



**TUGAS AKHIR - KI091391**

# ***Rancang Bangun Workflow Management System pada kasus ERP***

Chairaja Almas Djeni  
NRP 5110100709

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

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



**FINAL PROJECT - KI091391**

# ***DESIGN ARCHITECTURE OF WORKFLOW MANAGEMENT SYSTEM APPLICATION***

Chairaja Almas Djeni  
NRP 5110100709

Advisor  
Prof. Drs.Ec. Ir. Riyanarto Sarno, M.Sc, Ph.D  
Dwi Sunaryono, S.Kom, M.Kom

INFORMATICS DEPARTMENT  
Faculty of Information Technology  
Sepuluh Nopember Institute of Technology  
Surabaya 2014

# Rancang Bangun *Workflow Management System* pada kasus ERP

Nama Mahasiswa : Chairaja Almas Djeni  
NRP : 5110 100 709  
Jurusan : Teknik Informatika FTIf-ITS  
Dosen Pembimbing 1 : Prof. Drs.Ec. Ir. Rivanarto Sarno, M.Sc,  
Ph.D  
Dosen Pembimbing 2 : Dwi Sunaryono, S.Kom, M.Kom

## ABSTRAKSI

Para perusahaan umumnya membutuhkan aplikasi yang dapat dikonfigurasi dengan mudah. Hal ini dapat terwujud jika aplikasi tersebut didukung oleh sistem *Workflow Management System* (WFMS). WFMS berfungsi untuk menyimpan berbagai *workflow* yang sama dan bervariasi dari proses bisnis. Penelitian ini berfokus pada desain arsitektur yang dapat menyimpan berbagai *workflow* yang sama dan bervariasi. Untuk mengakomodir berbagai variasi dari *workflow*, penelitian ini menggunakan pendekatan ontologi dari *Web Ontology for Service* (OWL-S) untuk menganotasikan *workflow* agar dapat dilakukan *query* secara semantik. WFMS yang dibangun juga menyimpan *event log* yang dihasilkan oleh aktivitas dari *workflow*. *Event log* ini menggunakan standar *event log* pada *process mining* sehingga bisa dianalisis lebih lanjut menggunakan aplikasi *process mining* seperti ProM. Setelah proses ujicoba, arsitektur WFMS ini terbukti menghasilkan akurasi yang tinggi dari *query* OWL-S dan *event log* yang bisa dianalisis menggunakan aplikasi ProM.

**Kata kunci:** *BPEL, Workflow, Workflow Management System*

*[Halaman ini sengaja dikosongkan]*

# Design Architecture of *Workflow Management System Applications*

Student Name : Chairaja Almas Djeni  
NRP : 5110 100 709  
Major : Teknik Informatika FTIf-ITS  
Advisor 1 : Prof. Drs.Ec. Ir. Rryanarto Sarno, M.Sc,  
Ph.D  
Advisor 2 : Dwi Sunaryono, S.Kom, M.Kom

## ABSTRACTION

*Most companies need application which is easily configurable. This need can be fulfilled if the application is supported by Workflow Management System. WFMS will be functioned as workflow storage especially the common one. For accomodate every business process variation, this architecture use the ontology approach using Web Ontology for Services (OWL-S) for annotating workflow description. This application also generate event log form the activity of workflow. This event log is standardized using process mining standard. So, this event log can be used for analytical purposes using process mining.*

**Keywords:** *Business Process Execution Language, Workflow Management System, Workflow*

*[Halaman ini sengaja dikosongkan]*

## LEMBAR PENGESAHAN

Rancang Bangun *Workflow Management System* pada kasus  
*ERP*

## TUGAS AKHIR

Ditajukan 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 :

**CHAIRAJA ALMAS DJENI**  
NRP : 5110 100 709

Disetujui oleh Dosen Pembimbing Tugas Akhir :

Prof. Drs.Ec. Ir. Rryanarto Sarno, M.Sc., Ph.D.  
NIP. 195908031986011001

(pembimbing 1)

Dwi Sunaryono, S Kom, M. Kom  
NIP. 197205281997021001

(pembimbing 2)

**SURABAYA**  
**JULI 2014**

## KATA PENGANTAR

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

### *“Rancang Bangun Workflow Management System pada kasus 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 Dwi selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberikan petunjuk selama proses pengerjaan Tugas Akhir ini.
3. Bapak, Ibu dosen Jurusan Teknik Informatika ITS yang telah banyak memberikan ilmu dan bimbingan yang tak ternilai harganya bagi penulis.
4. Seluruh staf dan karyawan FTif ITS yang banyak memberikan kelancaran administrasi akademik kepada penulis.
5. Teman-teman administrator Laboratorium Pemrograman yaitu Endang, Galang, Dmitri, Fauzi, Nada, dan Adi yang selalu menyemangati ketika penulis menyusun Tugas Akhir ini.
6. Teman-teman seperjuangan anak didik Tugas Akhir pak Riyanarto yaitu Galang, Fernandes, Bagus, Firman, Dadang, dan Farid yang selalu berjuang bersama dan saling menyemangati.



7. 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.
8. 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, Juli 2014

Chairaja Almas Djani

## DAFTAR ISI

LEMBAR PENGESAHAN.....	vii
ABSTRAKSI.....	ix
ABSTRACTION.....	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR .....	xix
DAFTAR TABEL .....	xxi
DAFTAR KODE SUMBER .....	xxiii
BAB I PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Tujuan.....	3
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. <i>Business Process Execution Language</i> .....	9
2.2. Standar Log pada <i>Process Mining</i> .....	13
2.3. <i>Workflow Management System</i> .....	14
2.4. <i>Web Ontology Language for Services</i> .....	15
2.5. <i>SPARQL Ontology Query Language</i> .....	17
2.6. <i>Semantic Similarity</i> .....	18
BAB III METODOLOGI PEMECAHAN MASALAH.....	21
3.1. <i>BPEL Server Engine</i> .....	21
3.2. Arsitektur WFMS.....	22
3.3. <i>Event Log Generator</i> .....	25
3.4. Metode Pencarian Menggunakan OWL-S.....	28
BAB IV ANALISIS DAN PERANCANGAN SISTEM .....	31
4.1. Analisis.....	31
4.1.1. Cakupan Permasalahan.....	31

4.1.2.	Deskripsi Umum Sistem.....	31
4.1.3.	Spesifikasi Kebutuhan Perangkat Lunak.....	33
4.1.4.	Aktor.....	33
4.1.5.	Kasus Penggunaan.....	34
4.2.	Perancangan Sistem.....	43
4.2.1.	Perancangan Kelas.....	43
4.2.2.	Perancangan Antarmuka Pengguna.....	46
<b>BAB V IMPLEMENTASI .....</b>		<b>53</b>
5.1.	Implementasi Lapisan Antarmuka.....	53
5.1.1.	Implementasi Halaman Utama .....	53
5.1.2.	Implementasi Halaman Detail <i>Workflow</i> .....	54
5.1.3.	Implementasi Halaman Unggah <i>Workflow</i> .....	55
5.1.4.	Implementasi Halaman Pencarian <i>Workflow</i> .....	55
5.2.	Implementasi Lapisan Kontrol .....	56
5.2.1.	Kelas Unggah.....	56
5.2.2.	Kelas <i>UserOperation</i> .....	56
5.2.3.	Kelas <i>FindWorkflow</i> .....	57
5.2.4.	Kelas <i>WorkflowDetails</i> .....	60
5.3.	Implementasi Lapisan Data .....	61
5.3.1.	Kelas <i>Database Operation</i> .....	61
5.3.2.	Kelas <i>EventLog</i> .....	61
5.3.3.	Kelas <i>Instance</i> .....	61
5.3.4.	Kelas <i>OWLQuery</i> .....	62
5.3.5.	Kelas <i>Wordnet</i> .....	62
5.3.6.	Kelas <i>WorkflowDetails</i> .....	62
<b>BAB VI PENGUJIAN DAN EVALUASI .....</b>		<b>63</b>
6.1.	Lingkungan Pengujian.....	63

6.2.	Skenario Pengujian.....	63
6.2.1.	Pengujian Fungsionalitas.....	65
6.2.2.	Pengujian Pada Studi Kasus .....	72
6.2.3.	Pengujian Kegunaan.....	77
6.3.	Evaluasi Pengujian .....	80
6.3.1.	Evaluasi Pengujian Fungsionalitas .....	80
6.3.2.	Evaluasi Pengujian Studi Kasus .....	81
BAB VII KESIMPULAN DAN SARAN .....		83
7.1.	Kesimpulan.....	83
7.2.	Saran.....	84
7.2.1.	Peningkatan Metode Similaritas .....	84
DAFTAR PUSTAKA.....		85
Lampiran A. Diagram.....		87
Lampiran B. Kuisisioner .....		91
Lampiran C. Detail Percobaan .....		101
BIODATA PENULIS.....		103

*[Halaman ini sengaja dikosongkan]*

## DAFTAR GAMBAR

Gambar 2.1 Contoh <i>message</i> pada BPEL.....	10
Gambar 2.2 Contoh <i>port type</i> pada BPEL.....	11
Gambar 2.3 Contoh <i>partner link type</i> di BPEL.....	11
Gambar 2.4 Contoh <i>partner link</i> di BPEL.....	12
Gambar 2.5 Bagian-bagian dari <i>service profile</i> pada OWL-S.....	15
Gambar 2.6 Hubungan atau <i>overlap</i> antara WSDL dan OWL-S	17
Gambar 2.7 Contoh <i>query</i> SPARQL.....	17
Gambar 3.1 Arsitektur WFMS yang dipakai.....	24
Gambar 3.2 Alur dari arsitektur WFMS.....	25
Gambar 3.3 Alur metode pembuatan <i>event log</i> pada WFMS.....	27
Gambar 3.4 <i>Query</i> SPARQL yang digunakan .....	28
Gambar 4.1 Alur aplikasi WFMS yang dibangun .....	32
Gambar 4.2 Diagram kasus penggunaan sistem.....	34
Gambar 4.3 Diagram urutan mengunggah BPEL ke dalam aplikasi. ....	36
Gambar 4.4 Diagram aktivitas mengunggah BPEL ke dalam aplikasi .....	37
Gambar 4.5 Diagram aktivitas melihat informasi <i>workflow</i> .....	38
Gambar 4.6 Diagram urutan melihat informasi <i>workflow</i> .....	39
Gambar 4.7 Diagram aktivitas mencari <i>workflow</i> .....	40
Gambar 4.8 Diagram urutan mencari <i>workflow</i> .....	41
Gambar 4.9 Diagram aktivitas mengunduh <i>event log</i> .....	42
Gambar 4.10 Diagram urutan mengunduh <i>event log</i> .....	43
Gambar 4.11 Rancangan halaman utama pada sistem .....	47
Gambar 4.12 Rancangan antarmuka halaman informasi detail...	48
Gambar 4.13 Rancangan antarmuka halaman pencarian <i>workflow</i> .....	50
Gambar 4.14 Rancangan antarmuka halaman unggah <i>workflow</i> .....	51
Gambar 5.1 Antar muka halaman utama.....	53
Gambar 5.2 Antarmuka halaman informasi detil <i>workflow</i> .....	54
Gambar 5.3 Antarmuka halaman unggah <i>workflow</i> .....	55
Gambar 5.4 Antarmuka halaman pencarian <i>workflow</i> .....	56

Gambar 6.1 Proses bisnis <i>sale order</i> .....	64
Gambar 6.2 Hasil pengunggahan <i>workflow</i> .....	66
Gambar 6.3 Hasil pengujian melihat informasi <i>workflow</i> .....	68
Gambar 6.4 Hasil uji fitur pencarian <i>workflow</i> .....	70
Gambar 6.5 Hasil <i>event log</i> yang dibuka di aplikasi ProM .....	72
Gambar 6.6 <i>Workflow</i> yang digunakan dalam pengujian.....	76
Gambar 6.7 Hasil <i>workflow</i> yang telah terimplementasi.....	77

## DAFTAR KODE SUMBER

Kode Sumber 5.1 Fungsi signup pada Kelas UserOperation.....	57
Kode Sumber 5.2 Fungsi calculateScore pada Kelas FindWorkflow.....	58
Kode Sumber 5.3 Fungsi doOWL pada Kelas FindWorkflow .....	60
Kode Sumber 5.4 Fungsi meta_generator pada Kelas WorkflowDetails.....	61



*[Halaman ini sengaja dikosongkan]*

## DAFTAR TABEL

Tabel 2.1 Contoh <i>event log</i> sesuai standar <i>process mining</i> .....	14
Tabel 2.2 Contoh penggunaan rumus similaritas .....	19
Tabel 3.1 Tabel berbagai tingkat <i>web service mining</i> .....	26
Tabel 3.2 Contoh pemakaian rumus dari Persamaan 3.1 .....	30
Tabel 4.1 Daftar kebutuhan fungsional perangkat lunak.....	33
Tabel 4.2 Daftar kode diagram kasus penggunaan.....	34
Tabel 4.3 Spesifikasi kasus penggunaan mengunggah BPEL ke WFMS .....	35
Tabel 4.4 Spesifikasi kasus penggunaan melihat informasi <i>workflow</i> .....	37
Tabel 4.5 Spesifikasi kasus penggunaan menampilkan <i>event log</i> .....	39
Tabel 4.6 Spesifikasi kasus mengunduh <i>event log</i> .....	41
Tabel 4.7 Spesifikasi atribut antarmuka halaman utama.....	47
Tabel 4.8 Spesifikasi atribut antarmuka halaman informasi detail .....	49
Tabel 4.4.9 Spesifikasi atribut antarmuka halaman informasi detail.....	50
Tabel 4.10 Spesifikasi atribut antarmuka halaman unggah <i>workflow</i> .....	51
Tabel 6.1 Pengujian fitur mengunggah BPEL.....	65
Tabel 6.2 Pengujian fitur melihat informasi <i>workflow</i> .....	67
Tabel 6.3 Pengujian fitur mencari <i>workflow</i> .....	69
Tabel 6.4 Pengujian fitur mengunduh <i>event log</i> .....	70
Tabel 6.5 Tabel hasil perhitungan <i>threshold</i> .....	74
Tabel 6.6 Tabel hasil pengujian pada berbagai proses bisnis ERP .....	75
Tabel 6.7 Daftar pertanyaan kuesioner.....	78
Tabel 6.8 Daftar penguji plugin .....	79
Tabel 6.9 Hasil kuesioner .....	79
Tabel 6.10 Rangkuman hasil pengujian .....	81

*[Halaman ini sengaja dikosongkan]*

# **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**

Salah satu masalah terbesar dalam proses produksi perangkat lunak adalah besarnya kesenjangan antara pihak manajemen industri dan pihak pengembang perangkat lunak dalam mendesain proses bisnis. Kesenjangan ini disebabkan bedanya standar yang digunakan oleh pihak manajemen industri dan pihak pengembang perangkat lunak. Pihak manajemen industri atau perusahaan memiliki standar dalam membuat proses bisnis yaitu *Business Process Model and Notation* (BPMN) [1]. Sementara, pihak pengembang menggunakan standar *Business Process Execution Language* (BPEL) [2] untuk mendefinisikan berbagai proses bisnis. Perbedaan ini membuat para pengembang menafsirkan proses bisnis yang berstandar BPMN secara intuitif ke BPEL. Tetapi, berkembangnya proses pemetaan antara standar BPMN dan BPEL memungkinkan konversi langsung antara BPMN dan BPEL. Hal ini memperkecil kesenjangan antara pihak manajemen dan pihak pengembang. Sehingga proses bisnis dari pihak manajemen dengan standar BPMN dapat langsung dimengerti oleh pihak pengembang karena bisa diubah ke standar BPEL.

Terobosan terbaru ini membuat kesenjangan antara pihak manajemen dan pengembang *workflow* semakin sempit. Tetapi terdapat permasalahan baru yang timbul. Perusahaan, dalam hal ini pihak manajemen, sering mengubah ubah proses bisnis. Sehingga, aplikasi yang diciptakan para pengembang dituntut untuk lebih mudah beradaptasi terhadap perubahan. Aplikasi yang bisa dengan mudah beradaptasi dengan perubahan proses bisnis

bisa terwujud dengan menggunakan *Workflow Management System* (WFMS). Dengan WFMS, aplikasi yang dibangun bisa menggunakan orkestrasi BPEL dan mudah diatur kembali mengikuti perkembangan proses bisnis yang dibuat oleh pihak manajemen.

Tetapi, terdapat dua buah masalah dalam pengembangan aplikasi berbasis WFMS. Permasalahan pertama adalah masalah *reusability workflow*. Masalah ini timbul akibat sulitnya melakukan pencarian terhadap *workflow* yang sudah pernah diciptakan. Sehingga *workflow* yang sudah pernah diciptakan sulit untuk ditemukan dan digunakan kembali. Hal ini terjadi akibat tidak adanya bagian *profile* atau *description* pada standar orkestrasi *workflow* seperti *Web Service Description Language* (WSDL) atau BPEL. Standar-standar ini hanya menyediakan bagian implementasi yang memberikan penjelasan tentang detail teknis dari suatu *workflow* atau *service*, tetapi tidak bisa menjelaskan deskripsi dari *service* tersebut.

Permasalahan kedua adalah sulitnya menganalisis aktivitas dari *workflow* atau *service*. Salah satu teknik analisis *workflow* yang sedang berkembang adalah teknik *process mining*. Teknik ini bisa memberikan analisis beban dan kesalahan dari aktivitas *workflow*. Sayangnya, teknik ini membutuhkan *log* dari *workflow* yang sesuai dengan standar *process mining*. Beberapa WFMS terkemuka seperti Microsoft Biztalk, Oracle SOA Suite, dan IBM Websphere belum memiliki standar *log* berbasis *process mining*. Oleh karena itu, arsitektur WFMS yang dibangun ingin memberikan kemampuan terhadap WFMS untuk menghasilkan *log* berstandar *process mining*.

Dengan kata lain, tugas akhir ini mencoba membuat *Workflow Management System* berbasiskan *workflow engine* yang bisa menjalankan proses bisnis dengan standar BPEL, menghasilkan informasi tentang keadaan *workflow*, menghasilkan *file event log* yang sesuai dengan standar *process mining*, dan melakukan pencarian *workflow*.

## 1.2. Tujuan

Tujuan dari Tugas Akhir ini adalah sebagai berikut:

1. Membuat *Workflow Management System* yang mampu menjalankan *file* BPEL.
2. Membuat *Workflow Management System* yang mampu memberikan informasi keadaan berbagai *workflow* yang sedang berjalan.
3. Menghasilkan *file event log* yang berisi berbagai aktivitas yang dilakukan oleh *workflow*.
4. Meningkatkan *reusability workflow* dengan fitur pencarian *workflow*.

## 1.3. Rumusan Permasalahan

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

1. Bagaimana menjalankan komposisi *workflow* yang berada dalam standar BPEL.
2. Bagaimana menyediakan informasi terkait keadaan *workflow* di *workflow management system*.
3. Bagaimana mengubah bentuk *log format* dari *event* yang terjadi sesuai standar process mining.
4. Bagaimana konsep yang akan diterapkan pada fitur pencarian *workflow*.

## 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. *File* BPEL yang diterima oleh *Workflow Management System* sudah valid.
3. *Event log* yang dihasilkan berekstensi csv.

4. BPEL yang diunggah wajib dikompresi menggunakan standar zip.
5. *File* BPEL yang diterima wajib dilengkapi dengan berbagai *file* sesuai standar ODE *server*, seperti *Web Service Description Language* (WSDL) dan *web deployment descriptor*.

## 1.5. Metodologi

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

### 1. Studi literatur

Pada tahap ini, berbagai literatur yang dibutuhkan untuk membangun suatu *Workflow Management System* yang berbasis BPEL. Berbagai informasi tersebut juga meliputi hal-hal teknis dari *platform* yang akan digunakan. Rincian berbagai literatur yang akan dipelajari adalah sebagai berikut:

- a. *Business Process Execution Language* (BPEL)
- b. *Workflow management system*
- c. Standar *log* pada *event process mining*
- d. *Web Ontology Language for Services* (OWL-S)

### 2. Analisis dan Perancangan Sistem

Proses analisis dari perangkat lunak dalam tugas akhir ini adalah dengan pemecahan masalah yang telah dirumuskan dalam bab rumusan masalah. Contohnya seperti bagaimana menjalankan *workflow* dan membentuk informasi di *workflow management system*. Tahap analisis ini harus dilakukan terlebih dahulu terhadap konsep-konsep tersebut.

