

**APLIKASI *WORKFLOW REPOSITORY* PADA *ENTERPRISE RESOURCE PLANNING* PONDOK PESANTREN
MENGUNAKAN *RESOURCE DESCRIPTION FRAMEWORK***

SKRIPSI

Oleh:

RETNO ASTRININDITYA

NIM. 13650087



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2018**

**APLIKASI WORKFLOW REPOSITORY PADA ENTERPRISE
RESOURCE PLANNING PONDOK PESANTREN
MENGUNAKAN RESOURCE
DESCRIPTION FRAMEWORK**

SKRIPSI

Diajukan kepada:

Universitas Islam Negeri Maulana Malik Ibrahim Malang

**Untuk memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)**

Oleh :

**RETNO ASTRININDITYA
NIM. 13650087**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2018**

LEMBAR PERSETUJUAN

APLIKASI *WORKFLOW REPOSITORY* PADA *ENTERPRISE RESOURCE PLANNING* PONDOK PESANTREN MENGGUNAKAN *RESOURCE DESCRIPTION FRAMEWORK*

SKRIPSI

Oleh :

RETNO ASTRININDITYA

NIM. 13650087

Telah Diperiksa dan Disetujui untuk Diuji
Tanggal: 20 Desember 2017

Dosen Pembimbing I,



M. Ainul Yaqin, M.Kom
NIP.19761013 200604 1 004

Dosen Pembimbing II,



Syahiduz Zaman, M.Kom
NIP. 19700502 200501 1 005

Mengetahui,

Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi

Universitas Islam Negeri Maulana Malik Ibrahim Malang



Dr. Cahya Crys dian

NIP. 19740424 200901 1 008

LEMBAR PENGESAHAN

APLIKASI *WORKFLOW REPOSITORY* PADA *ENTERPRISE RESOURCE PLANNING* PONDOK PESANTREN MENGGUNAKAN *RESOURCE DESCRIPTION FRAMEWORK*

SKRIPSI

Oleh:

RETNO ASTRININDITYA
NIM. 13650087

Telah Dipertahankan di Depan Dewan Penguji Skripsi dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 29 Desember 2017

Susunan Dewan Penguji		Tanda Tangan
Penguji Utama	<u>Ajib Hanani, M.T</u> NIDT. 19840731 20160801 1 076	()
Ketua Penguji	<u>Dr. Suhartono, M.Kom</u> NIP. 19680519 200312 1 001	()
Sekretaris Penguji	<u>M. Ainul Yaqin, M.Kom</u> NIP. 19761013 200604 1 004	()
Anggota Penguji	<u>Syahiduz Zaman, M.Kom</u> NIP. 19700502 200501 1 005	()

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang




Dr. Cahyo Crysdiyan
NIP. 19740424 200901 1 008

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini:

Nama : Retno Astrininditya

NIM : 13650087

Jurusan : Teknik Informatika

Fakultas : Sains dan Teknologi

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 20 Desember 2017

Yang membuat pernyataan



Retno Astrininditya

NIM. 13650087

MOTTO

**“ Waktu itu bagaikan pedang, jika kamu tidak memanfaatkannya
menggunakan untuk memotong, ia akan memotongmu
(menggilasmu)”
(H.R. Muslim)**



HALAMAN PERSEMBAHAN

الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ

Dengan mengucapkan syukur Alhamdulillah kehadiran Allah SWT yang telah melimpahkan Rahmat serta Hidayah-Nya, karya sederhana ini ku persembahkan untuk orang-orang yang selalu memberi energi positif dalam hidupku.

Ibuku, Yuni Sulistyowati Arief dan Ayahku, Muhartoyo. Terima kasih telah membesarkan, mendidik, serta mencurahkan kasih sayang yang berlimpah serta dukungan padaku. Selalu dan selalu menyertakan doa terbaiknya untuk seorang anak yang akan terus berusaha membahagiakan kalian.

Kakakku, Dian Anugerah Dewantoro dan Adikku Nabilah Rizky Amalia. Terimakasih telah menjadi tawa dan senyum dalam hidupku. Kasih sayang kalian akan selalu menjadi kobar semangatku.

Bapak Ibu Dosen Teknik Informatika, terutama Bapak M. Ainul Yaqin dan Bapak Syahiduz Zaman. Terimakasih atas kesabaran dan keikhlasan dalam mendidik saya, semoga ilmu yang beliau amalkan akan bermanfaat bagi seluruh mahasiswa.

Terimakasih kepada seluruh sahabat, PEPIPO (Fian, Imamah, Rina, Arif, Andra) yang selalu ada dalam suka dan duka, Ridho yang senantiasa memberi semangat dan nasehat, serta Nandya yang tidak pernah lelah memberi dukungan. Dan seluruh teman-teman Teknik Informatika UIN Maliki Malang angkatan 2013 atas masa yang sangat berharga selama ini.

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Segala puji bagi Allah SWT tuhan semesta alam, karena atas segala rahmat dan karunia-Nya sehingga penulis mampu menyelesaikan skripsi dengan baik dan lancar. Shalawat serta salam selalu tercurah kepada tauladan terbaik Nabi Muhammad SAW yang telah membimbing umatnya dari zaman kebodohan menuju islam yang *rahmatan lil alamiin*.

Dalam menyelesaikan skripsi ini, banyak pihak yang telah memberikan bantuan baik secara moril, nasihat dan semangat maupun materiil. Atas segala bantuan yang telah diberikan, penulis ingin menyampaikan doa dan ucapan terimakasih yang sedalam-dalamnya kepada:

1. Bapak Dr. Cahyo Crysdiyan, selaku Ketua Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang
2. Bapak M. Ainul Yaqin, M.Kom, selaku dosen pembimbing I yang telah meluangkan waktu untuk membimbing, memotivasi, dan mengarahkan dan memberi masukan kepada penulis dalam pengerjaan skripsi ini hingga akhir.
3. Bapak H. Syahiduz Zaman, M.Kom, selaku dosen pembimbing II yang senantiasa memberi masukan dan nasihat serta petunjuk dalam penyusunan skripsi ini.

4. Ayah, Ibu, Kakak, Adik serta keluarga besar tercinta yang selalu memberi dukungan yang tak terhingga serta doa yang senantiasa mengiringi setiap langkah penulis.
5. Segenap Dosen Teknik Informatika yang telah memberikan bimbingan keilmuan kepada penulis selama masa studi.
6. Teman-teman Pepipo dan Skripsi Sukses yang telah berjuang bersama dan memberi bantuan.
7. Teman – teman seperjuangan Teknik Informatika angkatan 2013.

Berbagai kekurangan dan kesalahan mungkin pembaca temukan dalam penulisan skripsi ini, untuk itu penulis menerima segala kritik dan saran yang membangun dari pembaca sekalian. Semoga apa yang menjadi kekurangan bisa disempurnakan oleh peneliti selanjutnya dan semoga karya ini senantiasa dapat memberi manfaat. Aamiin. *Wassalamualaikum Wr.Wb*

Malang, 20 Desember 2017

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGAJUAN.....	ii
LEMBAR PERSETUJUAN.....	iii
LEMBAR PENGESAHAN	iv
PERNYATAAN KEASLIAN TULISAN	v
MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR	xii
DAFTAR TABEL.....	xiv
ABSTRAK	xv
ABSTRACT.....	xvi
ملخص البحث.....	xvii
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Tujuan Penelitian.....	3
1.4. Manfaat Penelitian.....	3
1.5. Batasan Masalah.....	4
1.6. Sistematika Penulisan.....	4
BAB 2 KAJIAN PUSTAKA	6
2.1. Business Process Management Suite (BPMS)	6
2.2. Workflow Management System (WfMS)	8
2.3. Workflow Repository	9
2.4. BPMN.....	11
2.5. XML Process Definition Language (XPDL).....	17
2.6. Metadata	21
2.7. Web Semantik	22
2.8. <i>Resource Description Framework</i>	25
2.9. RDF API for PHP	27

