginkan. SQL digunakan untuk melakukan query data berupa database yang terdiri satu atau beberapa tabel. Sedangkan SPARQL digunakan untuk melakukan query pada rdf yang berupa triples (subjek, predikat, dan objek), konjungsi, dan disjungsi. Berikut merupakan contoh query SPARQL yang menampilkan data *event* yang dieksekusi oleh *person*

**PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>**

**PREFIX owl: <http://www.w3.org/2002/07/owl#>**

**PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>**

**PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>**

**PREFIX db: <http://biostorm.stanford.edu/db\_table\_classes/DSN\_jdbc.mysql.//localhost.3306/CreditApp#>**

**SELECT ?subject ?person**

**WHERE { ?subject db:event.CONDUCTEDBY ?person.**

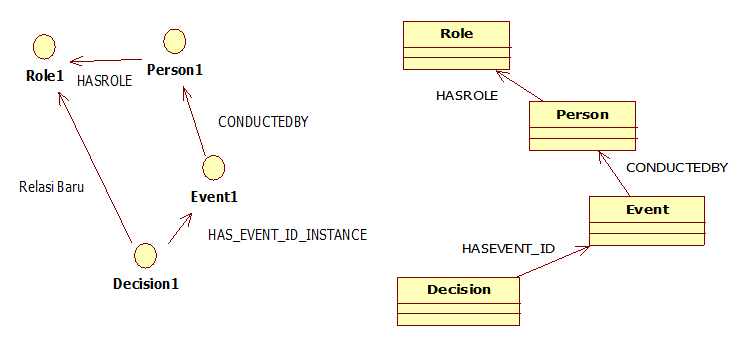
**FILTER (?person = "Farid")**

**}**

Pada tugas akhir ini SPARQL akan digunakan sebagai salah satu alat untuk menggali/melakukan query data yang ada di ontologi event logs. SPARQL memiliki kelebihan dalam menggali data pada file ontologi dikarenakan rdf memiliki sifat triples yang berarti relasi antar objek berupa kata kerja dan hal ini tidak dimiliki oleh query SQL.

### Kelebihan Ontologi dibandingkan Relational Database

* Proses query menggunakan ontologi yang berupa graph pattern menjadi lebih simple dikarenakan menggunakan subjek, predikat, dan objek dalam mendapatkan hasil.
* Informasi yang disimpan dalam ontologi menjadi lebih eksplisit dan lebih mudah dipahami.
* Dapat membentuk relasi baru antar individu berdasarkan knowledge base yang didapatkan. Serta memudahkan proses query yang melibatkan relasi banyak tabel. Lihat Gambar 7.
* Sebuah record dalam database dianggap sebagai object/instance sehingga lebih object oriented dalam menggali informasi. Lihat Gambar 6.

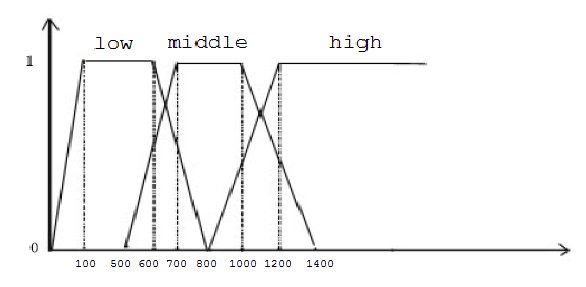


*Gambar 7. Query yang melibatkan banyak tabel antara ontologi dan relational*

### Pembobotan Menggunakan Fuzzy Membership Function Pada Ontology da Relational Database

Pembobotan menggunakan Fuzzy Membership Function merupakan pembobotan yang diterapkan pada sebuah value sehingga bobot pada value tersebut berada diantara nilai [0,1].

Contoh membership function diterapkan pada besarnya jumlah loan amount

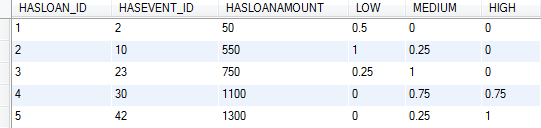


*Gambar 8. Fuzzy Membership Function*

Dengan a=0, b=100, c=600, d=800

Dengan a=500, b=700, c=1000, d=1400

Dengan a=800, b=1200



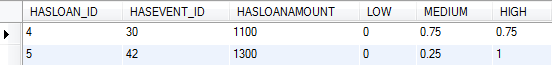
*Gambar 9. Database Loan Amount*

Query Relatinal Database untuk mendapatkan loan yang memiliki membership HIGH >= 0.25