Proses desain dari perangkat lunak ini akan dibuat berdasarkan hasil olahan dari tahap analisis. Hasil dari tahap analisis yang berupa konsep-konsep akan

diterjemahkan ke dalam desain implementasi yang lebih teknis. Seluruh desain teknis ini meliputi tahap pencarian *workflow* sampai implementasi *workflow*.

### 3. Implementasi

Pengembangan aplikasi tugas akhir ini berupa aplikasi *web*. Masukan dari aplikasi ini berupa *file* BPEL yang sudah valid. Dan *output* yang dihasilkan, berupa *workflow* yang dijalankan dan *event log*.

Beberapa hal yang diperlukan dalam implementasi ini adalah :

- a. IDE Eclipse Kepler.
- b. Bahasa Pemrograman Java.
- c. Java Development Kit 7 (JDK).
- d. Java Runtime Environment (JRE).
- e. Eclipse BPEL Editor
- f. Apache ODE *server*
- g. Spring Framework

### 4. Pengujian dan evaluasi

Pengujian aplikasi pada Tugas Akhir ini akan menggunakan *black box testing*, dimana masukan berupa *file* komposisi *workflow* BPEL, dan akan dilihat apakah hasil yang dikeluarkan seperti *event log* dan *web service* dari *workflow* sesuai dengan BPEL masukan. Selain itu juga program akan diuji apakah bisa berjalan dengan seharusnya.

Tahapan-tahapan dari pengujian adalah sebagai berikut:

- a. Pengujian hasil *event log* yang dihasilkan.
- b. Pengujian *metadata* dan fitur pencarian.

### 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 aplikasi *web*.

### **Bab V   Implementasi**

Bab ini berisi implementasi dari perancangan perangkat lunak pada aplikasi *web* dan implementasi fitur-fitur penunjang aplikasi *web*.

### **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 aplikasi *web*.

### **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 algoritma *Business Process Execution Language* (BPEL), *Workflow Management System* (WFMS), standar *log* pada *process mining*, *Web Ontology Language for Services* (OWL-S).

### **2.1. *Business Process Execution Language***

BPEL [2] merupakan sebuah bahasa orkestrasi yang dicetuskan oleh OASIS. OASIS merupakan sebuah organisasi yang dibentuk oleh beberapa *vendor* terkemuka di bidang aplikasi *web service orchestration*, seperti Microsoft dan IBM. Bahasa orkestrasi ini merupakan bahasa yang dibentuk untuk membuat suatu orkestrasi dari *web service*.

BPEL merupakan suatu bahasa yang menggunakan standar *Extensible Markup Language* (XML), sehingga berbagai atribut di dalam BPEL akan direpresentasikan dalam *tag-tag* XML. Bahasa ini lahir karena kebutuhan baru terhadap *Service Oriented Architecture* (SOA) yaitu kemampuan berbagai *web service* dalam orkestrasi. Selama ini, *web service* sangat tergantung terhadap *vendor* pembuatnya, sehingga mengurangi faktor *interoperability* pada SOA. Sebelum BPEL, sudah ada beberapa standar yang ada.

BPEL merupakan bahasa orkestrasi yang memiliki struktur yang lengkap, beberapa di antaranya adalah:

#### 1. *Web Service Description Language* (WSDL) *Definition*

Bagian ini akan menjelaskan tentang seluruh bagian dari WSDL. Bagian-bagian ini akan menjelaskan berbagai hal yang berhubungan dengan tipe data, deklarasi *partner links*, dan *input output* dari berbagai operasi. Terdapat beberapa bagian penting di WSDL *definition* di antaranya:

- *Message*

Bagian ini menjelaskan berbagai *input* atau *output* yang akan dipakai dalam BPEL. Bagian ini berguna untuk mengetahui parameter apa saja yang harus dikirim dan diterima oleh sisi klien atau *server*. Contoh *message* dapat dilihat di Gambar 2.1.

```
<wsdl:message name="POMessage">
  <wsdl:part name="customerInfo" type="sns:customerInfoType" />
  <wsdl:part name="purchaseOrder" type="sns:purchaseOrderType" />
</wsdl:message>
```

**Gambar 2.1** Contoh *message* pada BPEL

- *Port type* dan *partner link type*

Di dalam sebuah BPEL, terkadang terdapat lebih dari satu operasi atau *web service* yang digunakan. Untuk bisa membedakan antara satu operasi dan operasi lainnya, penggunaan *port type* diperlukan. Sebuah *port type* akan berkoordinasi dengan satu operasi. Di dalam *port type* akan dideklarasikan operasi yang dipakai dan *message* yang dipakai. Contoh *port type* dapat dilihat di Gambar 2.2.

Di dalam WSDL, berbagai operasi memang dideklarasikan. Tetapi, tidak semua operasi yang digunakan dalam BPEL berasal dari satu tempat yang sama. Terkadang untuk satu aktivitas, dua *web service* dari tempat yang berbeda digunakan. Kedua *web service* ini akan disatukan menjadi sebuah *partner link type*. *Partner link type* terdiri dari *role* dan *port type*. Contoh *partner link type* dapat dilihat di Gambar 2.3.

```

<wsdl:portType name="shippingPT">
  <wsdl:operation name="requestShipping">
    <wsdl:input message="pos:shippingRequestMessage" />
    <wsdl:output message="pos:shippingInfoMessage" />
    <wsdl:fault name="cannotCompleteOrder"
      message="pos:orderFaultType" />
  </wsdl:operation>
</wsdl:portType>

```

**Gambar 2.2** Contoh *port type* pada BPEL

```

<plnk:partnerLinkType name="invoicingLT">
  <plnk:role name="invoiceService"
    portType="pos:computePricePT" />
  <plnk:role name="invoiceRequester"
    portType="pos:invoiceCallbackPT" />
</plnk:partnerLinkType>

```

**Gambar 2.3** Contoh *partner link type* di BPEL

## 2. *Process Definition*

Bagian ini menjelaskan alur dari aktivitas yang dilakukan di dalam BPEL. Bagian ini juga menjelaskan tentang berbagai variabel yang dipakai di BPEL. Bagian ini terdiri dari beberapa bagian utama yaitu:

- *Partner Link*

Bagian ini menjelaskan berbagai *partner link* yang dipakai oleh alur proses BPEL. Sebuah *partner link* berisi berbagai *partner link type* dan sebuah *role*. Saat sebuah aktivitas memanggil *partner link* ini, seluruh operasi yang berada di *partner link type* miliknya akan dipanggil secara otomatis. Contoh *partner link* bisa dilihat di Gambar 2.4.

- *Variables*

Bagian ini akan berisi deklarasi variabel yang dipakai di dalam alur aktivitas dari BPEL. Tipe dari berbagai variabel ini harus berasal dari tipe-tipe yang sudah dikenali pada bagian WSDL *definition*.

- *Sequence*

Bagian ini merupakan bagian inti dari BPEL. Bagian ini mendeskripsikan alur dari rangkaian aktivitas atau *workflow* yang ada di BPEL. Berbagai aktivitas yang dapat dipakai di dalam struktur BPEL adalah:

- <receive>
- <reply>
- <invoke>
- <assign>
- <throw>
- <exit>
- <wait>
- <empty>
- <sequence>
- <if>
- <while>
- <repeatUntil>
- <forEach>
- <pick>
- <flow>
- <scope>
- <compensate>
- <compensateScope>
- <rethrow>
- <validate>
- <extensionActivity>

```
<partnerLinks>
  <partnerLink name="purchasing"
    partnerLinkType="lns:purchasingLT" myRole="purchaseService" />
  <partnerLink name="invoicing" partnerLinkType="lns:invoicingLT"
    myRole="invoiceRequester" partnerRole="invoiceService" />
</partnerLinks>
```

**Gambar 2.4** Contoh *partner link* di BPEL

## 2.2. Standar *Log* pada *Process Mining*

*Process mining* merupakan sebuah teknik penggalian informasi yang menggunakan *log* sebagai salah satu bahan utama di dalam metodenya [3]. *Process mining* mempunyai tiga bagian utama, yaitu *discovery*, *conformance*, dan *enhancement*. *Log* merupakan bahan utama dari proses *discovery*. Pada proses *discovery* ini, *log* akan diproses menjadi sebuah model. Selanjutnya, berbagai informasi penting, seperti kinerja, *bottleneck*, dan analisis beban dari perangkat lunak bisa diketahui dari analisis terhadap model tersebut. Untuk itulah, tugas akhir ini memiliki fitur untuk menghasilkan *log* yang dapat digunakan untuk metode *process mining*.

*Event log* yang digunakan dalam *process mining* sangat jauh berbeda dengan standar *log* yang dikenal selama ini. Di dalam konsep *process mining*, *event log* memiliki 3 atribut wajib, yaitu *case id*, *activity*, dan *timestamp* [4]. *Case id* merupakan atribut yang menunjukkan *id* dari suatu proses. *Case id* dibutuhkan untuk membedakan berbagai proses yang menggunakan aktivitas yang sama. *Activity* merupakan atribut untuk menunjukkan nama dari aktivitas yang dijalankan. *Timestamp* adalah atribut yang menunjukkan waktu terjadinya sebuah aktivitas. *Timestamp* bisa terbagi menjadi dua, yaitu waktu mulainya sebuah aktivitas, dan berakhirnya sebuah aktivitas. Atribut *event log* selain yang disebutkan diatas, bersifat opsional. Contoh *event log* yang benar bisa dilihat di Tabel 2.1.

Permasalahan terbesar di dalam *event log* adalah *log* pada sistem informasi atau *workflow application* jarang yang berstandar pada format ini. Sehingga menyulitkan proses analisis secara otomatis dari sistem *workflow*. Maka, salah satu permasalahan yang ingin diselesaikan dalam tugas akhir ini yaitu menciptakan sistem yang bisa menghasilkan *event log* yang menggunakan standar ini.



Case id	Event id	Properties	
		Timestamp	Activity
1	35654423	30-12-2010:11.02	Register request
	35654424	31-12-2010:10.06	Examine thoroughly
	35654425	05-01-2011:15.12	Check ticket
	35654426	06-01-2011:11.18	Decide
	35654427	07-01-2011:14.24	Reject request

**Tabel 2.1** Contoh *event log* sesuai standar *process mining*

### 2.3. *Workflow Management System*

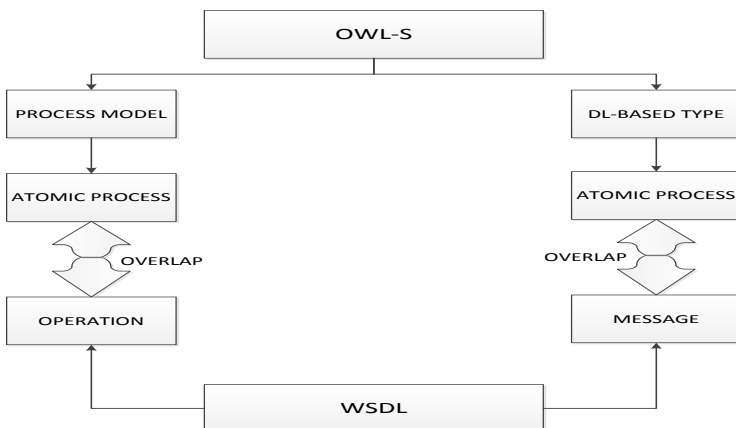
*Workflow Management System* memiliki definisi yang sangat luas dan beragam. Berdasarkan Van der Aalst dan Van Hee, WFMS adalah ide, metode, teknik, dan perangkat lunak yang digunakan untuk mendukung proses bisnis yang terstruktur [5]. Sedangkan berdasarkan Workflow Management Coalition (WFMC), WFMS adalah sebuah sistem yang berkemampuan untuk mendefinisikan, membuat, dan manajemen eksekusi dari sebuah alur kerja atau *workflow* yang berasal dari penggunaan sebuah perangkat lunak atau *source code* dan berjalan baik pada *workflow engine* yang sama maupun berbeda [6]. Berbagai definisi ini mensyaratkan bahwa sebuah WFMS harus menjalankan beberapa fungsi utama, yaitu menjalankan sebuah proses bisnis atau *workflow*, manajemen berbagai proses bisnis atau *workflow* yang ada, dan membuat serta mendefinisikan proses bisnis.

Pada tugas akhir ini, WFMS yang dibangun, memiliki dua fungsi, yaitu menjalankan *workflow* atau proses bisnis yang sudah didefinisikan di dalam standar BPEL dan manajemen berbagai *workflow* yang ada melalui metadata. Kedua fungsi ini dipilih karena merupakan area permasalahan yang ingin dipecahkan yaitu konsep *reusability* pada *workflow*, konsep *event log* pada *process mining*, dan konsep eksekusi *workflow* berbasis BPEL.

## 2.4. *Web Ontology Language for Services*

*Web Ontology Language for Services* (OWL-S) [7] adalah sebuah standar untuk mendeskripsikan sebuah *web service* yang mengutamakan kemampuan semantik dari deskripsi sebuah *web service*. Sebelum OWL-S terbentuk, terdapat sebuah standar deskripsi untuk *web service* yaitu *Web Service Description Language* (WSDL) [8]. WSDL mendeskripsikan sebuah *web service* dari sisi informasi teknis seperti berbagai operasi, masukan, keluaran, tipe data, dan *port*. Kekurangan dari WSDL adalah tidak adanya informasi yang bisa digali untuk bisa menciptakan kemampuan semantik pada *web service*. Oleh karena itu, OWL-S muncul untuk memberikan kemampuan semantik pada *web service*.

OWL-S terbagi pada tiga bagian besar, yaitu *service profile*, *service grounding*, dan *service model*. *Service profile* adalah bagian yang berisi tentang deskripsi dari *web service* seperti nama, deskripsi singkat, pengembang dari sebuah *web service*. *Service grounding* adalah bagian yang berisi berbagai informasi tentang bagaimana cara berhubungan dengan *web service* tersebut. *Service model* menggambarkan model dari operasi dari suatu *web service*.

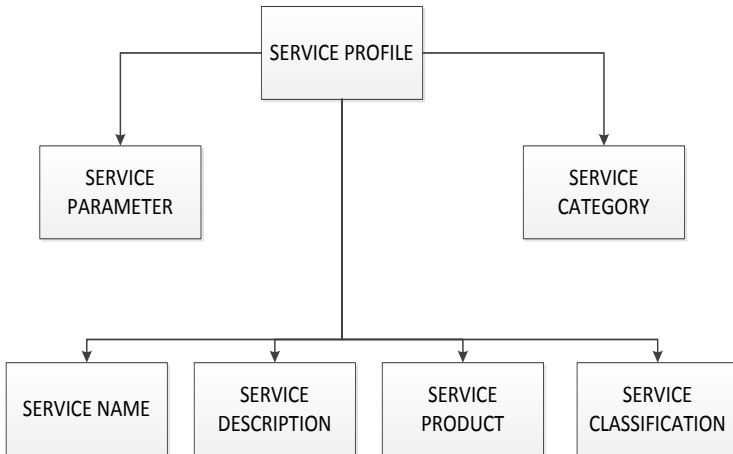


**Gambar 2.5** Bagian-bagian dari *service profile* pada OWL-S

*Service profile* merupakan bagian dari OWL-S yang *queryable*. *Service profile* mempunyai beberapa bagian seperti yang ditunjukkan oleh Gambar 2.5. Pada Gambar 2.5, ada beberapa bagian penting yang harus dijelaskan. Pertama, bagian deskripsi singkat seperti *service classification*, *service product*, *service name*, *text description* merupakan bagian yang diisi secara manual oleh pengembang *web service* ketika membuat *file* OWL-S. Bagian yang berada di sebelah kiri, berisi berbagai properti dari proses, seperti *parameter*, *condition*, *result*, *input*, dan *output*. Berbagai bagian ini dibutuhkan untuk mengetahui model dari proses secara singkat, sedangkan penjelasan terhadap model lebih lanjut akan diberikan oleh bagian *service model*.

*Service grounding* berisi tentang berbagai informasi tentang model secara teknis. Bagian ini diciptakan agar OWL-S bisa melakukan *automatic composition* secara semantik. Berbagai informasi seperti berbagai tipe *input* dan *output* baik *simple type* maupun *complex type*, *precondition* yaitu berbagai kondisi yang harus dipenuhi untuk bisa mengakses sebuah *web service*, dan *process*. *Process* pada *service ground* menjelaskan tentang berbagai informasi seperti, *atomic process*, *composite process*, dan *parameter*. *Atomic process* merupakan proses yang tidak memiliki aktivitas yang berasal dari *web service* yang lain. Sedangkan *composite service* memiliki aktivitas yang berasal dari *web service* lainnya.

*Service grounding* dan *service model* bisa dibuat menggunakan WSDL yang sudah ada dari *web service*. Hal ini dimungkinkan karena ada informasi yang sama atau *overlap* di antara OWL-S dan WSDL seperti yang ditunjukkan pada Gambar 2.6. Informasi tentang *atomic process* pada *service model* dan *service grounding* dapat diambil dari informasi *operation* pada WSDL. Hal ini disebabkan setiap operasi dalam WSDL bersifat *atomic*. Sementara bagian *input* dan *output* pada OWL-S bisa dihasilkan menggunakan bagian *message* pada WSDL. Pada tugas akhir ini, proses konversi berbagai informasi dari WSDL ke OWL-S akan menggunakan bantuan dari OWL-S API [9].



**Gambar 2.6 Hubungan atau overlap antara WSDL dan OWL-S**

## 2.5. SPARQL Ontology Query Language

SPARQL merupakan bahasa *query* dari standar ontologi [10]. Bahasa ini merupakan salah satu bahasa yang bisa melakukan *query* terhadap *file* OWL-S atau ontologi. Contoh *query* bisa dilihat di Gambar 2.7. Setiap *query* diawali oleh *prefix* yang berisi berbagai konstanta yang digunakan dalam *query*. Setiap konstanta akan merepresentasikan sebuah *file* ontologi yang berisi deklarasi kelas-kelas yang ada di *file* ontologi yang akan diakses menggunakan *query*. Setiap variabel akan diawali oleh karakter tanda tanya (?). Setiap *statement* di dalam *query* akan terdiri dari tiga elemen, yaitu dua variabel yang merepresentasikan nilai dari *triplet* dan satu *statement* yang menyatakan hubungan atau relasi yang digunakan di dalam *statement*.

```

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox
WHERE
{ ?x foaf:name ?name .
  ?x foaf:mbox ?mbox }
  
```

**Gambar 2.7 Contoh query SPARQL**

## 2.6. *Semantic Similarity*

Penelitian kali ini menggunakan teori *semantic similarity* untuk mencari nilai kedekatan antara deskripsi sebuah *service* dan *keyword* yang diberikan. *Semantic similarity* merupakan metode yang sangat luas dan bervariasi tergantung dari definisi yang diterapkan oleh metode tersebut. Di dalam penelitian kali ini, definisi *semantic similarity* yang digunakan mengutamakan jumlah kata yang sama daripada kata yang sinonim [11]. Persamaan yang digunakan dapat dilihat di Persamaan 2.1.

Rumus yang terdapat pada Persamaan 2.1 merupakan rumus yang dipakai untuk menentukan nilai similaritas antara dua buah kalimat. Fungsi  $\text{sem}(n_1, n_2)$  merupakan fungsi yang menentukan nilai similaritas antara dua buah kalimat. Variabel  $n_1$  dan  $n_2$  menunjukkan kalimat 1 dan kalimat 2. Variable  $w_1$  menunjukkan kumpulan kata yang terdapat pada  $n_1$ , begitu juga  $w_2$ . Fungsi  $\text{synonym}(s, t)$  merupakan fungsi yang memeriksa hubungan antara dua kata, apakah berhubungan erat seperti sinonim atau tidak. Dengan kata lain, rumus pada Persamaan 2.1 menghitung nilai similaritas secara semantik dengan membandingkan kata-kata dari kedua kalimat. Kata-kata yang sama atau terdapat di dua kalimat yang berbeda, akan diberi nilai 1 dan kata-kata yang memiliki hubungan erat seperti sinonim akan diberi nilai 0,75. Seluruh nilai yang didapat akan ditambah dan dibagi dengan jumlah kata terbesar di antara  $n_1$  dan  $n_2$ .