BAB 3	METODOLOGI PENELITIAN	29
3.1.	Desain Penelitian	29
3.1.1	Gambaran Umum Sistem	29
3.1.2	Sumber Data	29
3.2.	Rancangan Sistem	30
3.2.1	Input Workflow	31
3.2.2	Penyimpanan workflow pada workflow repository	32
3.2.3	Query data	34
3.2.4	Parsing	35
3.2.5	Penulisan sintak RDF	39
3.2.6	SPARQL	42
3.2.7	Interpretasi	44
3.3.	Pengujian Sistem	44
BAB 4	HASIL DAN PEMBAHASAN	47
4.1.	Workflow sebagai data uji	47
4.2.	Langkah Uji Coba	51
4.3.	Hasil Uji Coba	60
4.4.	Pembahasan	63
BAB 5	KESIMPULAN DAN SARAN	70
5.1.	Kesimpulan	70
5.2.	Saran	71
DAFTAR PUSTAKA	72
LAMPIRAN	75

DAFTAR GAMBAR

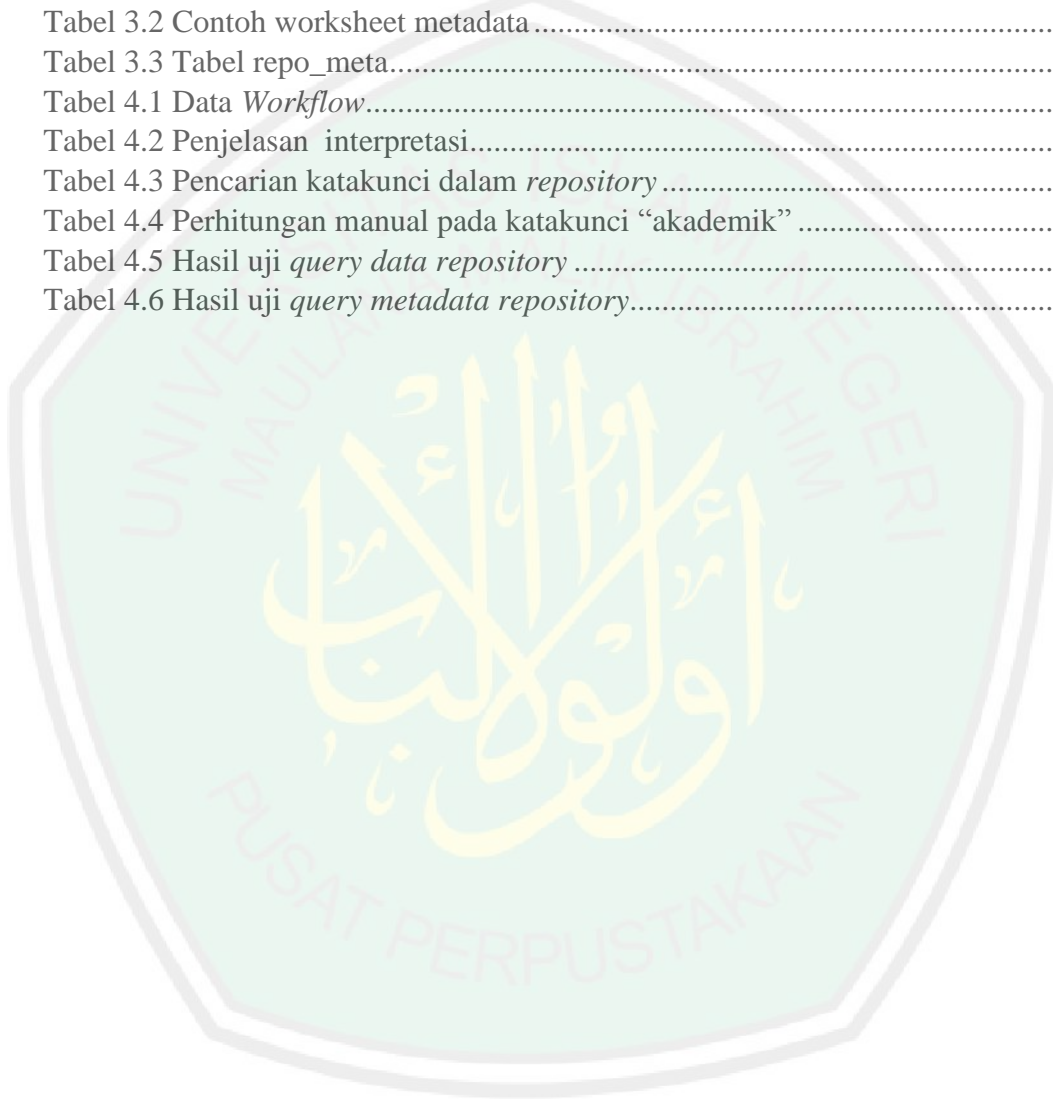
Gambar 2.1 Arsitektur WfMS.....	9
Gambar 2.2 Notasi <i>Event</i>	13
Gambar 2.3 Notasi <i>Activity</i>	14
Gambar 2.4 Notasi <i>Gateway</i>	14
Gambar 2.5 Notasi <i>sequence flow</i>	15
Gambar 2.6 Notasi <i>message flow</i>	15
Gambar 2.7 Notasi <i>association</i>	15
Gambar 2.8 Notasi data.....	16
Gambar 2.9 Notasi artifak.....	17
Gambar 2.10 Notasi <i>Swimlannes</i>	17
Gambar 2.11 Pemodelan BPMN.....	20
Gambar 2.12 Kode XPDL dari BPMN	21
Gambar 2.13 <i>Semantic Web Layer Cake</i>	22
Gambar 2.14 Contoh format RDF.....	24
Gambar 2.15 Penggambaran RDF	27
Gambar 3.1 Rancangan Sistem	31
Gambar 3.2 Contoh dokumen *.xpdL	32
Gambar 3.3 <i>Entity Relational Diagram</i>	32
Gambar 3.4 Form input data <i>workflow</i>	33
Gambar 3.5 Kode Sumber Simpan Data <i>Workflow</i>	33
Gambar 3.6 Kode sumber Query Data.....	34
Gambar 3.7 Hasil <i>query</i> data.....	35
Gambar 3.8 Alur parsing element	37
Gambar 3.9 Kode Sumber <i>Parsing Metadata Workflow</i>	38
Gambar 3.10 Penyusunan skema	39
Gambar 3.11 Contoh format RDF.....	40
Gambar 3.12 Sumber Kode Mengambil Data dari MySQL ke Array	40
Gambar 3.13 Sumber Kode Konversi Array ke XML	42
Gambar 3.14 Sumber Kode Menyimpan Data ke Sebuah File RDF	42
Gambar 3.15 Kode sumber SPARQL	43
Gambar 3.16 Contoh interpretasi ontologi.....	44
Gambar 4.1 Pemodelan <i>workflow</i> BPMN.....	48
Gambar 4.2 Dokumen XPDL.....	49
Gambar 4.3 Halaman awal <i>Workflow Repository</i>	51
Gambar 4.4 Form Data <i>Workflow</i>	52
Gambar 4.5 Tabel Data <i>Workflow</i>	52
Gambar 4.6 Dokumen RDF yang dihasilkan	53
Gambar 4.7 Validasi RDF Sukses.....	54
Gambar 4.8 Hasil triples data model RDF	55
Gambar 4.9 Hasil graph data model RDF.....	56
Gambar 4.10 Tabel metadata <i>workflow</i>	58