**SELECT \***

**FROM loan**

**WHERE loan.high >= 0.25**

****

*Gambar 10. Hasil Query Relational Database*

Query Ontologi untuk mendapatkan loan yang memiliki membership HIGH >= 0.25

**SELECT ?loan ?high**

**WHERE { ?loan db:loan.HIGH ?high**

**FILTER(?high >= 0.25)}**

****

*Gambar 11. Hasil Query Ontologi*

Deteksi Fraud Wrong Decision Menggunakan pembobotan

**Loan kurang dari 500 seharusnya dilakukan perform checks for small amount**

Query berikut digunakan untuk deteksi fraud dengan bobot Low >= 0.5

**SELECT ?case ?act**

**WHERE {?loan db:loan.HASEVENT\_ID\_INSTANCE ?eventins.**

**?loan db:loan.HASLOANAMOUNT ?amt.**

**?loan db:loan.LOW ?bobot.**

**?eventins db:event.HASCASE\_ID ?case.**

**?events db:event.HASCASE\_ID ?case.**

**?events db:event.HASACTIVITY ?act**

**FILTER(?amt<=500)**

**FILTER(?bobot>=0.5)**

**FILTER(?act = "perform\_checks\_for\_large\_amount")**

**}**

# RINGKASAN ISI TUGAS AKHIR

Aplikasi yang akan dibangun akan menerima sebuah inputan sebuah ontologi event log dan sebuah aturan proses bisnis yang kemudian akan dianalisa setiap *case* yang terdapat di dalam ontologi event log berdasarkan aturan yang ada. *Event logs* dapat diperoleh dari proses bisnis ERP yang telah dikonversikan menjadi bentuk ontologi. Setiap *case* yang melanggar sebuah aturan akan diindikasikan sebagai sebuah *fraud* dengan tingkatan tertentu.

Analisa setiap *case* untuk mendeteksi *fraud* dilakukan berdasarkan empat tipe investigasi. Yaitu analisa *control flow* untuk mendeteksi apakah ada aktifitas yang terlewati atau tidak dieksekusi berdasarkan proses model, analisa *resource* untuk memeriksa apakah setiap event yang ada dalam sebuah *case* dieksekusi oleh aktor yang benar, analisa *throughput time* untuk memeriksa apakah lama waktu sebuah event dieksekusi sesuai, dan analisa *decision point* untuk memeriksa apakah eksekusi sebuah event sesuai dengan data yang ada. Gambar 7 menunjukkan arsitektur sistem yang akan dibangun.



*Gambar 8. Arsitektur Sistem*

# METODOLOGI

## Penyusunan proposal tugas akhir

Proposal tugas akhir ini berisi rencana pembangunan sebuah perangkat lunak yang dapat menggali data *event logs* proses bisnis ERP guna mendeteksi adanya *fraud*. *Event logs* yang akan digali datanya adalah berupa ontologi yang memiliki kunggulan *triples* (terdiri dari subjek, predikat, dan objek dalam menggambarkan sebuah konsep).

Pada bab tiga telah dijelaskan mengenai latar belakang serta isu yang terjadi saat ini mengenai penggalian data pada proses bisnis. Penerapan proses mining memiliki nilai tambah dalam penggalian data. Pada bab empat dijelaskan rumusan masalah yang akan dibahas pada tugas akhir ini yang berisi permasalahan yang harus dipecahkan. Pada bab lima dan enam dijelaskan batasan masalah serta tujuan utama. Pada bab tujuh dijelaskan tentang manfaat yang didapat dalam tugas akhir ini.

Pada bagian tinjauan pustaka dijelaskan tentang literatur yang mendukung dalam pengerjaan tugas akhir yaitu metode yang digunakan. Metode yang digunakan adalah tentang proses mining pada ontologi event log. Selanjutnya pada bagian ringkasan tugas akhir dijelaskan tentang arsitektur sistem serta fitur yang akan dibangun. Pada bagian metodologi dijelaskan tentang tahap-tahap pengerjaan tugas akhir ini seperti penyusunan proposal tugas akhir, studi literatur, analisis dan desain perangkat lunak, implementasi, pengujian dan evaluasi, dan penyusunan buku tugas akhir. Pada bagian terakhir akan menjelaskan tentang jadwal kegiatan serta daftar pustaka.