Contoh dari penggunaan rumus dari Persamaan 2.1 dapat dilihat pada Tabel 2.2. Pada *service name*, terdapat dua buah kalimat yaitu *service name* dari *keyword* dan dari *file OWL-S*. Di antara kedua kalimat tersebut, terdapat satu pasang kata yang sinonim yaitu *pick* dan *collect* dan dua pasang kata lain yang sama. Sehingga, nilai kesamaan antara kalimat *service name* pada *keyword* dan kalimat *service name* dari deskripsi *service OWL-S* bernilai 0,917 yang berasal dari 1 pasang kata sinonim yang bernilai 0,75 dan 2 pasang kata yang sama secara leksikal yang bernilai 2 dan kedua angka tersebut dibagi terhadap jumlah kata yaitu 3.

Tabel 2.2 juga menampilkan contoh penghitungan dua kalimat yang lebih panjang yaitu *description* dari *keyword* dan deskripsi *service* OWL-S. Kedua kalimat ini memiliki nilai similaritas 0,937. Nilai ini berasal dari 2 pasang kata semantik yang bernilai 1,5 serta 6 pasang kata yang sama secara leksikal yang bernilai 6.

	Service Name	Description
Keyword	Pick Material Service	This service pick material and automate storage record
OWL-S	Collect Material Service	This service pick material and automate storage record
Score	0,917	0,937

**Tabel 2.2 Contoh penggunaan rumus similaritas**

$$sem(n1, n2) = \frac{1.0, |w1 \cap w2| + 0,75 \cdot \sum_{s \in w1 \setminus w2} synonym(s, t)}{\max(|w1|, |w2|)} \quad (2.1)$$

*[Halaman ini sengaja dikosongkan]*

## **BAB III**

### **METODOLOGI PEMECAHAN MASALAH**

Bab ini menjelaskan informasi tentang berbagai metode yang dilakukan dari menjalankan komposisi *workflow* dalam BPEL sampai menghasilkan *metadata* OWL-S untuk *workflow*, untuk menyelesaikan masalah yang dijelaskan pada bagian rumusan masalah.

#### **3.1. BPEL Server Engine**

Permasalahan pertama dalam tugas akhir ini adalah pemilihan BPEL *engine* sebagai dasar dari WFMS. BPEL *engine* adalah sebuah aplikasi yang menjalankan dan mengatur komposisi *workflow* di BPEL. BPEL merupakan sebuah bahasa komposisi, sehingga implementasi dari sebuah BPEL juga berbeda dari implementasi *server* pada kebanyakan. Ada banyak tipe dan *vendor* yang mengeluarkan perangkat lunak *server engine* dari BPEL, beberapa di antaranya adalah IBM Websphere, Oracle BPEL Process Manager, Microsoft Biztalk, dan Apache Orchestration Director Engine (ODE).

Karena banyaknya *engine* yang ada, kriteria dari *server engine* yang akan digunakan harus dibentuk terlebih dahulu. Kriteria pertama dari *server engine* adalah kemampuan yang tinggi pada modifikasi dan modularitas. Dengan kata lain, *server engine* harus yang dipilih harus bisa dimodifikasi sedemikian rupa dan memiliki *interface* untuk bisa mengakses modul-modulnya sehingga bisa memenuhi kebutuhan dari tugas akhir ini. Berdasarkan kriteria pertama ini, tipe *server engine* berbasis komersil atau tidak *open source* tentu mempunyai kemungkinan kecil untuk dipilih. Kriteria kedua adalah tersedianya aplikasi *editor* berbasis BPEL dan penggunaan standar BPEL di dalam *server engine*. Kriteria kedua ini berfungsi untuk melengkapi alur WFMS yang berada pada tugas akhir yang hanya dimulai dari tahap pengunggahan *workflow*. Kriteria ini juga menyingkirkan berbagai aplikasi yang menerjemahkan BPEL ke bahasa



komposisi mereka masing-masing. Dengan mengubah BPEL ke bahasa komposisi lain, aplikasi *editor* tentunya juga tidak berbasis ke BPEL, sehingga menyulitkan pengguna dalam merangkai komposisi *workflow*.

Dari kedua kriteria diatas, Apache ODE terpilih menjadi *server engine* yang digunakan dalam tugas akhir ini. Dari kedua kriteria, *server engine* ini dinilai bisa memenuhi kedua kriteria tersebut. Dari kriteria pertama, Apache ODE memiliki kemampuan modifikasi yang tinggi karena bersifat *open source*. Selain bersifat *open source*, *server* ini juga memiliki modularitas yang tinggi karena kemampuan untuk mengakses modul manajemen *workflow* dan manajemen *instance* melalui *web service*.

### 3.2. Arsitektur WFMS

Pada penelitian ini, arsitektur WFMS yang digunakan dapat dilihat pada Gambar 3.1. Aplikasi yang dibangun berupa *web application* yang menggunakan bahasa Java dan *framework* yang digunakan adalah Spring. Terdapat beberapa hal penting dari arsitektur yang diajukan pada penelitian ini, yaitu arsitektur WFMS mempunyai dua buah *repository* yaitu BPEL *repository* dan OWL-S *repository*. Di dalam arsitektur sistem ini, *workflow engine* yang dipakai adalah Apache ODE. Penyimpanan *event log* yang dihasilkan menggunakan basis data relasional.

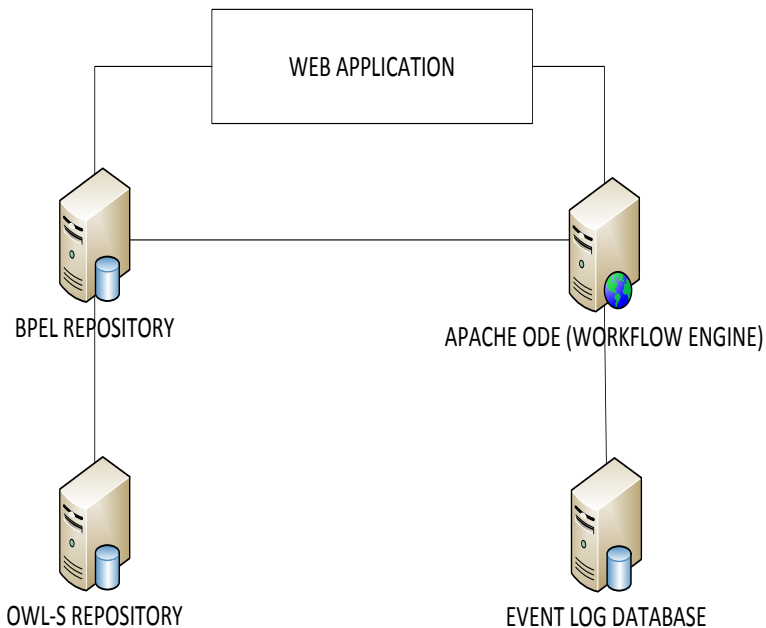
Berbeda dari konsep arsitektur WFMS lainnya seperti SHIWA, arsitektur ini menjadikan BPEL sebagai standar WFMS. WFMS seperti SHIWA menggunakan teknologi seperti *tavern* untuk *workflow*. BPEL dipilih karena independen terhadap *platform*. Dengan kata lain, berbagai *workflow* atau *web service* yang dibuat menggunakan bahasa pemrograman yang berbeda-beda, bisa dikomposisi menggunakan standar bahasa orkestrasi BPEL. Pemilihan BPEL ini juga disebabkan tingkat logika dan aktivitas yang tersedia cukup lengkap untuk sebuah bahasa orkestrasi.

Kedua tempat penyimpanan atau *repository* berfungsi untuk menampung *workflow* yang diunggah oleh pengguna. Arsitektur ini memisahkan BPEL *repository* dan OWL-S *repository* karena sifatnya yang berbeda. BPEL *repository* bersifat tertutup dan hanya bisa diakses oleh aplikasi web. Sifat tertutup ini dikarenakan BPEL berisi implementasi dari orkestrasi dan bersifat sensitif. Beberapa *tenant* mungkin ingin merahasiakan implementasi dan variasi *workflow* yang dipakai di perusahaan. Tetapi, OWL-S berfungsi sebagai *advertisement* yang berisi profil dari *service* sehingga bersifat tidak sensitif. Berbeda dengan WFMS lain, arsitektur ini memperbolehkan OWL-S *repository* diakses untuk umum. BPEL *repository* terisi ketika pengguna mengunggah *workflow* ke dalam BPEL. Sementara itu OWL-S *repository* terisi ketika BPEL dikonversi ke OWL-S menggunakan OWL-S API. BPEL yang telah diunggah oleh pengguna akan diubah menjadi *web service* dan memiliki WSDL. WSDL dapat dikonversi ke OWL-S karena memiliki beberapa kesamaan bagian.

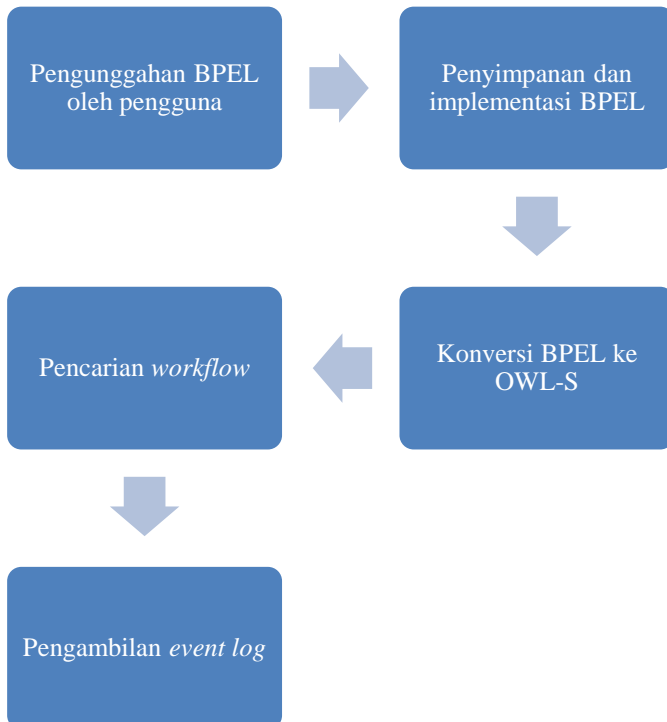
Apache ODE dipilih menjadi *workflow engine* pada arsitektur ini karena memiliki beberapa kelebihan. Pertama, sifat Apache ODE yang *open source* membuat *workflow engine* ini mudah untuk dipakai ke dalam aplikasi. Kedua, *log* aktivitas yang dibentuk oleh Apache ODE bisa dibentuk menjadi standar *process mining*.

Dengan kata lain, arsitektur ini menawarkan alur dari pengunggahan *workflow* sampai implementasi *workflow* menjadi *web service* seperti yang ada di Gambar 3.2. Tahap pertama adalah pengunggahan BPEL oleh pengguna. Pada tahap ini, pengguna bisa mengunggah *file* BPEL yang telah dikompresi dengan standar zip. Tahap kedua adalah penyimpanan dan implementasi BPEL. *File* BPEL yang telah diunggah akan disimpan ke dalam *repository* dan diimplementasi menjadi *web service* oleh *workflow engine*. Tahap ketiga adalah konversi BPEL menjadi OWL-S. Pada tahap ini, *web service* yang telah diimplementasi akan diubah OWL-S dan disimpan di OWL-S

*repository*. Tahap keempat adalah pencarian *workflow*. Pengguna yang ingin mencari *workflow* yang dibutuhkan untuk komposisi bisa mencari dengan memberikan *keyword* dari *service name* dan *description*. Tahap akhir adalah pengambilan *event log*. Setiap aktivitas yang dilakukan oleh *workflow* akan dicatat oleh sistem. Hal ini dimungkinkan karena arsitektur ini dilengkapi dengan metode *event log generator* yang dijelaskan pada sub Bab 3.3.



**Gambar 3.1 Arsitektur WFMS yang dipakai**



**Gambar 3.2** Alur dari arsitektur WFMS

### 3.3. *Event Log Generator*

Permasalahan kedua dalam tugas akhir ini adalah metode untuk menghasilkan *event log* atau *event log generator*. Terdapat berbagai metode yang telah dikembangkan dan diwacanakan untuk bisa menghasilkan *event log* dari komposisi *web service*. Salah satunya adalah konsep *web service mining* yang ingin mengubah konsep dari *Service Oriented Architecture Protocol (SOAP) header* dengan menambah informasi tentang *event log attribute* [12]. Konsep ini menjelaskan bahwa keterbatasan informasi dari *header* dari protokol SOAP telah menyebabkan sulitnya pembentukan *event log* dari *web service*. Untuk itulah, konsep ini ingin menambahkan informasi dengan *tag*

<WSHeaders> yang bisa berisi informasi seperti *choreography protocol, choreography name, choreography case*.

Selain konsep *web service mining* diatas, terdapat konsep yang lain dengan memanfaatkan berbagai *logging facilities* [13]. Konsep ini membedakan berbagai tipe *web service mining* yang bisa dilakukan berdasarkan *logging facilities* yang tersedia. Terdapat lima tingkatan *web service mining* yang tersedia berdasarkan konsep ini seperti yang ditunjukkan Tabel 3.1.

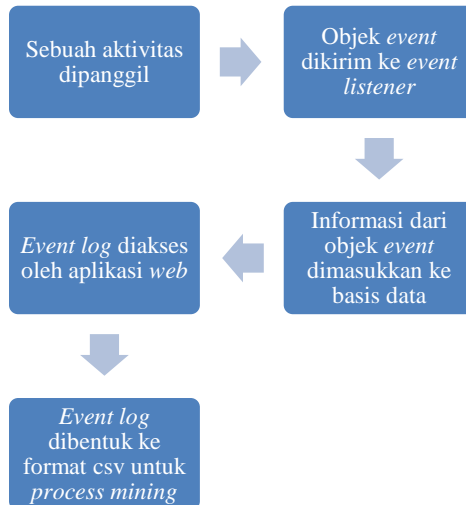
**Tabel 3.1** Tabel berbagai tingkat *web service mining*

Level	Logged information	Logging facilities	Mining opportunities
1	- consumer IP - invoked WS - timestamp - HTTP status code	- Web server	- service utilization - consumer tracking - failure rates
2	- level 1 + - SOAP request & response - timestamps	- HTTP listener & logger	- level 1 + - WS execution time - analysis of SOAP messages
3	- invoked WS & operation - SOAP request & response - timestamps	- WS container - SOAP handlers	- level 2
4	- level 3 + - consumer-side activity	- WS container - SOAP handlers	- level 3 + - client-side activity
5	- level 4 + - workflow information	- level 4 + - Web services	- level 4 + - Web services workflows

Pada tingkat pertama, *logging facilities* hanya tersedia pada *web server* tempat *service* berada. Informasi yang bisa didapat hanya alamat IP dari klien, *web service* yang dijalankan, *timestamp*, dan *HTTP status code*. Dengan informasi yang sedikit ini, tentu *event log* belum bisa dibentuk. Berbagai tingkat yang ada di konsep ini, belum menjelaskan bagaimana informasi *event log* bisa terbentuk. Konsep ini hanya menjelaskan bagaimana informasi-informasi yang terdapat pada *web service* dan protokol SOAP bisa berguna untuk *web service mining*.

Oleh karena itu, tugas akhir ini menggunakan sistem *event log generator* pada *server* Apache ODE. Sistem ini merupakan sebuah *logging facilities* yang diberikan oleh Apache ODE. Konsep ini sangat berbeda dari kedua konsep yang telah

disebutkan diatas. Sistem ini tidak mengubah struktur informasi dari SOAP *header*. Sistem ini juga tidak memanfaatkan HTTP *logging*. Sistem ini memanfaatkan implementasi dari kelas *event listener* yang disediakan dari Apache ODE. Apache ODE memberikan sebuah kelas abstrak yang dapat diimplementasikan oleh pengguna. Metode dari kelas ini akan dipanggil setiap suatu aktivitas akan dipanggil. Implementasi yang dipakai pada tugas akhir ini bekerja dengan alur yang digambarkan pada Gambar 3.3.



**Gambar 3.3 Alur metode pembuatan *event log* pada WFMS**

Pada alur ini, terdapat beberapa bagian yang penting. Bagian pengiriman objek *event* terjadi pada sistem *event listener*. Setelah objek diterima, berbagai informasi dari *event* tersebut dapat diakses. Seluruh informasi akan dikirim ke basis data, tetapi hanya beberapa informasi penting seperti *case id*, *activity name*, *status*, dan *timestamp*. Setelah semua *event log* berada di basis data, *event log* bisa dilihat melalui web. *Event log* juga bisa diunduh ke dalam format csv. Csv dipilih karena merupakan tipe data yang bisa diproses ke dalam *process mining*. Selain itu, format csv juga dipilih karena mudah dan terdapat aplikasi yang

bisa mengubah format csv ke format data di *process mining*, seperti format mxml.

### 3.4. Metode Pencarian Menggunakan OWL-S

Metode dari pencarian berbasis OWL-S adalah sistem *query* ontologi. Metode ini berfungsi mengambil seluruh informasi yang dibutuhkan dari *file* berstandar OWL-S. Metode *query* ontologi yang dipakai adalah SPARQL. Dengan menggunakan *query* SPARQL, berbagai elemen pada *file* dengan standar OWL-S bisa diakses dan diseleksi untuk tahap selanjutnya.

Seperti yang dijelaskan pada bab sebelumnya, OWL-S terdiri dari tiga bagian, yaitu *profile*, *grounding*, dan *process*. Di dalam *query* yang dipakai di tugas akhir ini, bagian *profile* menjadi bagian satu satunya yang akan diambil. Hal ini dikarenakan bagian *profile* mengandung bagian semantik dari *web service*, yaitu *service name* dan *description*. Untuk mengambil bagian *profile*, maka digunakan *query* seperti pada Gambar 3.3. Dengan menggunakan *query* ini, *service name* dan *description* dari seluruh *web service* direpresentasikan dalam *file* OWL-S akan didapatkan.

```
PREFIX service:<http://www.daml.org/services/owl-s/1.2/Service.owl#>
PREFIX profile:<http://www.daml.org/services/owl-s/1.2/Profile.owl#>

SELECT ?subject ?name ?desc
WHERE { ?subject service:presents ?object.
?object profile:serviceName ?name.
?object profile:textDescription ?desc.
}
```

**Gambar 3.4 Query SPARQL yang digunakan**

Setelah mendapatkan seluruh *service name* dan *description* dari OWL-S, langkah selanjutnya adalah menghitung *similarity score* dari seluruh data terhadap *keyword* yang diinginkan pengguna. *Keyword* yang dimasukkan terbagi menjadi dua tipe, yaitu *service name* dan *description*. *Service name* pada OWL-S akan dicocokkan terhadap *service name* dari *keyword*, begitu juga dengan *description* dari OWL-S dan *keyword*. Perhitungan ini menggunakan konsep *semantic similarity* yang menghitung nilai *semantic similarity* dari jumlah kata yang sama dan memiliki hubungan yang tinggi. Rumus dari nilai *semantic similarity* antara OWL-S dan *keyword* dapat dilihat pada Persamaan 3.1

Persamaan 3.1 merupakan rumus yang dipakai pada penelitian ini untuk menghitung nilai kedekatan antara *keyword* yang diberikan dan sebuah OWL-S. Pada Persamaan 3.1, terdapat lima variabel dan satu fungsi. Fungsi *sem* adalah fungsi yang menghitung *semantic similarity* dari dua kalimat seperti yang dijelaskan di sub Bab 2.6. Variabel *nameOWL* adalah *service name* dari file OWL-S dan *nameKey* merupakan *service name* pada *keyword*. Variabel *descOWL* adalah *service description* yang ada pada file OWL-S dan *descKey* adalah *description* pada *keyword*. Dengan kata lain, nilai similaritas antara file OWL-S dan *keyword* adalah nilai rata-rata dari nilai *semantic similarity* antara *service name* pada OWL-S dan *keyword* dan nilai *semantic similarity* antara *description* pada OWL-S dan *keyword*.