Gambar 4.11 Query kata “akademik” pada <i>repository</i> data	59
Gambar 4.12 Query kata “akademik” pada <i>repository</i> data	59
Gambar 4.13 Grafik perbandingan nilai <i>precision</i>	63
Gambar 4.14 Grafik perbandingan nilai <i>recall</i>	64



DAFTAR TABEL

Tabel 2.1 Deskripsi <i>Definition Header</i>	19
Tabel 2.2 Deskripsi <i>Redefinable Header</i>	20
Tabel 3.1 Tabel <i>repo_data</i>	32
Tabel 3.2 Contoh worksheet metadata	36
Tabel 3.3 Tabel <i>repo_meta</i>	36
Tabel 4.1 Data <i>Workflow</i>	50
Tabel 4.2 Penjelasan interpretasi.....	57
Tabel 4.3 Pencarian katakunci dalam <i>repository</i>	60
Tabel 4.4 Perhitungan manual pada katakunci “akademik”	60
Tabel 4.5 Hasil uji <i>query data repository</i>	61
Tabel 4.6 Hasil uji <i>query metadata repository</i>	62



ABSTRAK

Astrininditya, Retno. 2017. **Aplikasi *Workflow Repository* pada *Enterprise Resource Planning* Pondok Pesantren menggunakan *Resource Description Framework***. Skripsi. Jurusan Teknik Informatika. Fakultas Sains dan Teknologi. Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Pembimbing: (I) M. Ainul Yaqin, M.Kom, (II) Syahiduz Zaman, M.Kom

Kata Kunci: *Workflow repository*, XPDL, data, metadata, RDF

Kuantitas *workflow* yang ada dalam perangkat lunak perusahaan besar seperti dalam *Enterprise Resource Planning* (ERP) tentu sangat banyak jumlah dan jenisnya. *Workflow* dibutuhkan untuk menjadi acuan aktivitas setiap *tenant* yang bergabung dalam BPMS. Kebutuhan *workflow* dari setiap *tenant* BPMS cukup beragam namun tidak jarang antara satu sama lain memiliki kemiripan, oleh karena itu dibutuhkan penyimpanan *workflow* yang dapat mengakomodasi kebutuhan untuk para *tenant*. Sebagian besar *Workflow Management System* (WfMS) yang tersedia memiliki penyimpanan yang belum teratur, sehingga hasil *workflow discovery* mempunyai *precision*, *recall*, *accuracy* yang rendah. Untuk menangani masalah tersebut, maka dibangun *workflow repository* yang mempunyai dua layer. Layer pertama menyimpan data *workflow*, layer yang kedua digunakan untuk menyimpan metadata *workflow*. Metadata *repository* dapat menampung data dari suatu data *repository*. Metadata yang dibangun disimpan dalam bentuk *Resource Description Framework* (RDF).

Penelitian ini menyimpan *workflow* dalam dua lapis penyimpanan, yakni data *repository* dan metadata *repository*. Untuk membandingkan kedua *repository* tersebut maka dilakukan beberapa *query* menggunakan SQL dan SPARQL. Berdasarkan hasil uji coba yang dilakukan, *query* SPARQL dalam metadata *repository* mampu memberi hasil yang lebih baik daripada data *repository* yaitu hasil *precision* dari 35% menjadi 87,7%, hasil *recall* dari 35% menjadi 92,1%.

ABSTRACT

Astrininditya, Retno. 2017. *Workflow Repository Application on Enterprise Resource Planning Pondok Pesantren using Resource Description Framework*. Minithesis. Informatics Engineering Department. Faculty of Science and Technology. State Islamic University Maulana Malik Ibrahim Malang.

Advisors: (I) M. Ainul Yaqin, M.Kom, (II) Syahiduz Zaman, M.Kom

Keywords: Workflow repository, XPDL, data, metadata, RDF

The quantity of workflows that exist in large enterprise software such as in the Enterprise Resource Planning (ERP) is certainly much the numbers and types. Workflow is needed to become a reference activity of every tenant who joined in the BPMS. The workflow requirements of each BPMS tenant are quite diverse but not infrequently that between one and other has similarities, therefore it takes workflow storage that can accommodate the needed of the tenants. Most of the available Workflow Management System (WfMS) have irregular storage, so workflow discovery results in low precision, recall, and accuracy. To handle the problem, then built a workflow repository that has two layers. The first layer stores the workflow data, the second layer is used to store the workflow metadata. Metadata repository can hold data from a data repository. The built metadata is stored in the Resource Description Framework (RDF) format.

This research stores workflows in two layers of storage, that is data repository and metadata repository. To compare both of that repositories, researcher tries to query using SQL and SPARQL. Based on the experimental results, SPARQL query in metadata repository had better result than data repository which is precision results from 35% to 87,7%, and the recall from 35% to 92,1%.

ملخص البحث

أسترينديتيا، ريتنو. 2017. تطبيق وركفلو ريفوسيطاري (*Workflow Repository*) على انتيرفرس رصوج فلانيع (*Enterprise Resource Planning*) معهد المؤسسة يستخدم رصوج دسجرفسون فريموروك (*Resource Description Framework*). أطروحة قسم المعلوماتية. كلية العلوم والتكنولوجيا. جامعة الدولة الإسلامية مولانا مالك إبراهيم مالانج.

المستشار: (1) محمد عين اليقين، الماجستير، (2) شاهد الزمان، الماجستير

الكلمات الرئيسية: مستودع سير العمل، كدل، البيانات، البيانات الوصفية، ردف

كمية وركفلو الموجودة في الشركات الكبيرة مثل البرمجيات في انتيرفرس رصوج فلانيع (ERP) التأكيد كثيرا في العدد والنوع. يلزم وركفلو مرجع نشاط سير العمل في كل المستأجر الذي انضم إلى BPMS. احتياجات وركفلو لكل المستأجرين BPMS متنوعة جداً ولكن ليس من غير المألوف بين بعضها البعض بالتشابه، ولذلك التخزين وركفلو المطلوبة التي يمكن أن تستوعب احتياجات المستأجرين. لدى معظم "نظام إدارة سير العمل" (*Workflow Management System*) (WfMS) المتوفرة في المخزن لم يكن بانتظام، حتى اكتشاف وركفلو دسقفري له نتيجة الدقة (*Precision*)، استدعاء (*recall*)، أذكر (*accuracy*) منخفضة. للتعامل مع هذه المشاكل، ثم تبنء مستودع وركفلو الذي يحتوي على طبقتين. طبقة أول طبقة ميبمبان بيانات وركفلو، الطبقة الثانية تستخدم لتخزين بيانات تعريف وركفلو. مستودع البيانات الوصفية يمكن أن تأخذ البيانات من مستودع بيانات. بيانات التعريف المخزنة في النموذج بنيت في رصوج دسجرفسون فريموروك.

يوفر وركفلو هذا البحث في طبقتين، أي مستودع (*Repository*) تخزين البيانات ومستودع (*Repository*) البيانات الوصفية. لمقارنة كل المستودع (*Repository*) ثم تنفيذ عدة استعلامات باستخدام سقل (SQL) و سباركل (SPARQL). استناداً إلى نتائج التجارب التي أجريت في سباركل (SPARQL)، الاستعلام مستودع (*Repository*) البيانات الوصفية قادرة على إعطاء نتائج أفضل من مستودع (*Repository*) البيانات هو النتائج الدقة (*precision*) من 35% إلى 87.7%، نتائج استدعاء (*recall*) 35% و 92.1% في المائة.