## Studi Literatur

Studi literatur yang digunakan untuk menunjang pengerjaan tugas akhir ini adalah:

1. Penggunaan metode proses mining dalam pengalian data.
2. Mengubah data *event logs* menjadi bentuk ontologi.
3. Penggunaan query SPARQL dalam penggalian data ontologi.
4. Mendeteksi *case* yang mengandung *fraud* dengan analisa *control flow, role resource, throughput time,* dan *decision point*.

## Analisis dan Desain Perangkat Lunak

Analisis dan desain perangkat lunak yang dilakukan adalah berdasarkan rumusan masalah yang telah dijelaskan sebelumnya. Kebutuhan fungsional dari perangkat lunak adalah dapat menganalisa *event logs* dalam bentuk ontologi menggunakan metode proses mining untuk mendeteksi adanya *fraud* berdasarkan aturan proses bisnis yang ada dengan tepat dan tingkatan tertentu.

## Implementasi Perangkat Lunak

Aplikasi yang akan dibangun nantinya akan menerima masukan sebuah ontologi event log dan menghasilkan keluaran berupa data case yang mengandung *fraud*. Dalam pengembangan aplikasi dibutuhkan:

1. *IDE Eclipse Kepler*
2. Java Development Kit 7.0
3. Java Run Time Environment
4. Protégé 4.2 dan 3.4
5. Librari ProM

## Pengujian dan Evaluasi

Pengujian dan evaluasi menggunakan metode *blacbox testing* dengan menggunakan ontologi event log sebagai masukan pada aplikasi yang akan dibangun. Keluaran yang dihasilkan adalah berupa data *case* yang mengandung *fraud* dan akan diperiksa apakah *case* yang terdeteksi benar-benar mengandung *fraud* dan melanggar aturan proses bisnis yang telah ditentukan sebelumnya.

## Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat. Sistematika penulisan buku tugas akhir secara garis besar antara lain:

1. Pendahuluan
   1. Latar Belakang
   2. Rumusan Masalah
   3. Batasan Tugas Akhir
   4. Tujuan
   5. Metodologi
   6. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Pengujian dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka

# JADWAL KEGIATAN

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tahapan | Bulan (Tahun 2013) | | | | | | | | | | | | |
| Oktober | | | | Nopember | | | | Desember | | | |
| Analisa kebutuhan dan studi literatur |  |  |  |  |  |  |  |  |  |  |  |  |
| Perancangan system |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementasi |  |  |  |  |  |  |  |  |  |  |  |  |
| Uji coba dan evaluasi |  |  |  |  |  |  |  |  |  |  |  |  |
| Penyusunan buku |  |  |  |  |  |  |  |  |  |  |  |  |

# DAFTAR PUSTAKA

x

|  |  |
| --- | --- |
| [1] | Michael Alles, Miklos Vasarhelyi Mieke Jans, "Process Mining of Event Logs in Internal Auditing," p. 29, 2011. |
| [2] | E. Lundin, H. Kvarnstr, and E. Jonsson., "A Synthetic Fraud Data," , 2002. |
| [3] | H Kvarnstrom and E. Jonsson, "Synthesizing test data for fraud detection systems ," , 2003. |
| [4] | A.K.A. de Medeiros W.M.P. van der Aalst, "Process Mining and Security: Detecting Anomalous Process Execution and Checking Process Conformance". |
| [5] | W.M.P. van der Aalst, A.J.M.M. Weijters, and L. Maruster, "Workflow mining: Discovering process models from event logs," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1128–1142, 2004. |
| [6] | A. Rozinat, A.K. Alves de Medeiros, C.W. G¨unther, A.J.M.M. Weijters, and W.M.P. van der Aalst, "Towards an Evaluation Framework for Process Mining Algorithms," *BPM Center*, 2007. |
| [7] | Wil M.P. van der Aalst, *Process Mining: Discovery, Conformance and Enhancement of Business Processes*.: Springer, 2011. |
| [8] | Yeni Anistyasari, "Ontologi Berbobot untuk Pencarian Nama Mata Kuliah Berdasarkan Isi Dokumen pada Sistem Pengelolaan Muatan Pembelajaran," Surabaya, 2011. |
| [9] | Riyanarto Sarno Rahadian Dustrial Dewandono, *Process Sequence Mining for Fraud Detection Using Complex Event Processing*., 2013, p. 29. |

x