Contoh dari Persamaan 3.1 dapat dilihat pada Tabel 3.2. Pada Tabel 3.2, terdapat sebuah *keyword* dan sebuah deskripsi *service* OWL-S. Nilai *score* merepresentasikan similaritas antara *keyword* dan OWL-S yang bernilai 0,927. Nilai ini didapat dari nilai rata-rata dari nilai similaritas *service name* yang bernilai 0,917 dan *description* yang bernilai 0,937.

Setelah seluruh OWL-S dibandingkan dengan *keyword*, langkah terakhir adalah memilih *service* yang berbentuk OWL-S yang akan ditampilkan ke pengguna. Penelitian ini menggunakan



sistem *threshold* untuk menentukan nilai terendah dari OWL-S dan *keyword* dari *service* yang akan ditampilkan. Sehingga, *service* atau OWL-S yang memiliki nilai similaritas lebih rendah dari *threshold*, tidak akan ditampilkan oleh sistem. Untuk menentukan nilai *threshold*, maka akan ditentukan pada pengujian yang dijelaskan pada bab pengujian.

$$score = \frac{sem(name_{OWL}, name_{Key}) + sem(desc_{OWL}, desc_{Key})}{2} \quad (3.1)$$

**Tabel 3.2 Contoh pemakaian rumus dari Persamaan 3.1**

	service name	Description
Keyword	pick material service	this service pick material and automate storage record
OWL-S	collect material service	this service pick material and automate storage record
Sem(n1,n2)	0.917	0.937
Score	0.927	

## **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 *Unified Modelling Language* (UML).

#### **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**

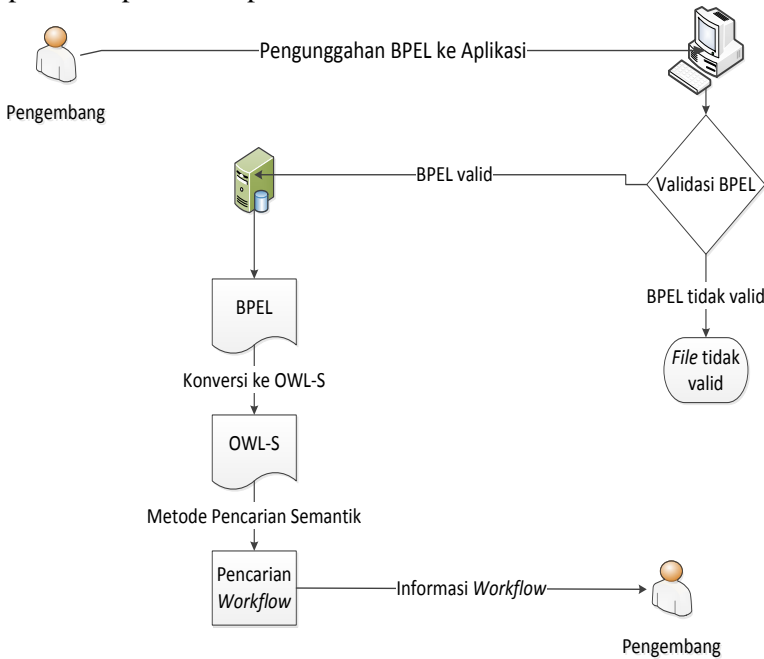
Permasalahan utama yang dihadapi oleh tugas akhir ini adalah membuat arsitektur WFMS yang bisa meningkatkan konsep *reusability* pada *workflow* dan menghasilkan *event log* yang sesuai standar *process mining*. Terdapat dua metode yang dipakai oleh arsitektur untuk mengatasi kedua masalah tersebut. Metode pertama menggunakan *semantic similarity* dan OWL-S. Permasalahan yang timbul tentang *threshold* dan rumus yang dipakai untuk menghitung nilai similaritas antara *keyword* dan *service* yang dicari. Permasalahan juga terdapat pada pemisahan dua *repository* yaitu BPEL *repository* dan OWL-S *repository*.

##### **4.1.2. Deskripsi Umum Sistem**

Aplikasi ini dibangun untuk memberikan arsitektur baru terhadap WFMS. WFMS yang dibangun mempunyai beberapa fungsi utama. Pertama, WFMS mampu mengimplementasikan *workflow* dalam standar BPEL menjadi *web service*. Kedua,

WFMS mampu meningkatkan konsep *reusability* dari *workflow*. Dan ketiga, WFMS mampu menghasilkan *event log* sesuai standar *process mining*. *Input* dari sistem ini adalah *file workflow* BPEL dan *keyword* deskripsi pencarian. *Output* dari sistem ini adalah *event log* dari aktivitas dan informasi tentang *workflow*.

Konsep yang dibangun oleh sistem ini memiliki beberapa perbedaan dengan WFMS lainnya. Pertama, anotasi dari *workflow* menggunakan pendekatan ontologi. WFMS lainnya menyimpan anotasi dari *workflow* menggunakan basis data relasional yang bersifat tertutup. Sehingga menyulitkan pengguna untuk melakukan *query* ke *repository*. Kedua, proses pembuatan *event log* juga mengikuti aturan dari *process mining*. Hal ini dikarenakan berbagai *event log* yang dihasilkan WFMS lain tidak bisa dianalisis langsung oleh aplikasi *process mining*. Alur aplikasi dapat dilihat pada Gambar 4.1.



**Gambar 4.1** Alur aplikasi WFMS yang dibangun

### 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.

**Tabel 4.1 Daftar kebutuhan fungsional perangkat lunak**

<b>Kode Kebutuhan</b>	<b>Kebutuhan Fungsional</b>	<b>Deskripsi</b>
TA-F0001	Mengunggah BPEL	Pengguna dapat mengunggah <i>workflow</i> dalam bentuk BPEL
TA-F0002	Melihat informasi <i>workflow</i>	Pengguna dapat melihat informasi <i>workflow</i> yang telah diunggah
TA-F0003	Mencari <i>workflow</i>	Pengguna dapat mencari <i>workflow</i> menggunakan <i>semantic workflow query</i>
TA-F0004	Mengunduh <i>event log</i>	Pengguna dapat mengunduh <i>event log</i> dari aktivitas <i>workflow</i>

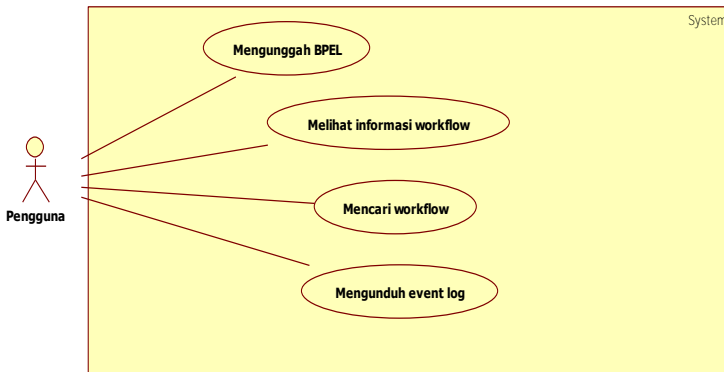
#### 4.1.4. Aktor

Aktor mendefinisikan entitas-entitas yang terlibat dan berinteraksi langsung dengan sistem. Entitas ini bisa berupa manusia maupun sistem atau perangkat lunak yang lain. Pada

tugas akhir ini, aktor yang ada di dalam sistem ini adalah pengembang *workflow*.

#### 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 4.2. Daftar kode diagram kasus penggunaan sistem dapat dilihat pada Tabel 4.2.



**Gambar 4.2 Diagram kasus penggunaan sistem**

**Tabel 4.2 Daftar kode diagram kasus penggunaan**

Kode Kasus Penggunaan	Nama
TA-UC0001	Mengunggah BPEL
TA-UC0002	Melihat informasi <i>workflow</i>
TA-UC0003	Mencari <i>workflow</i>
TA-UC0004	Mengunduh <i>event log</i>

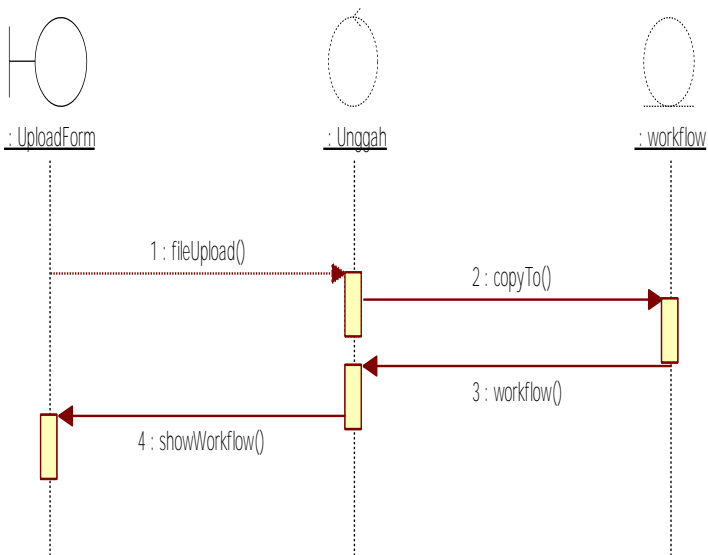
#### 4.1.5.1. Mengunggah BPEL ke WFMS

Pada kasus penggunaan ini, WFMS akan menerima perintah dari pengguna untuk mengunggah *file* BPEL ke aplikasi WFMS. Kemudian, BPEL akan dimasukkan ke *repository* dan disimpan oleh aplikasi. BPEL yang sudah disimpan, akan diimplementasikan sebagai *web service*. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 4.3. Diagram aktivitas dan diagram urutan dari kasus penggunaan ini bisa dilihat pada Gambar 4.3 dan Gambar 4.4.

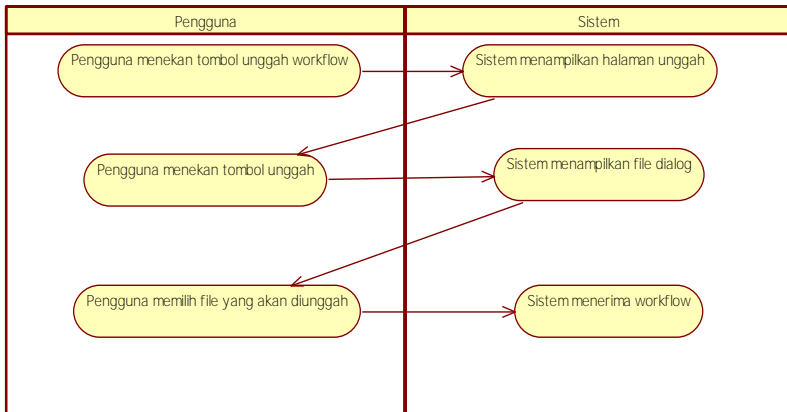
**Tabel 4.3 Spesifikasi kasus penggunaan mengunggah BPEL ke WFMS**

<b>Nama</b>	Mengunggah BPEL ke WFMS
<b>Kode</b>	TA-UC0001
<b>Deskripsi</b>	Mengunggah <i>file</i> BPEL ke aplikasi web oleh pengguna
<b>Tipe</b>	Fungsional
<b>Pemicu</b>	Pengguna menekan tombol perintah unggah pada aplikasi web WFMS
<b>Aktor</b>	Pengguna
<b>Kondisi Awal</b>	Pengguna telah ada di halaman unggah <i>workflow</i>
<b>Aliran:</b>	
<b>- Kejadian Normal</b>	<ol style="list-style-type: none"> <li>1. Pengguna menekan tombol unggah yang terdapat pada halaman unggah <i>workflow</i></li> <li>2. Sistem menampilkan <i>page</i> untuk melakukan unggah <i>workflow</i></li> <li>3. Pengguna memilih <i>file</i> yang akan diunggah menggunakan <i>file dialog</i></li> <li>4. Pengguna menekan tombol <i>open</i></li> </ol>

	<ol style="list-style-type: none"> <li>5. Sistem menyimpan <i>file</i> BPEL ke <i>repository</i></li> <li>6. Sistem menampilkan <i>form</i> deskripsi OWL-S</li> <li>7. Sistem menyimpan deskripsi OWL-S ke <i>repository</i></li> <li>8. Sistem menampilkan <i>file</i> yang telah dipilih ke dalam <i>workspace</i></li> </ol>
<b>- Kejadian Alternatif</b>	Tidak ada
<b>Kondisi Akhir</b>	Sistem menampilkan <i>file</i> yang telah diimpor ke dalam sistem
<b>Kebutuhan Khusus</b>	Tidak ada



**Gambar 4.3 Diagram urutan mengunggah BPEL ke dalam aplikasi.**



**Gambar 4.4 Diagram aktivitas mengunggah BPEL ke dalam aplikasi**

#### 4.1.5.2. Melihat Informasi *Workflow*

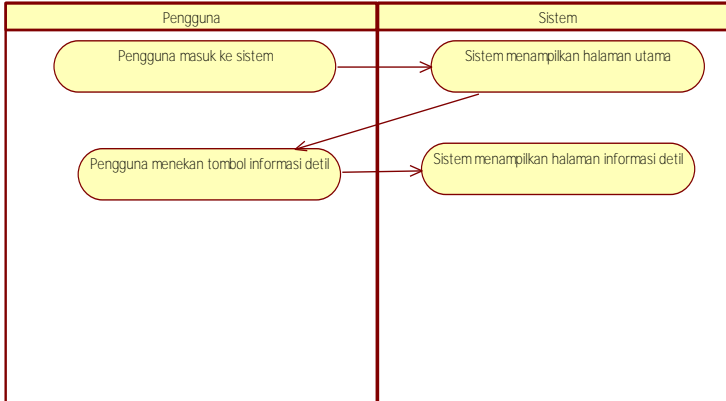
Pada kasus penggunaan ini, sistem akan menerima perintah dari pengguna untuk melihat informasi terkait tentang *workflow* seperti *instance*, *event log*, dan deskripsi. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 4.4. Diagram aktivitas dan diagram urutan dari kasus penggunaan ini bisa dilihat pada Gambar 4.5 dan Gambar 4.6.

**Tabel 4.4 Spesifikasi kasus penggunaan melihat informasi *workflow***

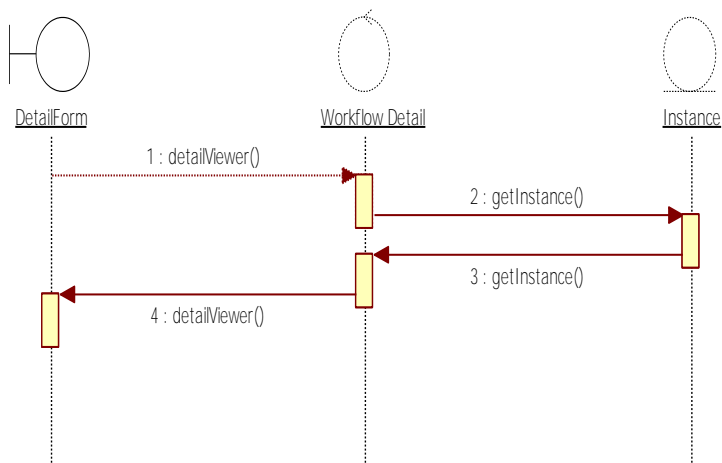
<b>Nama</b>	Melihat Informasi <i>Workflow</i>
<b>Kode</b>	TA-UC0002
<b>Deskripsi</b>	Melihat berbagai status dan keadaan dari <i>workflow</i> seperti <i>nstance</i> , <i>event log</i> , dan deskripsi dari <i>workflow</i> .
<b>Tipe</b>	Fungsional



<b>Pemicu</b>	Pengguna menekan tombol perintah melihat detail <i>workflow</i> pada halaman utama
<b>Aktor</b>	Pengguna
<b>Kondisi Awal</b>	Pengguna berada pada halaman utama
<b>Aliran:</b>	
<b>- Kejadian Normal</b>	<ol style="list-style-type: none"> <li>1. Pengguna menekan tombol perintah detail <i>workflow</i> pada halaman utama</li> <li>2. Sistem menampilkan <i>window</i> dari <i>page</i> yang berisi berbagai informasi dari <i>workflow</i></li> </ol>
<b>- Kejadian Alternatif</b>	Tidak ada
<b>Kondisi Akhir</b>	Sistem menampilkan informasi <i>workflow</i> pada halaman web.
<b>Kebutuhan Khusus</b>	Tidak ada



**Gambar 4.5** Diagram aktivitas melihat informasi *workflow*



**Gambar 4.6 Diagram urutan melihat informasi *workflow***

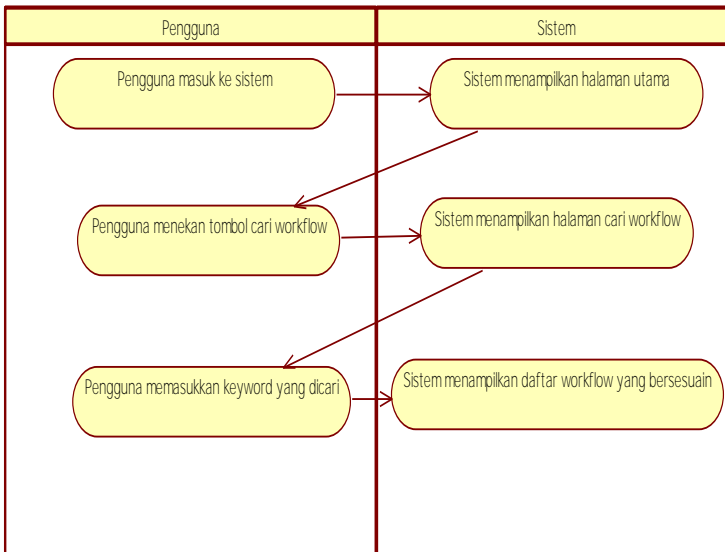
#### 4.1.5.3. Mencari *workflow*

Sistem dapat melakukan pencarian *workflow* menggunakan metode *semantic OWL-S query*. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 4.5. Diagram aktivitas dan diagram urutan dari kasus penggunaan ini bisa dilihat pada Gambar 4.7 dan Gambar 4.8.