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Selama tahun 1990-an, sistem *Enterprise Resource Planning* (ERP) diperkenalkan sebagai “*integrated suites*” yang mengotomatisasi inti kegiatan perusahaan seperti keuangan, sumber daya manusia, manufaktur, pasokan, dan distribusi. Sistem ERP membantu perusahaan berbagi data yang bersifat umum dan berjalan pada seluruh perusahaan yang memungkinkan akses informasi dalam suatu lingkungan secara *real-time*. Solusi ERP dirancang untuk memecahkan fragmentasi informasi dalam bisnis skala besar dan mengintergrasikan semua informasi yang mengalir dalam suatu perusahaan [1].

Daya dari suatu sistem ERP dapat dilihat dari banyaknya *workflow* yang disediakan. Perusahaan yang bergabung dalam ERP tentu menginginkan jika suatu saat *workflow* yang telah dibuatnya dapat bertransformasi sesuai perkembangan perusahaan. Menurut Atmagi *et al* [2] *workflow* merupakan hal yang paling menarik pada perkembangan teknologi SOA. Dengan menggunakan *workflow*, pengguna dapat melakukan intervensi/perubahan alur kerja sistem secara komposisi/orkestrasi. Akan tetapi kuantitas *workflow* dalam perangkat lunak yang digunakan perusahaan adalah besar. Setiap modul mungkin memiliki *workflow* yang mirip/variasi dari *workflow* yang lain. Untuk terus mendukung pemeliharaan dan perbaikan secara berkelanjutan disarankan menggunakan *workflow repository*. Selain hanya untuk disimpan, *workflow* yang sudah ada pada *repository* dapat digunakan kembali (*reuse*) untuk digunakan dalam pembuatan *workflow* baru.

Workflow yang sudah dibuat nantinya dapat dipakai dalam pembuatan BPEL (*Business Process Execution Language*) pada *tenant* yang lain.

Setiap data memiliki metadata, begitu juga dengan *workflow*. Metadata merupakan informasi dari sebuah data. Dalam *booklet* yang diterbitkan NISO Press [3] menjelaskan bahwa alasan penting dalam pembuatan metadata deskriptif adalah untuk memfasilitasi pencarian dari informasi yang relevan. Selain digunakan untuk pencarian, metadata juga dapat membantu mengatur elektronik *resource*, memfasilitasi interoperabilitas serta integrasi penyaluran *resource*, menyediakan identifikasi digital dan mendukung pengarsipan atau pemeliharaan. Bahasa yang merepresentasikan metadata tentang *web resource* dikenal sebagai RDF (*Resource Description Framework*). Selama ini *user* mengalami kesulitan dalam mencari *workflow* yang sesuai dengan kebutuhan dan penyimpanannya belum teratur. *Workflow repository* yang mengandung metadata dapat memberi informasi suatu data yang diinginkan *user*.

RDF mengandung sintaks XML, ia juga memperluas struktur yang menghubungkan dari web menggunakan URI untuk menjadi penamaan dalam hubungan antara dua atau tiga ujung link. Struktur RDF juga dapat digambarkan dalam bentuk skema (RDFS). RDFS merupakan bahasa yang digunakan untuk menulis ontologi. Ontologi adalah model atau bagian yang relevan dari dunia, berisi daftar tipe dari sebuah objek, relasi yang dapat mengkoneksikan satu sama lain, dan cara mengatasi bagaimana objek dan relasi dapat dikombinasikan. Dalam Katsumi [4] menerangkan bahwa para pendukung ontologi sering mengklaim jika salah satu manfaat ontologi adalah dapat digunakan kembali (*reusable*). Prinsip tersebut

memang benar adanya, mengingat bahwa ada banyak konsep generik cenderung muncul dan kemudian digunakan kembali.

Berdasar latar belakang yang telah diuraikan, perlu dilakukan penelitian untuk mengukur pengaruh penerapan ontologi maka pencarian *workflow* dalam *workflow repository* menjadi lebih optimal.

1.2. Rumusan Masalah

Dari latar belakang yang telah dipaparkan diatas dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimana menyimpan *workflow* dalam *repository*?
2. Bagaimana menerapkan RDF untuk menghasilkan metadata?

1.3. Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

1. Menyediakan *workflow repository* yang berguna untuk ERP pondok pesantren.
2. Membuat *metadata workflow repository* dengan ontologi RDF

1.4. Manfaat Penelitian

Manfaat yang ditawarkan dalam penelitian ini adalah:

1. Mengurangi biaya pengembangan modul-modul baru
2. Mengurangi waktu penemuan *workflow* dalam *repository*.

1.5. Batasan Masalah

Untuk menghindari meluasnya permasalahan yang ada, serta keterbatasan ilmu dan kemampuan yang dimiliki peneliti maka batasan masalah pada penelitian ini adalah:

1. Data uji menggunakan format *XML Process Definiton Language* (XPDL).
2. Data yang akan diontologikan menggunakan ontologi RDF.
3. Pemodelan *workflow* menggunakan Bizagi Process Modeller.
4. Data uji penelitian berasal dari beberapa pondok pesantren yakni:
 - a. Pondok Pesantren Annur 1 Bululawang Kab. Malang
 - b. Pondok Pesantren Tahfidz Ar Rohmah 2 Batu
 - c. Pondok Pesantren Modern Ar Rifai Malang
 - d. Pondok Pesantren Mahasiswa Luhur Malang
 - e. Pondok Pesantren Salaf Anwarul Huda Malang

1.6. Sistematika Penulisan

Adapun sistematika pada skripsi ini adalah sebagai berikut:

BAB 1 PENDAHULUAN

Bab ini berisi tentang latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika penulisan.

BAB 2 KAJIAN PUSTAKA

Bab ini berisi tentang teori acuan yang berkaitan dengan permasalahan yang dibahas serta teori acuan dalam analisis dan pembuatan pemecahan permasalahan

yang dibahas. Teori acuan tersebut yaitu BPMS, WFMS, *workflow repository*, BPMN, XPD, metadata, web semantik, RDF, RAP.

BAB 3 METODOLOGI PENELITIAN

Bab ini berisi tentang metode yang dipakai dalam penelitian *workflow repository* pada ERP pondok pesantren menggunakan RDF. Tahapan dari penelitian ini dimulai dari input data *workflow*, *parsing*, *generate* RDF hingga metode pengujian sistem.

BAB 4 HASIL DAN PEMBAHASAN

Bab ini menjelaskan tentang hasil dan pembahasan dari sistem yang telah dibangun berdasar dari skenario pengujian. Pengujian sistem dilakukan dengan memasukkan model proses bisnis ERP Pondok Pesantren ke dalam *repository* data sebagai data uji, kemudian melakukan *parsing*, lalu menyimpannya ke dalam *repository* metadata, mencari proses bisnis yang diinginkan dengan *query* hingga dapat diperoleh informasinya dan diunduh. Selanjutnya akan dipaparkan hasil pengujian serta pembahasan dan evaluasi dari kinerja sistem pada penelitian ini.

BAB 5 KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan dan saran yang diharapkan dapat bermanfaat dalam pengembangan program aplikasi.

BAB 2

KAJIAN PUSTAKA

2.1. Business Process Management Suite (BPMS)

Business Process Management Suite adalah *software* yang dapat membantu manajerial memberikan perbaikan terus menerus untuk proses bisnis dengan menyediakan proses penemuan dan alat pemodelan, mesin *business rules*, mesin *workflow*, simulasi dan alat pengujian. Proses bisnis profesional, analis bisnis, *developer* dan semua pekerja IT dapat memanfaatkan BPMS secara efektif untuk meningkatkan produktivitas organisasi, efisiensi, efektivitas dan profitabilitas. Menurut Chang [5] BPMS mampu mengintegrasikan manusia, sistem, dan data organisasi yang memiliki kemampuan sebagai berikut:

1. Pendekatan dalam merancang solusi bisnis proses.
2. Kemampuan untuk mengintegrasikan orang dan sistem yang berpartisipasi dalam proses bisnis.
3. Kemampuan untuk mensimulasikan proses bisnis.