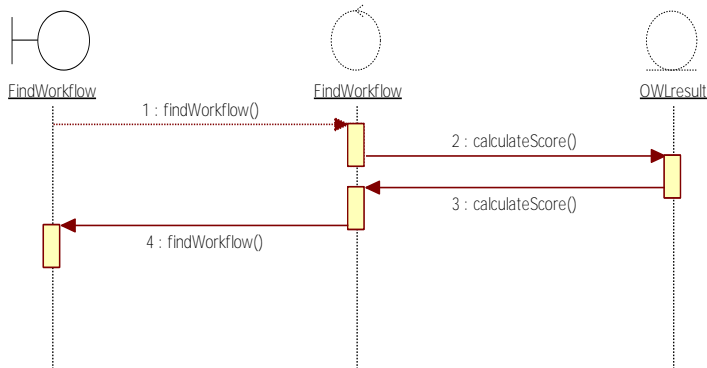
**Tabel 4.5 Spesifikasi kasus penggunaan menampilkan *event log***

<b>Nama</b>	Mencari <i>workflow</i>
<b>Kode</b>	TA-UC0003
<b>Deskripsi</b>	Sistem dapat melakukan pencarian terhadap <i>workflow</i>
<b>Tipe</b>	Fungsional
<b>Pemicu</b>	Pengguna menekan tombol cari <i>workflow</i> pada halaman utama

<b>Aktor</b>	Pengguna
<b>Kondisi Awal</b>	Pengguna berada pada halaman utama
<b>Aliran:</b>	
<b>- Kejadian Normal</b>	<ol style="list-style-type: none"> <li>1. Pengguna menekan tombol cari <i>workflow</i></li> <li>2. Sistem menampilkan <i>form pencarian</i></li> <li>3. Pengguna mengisi <i>keyword</i> pada form yang telah disediakan</li> <li>4. Sistem menampilkan berbagai <i>workflow</i> yang terkait dengan <i>keyword</i> dari pengguna</li> </ol>
<b>- Kejadian Alternatif</b>	Tidak ada
<b>Kondisi Akhir</b>	Sistem menampilkan berbagai <i>workflow</i> yang terkait dengan <i>keyword</i>
<b>Kebutuhan Khusus</b>	Tidak ada



**Gambar 4.7** Diagram aktivitas mencari *workflow*



**Gambar 4.8 Diagram urutan mencari *workflow***

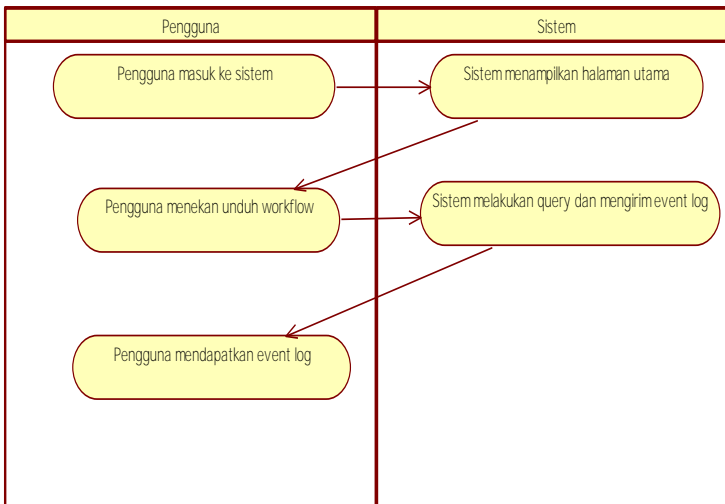
#### 4.1.5.4. Mengunduh *event log*

Sistem dapat menghasilkan *event log* yang berstandar *process mining*. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 4.6. Diagram aktivitas dan diagram urutan dari kasus penggunaan ini bisa dilihat pada Gambar 4.9 dan Gambar 4.10,

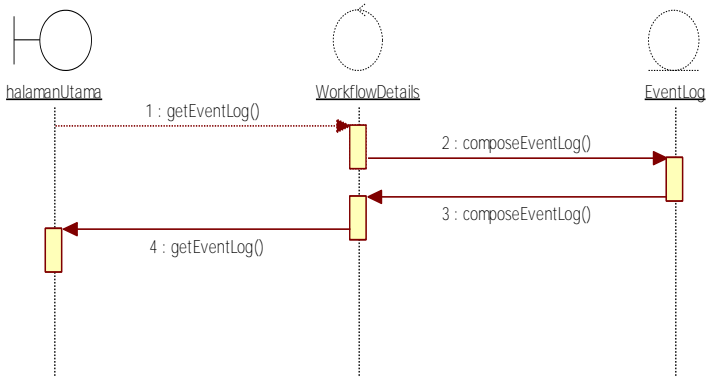
**Tabel 4.6 Spesifikasi kasus mengunduh *event log***

<b>Nama</b>	Mengunduh <i>event log</i>
<b>Kode</b>	TA-UC0004
<b>Deskripsi</b>	Pengguna dapat mengunduh <i>event log</i> dari aktivitas yang telah terjadi terhadap <i>workflow</i>
<b>Tipe</b>	Fungsional
<b>Pemicu</b>	Pengguna menekan tombol unduh <i>event log</i> pada halaman utama
<b>Aktor</b>	Pengguna
<b>Kondisi Awal</b>	Pengguna telah berada pada halaman utama
<b>Aliran:</b>	
<b>- Kejadian Normal</b>	1. Pengguna menekan tombol unduh <i>event log</i> pada <i>workflow</i> yang diinginkan.

	<ol style="list-style-type: none"> <li>2. Sistem melakukan <i>query</i> ke basis data</li> <li>3. Sistem mengaktifkan pengunduhan <i>event log</i></li> <li>4. <i>Event log</i> selesai diunduh</li> </ol>
<b>- Kejadian Alternatif</b>	Tidak ada
<b>Kondisi Akhir</b>	Sistem selesai mengunduh <i>event log</i>
<b>Kebutuhan Khusus</b>	Tidak ada
<b>Kebutuhan Khusus</b>	Tidak ada



**Gambar 4.9** Diagram aktivitas mengunduh *event log*



**Gambar 4.10** Diagram urutan mengunduh *event log*

## 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 Kelas

Perancangan diagram kelas berisi rancangan dari kelas-kelas yang digunakan untuk membangun sistem. Pada subbab ini, hubungan dan perilaku antar kelas 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. 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.

#### 4.2.1.1. Perancangan Kelas Lapisan Antarmuka

Perancangan kelas lapisan antarmuka meliputi perancangan berbagai halaman yang terdapat di aplikasi *web* ini. Halaman halaman ini meliputi halaman utama, halaman unggah *workflow*, halaman pencarian *workflow*, dan halaman informasi detail dari *workflow*.

Halaman utama merupakan halaman yang didapat pengguna setelah berhasil masuk sistem. Halaman unggah *workflow* merupakan halaman yang bertugas untuk membantu pengguna dalam mengunggah *file workflow* BPEL. Halaman pencarian *workflow* merupakan halaman yang berisi *keyword* yang dimasukkan pengguna pada proses *semantic query* pada OWL-S. Halaman informasi detail dari *workflow* merupakan halaman yang berisi berbagai informasi tentang aktivitas *workflow*.

#### 4.2.1.2. Perancangan Kelas Lapisan Kontrol

Lampiran Diagram A.1 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 FindWorkflow, Unggah, UserOperation, WorkflowDetails.

Kelas FindWorkflow adalah kelas yang bertanggung jawab pada metode pencarian *workflow* menggunakan *semantic similarity*. Kelas ini bertanggung jawab terhadap pengambilan *keyword* dari pengguna, mengakses OWL-S *repository*, melakukan *query* SPARQL terhadap OWL-S, melakukan perhitungan semantik terhadap deskripsi *workflow* dan menghasilkan daftar *service* atau *workflow* yang akan ditampilkan ke pengguna. Kelas UserOperation adalah kelas yang bertanggung jawab terhadap operasi terhadap status pengguna, seperti *login*, daftar pengguna dan *logout*.

Kelas unggah merupakan kelas yang bertanggung jawab terhadap proses pengunggahan *file workflow* BPEL dan pemindahan ke BPEL *repository*. Kelas `WorkflowDetails` merupakan kelas yang bertanggung jawab pada proses pembentukan informasi dari *workflow*, seperti mengakses Apache ODE, mengakses informasi dari basis data, dan menghasilkan *metadata* OWL-S dengan penambahan deskripsi dan konversi WSDL ke OWL-S.

#### 4.2.1.3. Diagram Kelas Lapisan Data

Diagram kelas pada lapisan data ditunjukkan oleh Diagram A.2. 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 `databaseOperation`, `EventLog`, `Instance`, `metadata`, `OWLquery`, `OWLresult`, `pengguna`, `UploadedFile`, `wordnet`, `workflow`, dan `workflowDetailsModel`.

Kelas `databaseOperation` merupakan kelas yang bertugas melakukan *query* ke basis data. *Query* tersebut bervariasi, seperti pada saat pendaftaran pengguna, *login* pengguna, pengunggahan *workflow* BPEL, dan pendaftaran aktivitas *workflow*. Kelas `EventLog` merupakan kelas yang berfungsi menampung seluruh informasi *event log* dari basis data. Kelas ini memiliki atribut yang terkait dengan standar *event log* dari *process mining* seperti *process id* dan *activity name*. Kelas `instance` berfungsi untuk menampung informasi tentang *instance activity* dari Apache ODE. Kelas ini memiliki atribut *instance* seperti *activity name*, *status*, *fault error*, dan *timestamp*.

Kelas `metadata` merupakan kelas yang menampung informasi deskripsi yang dibutuhkan oleh *profile* OWL-S. Kelas ini memiliki atribut *service name* dan *description*. Kelas `OWLquery` dan `OWLresult` merupakan kelas yang



bertanggung jawab dalam memuat informasi yang dibutuhkan oleh pencarian semantik *workflow*. `OWLresult` dilengkapi dengan sebuah *list* yang berisi *service* yang akan ditampilkan kepada pengguna. Kelas pengguna merupakan kelas yang digunakan saat pendaftaran pengguna dan *login* dari pengguna. Kelas ini memiliki atribut berupa nama dan kata sandi dari pengguna.

Kelas `UploadedFile` merupakan kelas yang berfungsi mengambil *file* yang diunggah oleh pengguna pada saat pengunggahan *workflow*. Kelas `wordnet` merupakan kelas yang menampung seluruh atribut yang akan dipakai pada saat proses semantik dari pencarian *workflow*. Beberapa atribut tersebut di antaranya adalah konfigurasi metode wu-palmer dan indeks dari basis data *wordnet*. Kelas `workflow` merupakan kelas yang bertugas menampung informasi tentang *workflow* saat pengunggahan yaitu nama dan tanggal pengunggahan. Sementara kelas `workflowDetails` bertugas memuat informasi detail dari *workflow* seperti nama, tanggal, daftar dari *instance workflow*, dan potongan *event log* terbaru dari *workflow* tersebut.

## 4.2.2. Perancangan Antarmuka Pengguna

Bagian ini membahas mengenai perancangan antarmuka pada sistem. Terdapat beberapa halaman pada aplikasi ini, yaitu halaman utama, halaman informasi detail *workflow*, halaman unggah *workflow* dan halaman pencarian *workflow*.

### 4.2.2.1. Halaman utama pada sistem

Halaman ini merupakan halaman yang didapat oleh pengguna setelah berhasil *login* ke sistem. Halaman ini berisi informasi seluruh *workflow* BPEL yang pernah diunggah oleh pengguna. Selain itu, di halaman utama ini terdapat pilihan-pilihan fitur yang diberikan oleh sistem ini, seperti menghasilkan *event log*, melihat informasi detail *workflow*, pencarian *workflow*, dan *generate* metadata OWL-S dari *workflow*. Rancangan halaman utama ini dapat dilihat pada Gambar 4.13. Penjelasan

mengenai atribut-atribut yang terdapat pada halaman ini bisa dilihat pada Tabel 4.8.



**Gambar 4.11** Rancangan halaman utama pada sistem

**Tabel 4.7** Spesifikasi atribut antarmuka halaman utama

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
1	Nama <i>workflow</i>	<i>Text</i>	Menampilkan nama <i>workflow</i> di table	<i>String</i>
2	Tanggal <i>workflow</i>	<i>Date</i>	Menampilkan tanggal pengunggahan <i>workflow</i>	<i>Date</i>
3	Informasi detil	<i>Button</i>	Menuju halaman detil <i>workflow</i> yang dipilih	<i>Button</i>
4	<i>Generate Metadata</i>	<i>Button</i>	Menuju halaman pembuatan deskripsi OWL-S	<i>Button</i>
5	<i>Generate Event log</i>	<i>Button</i>	Menuju aktivitas pengunduhan <i>event log</i>	<i>Button</i>

6	Unggah workflow	Link	Menuju halaman pengunggahan workflow	Link
7	Cari workflow	Link	Menuju halaman pencarian workflow	Link

#### 4.2.2.2. Halaman Informasi Detail Workflow

Halaman ini muncul ketika pengguna menekan tombol informasi detail pada tabel daftar workflow pada halaman utama. Pada halaman ini, terdapat beberapa informasi terkait workflow, seperti *instance management* dan *event log* terbaru. Rancangan halaman informasi detail workflow ini dapat dilihat pada Gambar 4.12. Penjelasan mengenai atribut-atribut yang terdapat pada halaman ini bisa dilihat pada Tabel 4.9.

The screenshot shows a web interface for workflow details. At the top, there is a green navigation bar with three links: 'halaman utama', 'unggah workflow', and 'cari workflow'. Below this is a dark header with the title 'informasi detil workflow'. The main content area is divided into three sections:

- Informasi Workflow:** A table with two rows: 'Nama Workflow' with value 'bpei-almas' and 'Tanggal pembuatan' with value '2014-02-20 01:49:04.0'.
- Manajemen Instance:** A table with columns: Process ID, Instance ID, Status, Start, Last Active, Error Since, and Fa. It contains three rows of instance data, all with a status of 'COMPLETED'.
- Event Log:** A table with columns: Process ID, Case ID, activity, status, and timestamp. It contains two rows of event data.

Gambar 4.12 Rancangan antarmuka halaman informasi detail

**Tabel 4.8 Spesifikasi atribut antarmuka halaman informasi detail**

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
1	Nama <i>workflow</i>	<i>String</i>	Menunjukkan nama dari <i>workflow</i>	<i>String</i>
2	Tanggal pengunggahan	<i>String</i>	Menunjukkan nilai tanggal pengunggahan <i>workflow</i>	<i>String</i>
3	<i>Process Id</i>	<i>String</i>	Menunjukkan <i>process id</i> dari <i>nstance</i>	<i>String</i>
4	<i>Instance ID</i>	<i>Integer</i>	Menunjukkan nilai <i>nstance ID</i>	<i>Integer</i>
5	<i>Status</i>	<i>String</i>	Menunjukkan status <i>Instance</i>	<i>String</i>
6	<i>Case ID</i>	<i>Integer</i>	Menunjukkan nilai <i>Case ID</i>	<i>Integer</i>
7	<i>Activity</i>	<i>String</i>	Menunjukkan nama aktivitas	<i>String</i>
8	<i>Timestamp</i>	<i>Date</i>	Menunjukkan tanggal terjadinya aktivitas	<i>Date</i>

#### 4.2.2.3. Halaman Pencarian *Workflow*

Halaman ini muncul ketika pengguna menekan *link* cari *workflow* pada *header panel*. Halaman ini mempunyai fungsi untuk menerima *keyword* yang dimasukkan pengguna. Rancangan halaman pencarian *workflow* ini dapat dilihat pada Gambar 4.13. Penjelasan mengenai atribut-atribut yang terdapat pada halaman ini bisa dilihat pada Tabel 4.9.



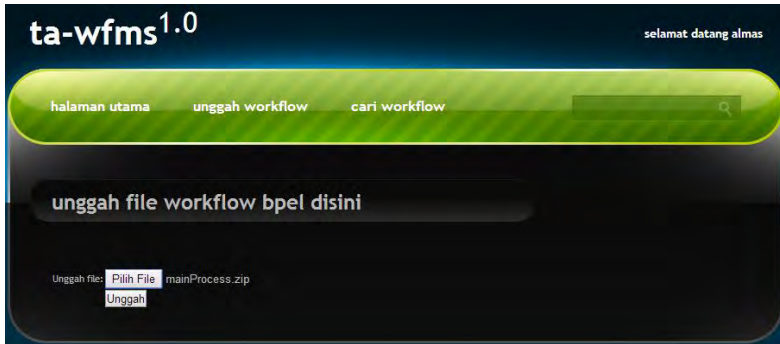
**Gambar 4.13** Rancangan antarmuka halaman pencarian *workflow*

**Tabel 4.4.9** Spesifikasi atribut antarmuka halaman informasi detail

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
1	<i>Service Name</i>	<i>Text</i>	Tempat mengisi <i>keyword</i> dari <i>service name</i>	<i>Text</i>
2	<i>Description</i>	<i>Text</i>	Menfinalisasi <i>file BPEL</i> yang dipilih	<i>Text</i>
3	<i>Find</i>	<i>Button</i>	Mengaktifkan fitur pencarian <i>workflow</i>	<i>Button</i>
4	Unggah <i>workflow</i>	<i>Link</i>	Menuju halaman pengunggahan <i>workflow</i>	<i>Link</i>
5	Halaman utama	<i>Link</i>	Menuju halaman utama	<i>String</i>

#### 4.2.2.4. Halaman Unggah *workflow*

Halaman ini muncul ketika pengguna menekan *link* unggah *workflow* yang ada pada *header*. Halaman ini mempunyai fungsi untuk mempermudah pengguna dalam mengunggah *workflow*. Rancangan halaman ini dapat dilihat pada Gambar 4.14. Penjelasan mengenai atribut-atribut yang terdapat pada halaman ini bisa dilihat pada Tabel 4.10,



**Gambar 4.14** Rancangan antarmuka halaman unggah *workflow*

**Tabel 4.10** Spesifikasi atribut antarmuka halaman unggah *workflow*

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
1	Pilih File	Button	Menampilkan <i>file dialog</i>	Button
2	Upload	Button	Menfinalisasi <i>file BPEL</i> yang dipilih	Button
3	Unggah <i>workflow</i>	Link	Menuju halaman pengunggahan <i>workflow</i>	Link
4	Cari <i>workflow</i>	Link	Menuju halaman pencarian <i>workflow</i>	Link
5	Halaman utama	Link	Menuju halaman utama	String

*[Halaman ini sengaja dikosongkan]*

## BAB V IMPLEMENTASI

Bab ini membahas tentang implementasi dari perancangan sistem. Bab ini berisi implementasi dari setiap kelas pada aplikasi ini. Bahasa 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. Implementasi Halaman Utama

Halaman utama memiliki beberapa fitur atau fungsi. Fungsi pertama adalah menampilkan data *workflow* yang diunggah oleh pengguna. Kedua, memberikan pilihan untuk melihat informasi detail dan *generate* metadata OWL-S. Gambar halaman utama dapat dilihat pada Gambar 5.1.



The screenshot shows a web interface with a green header bar containing navigation links: 'halaman utama', 'unggah workflow', and 'cari workflow'. Below the header, there is a section titled 'daftar workflow anda' containing a table with four rows of workflow data. Each row has three buttons: 'Informasi Detil', 'Bentuk metadata OWL-S', and 'Unduh Event Log'.

nama	tanggal			
bpel-aimas	2014-02-20 01:49:04.0	Informasi Detil	Bentuk metadata OWL-S	Unduh Event Log
halo	2014-02-20 00:00:00.0	Informasi Detil	Bentuk metadata OWL-S	Unduh Event Log
BPELaimas	2014-04-05 22:00:54.0	Informasi Detil	Bentuk metadata OWL-S	Unduh Event Log
mainProcess	2014-06-30 14:59:55.0	Informasi Detil	Bentuk metadata OWL-S	Unduh Event Log

Gambar 5.1 Antar muka halaman utama



### 5.1.2. Implementasi Halaman Detail *Workflow*

Halaman detail *workflow* pada aplikasi ini berfungsi untuk menampilkan berbagai informasi terkait keadaan dan aktivitas yang telah terjadi pada *workflow*. Pada halaman ini terdapat tiga jenis informasi *workflow*, yaitu informasi umum, informasi *instance management*, dan potongan *event log* terbaru.

Informasi umum meliputi nama dan tanggal pengunggahan *workflow*. *Instance management* berisi tentang aktivitas yang sedang dan telah terjadi pada *workflow* tersebut. Gambar 5.2 menunjukkan implementasi dari halaman ini.

halaman utama    unggah workflow    cari workflow

#### informasi detil workflow

**Informasi Workflow**

Nama Workflow	bpel-almas
Tanggal pembuatan	2014-02-20 01:49:04.0

**Manajemen Instance**

Process ID	Instance ID	Status	Start	Last Active	Error Since	Failure
{http://almasWorld}bpel-almas-3	802	COMPLETED	Thu May 29 18:04:51 ICT 2014	Thu May 29 18:04:52 ICT 2014	-----	-----
{http://almasWorld}bpel-almas-3	352	COMPLETED	Tue Apr 29 16:43:48 ICT 2014	Tue Apr 29 16:43:48 ICT 2014	-----	-----
{http://almasWorld}bpel-almas-3	351	COMPLETED	Tue Apr 29 15:10:52 ICT 2014	Tue Apr 29 15:10:53 ICT 2014	-----	-----

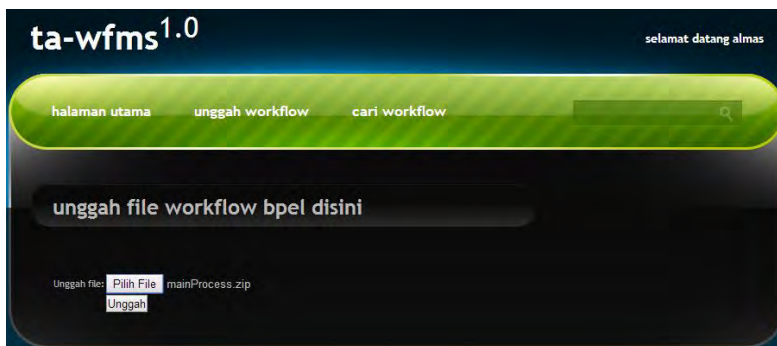
**Event Log**

Process ID	Case ID	activity	status	timestamp
{http://almasWorld}bpel-almas	5551	receiveInput	start	2014-03-01 21:08:06.0
{http://almasWorld}bpel-almas	5551	receiveInput	complete	2014-03-01 21:08:08.0

**Gambar 5.2** Antarmuka halaman informasi detil *workflow*

### 5.1.3. Implementasi Halaman Unggah *Workflow*

Halaman unggah *workflow* berfungsi untuk mempermudah pengguna dalam mengunggah *workflow* BPEL ke sistem. Halaman ini mengaktifkan *file dialog* yang berfungsi mengambil alamat *file* yang akan dipilih oleh pengguna. Setelah itu, nama *file* yang dimaksud akan muncul ke sistem. Gambar antar muka dapat dilihat pada Gambar 5.3.