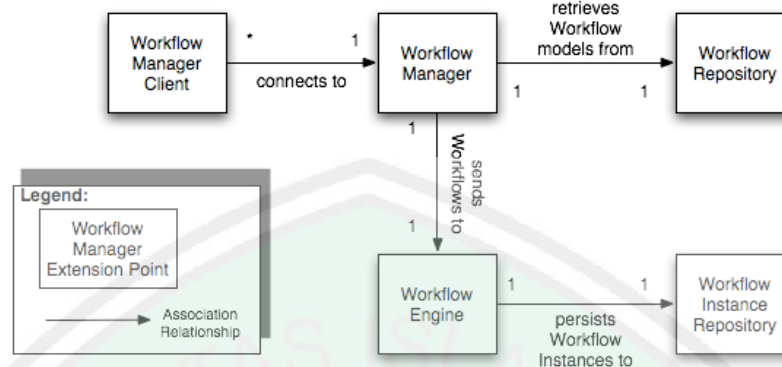
Susilo [6] menyatakan bahwa sudah dilengkapi dengan notasi-notasi BPMN versi 2.0, perlengkapan untuk membuat form dan tools untuk mendesain database, tinggal drag and drop saja. Kedua BPMS portal, yakni bagian dimana kita bisa mengakses aplikasi hasil rancangan kita. Keseluruhan aplikasi dijalankan dari dan ditampilkan di BPMS portal.

Perbedaan antara Pemrograman Biasa dan BPMS	
Pemrograman Biasa	BPMS
<p>1. Desain sistem</p> <ul style="list-style-type: none"> • Survey dan diskusi untuk mengumpulkan informasi tentang aplikasi seperti apa yang dibutuhkan dan diinginkan oleh user. • Desain flowchart, database harus sudah matang. Karena menjadi acuan untuk peng-coding-an. Jika desain salah, maka semua yang dilakukan akan sia-sia. • Rawan terhadap proses-proses penting yang belum terpikirkan. • Apabila sistem yang dibangun adalah sistem yang terintegrasi, maka kesalahan satu bagian akan berpengaruh pada bagian lainnya. 	<ul style="list-style-type: none"> • Survey dan diskusi dengan user tetap harus dilakukan. • Desain alur proses bisa langsung dibuat, tanpa harus menunggu selesai. • Tak perlu takut salah, karena desain bisa langsung dieksekusi. Kita bisa langsung simulasikan apakah sistem sudah berjalan sesuai dengan keinginan user atau belum. • Karena berbasis proses, maka apabila ada tambahan proses-proses yang belum sempat terpikirkan, bisa langsung ditambahkan.
<p>2. Peng-coding-an dan Pengujian Sistem</p> <p>Agar menjadi aplikasi, logika proses harus dituliskan dalam script kode pemrograman. Setelah proses coding selesai, aplikasi harus diuji kestabilan dengan menghilangkan bug-bug yang ada.</p>	<p>Di BPMS tidak perlu ada penulisan script kode pemrograman. Model proses yang telah digambarkan bisa langsung dieksekusi menjadi aplikasi sistem informasi.</p>
<p>3. Implementasi</p> <p>Aplikasi sistem informasi harus benar-benar sudah fix sebelum dipakai untuk mendukung aktivitas bisnis.</p>	<p>Tak perlu menunggu fix, aplikasi sudah bisa dijalankan. Sembari terus disesuaikan dengan perkembangan proses bisnis yang ada.</p>
<p>4. Evaluasi dan Perbaikan Proses Bisnis</p> <p>Untuk melakukan perbaikan aplikasi, user sangat tergantung pada programmer/vendor sebelumnya. Walaupun source code diberikan, programmer/ vendor yang baru belum tentu bisa mengubah aplikasi. Sehingga kalau ingin perbaikan atau penambahan pada proses bisnis harus menggunakan programmer lama atau membangun sistem lagi dari awal.</p>	<ul style="list-style-type: none"> • Semua proses bisnis yang terjadi bisa ditampilkan step by step, sehingga bisa dilakukan evaluasi setiap saat. Sistem bisa mengetahui proses mana yang berjalan lambat ataupun sering stag. • Perbaikan proses bisnis menuju ke proses yang paling efisien bisa dilakukan setiap saat, tanpa harus tergantung pada satu programmer atau satu vendor tertentu.

2.2. Workflow Management System (WfMS)

Workflow merupakan kunci dari komponen teknologi lainnya yang penting untuk BPMS. Dari WfMS, BPMS memperoleh *tool* untuk digunakan membuat desain proses yang memungkinkan solusi dari pemodelan manajemen proses bisnis. Selain itu juga memberikan BPMS kemampuan untuk mengintegrasikan manusia ke dalam pemecahan masalah manajemen proses bisnis. *Workflow Management Coalition* (WfMC) mendefinisikan *workflow* adalah otomatisasi dari proses bisnis secara keseluruhan atau sebagian selama dokumen, informasi, atau tugas yang telah melewati atau lolos dari satu partisipan menuju partisipan lain dengan tindakan tertentu berdasar pada satu set aturan prosedural. Definisi tersebut masih sangat umum jika digunakan dalam banyak variasi skenario *workflow*.

WfMC juga mendefinisikan WfMS sebagai sistem yang mendefinisikan, menciptakan, dan mengelola pelaksanaan *workflow* melalui penggunaan perangkat lunak, yang berjalan pada satu atau lebih mesin *workflow*, yang mampu menafsirkan definisi proses, berinteraksi dengan partisipan *workflow* yang lain jika diperlukan melibatkan penggunaan alat-alat IT dan aplikasi. Chang [5] menambahkan jika definisi dari *workflow* dan WfMS terdengar mirip dengan definisi dari BPM dan BPMS. Pada nyatanya, fungsi dari WfMS identik untuk beberapa komponen dari BPMS. Seperti BPMS, WfMS memberi layanan untuk mengintegrasikan manusia dengan aplikasi, jadi mereka dapat berpartisipasi dalam sebuah proses.



Gambar 2.1 Arsitektur WfMS

Pada Gambar 2.1 ditunjukkan bagan tentang arsitektur dari *workflow manager* yang memuat *workflow repository* guna mengelola model *workflow* dan *workflow engine* guna memproses *workflow instances* yang didalamnya mengandung metadata. Pada dasarnya, *workflow repository* merupakan penyimpanan dari *workflow* model abstrak yang bisa berubah menjadi *workflow instances* dengan menggunakan *engine extension* [7].

2.3. Workflow Repository

Workflow repository merupakan sebuah sistem yang digunakan untuk menyimpan dan menggambarkan *workflow* proses bisnis kedalam sebuah penyimpanan (*repository*). Menurut Liu [8], repositori berfungsi untuk berbagi informasi dalam *database* yang berisi tentang *workflow* yang telah dibuat untuk diproduksi dan digunakan oleh *enterprise*. Sebuah manajer *repository* mengimplementasikan layer dari control service untuk pemodelan, mengambil, dan mengelola objek dalam *repository* dalam DBMS. Selain hanya untuk disimpan, *workflow* yang sudah ada pada *repository* dapat diakses kembali untuk digunakan

dalam pembuatan *workflow* baru. Pencarian dalam *workflow repository* haruslah bermakna dan efisien. Dalam penelitian yang dilakukan oleh Yan *et al* [9] didapat motivasi dari pembangunan sistem *BPM Repositories*, yang pertama yakni repositori menawarkan penyimpanan dan kemudahan pencarian bisnis proses meliputi dari navigasi untuk pencarian terhadap kumpulan proses bisnis. Kedua, repositori dapat digunakan untuk membangun kembali model proses pada perusahaan spesifik. Penelitian mereka juga fokus terhadap mekanisme *reuse*, seperti dalam pemilihan konfigurasi dan menggunakan ontologi untuk menelaraskan terminologi antar proses.

Deelman *et al* [10] mengatakan bahwa jika terjadi kegagalan pada *workflow*, user juga memiliki pilihan untuk melakukan *replan* pada *workflow* dan memindahkan komputasi ke *resource* yang lain. Menyimpan ulang *workflow* memanfaatkan fitur *reuse* dari data yang telah tersimpan sebelumnya, dimana hasilnya didaftarkan kembali dalam tabel data saat dihasilkan. Algoritma *reuse data* memangkas *workflow* berdasarkan pada file *output* dan intermediet yang ada dalam tabel data. *Workflow replanning* juga dapat membantu dalam kasus dimana user membuat kesalahan selama menghasilkan input deskripsi *workflow* yang mengarah kepada kegagalan, seperti dalam pembuatan argumen yang salah untuk kode aplikasi, atau struktur DAG yang tidak benar menghasilkan suatu proses kerja bisa berjalan saat semua masukan disajikan.

2.4. BPMN

Business Process Management Notation (BPMN) adalah metodologi baru yang dikembangkan oleh *Business Process Management Initiative* (BPMI) sebagai standar baru pada pemodelan proses bisnis, dan digunakan sebagai alat desain sistem yang kompleks [11]. Tujuan utama dari pembuatan BPMN ini adalah untuk menyediakan suatu notasi yang mudah dipahami oleh semua pengguna bisnis, mulai dari bisnis analis yang menciptakan konsep awal dari proses, para pengembang teknis yang bertanggung jawab untuk menerapkan teknologi yang akan dilakukan, dan akhirnya kepada pelaku bisnis yang akan mengelola dan memantau proses tersebut. Penelitian yang dilakukan oleh La Rosa et al [12] menyediakan tentang APPROMORE yang merupakan lanjutan *repository* untuk menganalisa dan menggunakan kembali model proses berskala besar. Membandingkan semua elemen permodelan dalam berbagai bahasa BPEL satu sama lain dengan melihat yang mendasari perekaman konsep setiap elemen. Sedangkan dalam penelitian yang akan peneliti buat menggunakan bahasa XPDL yang didalamnya mengandung metadata dari sebuah permodelan proses bisnis. XPDL didapat dari BPMN yang telah di *export*. Penelitian selanjutnya yang dilakukan oleh Puspa et al [13] membandingkan pemodelan antara BPMN dan UML. Hasilnya BPMN dinilai cenderung menggambarkan proses dari pandangan seorang analis bisnis dibandingkan dengan UML. Dari sekian banyak pemodelan proses bisnis yang ada, peneliti memilih BPMN sebagai bahasa yang digunakan karena merupakan proses bisnis yang paling unggul daripada yang lain.

Penotasian proses bisnis perlu mengacu pada aturan berikut:

1. Memodelkan kejadian-kejadian yang memulai proses, proses yang dilakukan dan hasil akhir dari aliran proses.
2. Keputusan bisnis atau percabangan aliran dimodelkan dengan *gateway*. Sebuah *gateway* mirip dengan simbol keputusan dalam flowchart.
3. Sebuah proses dalam aliran dapat mengandung sub proses, yang secara grafis dapat ditunjukkan dengan *Business Process Diagram* (BPD) lain yang tersambung melalui sebuah *hyperlink* ke simbol proses.
4. Jika sebuah proses tidak didetikan ke dalam sub proses, maka dianggap sebagai sebuah task, yaitu level proses paling rendah.
5. Sebuah tanda '+' pada simbol proses menunjukkan bahwa proses ini didekomposisi, jika tidak ada tanda '+', maka proses ini disebut sebuah *task*.

BPMN juga mendefinisikan *Business Process Diagram* (BPD) yang didasarkan pada teknik *flowchart* kemudian disesuaikan untuk mencapai model grafis dari operasi proses bisnis. Sebuah model proses bisnis selanjutnya merupakan jaringan dari objek grafis yang merupakan kegiatan atau pekerjaan dan *flow control* yang mendefinisikan *message* dari suatu kegiatan tersebut. Menurut Sari [14] penjelasan tentang keempat notasi tersebut adalah:

1. Notasi *Event*

Event direpresentasikan oleh lingkaran dan adalah sesuatu yang terjadi selama proses bisnis. *Event* mempengaruhi aliran proses dan biasanya memiliki penyebab (pemicu) atau dampak (hasil). *Event* lingkaran dengan pusat terbuka untuk memungkinkan internal *marker* untuk membedakan pemicu yang

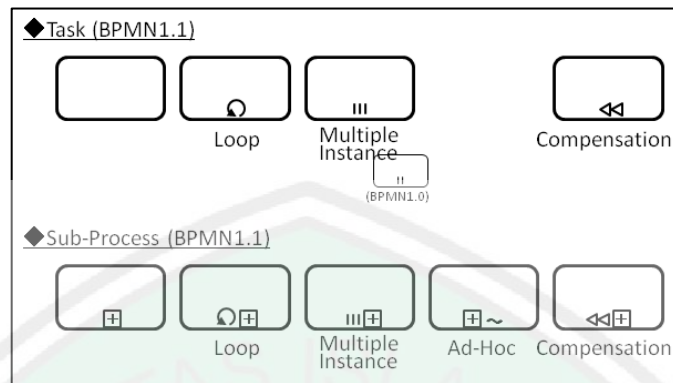
berbeda atau hasil. Ada tiga jenis *Event*, berdasarkan ketika mereka mempengaruhi aliran: *start*, *intermediate* dan *end*. Bentuk dari notasi event ada pada Gambar 2.2 berikut,

Types	Start			Intermediate			End
	Top-Level	Event Sub-Process Interrupting	Event Sub-Process Non-Interrupting	Catching	Boundary Interrupting	Boundary Non-Interrupting	
None	○			○			○
Message	✉	✉	✉	✉	✉	✉	✉
Timer	⌚	⌚	⌚	⌚	⌚	⌚	
Error		⚠			⚠		⚠
Escalation		⚠	⚠		⚠	⚠	⚠
Cancel				⊗	⊗		⊗
Compensation		⏪		⏪			⏪
Conditional	⏸	⏸	⏸	⏸	⏸	⏸	
Link				⏸			⏸
Signal	⚠	⚠	⚠	⚠	⚠	⚠	⚠
Terminate							⦿
Multiple	⊕	⊕	⊕	⊕	⊕	⊕	⊕
Parallel Multiple	⊕	⊕	⊕	⊕	⊕	⊕	

Gambar 2.2 Notasi *Event*

2. Notasi *Activity*

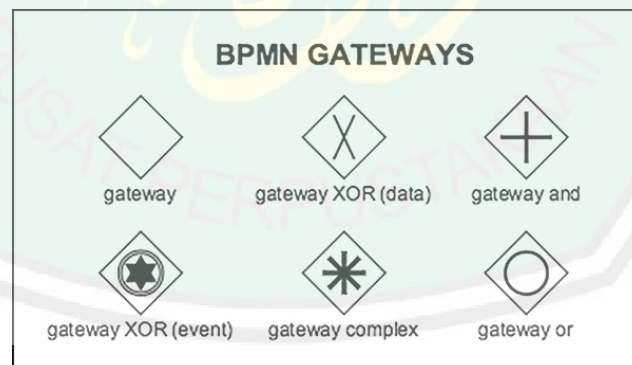
Activity diwakili oleh persegi panjang bulat sudut dan merupakan istilah umum untuk pekerjaan yang melakukan perusahaan. Kegiatan dapat berupa atom atau *nonatomic* (gabungan). Jenis-jenis aktiviti adalah: task dan sub-proses. Sub proses dibedakan dengan simbol kecil tanda plus di bagian tengah bawah dari bentuk. Berikut adalah Gambar 2.2 mengenai notasi *activity*