Gambar 5.3 Antarmuka halaman unggah *workflow*

### 5.1.4. Implementasi Halaman Pencarian *Workflow*

Halaman pencarian *workflow* merupakan halaman yang dipakai untuk menangkap *keyword* dan menampilkan hasil dari pencarian. Terdapat sebuah *form* pada halaman ini, yaitu *form* yang berfungsi mengambil *keyword* berupa *service name* dan *service description*. Gambar antar muka dapat dilihat pada Gambar 5.4.

Kriteria	Kata Kunci
nama layanan	<input type="text"/>
deskripsi	<input type="text"/>
Cari	

**Gambar 5.4** Antarmuka halaman pencarian *workflow*

## 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 mengikuti fungsionalitas dari sistem.

### 5.2.1. Kelas Unggah

Kelas Unggah merupakan kelas yang melakukan pengunggahan *file* ke *repository*. Kelas ini bertanggung jawab dalam dekompresi *file* dan pemindahan ke *repository*.

### 5.2.2. Kelas UserOperation

Kelas ini bertanggung jawab dari fungsi *login* pengguna dan pada saat pendaftaran *pengguna* baru. Fungsi utama dari kelas UserOperation adalah *signup* yang dapat dilihat pada Kode Sumber 5.1.

```

@RequestMapping(value=      "/sign",      method      =
RequestMethod.POST)
    public ModelAndView signUp(@ModelAttribute("pengguna")
    pengguna p, BindingResult result)
    {
        databaseOperation db = new databaseOperation();
        db.insertPengguna(p);
        return new ModelAndView("afterLogin", "command", p);
    }
        return driver;

```

**Kode Sumber 5.1 Fungsi signup pada kelas UserOperation**

### 5.2.3. Kelas FindWorkflow

Kelas FindWorkflow merupakan kelas yang bertanggung jawab dalam proses pencarian *workflow* berbasis *semantic OWL-S query*. Fungsi utama dari kelas ini adalah fungsi `calculateScore` yang menghitung nilai similaritas menggunakan basis data *wordnet* dan metode *wu-palmer* serta fungsi `doOWL` yang menggunakan *query SPARQL*. Fungsi `calculateScore` dapat dilihat pada Kode Sumber 5.2. Fungsi `doOWL` dapat dilihat pada Kode Sumber 5.3.

```

private OWLresult calculateScore(OWLquery q, OWLresult rs)
throws WSDLException
    {
        //readWSDL();
        String s = q.getServiceName();
        String s2 = rs.getServiceName();
        String[] r = s.split("(?=\p{Upper})");
        String[] r2 = s2.split("(?=\p{Upper})");
        /*
        System.out.print("      ");
        for (int i = 0; i < r.length; i++)
        {
            System.out.print(r[i] + " ");
        }
        System.out.println();
        */
        for (int i = 0; i < r2.length; i++)
        {
            if (r2[i].startsWith(" "))
            {
                r2[i] = r2[i].substring(1,
r2[i].length());

```

```

        }
        //System.out.println(r2[i]);
    }

    RelatednessCalculator rc = word.getRc();
    double[][]
matrix=rc.getNormalizedSimilarityMatrix(r, r2);
    double score = 0;
    boolean[] status = new
boolean[Math.max(r.length,r2.length)];
    for (int
i=0;i<Math.max(r.length,r2.length);i++)
    {
        status[i]=false;
    }
    for (int i=0;i<matrix.length;i++)
    {
        for (int
j=0;j<matrix[i].length;j++)
        {
            if (status[j]) continue;
            if
(r[i].equalsIgnoreCase(r2[j]))
            {
                status[j]=true;
                score+=1;

System.out.println("score = "+score+"yeah "+r[i]+"
"+r2[j]);

                break;
            }
            else
            if (matrix[i][j] > 0,7)
            {
                status[j]=true;
                score+=0,75;

                System.out.println("score = "+score+"hoho "+r[i]+"
"+r2[j]);

                break;
            }
        }
    }
    //System.out.print(matrix[i][j]+" ");
    }
    //System.out.println();
}

```

**Kode Sumber 5.2 Fungsi calculateScore pada kelas FindWorkflow**

```

private List<OWLresult> doOWL(List<OWLresult> owl_list, File
dir, OWLquery q) throws WSDLException, IOException
{
    OntModel                model                =
ModelFactory.createOntologyModel( OntModelSpec.OWL_MEM_MICRO
_RULE_INF);
    model=getOWL(model, dir);
    String                queryString            =
service:<http://www.daml.org/services/owl-
s/1.2/Service.owl#>\n"+
"PREFIX
profile:<http://www.daml.org/services/owl-
s/1.2/Profile.owl#>\n"+
"SELECT ?name ?desc\n"+
"WHERE { ?subject
service:presents ?object.\n"+
"?object
profile:serviceName ?name.\n"+
"?object
profile:textDescription ?desc.\n}";
    Query                query                =
QueryFactory.create(queryString);
    // Execute the query and obtain results
    QueryExecution        qe                =
QueryExecutionFactory.create(query, model);
    ResultSet results = qe.execSelect();
    System.out.println("++"+results.getRowNumber());
    while (results.hasNext())
    {
        QuerySolution qs = results.next();
        System.out.println((qs.getLiteral("name").getString()
));
        System.out.println((qs.getLiteral("desc").getString()
));
        OWLresult        rs                =
new
OWLresult(qs.getLiteral("name").getString(),
qs.getLiteral("desc").getString(), 0);
        rs = calculateScore(q, rs);
        //System.out.println(queryString);
        if (rs.getScore()<threshold) continue;
        owl_list.add(rs);
    }
    qe.close();
    return owl_list;
}

```

```
}

```

**Kode Sumber 5.3 Fungsi doOWL pada kelas FindWorkflow**

### 5.2.4. Kelas WorkflowDetails

Kelas `WorkflowDetails` berfungsi untuk mengambil berbagai informasi detail dari *workflow* yaitu potongan *event log*, *instance management* dan informasi umum serta menghasilkan metadata OWL-S. Rincian kode dari `meta_generator` dapat dilihat pada Kode Sumber 5.6.

```
@RequestMapping("/meta")
public ModelAndView meta_generator(HttpSession hs,
@RequestParam(value = "nama") String
nama, @RequestParam(value = "description") String desc)
throws Exception
{
    //URI uri = new URI(deploymentURL+fileName.substring(0,
URI (deploymentURL+fileName.substring(0, fileName.length()-
4));
    URI uri = new URI(deploymentURL+"bpel-almas?wsdl");
    WSDLService ws = new WSDLService(uri);
    URI ontURI;
    try
    {
        ontURI = new URI(owlsURI+nama+".owl");
        OWLOntology ont =
OWLFactory.createKB().createOntology(ontURI);
        OWLS.addOWLSImports(ont);
        WSDLTranslator wst = new WSDLTranslator(ont,
ws.getOperations().get(0), "");
        wst.setTextDescription(desc);

        wst.setServiceName(ws.getOperations().get(0).getName());

        //wst.addInput(ws.getOperations().get(0).getInput(0),
ws.getOperations().get(0).getInput(0).getName(),
ws.getOperations().get(0).getInput(0).getType(),
ws.getOperations().get(0).getInput(0).);

        //System.out.println(ws.getOperations().get(0).getInput(0)
).getType().toString());
        FileOutputStream fos = new
FileOutputStream(new File(owlsDir+nama+".owl"));
        wst.writeOWLS(fos);
    }
}
```

```

        fos.close();
    }
    catch (URISyntaxException e)
    {
        System.out.println("wrong URI format");
    }
    return new ModelAndView("redirect:/main.html");
}

```

**Kode Sumber 5.4 Fungsi `meta_generator` pada kelas `WorkflowDetails`**

### 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 Database Operation

Kelas Database Operation merupakan kelas lapisan data yang mempunyai fungsi-fungsi yang berhubungan dengan *query* basis data. Berbagai metode *query* basis data yang ada antara lain menambah data pengguna baru, *login* pengguna, memasukkan informasi *workflow*, menambah data *event log* baru, dan mengakses informasi detail dari *workflow*.

#### 5.3.2. Kelas EventLog

Kelas *EventLog* adalah kelas yang merepresentasikan seluruh atribut pada *event log* berstandar *process mining*. Kelas ini digunakan untuk menampilkan *event log* pada halaman informasi detail *workflow*

#### 5.3.3. Kelas Instance

Kelas *instance* merupakan kelas yang merepresentasikan berbagai atribut yang dimiliki oleh sebuah *instance*. Sebuah *instance* memiliki atribut seperti *process id*, *status*, dan *timestamp*.



#### **5.3.4. Kelas OWLQuery**

Kelas `OWLQuery` merupakan kelas yang berfungsi menangkap *keyword* dan dikirim ke metode penghitungan nilai similaritas pada lapisan kontrol. Kelas ini memiliki atribut *keyword* yang dimasukkan pengguna.

#### **5.3.5. Kelas Wordnet**

Kelas `Wordnet` merupakan kelas yang digunakan untuk keperluan akses basis data *wordnet* yang berisi daftar kata-kata yang dicari nilai similaritasnya. Kelas ini berisi atribut basis data *wordnet* dan metode yang dipakai yaitu metode *wu-palmer*.

#### **5.3.6. Kelas WorkflowDetails**

Kelas `workflowdetails` merupakan kelas yang menampung seluruh informasi detil dari keadaan suatu *workflow*. Kelas ini akan dipakai pada halaman informasi *workflow* untuk menampilkan seluruh data terkait *workflow activities*.

## **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 tiga. 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 <i>Device</i>	: Laptop
Sistem Operasi	: Microsoft Windows 8 Enterprise 64 bit
IDE Eclipse	: Eclipse Kepler

#### **6.2. Skenario Pengujian**

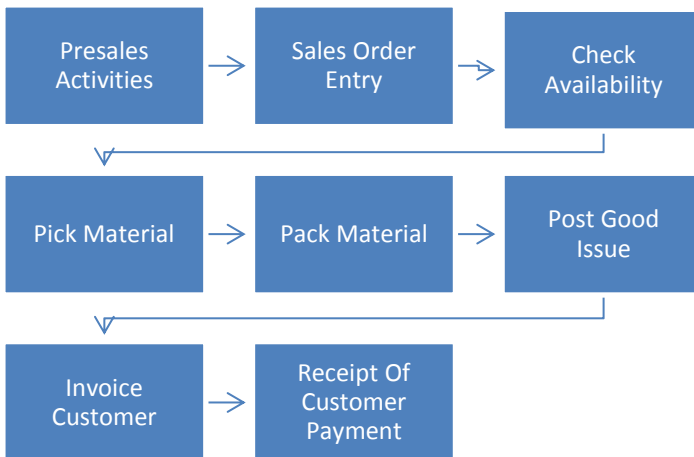
Pada bagian ini akan dijelaskan tentang skenario pengujian yang dilakukan. Pengujian dilakukan dalam tiga tahap yaitu pengujian kebutuhan fungsionalitas, pengujian pada studi kasus, dan pengujian kegunaan sistem. Pengujian kebutuhan fungsionalitas menggunakan metode kotak hitam (*black box*). Metode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

Kasus yang digunakan di riset ini adalah kasus dari proses bisnis *Enterprise Resource Planning*. Bagian yang digunakan merupakan bagian *sales order* dari proses bisnis ERP. Alur dari

*sale order* pada ERP dapat dilihat pada Gambar 6.1. *Sale order* merupakan alur proses bisnis dari ERP yang fokus pada pengadaan barang. Alur ini dimulai dari permintaan pengadaan dan berakhir pada pembayaran barang.

Terdapat delapan aktivitas yang di dalam alur ini. Aktivitas pertama, *presale activities* merupakan suatu aktivitas yang mengecek barang yang dibutuhkan dan kelengkapan dokumen pengadaan barang. *Sales order entry* merupakan aktivitas yang memasukkan data pengadaan dan memeriksa atribut pengadaan. *Check availability* adalah aktivitas yang mengecek ketersediaan barang yang mau diadakan di *inventory*. Aktivitas *pick material* dan *pack material* merupakan aktivitas yang mengambil daftar barang yang dibutuhkan. Aktivitas *post good issue* merupakan aktivitas yang mengurus transportasi dan penerimaan barang di gudang. Aktivitas *invoice customer* dan *receipt of customer payment* merupakan aktivitas yang mengurus masalah pembayaran terhadap barang setelah sampai di gudang.

Masing-masing aktivitas merupakan komposisi dari *web service* dibawahnya. Deskripsi berbagai aktivitas juga akan dimasukkan ke dalam kasus studi. Deskripsi ini didapat dari kumpulan penjelasan yang didapat dari ERP.



**Gambar 6.1** Proses bisnis *sale 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 sub Bab 4.1.5. Pengujian pada kebutuhan fungsionalitas dapat dijabarkan pada subbab berikut.

#### 6.2.1.1. Pengujian Fitur Mengunggah BPEL

Pengujian fitur mengunggah BPEL dilakukan dengan mengunggah sebuah sampel *file workflow* BPEL yang telah dikompresi dalam ekstensi zip. *File* yang telah dikompresi tersebut harus berisi *file* BPEL, berbagai WSDL yang dibutuhkan, dan sebuah ODE *Deployment Descriptor*. Rincian pengujian fitur ini dapat dilihat pada Tabel 6.1. Hasil pengujian bisa dilihat pada Gambar 6.2.

**Tabel 6.1 Pengujian fitur mengunggah BPEL**

<b>ID</b>	TA-UJ.UC0001
<b>Referensi Kasus Penggunaan</b>	TA-UC0001
<b>Nama</b>	Pengujian fitur mengunggah BPEL
<b>Tujuan Pengujian</b>	Menguji fitur untuk mengunggah BPEL.
<b>Skenario 1</b>	Mengunggah sebuah <i>file</i> BPEL dari aktivitas <i>presales activities</i> dari proses bisnis <i>Sales Order</i>
<b>Kondisi Awal</b>	Pengguna sudah berada pada halaman unggah <i>workflow</i>
<b>Data Uji</b>	Data uji merupakan <i>file workflow</i> dari aktivitas <i>presales activities</i> dari proses bisnis <i>Sales Order</i>
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pengguna menekan tombol unggah</li> <li>2. Pengguna memilih dokumen yang akan diunggah</li> </ol>

	3. Sistem menampilkan <i>file</i> yang sukses diunggah
<b>Hasil Yang Diharapkan</b>	<i>File workflow</i> BPEL dapat diunggah dengan sukses
<b>Hasil Yang Didapat</b>	<i>File workflow</i> BPEL dapat diunggah dengan sukses
<b>Hasil Pengujian</b>	Berhasil.
<b>Kondisi Akhir</b>	Sistem menampilkan nama <i>file workflow</i> yang telah sukses diunggah



**Gambar 6.2** Hasil pengunggahan *workflow*

### 6.2.1.2. Pengujian Fitur Melihat Informasi *Workflow*

Pengujian fitur melihat informasi *workflow* dilakukan dengan melihat aktivitas *workflow* yang telah dipanggil beberapa kali. Setiap pemanggilan akan mengaktifkan berbagai aktivitas yang ada di dalam *sequence workflow* tersebut. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 6.2. Hasil dari pengujian dapat dilihat pada Gambar 6.3.

**Tabel 6.2 Pengujian fitur melihat informasi *workflow***

<b>ID</b>	<b>TA-UJ.UC0002</b>
<b>Referensi Kasus Penggunaan</b>	TA-UC0002
<b>Nama</b>	Pengujian fitur untuk melihat informasi <i>workflow</i>
<b>Tujuan Pengujian</b>	<b>Menguji fitur untuk melihat informasi <i>workflow</i></b>
<b>Skenario 1</b>	<b>Melihat informasi <i>workflow</i> yang telah diunggah</b>
<b>Kondisi Awal</b>	Pengguna telah berada pada halaman utama
<b>Data Uji</b>	Data uji merupakan <i>file workflow</i> yang sudah diunggah pengguna
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pengguna menekan tombol informasi detail pada baris <i>workflow</i> yang diinginkan</li> <li>2. Sistem menampilkan halaman informasi detail</li> </ol>
<b>Hasil Yang Diharapkan</b>	Semua informasi detail berhasil ditampilkan pada halaman informasi detail <i>workflow</i>
<b>Hasil Yang Didapat</b>	Semua informasi detail berhasil ditampilkan pada halaman informasi detail <i>workflow</i>
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	Tampilan berbagai informasi detail <i>workflow</i> pada halaman informasi detail

halaman utama    unggah workflow    cari workflow

### informasi detail workflow

**Informasi Workflow**

Nama Workflow	bpel-almas
Tanggal pembuatan	2014-02-20 01:49:04.0

**Manajemen Instance**

Process ID	Instance ID	Status	Start	Last (Active)	Error Since	Fail
{http://almasWorld}bpel-almas-3	802	COMPLETED	Thu May 29 18:04:51 ICT 2014	Thu May 29 18:04:52 ICT 2014	-----	-----
{http://almasWorld}bpel-almas-3	352	COMPLETED	Tue Apr 29 16:43:48 ICT 2014	Tue Apr 29 16:43:48 ICT 2014	-----	-----
{http://almasWorld}bpel-almas-3	351	COMPLETED	Tue Apr 29 15:10:52 ICT 2014	Tue Apr 29 15:10:53 ICT 2014	-----	-----

**Event Log**

Process ID	Case ID	activity	status	timestamp
{http://almasWorld}bpel-almas	5551	receiveInput	start	2014-03-01 21:08:06.0
{http://almasWorld}bpel-almas	5551	receiveInput	complete	2014-03-01 21:08:08.0

Gambar 6.3 Hasil pengujian melihat informasi *workflow*

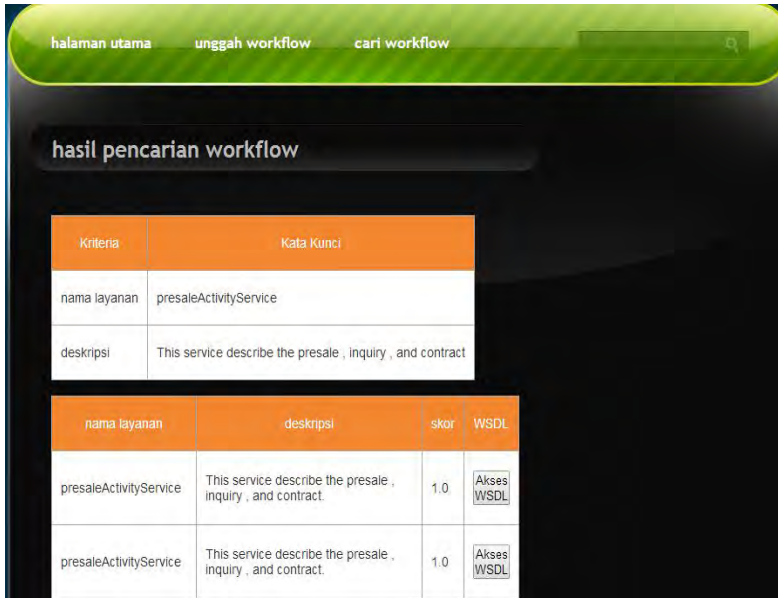
### 6.2.1.3. Pengujian Fitur Mencari *Workflow*

Pengujian fitur pencarian *workflow* merupakan pengujian terhadap kemampuan metode mencari *workflow* yang sesuai dengan *keyword* yang diberikan. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 6.3. Tabel tersebut bertujuan untuk menguji fungsionalitas sistem dalam melakukan pencarian *workflow*. Hasil pengujian dapat dilihat pada Gambar 6.4.