Gambar 2.3 Notasi Activity

3. Notasi Gateway

Gateway diwakili oleh bentuk berlian familiar dan digunakan untuk mengontrol divergensi dan konvergensi *sequential flow*. Dengan demikian, akan menentukan keputusan tradisional, *forking*, penggabungan, dan bergabung jalur. *Internal marker* akan menunjukkan jenis *control* perilaku seperti pada Gambar 2.4 berikut,

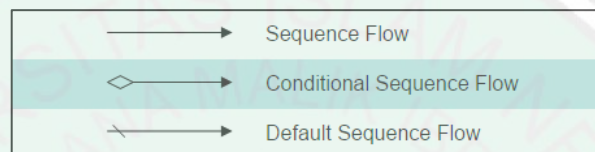


Gambar 2.4 Notasi Gateway

4. Penghubung

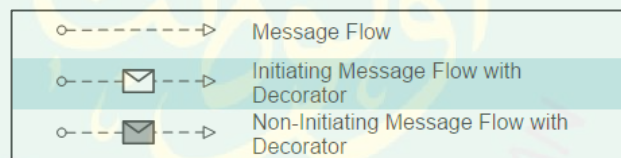
Notasi penghubung terdiri dari tiga jenis yaitu:

- a. *Sequence flow*, menunjukkan kegiatan yang dituju sebagai kelanjutan kegiatan sebelumnya. Gambar 2.5 berikut menunjukkan bentuk notasi *sequence flow*.



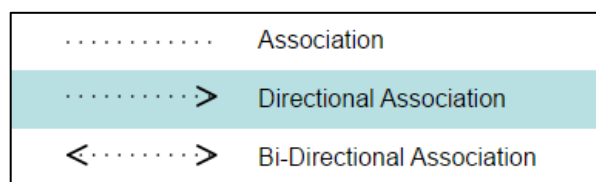
Gambar 2.5 Notasi *sequence flow*

- b. *Message flow*, yang menunjukkan adanya aliran pesan dimana proses yang dituju bukan kelanjutan proses sebelumnya. Gambar 2.6 berikut menunjukkan notasi *message flow*.



Gambar 2.6 Notasi *message flow*

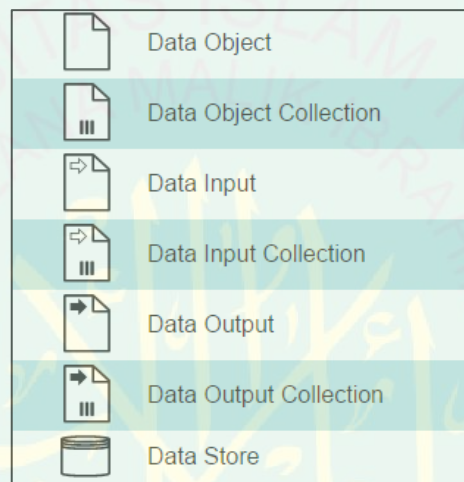
- c. *Association*, yang menunjukkan bentuk hubungan antara dua proses atau kegiatan. Gambar 2.7 menunjukkan notasi *association*.



Gambar 2.7 Notasi *association*

5. Data



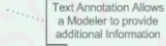
Notasi data bertujuan untuk mengidentifikasi elemen grafis yang spesifik secara visual. Menjelaskan apa yang dilakukan, mendeskripsikan tanda, jenis yang digunakan. Notasi data juga menggambarkan hubungan antar elemen grafis. Bentuk notasi data ditunjukkan dengan Gambar 2.8



Gambar 2.8 Notasi data

6. Artifak

BPMN dirancang untuk memungkinkan pemodelan dan alat-alat pemodelan fleksibel dalam memperluas notasi dasar dan dalam memberikan kemampuan untuk konteks sesuai dengan situasi pemodelan spesifik. Setiap jumlah artefak dapat ditambahkan ke diagram yang sesuai untuk konteks proses bisnis yang dimodelkan. Notasi artefak ditunjukkan dengan Gambar 2.9 berikut,

Notasi	Keterangan
	Data Object Data objek adalah mekanisme untuk menunjukkan bagaimana data yang diperlukan atau dihasilkan oleh aktivitas. Dihubungkan menggunakan asosiasi
	Group group ini dapat digunakan untuk keperluan dokumentasi atau analisis, tapi tidak mempengaruhi sequential flow.
	Annotation An <i>notation</i> adalah mekanisme untuk modeler untuk memberikan informasi teks tambahan untuk pembaca

Gambar 2.9 Notasi artefak

7. Swimlanes

Swimlanes digunakan untuk memvisualisasikan kemampuan fungsional atau tanggung jawab yang berbeda. Ditunjukkan pada Gambar 2.10 berikut,



Gambar 2.10 Notasi *Swimlanes*

2.5. XML Process Definition Language (XPDL)

XML Process Definition Language merupakan format standar yang dibuat oleh WfMC untuk pertukaran definisi proses bisnis antara produk *workflow* yang berbeda, yakni antara alat pemodelan yang berbeda dan manajemen *suite*. XPDL mendefinisikan skema dari XML untuk menentukan bagian deklaratif dari alur kerja proses bisnis.

XPDL dirancang untuk bertukar definisi proses, baik grafis dan semantik dari proses bisnis sebuah *workflow*. XPDL saat ini adalah format terbaik untuk pertukaran diagram BPMN dan dirancang khusus untuk menyimpan semua aspek dari BPMN. Atau dalam kata lain XPDL adalah serialisasi XML dari BPMN. XPDL mengandung unsur untuk menahan informasi grafis seperti posisi node X dan Y, serta aspek eksekusi yang akan digunakan untuk menjalankan proses. Menurut Handayani *et al* [15] elemen dari XPDL ada enam, yakni:

1. Elemen *<Package>* adalah paket yang mendefinisikan proses bisnis. Konsep paket ini berguna untuk mengurangi definisi *redundancy* yang dapat digunakan untuk berbagai macam proses.
2. Elemen *<WorkflowProcess>* yaitu elemen yang digunakan untuk mendefinisikan sub proses dari suatu proses tertentu.
3. Elemen *<ActivitySet>* menunjukkan kumpulan dari berbagai macam aktivitas dan transisi yang menghubungkan antar aktivitas tersebut.
4. Elemen *<BlockActivity>* berfungsi untuk melakukan eksekusi atas elemen *<ActivitySet>*
5. Elemen *<Participant>* digunakan untuk menspesifikasikan partisipan pada *workflow*, misalkan entitas yang dapat mengeksekusi aktivitas.
6. Elemen *<DataType>* dan *<DataField>* digunakan untuk menspesifikasikan data relevan *workflow*. Data ini digunakan untuk penentuan keputusan atau mereferensikan data diluar dari *workflow* dan ditransfer ke semua aktivitas dan *subflows*.

Dari sekian *element* yang ada maka peneliti mengambil beberapa *element* yang mendukung untuk pendefinisian metadata dari suatu *workflow*, *element* yang diambil antara lain:

a. *Package Definition Header*

Bagian *package Definition Header* menyimpan seluruh informasi umum dalam suatu *package*,

Tabel 2.1 Deskripsi *Definition Header*

	Deskripsi
XDPLVersion	Versi dari dokumen XPDL, versi terakhir adalah versi “2.2”.
Vendor	Mendefinisikan asal dari model proses dan berisi nama vendor, nama produk vendor, serta nomor rilis produk.
Created	Tanggal pembuatan <i>Package Definition</i> disimpan dalam bentuk basic atau extended yang telah ditentukan oleh ISO 8601. Misal: 1985-04-12T10:15:30Z memiliki arti pukul 15.30 dan tanggal 12 April 1985 GMT.
ModificationDate	Mendefinisikan tanggal dimana diagram terakhir diubah. Format dari waktunya sama seperti pada elemet Created.
Description	Deskripsi teks dari paket diagram .