Tabel 6.3 Pengujian fitur mencari *workflow*

<b>ID</b>	TA-UJ.UC0003
<b>Referensi Kasus Penggunaan</b>	TA-UC0003
<b>Nama</b>	Pengujian fitur mencari <i>workflow</i>
<b>Tujuan Pengujian</b>	Menguji fungsionalitas sistem dalam mencari <i>workflow</i>
<b>Skenario 1</b>	Melakukan pencarian berdasarkan <i>keyword</i> aktivitas <i>check availability</i> dari proses bisnis <i>sales order</i>
<b>Kondisi Awal</b>	Pengguna sudah berada pada halaman utama
<b>Data Uji</b>	Data uji berupa seluruh deskripsi <i>web service</i> berbasis OWL-S yang ada pada OWL-S <i>repository</i>
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pengguna menekan tombol cari <i>workflow</i></li> <li>2. Sistem menampilkan halaman pencarian <i>workflow</i></li> <li>3. Pengguna mengisi <i>keyword</i> yang diinginkan</li> <li>4. Pengguna menekan tombol <i>find</i></li> <li>5. Sistem menampilkan berbagai <i>workflow</i> yang sesuai dengan <i>keyword</i> pdari pengguna</li> </ol>
<b>Hasil Yang Diharapkan</b>	Dapat menampilkan berbagai <i>workflow</i> yang memiliki keterkaitan dengan <i>keyword</i> dari pengguna.
<b>Hasil Yang Didapat</b>	Dapat menampilkan berbagai <i>workflow</i> yang memiliki keterkaitan dengan <i>keyword</i> dari pengguna.
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	Daftar seluruh <i>workflow</i> yang terkait dengan <i>keyword</i> dari pengguna





**Gambar 6.4 Hasil uji fitur pencarian *workflow***

#### 6.2.1.4. Pengujian Fitur Mengunduh *Event Log*

Pegujian fitur mengunduh *event log* dilakukan dengan mengunduh *event log* dari *workflow sale order* yang sudah pernah diunggah oleh pengguna. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 6.4. Untuk hasil pengujian pada skenario 1 dapat dilihat pada Gambar 6.5.

**Tabel 6.4 Pengujian fitur mengunduh *event log***

<b>ID</b>	<b>TA-UJ.UC0004</b>
<b>Referensi Kasus Penggunaan</b>	TA-UC0004
<b>Nama</b>	Pengujian fitur mengunduh <i>event log</i>
<b>Tujuan Pengujian</b>	Menguji fungsionalitas sistem dalam mengunduh <i>event log</i>

<b>Skenario 1</b>	Mengunduh <i>event log</i> dari <i>workflow</i> proses bisnis <i>sales order</i> .
<b>Kondisi Awal</b>	Pengguna berada pada halaman utama
<b>Data Uji</b>	-
<b>Langkah Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pengguna menekan tombol unduh <i>event log</i> pada <i>workflow</i> yang diinginkan</li> <li>2. Sistem menginisiasi prosedur pengunduhan <i>workflow</i></li> <li>3. <i>Event log</i> selesai diunduh</li> </ol>
<b>Hasil Yang Diharapkan</b>	<i>Event log</i> berhasil diunduh
<b>Hasil Yang Didapat</b>	<i>Event log</i> berhasil diunduh
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	<i>Event log</i> berhasil diunduh oleh pengguna



Gambar 6.5 Hasil *event log* yang dibuka di aplikasi ProM

## 6.2.2. Pengujian Pada Studi Kasus

### 6.2.2.1. Studi Kasus Proses Bisnis *Sale Order*

Setiap aktivitas dari alur *sales order* pada Gambar 4.1 akan dijadikan 30 buah *web service* yang memiliki kemiripan tinggi. *Web service* itu bersifat *composite*, atau memiliki bagian lagi di dalamnya. Dalam pengujian ini, digunakan 240 buah *web service* untuk menguji metode *similarity score* yang diajukan di tugas akhir ini. Seluruh *web service* telah dikonversi ke dalam OWL-S memiliki *service name* dan *description* dalam bahasa Inggris.

Skenario pengujian memecah *web service* tersebut ke dalam delapan tipe sesuai dengan aktivitasnya masing-masing. Hal ini dilakukan karena terdapat delapan *query* berbeda yang terdapat di pengujian. Masing-masing *query* mewakili satu aktivitas. Setelah *web service* dibagi menjadi delapan bagian, masing-masing *query* akan dijalankan. Sehingga nilai *similarity score* dari masing-masing *web service* akan muncul dan dicatat.

Setelah semua data selesai diuji dan mendapat nilai, akurasi dari metode ini dapat dihitung sesuai rumus pada Persamaan 6.1. Nilai *True Positive* (TP) merupakan jumlah data

yang memiliki nilai di atas *threshold* dan berasal dari daftar *web service* pada aktivitas tersebut. Nilai *True Negative* (TN) merupakan jumlah data yang memiliki nilai dibawah *threshold* dan tidak berasal dari daftar *web service* pada aktivitas tersebut. Nilai *False Positive* (FP) merupakan jumlah data yang memiliki nilai diatas *threshold* tetapi tidak berasal dari daftar *web service* pada aktivitas tersebut. Nilai *False Negative* (FN) merupakan jumlah data yang memiliki nilai di bawah *threshold* tetapi berasal dari daftar *web service* pada aktivitas tersebut. Hasil pengujian dari beberapa aktivitas dilampirkan dalam buku tugas akhir ini. Tabel C.1 merepresentasikan hasil pengujian pada aktivitas *presale activity*. Tabel C.2 merepresentasikan hasil pengujian aktivitas *sale order entry*. Tabel C.3 merepresentasikan hasil pengujian aktivitas *check availability*. Tabel C.4 merepresentasikan hasil pengujian aktivitas *pick material*. Hasil penghitungan seluruh *threshold* dapat dilihat pada Tabel 6.5. Dari Tabel 6.5, dapat disimpulkan bahwa *threshold* yang dipilih adalah 0,67 yang berasal dari rata-rata *threshold* terbaik tiap aktivitas.

Dari hasil pengujian, terdapat beberapa hal yang patut dicermati. Pertama, kasus FP terjadi karena beberapa deskripsi yang terdapat pada *web service* memiliki kata-kata yang sinonim dengan *keyword* tetapi pada letak yang berbeda. Perbedaan letak kata tidak diakomodir dalam perhitungan. Kasus seperti ini jarang terjadi, sehingga tidak menimbulkan perubahan yang berarti. Kedua, *threshold* yang tinggi tidak menjamin akurasi yang tinggi. Hal ini disebabkan perhitungan nilai semantik bergantung pada perbedaan jumlah kata. Dengan kata lain, jika perbedaan jumlah kata *keyword* dan OWL-S terlalu jauh, nilai semantik akan semakin kecil. Sehingga, kasus dengan nilai semantik yang kecil bisa saja berasal dari daftar positif, karena perbedaan jumlah kata pada OWL-S dan *keyword*. Oleh karena itu, disarankan pembatasan jumlah kata pada OWL-S dan *keyword* untuk menjaga akurasi yang tinggi.

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

**Tabel 6.5** Tabel hasil perhitungan *threshold*

No	Activity	Threshold terbaik
1	Presale Activity Service	0,7
2	Sale Order Entry Service	0,7
3	Check Availability Service	0,7
4	Pick Material Service	0,6
5	Pack Material Service	0,6
6	Post Good Issue Service	0,7
7	Invoice Customer Service	0,6
8	Receipt of Customer Payment Service	0,7
	Rata-rata	0,6625

### 6.2.2.2. Studi Kasus Proses Bisnis *Material Management, Controlling, Financial Account, Human Capital Management, dan Production Planning*

Pengujian pada studi kasus proses bisnis *material management, controlling, financial account, human capital management, dan production planning* berbeda dengan pengujian pada studi kasus *sales order*. Pengujian pada kasus studi *sales order* bertujuan untuk mendapatkan *threshold* dan mendapatkan akurasi tertinggi. Untuk pengujian pada bagian ini, pengujian ditujukan untuk menguji metode dan *threshold* yang telah diteapkan. Pada pengujian kali ini, masing-masing proses bisnis akan diterjemahkan ke dalam 30 *service* yang berbeda. Masing-masing *service* akan bersifat *composite* dan memiliki beberapa aktivitas di dalamnya. Untuk pengujian kali ini, *service* dari

aktivitas *composite* tersebut juga ikut masuk dalam perhitungan metode *semantic query*.

Proses bisnis *material management* merupakan bagian dari proses bisnis ERP yang mengurus bagian manajemen dari data dan keadaan barang di *inventory*. Proses bisnis *controlling* merupakan proses bisnis yang berfungsi mengumpulkan data dan menghasilkan laporan. Proses bisnis *financial account* bertugas mengawasi segala transaksi dalam perusahaan dan khususnya arus masuk dan keluar dari dana perusahaan. Proses bisnis, *human capital management* berfungsi untuk mengatur kebutuhan pegawai dan staf di perusahaan tersebut. Proses bisnis *production planning* bertugas untuk merancang proses tahap produksi barang sesuai spesifikasi. Semua proses bisnis ini akan direpresentasikan dalam *service* dan dikonversi ke OWL-S.

Setelah itu, seluruh data OWL-S tersebut akan dilakukan *query* menggunakan 5 *query* yang berbeda tergantung kepada tipe proses bisnis yang ada. Hasil pengujian dapat dilihat pada Tabel 6.6. Pada Tabel 6.6, dapat dilihat bahwa seluruh proses bisnis dapat terdeteksi dengan sempurna. Atribut *max score* menunjukkan nilai tertinggi yang didapatkan oleh data OWL-S ketika dilakukan *query* menggunakan *semantic query*, sementara *min score* menunjukkan skor terendah dan *average* menunjukkan skor rata-rata.

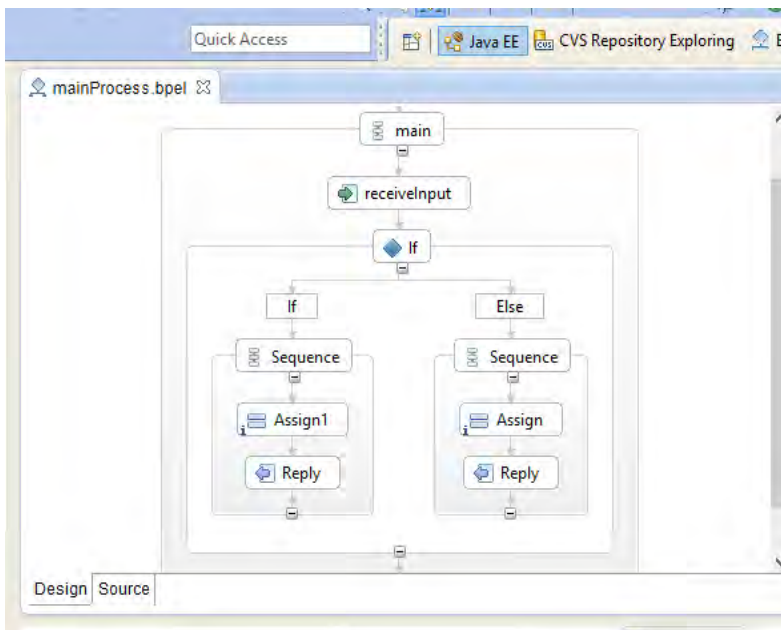
**Tabel 6.6 Tabel hasil pengujian pada berbagai proses bisnis ERP**

Proses Bisnis	Jumlah Terdeteksi	Max score	Min score	average score
Material Management	30	1	0,75	0,892968
Controlling	30	1	0,708	0,860396
Financial Account	30	1	0,748	0,934143
Human Capital Management	30	1	0,75	0,909112
Production Planning	30	1	0,7345	0,909755

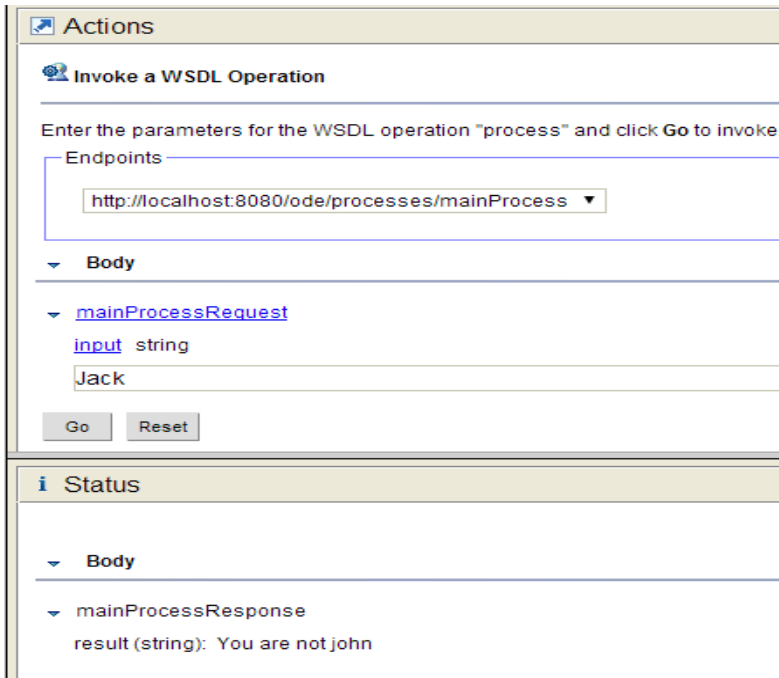
### 6.2.2.3. Studi Kasus *Workflow* yang Memiliki *Complex Sequence*

Pengujian ini menguji kemampuan WFMS dalam menjalankan *workflow* yang memiliki aktivitas *complex*. Di dalam pengujian ini, *workflow* yang dipakai adalah *workflow* yang tergambar pada Gambar 6.6. *Workflow* pada Gambar 6.6 memiliki satu aktivitas percabangan yang berisi kondisi yang menyeleksi *string* dari masukan dan mengembalikan *response* yang tergantung kepada masukan.

Hasil pengujian *workflow* ini menunjukkan bahwa WFMS bisa menjalankan berbagai *workflow*, baik *sequential* ataupun *complex*. Keberhasilan pengujian ini dapat dilihat pada Gambar 6.7.



Gambar 6.6 *Workflow* yang digunakan dalam pengujian



**Gambar 6.7 Hasil *workflow* yang telah terimplementasi**

### 6.2.3. Pengujian Kegunaan

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

Pengujian dilakukan oleh beberapa orang yang dipilih oleh penulis. Penulis memilih orang-orang yang mendalami 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 aplikasi yang telah dikembangkan. Uji coba yang dilakukan pengguna meliputi pengunggahan BPEL, pencarian *workflow*, pengunduhan *event log*, dan melihat detail *event log*. Rincian isi kuesioner yang diberikan pada pengguna dapat dilihat pada Tabel 6.7.

**Tabel 6.7 Daftar pertanyaan kuesioner**

No.	Pernyataan	Pilihan jawaban			
		Sangat setuju	Setuju	Tidak setuju	Sangat tidak setuju
1	Menurut saya, aplikasi ini mudah digunakan	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	Menurut saya, cukup mudah mengunggah <i>file</i> BPEL	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	Menurut saya, cukup mudah melihat informasi detail dari <i>workflow</i>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	Menurut saya, cukup mudah mengunduh <i>event log</i> dari aplikasi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	Menurut saya, cukup mudah menggunakan fitur pencarian <i>workflow</i>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	Menurut saya, informasi <i>workflow</i> yang ditampilkan cukup mudah dimengerti	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

No.	Pernyataan	Pilihan jawaban			
		Sangat setuju	Setuju	Tidak setuju	Sangat tidak setuju
7	Menurut saya, tampilan aplikasi ini membantu pengguna	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Tabel 6.8 Daftar penguji plugin

No.	Nama	Profesi	Jurusan
1	Aldy Syahdeini	Mahasiswa	Teknik Informatika
2	Qotrun Nada Haroen	Mahasiswa	Teknik Informatika
3	Mohammad Farid Naufal	Mahasiswa	Teknik Informatika
4	Endang Wahyu Pamungkas	Mahasiswa	Teknik Informatika
5	Azi Prastyo	Mahasiswa	Teknik Informatika
6	Bagus Ardiansyah	Mahasiswa	Teknik Informatika
7	Muhammad Farhan	Mahasiswa	Teknik Informatika
8	Bayu Aji Mahendra P.	Mahasiswa	Teknik Informatika
9	Fernandes Sinaga	Mahasiswa	Teknik Informatika
10	I Gusti Fauzi Geri S	Mahasiswa	Teknik Informatika

Tabel 6.9 Hasil kuesioner

No.	Pernyataan	Pilihan jawaban			
		Sangat setuju	Setuju	Tidak setuju	Sangat tidak setuju
1	Menurut saya, aplikasi ini mudah digunakan	40%	60%		
2	Menurut saya, cukup mudah mengunggah <i>file</i> BPEL	40%	60%		
3	Menurut saya, cukup mudah melihat informasi detil dari <i>workflow</i>	30%	60%	10%	

No.	Pernyataan	Pilihan jawaban			
		Sangat setuju	Setuju	Tidak setuju	Sangat tidak setuju
4	Menurut saya, cukup mudah mengunduh <i>event log</i> dari aplikasi	60%	40%		
5	Menurut saya, cukup mudah menggunakan fitur pencarian <i>workflow</i>	60%	40%		
6	Menurut saya, informasi <i>workflow</i> yang ditampilkan cukup mudah dimengerti	40%	60%		
7	Menurut saya, tampilan aplikasi ini membantu pengguna	50%	50%		

Penguji aplikasi WFMS ini adalah orang-orang yang pernah menggunakan aplikasi *ProM* untuk melakukan *process mining*. Daftar penguji *plugin* ini dapat dilihat pada Tabel 6.8. Hasil rekap kuesioner dapat dilihat pada Tabel 6.9. 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 6.10, 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 6.10 Rangkuman hasil pengujian**

<b>ID</b>	<b>Nama</b>	<b>Skenario</b>	<b>Hasil</b>
TA-UJ.UC0001	Pengujian fitur mengunggah BPEL	1	Berhasil
TA-UJ.UC0002	Pengujian fitur untuk melihat informasi <i>workflow</i>	1	Berhasil
TA-UJ.UC0003	Pengujian fitur mencari <i>workflow</i>	1	Berhasil
TA-UJ.UC0004	Pengujian fitur mengunduh <i>event log</i>	1	Berhasil

### **6.3.2. Evaluasi Pengujian Studi Kasus**

#### **6.3.2.1. Evaluasi Studi Kasus *Sale Order* , *Material Management*, *Controlling*, *Financial Account*, *Human Capital Management*, dan *Production Planning***

Dari studi kasus ini, didapat bahwa akurasi dari fitur pencarian *workflow* sangat tinggi. Terbukti dari berbagai percobaan pada berbagai proses bisnis yang memiliki akurasi yang tinggi. Percobaan pada berbagai proses bisnis juga ditujukan untuk mencoba formula nilai similaritas pada keadaan pencarian

yang berbeda-beda. Sehingga semakin meyakinkan bahwa akurasi metode ini tinggi pada berbagai keadaan pencarian.

#### **6.3.2.2. Evaluasi Pengujian Studi Kasus *Workflow* yang Memiliki *Complex Sequence***

Berdasarkan hasil pengujian, WFMS bisa menjalankan *workflow* terlepas dari tipe *workflow* tersebut. Hal ini dibuktikan dengan berhasilnya implementasi *workflow* yang diimplementasikan ke dalam *web service*.

#### **6.3.2.3. Evaluasi Pengujian Kegunaan**

Berdasarkan hasil kuesioner pada Tabel 6.9, dapat ditarik kesimpulan bahwa aplikasi cukup mudah digunakan dan dapat membantu pengguna dalam mengembangkan *workflow*. Hal ini dapat dilihat dari hasil kuisisioner yang 100% menyatakan setuju bahwa aplikasi ini mudah digunakan.

## **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**

Tugas akhir ini diajukan karena dilatar belakangi oleh beberapa masalah yang ada pada pengembangan *workflow*. Beberapa masalah ini terbagi dalam dua jenis masalah yaitu sulitnya melakukan penggunaan kembali *workflow* yang sudah ada atau *workflow reusability* dan sulitnya menganalisis aktivitas dari *workflow* menggunakan teknik *process mining*.

Tugas akhir ini menawarkan sebuah arsitektur WFMS untuk mengatasi kedua jenis masalah di atas. Arsitektur yang diajukan oleh penelitian ini mempunyai tujuan meningkatkan *reusability workflow* dengan metode pencarian *semantic query* dengan pendekatan ontologi OWL-S dan metode *event log generator* menggunakan *workflow engine* Apache ODE.

Permasalahan pertama, yaitu masalah sulitnya melakukan pencarian terhadap *workflow*, diselesaikan dengan menggunakan metode pencarian *workflow* menggunakan standar OWL-S sebagai basis *metadata* dan *semantic similarity* sebagai rumus penghitungan nilai hubungan antara *keyword* dan *web service*. Dengan hanya berbekal *keyword* berupa *service name* dan *description*, pengguna bisa mendapatkan berbagai *workflow* yang diinginkan. Metode ini berhasil membantu pengguna dalam mencari *workflow*, dibuktikan dari hasil pengujian yang memiliki akurasi tinggi yaitu 0,99. Hasil akurasi yang tinggi ini membuktikan bahwa metode ini dapat mencari *workflow* secara

akurat dan membantu proses pencarian *workflow* yang menjadi permasalahan pertama yang melatarbelakangi tugas akhir ini.

Permasalahan kedua, yaitu sulitnya menganalisis aktivitas *workflow*, terjadi akibat tidak adanya WFMS yang menghasilkan *log* berstandar *process mining*. Arsitektur yang diajukan pada tugas akhir ini, memiliki metode untuk menghasilkan *event log* berstandar *process mining*. Dengan menggunakan *workflow engine* dari Apache ODE, metode ini bisa menghasilkan *event log* berstandar *process mining* secara sempurna. Hal ini dibuktikan dari hasil pengujian, yang menunjukkan hasil *event log* yang memiliki atribut sesuai standar *process mining*.

Dengan kata lain, tugas akhir ini telah berhasil memberikan sebuah solusi baru untuk kedua masalah yang melatarbelakangi penelitian tugas akhir ini. Dengan metode pencarian yang akurat, permasalahan kesulitan pencarian *workflow* bisa diselesaikan. Dan dengan metode penghasilan *event log*, masalah sulitnya menganalisis aktivitas *workflow* dengan teknik *process mining* bisa teratasi.

## 7.2. Saran

### 7.2.1. Peningkatan Metode Similaritas

Metode similaritas pada pencarian *workflow* merupakan salah satu hal yang bisa ditingkatkan. Karena atribut yang dilihat hanya terbatas pada nama dan deskripsi dari *workflow*, sehingga penulis berharap ke depannya bisa menambahkan atribut pencarian seperti *rating*, *security*, dan atribut lainnya yang bisa dinilai secara kuantitatif.

## DAFTAR PUSTAKA

- [1] M. Owen and J. Raj, *BPMN and Business Process Management*. Popkin Software, 2003.
- [2] OASIS Web Services *Business Process Execution Language* (WSBPEL) TC. (2007, Apr.) OASIS. [Online]. <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.pdf>
- [3] W. v. d. Aalst, "Service Mining: Using Process Mining to Discover, Check, and Improve Service Behavior," in *IEEE TRANSACTIONS ON SERVICES COMPUTING*, 2011.
- [4] Anne. (2012, Feb.) fluxicon. [Online]. <http://fluxicon.com/blog/2012/02/data-requirements-for-process-mining/>
- [5] W. v. d. Aalst and K. M. v. Hee, *Workflow Management: Models, Methods, and Systems*. Cambridge, MA: MIT Press, 2002.
- [6] Workflow Management Coalition. (1999) Terminology and Glossary (WFMC-TC-1011). [Online]. <http://www.aiai.ed.ac.uk/project/wfmc/ARCHIVE/DOCS/glossary/glossary.html>
- [7] D. Martin, et al. (2004, ) <http://www.w3.org/Submission/OWL-S/>. [Online]. <http://www.w3.org/Submission/OWL-S/>
- [8] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana. (2001, Mar.) W3C. [Online]. <http://www.w3.org/TR/wsd1>
- [9] E. Sirin, M. Dänzer, and T. Möller. (2012, ) OWL-S API. [Online]. <http://on.cs.unibas.ch/owls-api/>
- [10] The SPARQL (1.1) Working Group. (2008) W3C. [Online]. <http://www.w3.org/TR/rdf-sparql-query/>



- [11] M. D. Remco Dijkman, B. v. Dongen, R. Käärik, and J. Mendling, "Similarity of business process models: Metrics and evaluation," *Information Systems*, vol. 36, no. 2, pp. 498-516, 2011.
- [12] W. Gaaloul, S. Bhiri, and C. Godart, "Research challenges and opportunities in web services mining," *Proc of System and Information Service Web*, 2006.
- [13] S. Dustdar and R. Gombotz, "Discovering web service workflows using web services interaction mining," *International Journal of Business Process Integration and Management*, vol. 1, pp. 256-266, Feb. 2007.

## BIODATA PENULIS



Penulis, **Chairaja Almas Djani**, lahir di Semarang, 28 Januari 1993. Penulis menempuh pendidikan dasar mulai kelas 1 sampai 6 di SD 82 Banda Aceh. Untuk pendidikan menengah, penulis tempuh di SMPN 1 Medan dan selanjutnya di SMA Negeri Modal Bangsa Aceh. Selanjutnya penulis melanjutkan pendidikan sarjana di Jurusan Teknik Informatika, Institut Teknologi

Sepuluh Nopember Surabaya. Selama kuliah penulis aktif menjadi administrator Laboratorium Pemrograman Teknik Informatika.

Penulis dalam menyelesaikan pendidikan S1 mengambil bidang minat Rekayasa Perangkat Lunak (*Software Engineering*) dan memiliki ketertarikan di bidang *Software Architecture*, *Workflow Engineering*, dan *Desktop Software Development*. Penulis dapat dihubungi melalui email: [chairaja.almas.djeni@gmail.com](mailto:chairaja.almas.djeni@gmail.com) dan [almas10@mhs.if.its.ac.id](mailto:almas10@mhs.if.its.ac.id)

## LAMPIRAN A. DIAGRAM

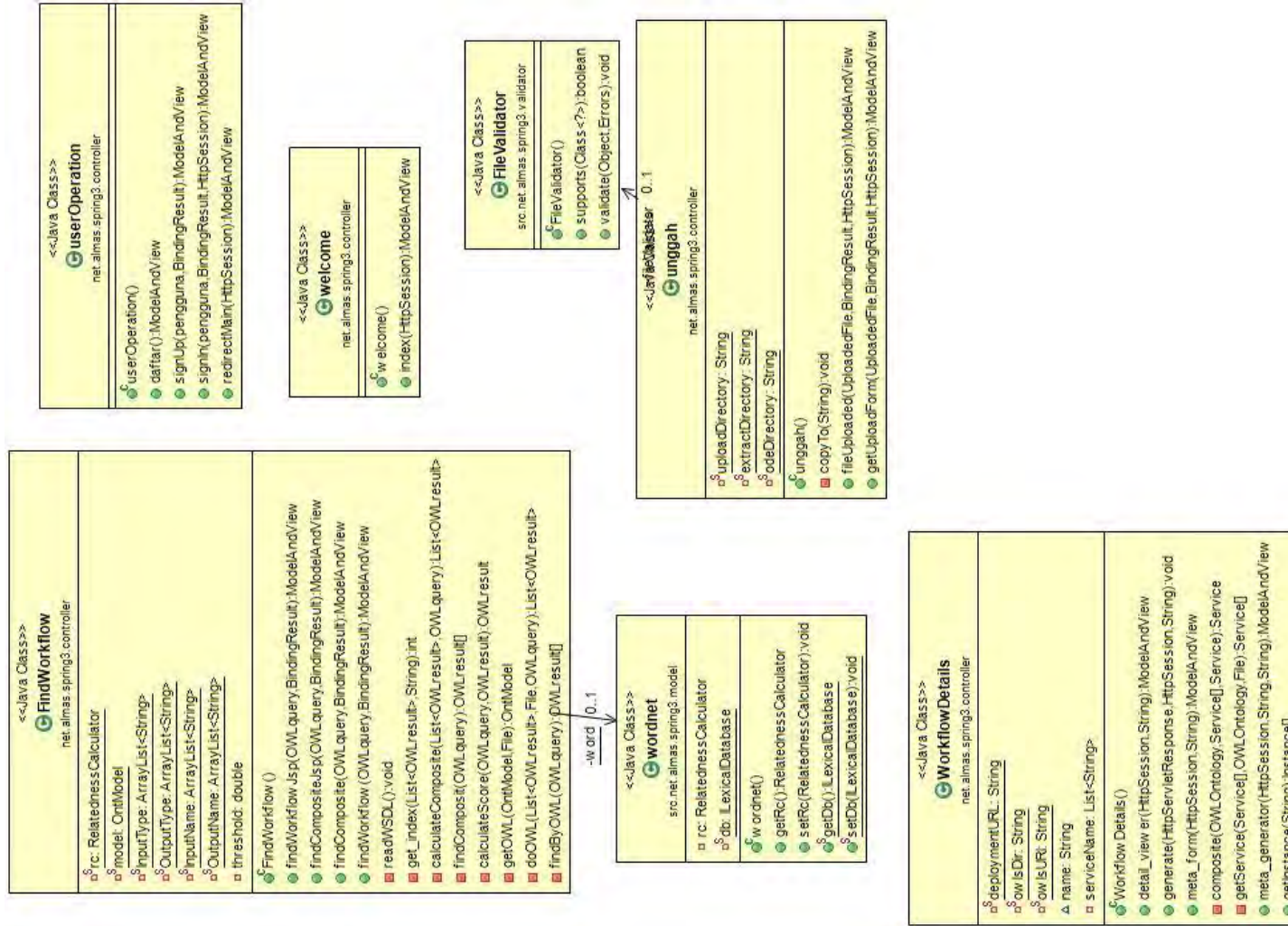


Diagram A.1 Kelas Diagram Kontrol

*[Halaman ini sengaja dikosongkan]*



*[Halaman ini sengaja dikosongkan]*



## LAMPIRAN B. KUISIONER

Nama : Aldy syahdeini  
 NRP : 511100122

No.	Pernyataan	Pilihan jawaban			
		Sangat setuju	Setuju	Tidak setuju	Sangat tidak setuju
1	Menurut saya, aplikasi ini mudah digunakan	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	Menurut saya, cukup mudah mengunggah file BPEL	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	Menurut saya, cukup mudah melihat informasi detail dari <i>workflow</i>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	Menurut saya, cukup mudah mengunduh <i>event log</i> dari aplikasi	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	Menurut saya, cukup mudah menggunakan fitur pencarian <i>workflow</i>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	Menurut saya, informasi <i>workflow</i> yang ditampilkan cukup mudah dimengerti	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	Menurut saya, tampilan aplikasi ini membantu pengguna	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>

Kuisisioner B.1 Kuisisioner Aldy Syahdeini

Qotrun Nada Haroen  
5110100130

No.	Pernyataan	Pilihan jawaban			
		Sangat setuju	Setuju	Tidak setuju	Sangat tidak setuju
1	Menurut saya, aplikasi ini mudah digunakan	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	Menurut saya, cukup mudah mengunggah file BPEL	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	Menurut saya, cukup mudah melihat informasi detail dari workflow	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	Menurut saya, cukup mudah mengunduh event log dari aplikasi	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	Menurut saya, cukup mudah menggunakan fitur pencarian workflow	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	Menurut saya, informasi workflow yang ditampilkan cukup mudah dimengerti	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	Menurut saya, tampilan aplikasi ini membantu pengguna	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Kuisisioner B.2 Kuisisioner Qotrun Nada Haroen



Mohammad Farid Naufal

No.	Pernyataan	Pilihan jawaban			
		Sangat setuju	Setuju	Tidak setuju	Sangat tidak setuju
1	Menurut saya, aplikasi ini mudah digunakan	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	Menurut saya, cukup mudah mengunggah <i>file</i> BPEL	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	Menurut saya, cukup mudah melihat informasi detail dari <i>workflow</i>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	Menurut saya, cukup mudah mengunduh <i>event log</i> dari aplikasi	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	Menurut saya, cukup mudah menggunakan fitur pencarian <i>workflow</i>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	Menurut saya, informasi <i>workflow</i> yang ditampilkan cukup mudah dimengerti	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	Menurut saya, tampilan aplikasi ini membantu pengguna	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**Kuisisioner B.3 Kuisisioner Muhammad Farid Naufal**

No.	Pernyataan	Pilihan jawaban			
		Sangat setuju	Setuju	Tidak setuju	Sangat tidak setuju
1	Menurut saya, aplikasi ini mudah digunakan	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	Menurut saya, cukup mudah mengunggah file BPEL	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	Menurut saya, cukup mudah melihat informasi detil dari <i>workflow</i>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	Menurut saya, cukup mudah mengunduh <i>event log</i> dari aplikasi	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	Menurut saya, cukup mudah menggunakan fitur pencarian <i>workflow</i>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	Menurut saya, informasi <i>workflow</i> yang ditampilkan cukup mudah dimengerti	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	Menurut saya, tampilan aplikasi ini membantu pengguna	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Nama : Endang Wahyu, P.

NRP : 5110100106

**Kuisiener B.4 Kuisiener Endang Wahyu Pamungkas**

Azi Prastyo  
5110100225

No.	Pernyataan	Pilihan jawaban			
		Sangat setuju	Setuju	Tidak setuju	Sangat tidak setuju
1	Menurut saya, aplikasi ini mudah digunakan	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	Menurut saya, cukup mudah mengunggah <i>file</i> BPEL	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	Menurut saya, cukup mudah melihat informasi detail dari <i>workflow</i>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	Menurut saya, cukup mudah mengunduh <i>event log</i> dari aplikasi	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	Menurut saya, cukup mudah menggunakan fitur pencarian <i>workflow</i>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	Menurut saya, informasi <i>workflow</i> yang ditampilkan cukup mudah dimengerti	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	Menurut saya, tampilan aplikasi ini membantu pengguna	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Kuisisioner B.5 Kuisisioner Azi Prastyo

Bagus Ardiansyah

5110100194

No.	Pernyataan	Pilihan jawaban			
		Sangat setuju	Setuju	Tidak setuju	Sangat tidak setuju
1	Menurut saya, aplikasi ini mudah digunakan	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	Menurut saya, cukup mudah mengunggah <i>file</i> BPEL.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	Menurut saya, cukup mudah melihat informasi detail dari <i>workflow</i>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	Menurut saya, cukup mudah mengunduh <i>event log</i> dari aplikasi	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	Menurut saya, cukup mudah menggunakan fitur pencarian <i>workflow</i>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	Menurut saya, informasi <i>workflow</i> yang ditampilkan cukup mudah dimengerti	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	Menurut saya, tampilan aplikasi ini membantu pengguna	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**Kuisiner B.6 Kuisiner Bagus Ardiansyah**

511100109  
M. Farhan, A

No.	Pernyataan	Pilihan jawaban			
		Sangat setuju	Setuju	Tidak setuju	Sangat tidak setuju
1	Menurut saya, aplikasi ini mudah digunakan	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	Menurut saya, cukup mudah mengunggah <i>file</i> BPEL	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	Menurut saya, cukup mudah melihat informasi detail dari <i>workflow</i>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
4	Menurut saya, cukup mudah mengunduh <i>event log</i> dari aplikasi	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	Menurut saya, cukup mudah menggunakan fitur pencarian <i>workflow</i>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	Menurut saya, informasi <i>workflow</i> yang ditampilkan cukup mudah dimengerti	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	Menurut saya, tampilan aplikasi ini membantu pengguna	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**Kuisisioner B.7 Kuisisioner Muhammad Farhan**



BAYU AJI MAHENDRA PUTRA

No.	Pernyataan	Pilihan jawaban			
		Sangat setuju	Setuju	Tidak setuju	Sangat tidak setuju
1	Menurut saya, aplikasi ini mudah digunakan	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	Menurut saya, cukup mudah mengunggah <i>file</i> BPEL	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	Menurut saya, cukup mudah melihat informasi detail dari <i>workflow</i>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	Menurut saya, cukup mudah mengunduh <i>event log</i> dari aplikasi	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	Menurut saya, cukup mudah menggunakan fitur pencarian <i>workflow</i>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	Menurut saya, informasi <i>workflow</i> yang ditampilkan cukup mudah dimengerti	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	Menurut saya, tampilan aplikasi ini membantu pengguna	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>

**Kuisisioner B.8 Kuisisioner Bayu Aji Mahendra Putra**

Nama : Fernandes Siraga

No.	Pernyataan	Pilihan jawaban			
		Sangat setuju	Setuju	Tidak setuju	Sangat tidak setuju
1	Menurut saya, aplikasi ini mudah digunakan	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	Menurut saya, cukup mudah mengunggah <i>file</i> BPEL	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	Menurut saya, cukup mudah melihat informasi detail dari <i>workflow</i>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	Menurut saya, cukup mudah mengunduh <i>event log</i> dari aplikasi	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	Menurut saya, cukup mudah menggunakan fitur pencarian <i>workflow</i>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	Menurut saya, informasi <i>workflow</i> yang ditampilkan cukup mudah dimengerti	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	Menurut saya, tampilan aplikasi ini membantu pengguna	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Kuisiner B.9 Kuisiner Fernandes Sinaga

I Gusti Fauzi Geri S. 5110100078

No.	Pernyataan	Pilihan jawaban			
		Sangat setuju	Setuju	Tidak setuju	Sangat tidak setuju
1	Menurut saya, aplikasi ini mudah digunakan	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	Menurut saya, cukup mudah mengunggah <i>file</i> BPEL	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	Menurut saya, cukup mudah melihat informasi detail dari <i>workflow</i>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	Menurut saya, cukup mudah mengunduh <i>event log</i> dari aplikasi	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	Menurut saya, cukup mudah menggunakan fitur pencarian <i>workflow</i>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	Menurut saya, informasi <i>workflow</i> yang ditampilkan cukup mudah dimengerti	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	Menurut saya, tampilan aplikasi ini membantu pengguna	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Kuisiонер B.10 Kuisiонер I Gusti Fauzi Geri



## LAMPIRAN C. DETAIL PERCOBAAN

**Tabel C.1 Hasil percobaan pada aktivitas *presale activities***

<i>Threshold</i>	TP	FP	TN	FN	<i>Accuracy</i>
0,1	30	210	0	0	0,125
0,2	30	210	0	0	0,125
0,3	30	138	72	0	0,425
0,4	30	12	198	0	0,95
0,5	30	6	204	0	0,975
0,6	30	0	210	0	1
0,7	30	0	210	0	1
0,8	24	0	210	6	0,975
0,9	24	0	210	6	0,975

**Tabel C.2 Hasil percobaan pada aktivitas *sale order***

<i>Threshold</i>	TP	FP	TN	FN	<i>Accuracy</i>
0,1	30	210	0	0	0,125
0,2	30	210	0	0	0,125
0,3	30	204	6	0	0,15
0,4	30	77	133	0	0,679166667
0,5	30	6	204	0	0,975
0,6	30	0	210	0	1
0,7	30	0	210	0	1
0,8	24	0	210	6	0,975
0,9	21	0	210	9	0,9625

**Tabel C.3 Hasil percobaan pada aktivitas *check availability***

<i>Threshold</i>	TP	FP	TN	FN	<i>Accuracy</i>
0,1	30	210	0	0	0,125
0,2	30	210	0	0	0,125
0,3	30	162	48	0	0,325
0,4	30	61	149	0	0,74583333
0,5	30	6	204	0	0,975
0,6	30	0	210	0	1
0,7	30	0	210	0	1
0,8	24	0	210	6	0,975
0,9	6	0	210	24	0,9

**Tabel C.4 Hasil percobaan pada aktivitas *pick material***

<i>Threshold</i>	TP	FP	TN	FN	<i>Accuracy</i>
0,1	30	210	0	0	0,125
0,2	30	210	0	0	0,125
0,3	30	204	6	0	0,15
0,4	30	48	162	0	0,8
0,5	30	18	192	0	0,925
0,6	30	0	210	0	1
0,7	24	0	210	6	0,975
0,8	23	0	210	7	0,97083333
0,9	12	0	210	18	0,925