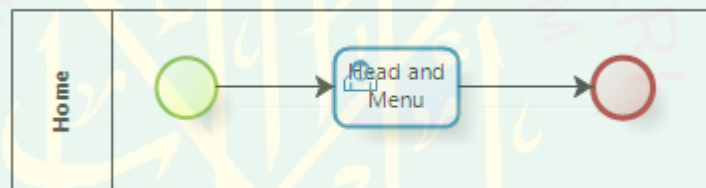
b. *Redefinable Header*

Redefinable Header meliputi atribut *header* yang ada dalam *Definition Header* dan dapat didefinisikan ulang. *Element* yang ada dalam *Redefinable Header* antara lain:

Tabel 2.2 Deskripsi Redefinable Header

	Deskripsi
Author	Nama pengarang pada package ini.
CountryKey	Kode negara yang berdasar pada ISO 3166. Dapat berupa tiga digit nomor kode negara atau dua kode alfabet negara.
Version	Versi dari Package Definiton ini.

Salah satu keuntungan dari penggunaan XPDL adalah proses deskripsi XML yang terdapat di XPDL dapat ditransformasikan ke dalam bentuk grafis sehingga pengguna lebih mudah dan cepat untuk memahami alur proses bisnis organisasi. Akan diilustrasikan pada dua gambar dibawah ini,

**Gambar 2.11 Pemodelan BPMN**

```

<?xml version="1.0" encoding="utf-8"?>
<Package xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" Id="75825e66-
ba88-4360-a071-83658958bcf4" Name="Diagram 1"
xmlns="http://www.wfmc.org/2009/XPDL2.2">
<PackageHeader>
<XPDLVersion>2.2</XPDLVersion>
<Vendor>Bizagi Process Modeler.</Vendor>
<Created>2016-11-05T00:39:01.8247066+07:00</Created>
<ModificationDate>2017-03-
02T19:12:05.0499293+07:00</ModificationDate>
<Description>Diagram 1</Description>
<Documentation />
</PackageHeader>
<RedefinableHeader>
<Author>navIras</Author>
<Version>1.0</Version>
<Countrykey>CO</Countrykey>
</RedefinableHeader>
<ExternalPackages />
<DataStores>
<DataStore Id="caa965df-49b8-4ca0-98e9-52d35ca79f6b"
Name="192.168.1.101" IsUnlimited="true">
<Object>
<Documentation />
</Object>
</DataStore>
</DataStores>

```

Gambar 2.12 Kode XPDL dari BPMN

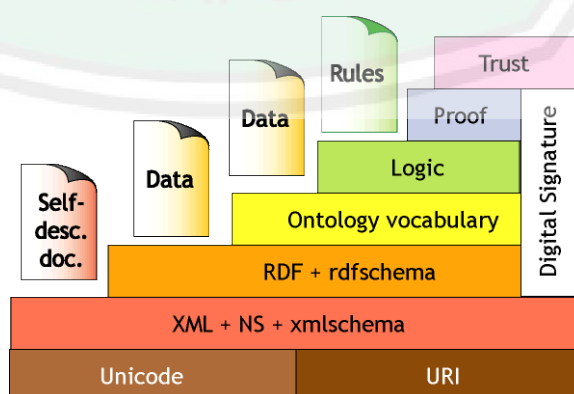
2.6. Metadata

Metadata adalah sebuah struktur informasi yang mendiskripsikan, menjelaskan, menempatkan, atau dalam kata lain membuatnya untuk mudah dikembalikan, digunakan atau mengelola sumber informasi. Metadata biasa disebut juga dengan data yang menerangkan tentang data atau informasi yang menerangkan tentang informasi. Metadata merupakan kunci untuk memastikan sumber tersebut akan bertahan dan berlanjut untuk dapat diakses lagi pada masa depan. Ada tiga macam tipe dari metadata [3], yaitu:

1. *Descriptive metadata* menjelaskan sumber untuk tujuan seperti penemuan dan identifikasi. Elemen yang termasuk antara lain *title*, *abstract*, *author*, dan *keyword*.
2. *Structural metadata* mengindikasikan bagaimana suatu objek diletakkan bersama-sama.
3. *Administrative metadata* menyediakan informasi untuk membantu mengelola sumber daya, seperti kapan dan bagaimana ia diciptakan, tipe file dan informasi teknis lainnya, serta siapa saja yang dapat mengakses.

2.7. Web Semantik

Pada mulanya *Hypertext Markup Language* (HTML) mulai diperkenalkan pada tahun 1990 oleh Tim Berners-Lee sebagai standar dalam mengatur sistem dokumen yang terdistribusi. Seiring berjalannya waktu maka untuk menghasilkan suatu halaman *web* yang dinamis dan cerdas dalam memecahkan masalah, munculah yang disebut dengan *web* semantik. Istilah *web* semantik memiliki pengertian yakni informasi yang terhubung secara global dengan suatu cara tertentu yang dapat dipahami oleh mesin (*machine understandable*).



Gambar 2.13 *Semantic Web Layer Cake*

Layer web semantik terdiri dari tujuh *layer* dan dijelaskan sebagai berikut:

1. URI dan *Unicode*

Uniform Resource Identifier (URI) memiliki fungsi menunjukkan lokasi dan identitas dari suatu *resources*.

2. XML + NS + *xmldata*

Extensible Markup Language dan *Namespace* serta *Schema* merupakan sintaks yang berfungsi untuk menyajikan struktur pada *web*.

3. RDF + RDFS

Resource Description Framework (RDF) dibangun oleh *World Wide Web Consortium* (W3C) merupakan data model untuk mendeskripsikan sebuah *resource* pada *web* yang menyediakan mekanisme untuk mengintegrasikan banyak skema metadata. RDF dirancang agar bisa dibaca dan dipahami oleh komputer. RDF pada umumnya ditulis dalam XML. Contoh penggunaan RDF antara lain:

- a. Mendeskripsikan properti untuk item belanja, seperti harga dan ketersediaan
- b. Mendeskripsikan penjadwalan untuk agenda *web*
- c. Mendeskripsikan informasi tentang halaman *web* (konten, *author*, tanggal dibuat dan tanggal modifikasi)

```

<?xml version="1.0"?>

<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:si="https://www.w3schools.com/rdf/">

<rdf:Description rdf:about="https://www.w3schools.com">
  <si:title>W3Schools</si:title>
  <si:author>Jan Egil Refsnes</si:author>
</rdf:Description>

</rdf:RDF>

```

Gambar 2.14 Contoh format RDF

4. Ontology

Ontologi merupakan kosakata untuk menggambarkan serangkaian konsep dalam domain (domain didefinisikan sebagai area spesifik subjek atau bidang pengetahuan) dan hubungan yang ada antara beberapa konsep. Dalam konteks *web services*, ontologi merupakan tokoh menonjol sebagai cara memberikan deskripsi semantik untuk *web services* yang dapat digunakan oleh aplikasi web dan sistem cerdas. Lalu Grubber [16] menjelaskan bahwa ontologi adalah suatu spesifikasi eksplisit dan konseptualisme. Ia menerangkan untuk sistem berbasis pengetahuan, eksistensi sesuatu berkenaan dengan bagaimana sesuatu tersebut dapat direpresentasikan.

Menurut Prasetyo dan Wiryana [17] Ontologi menangkap permasalahan-permasalahan dalam lingkup domain spesifik, contohnya masalah finansial, pengobatan dan biologi. Ontologi memiliki kemiripan dengan skema *database* atau diagram *class* dalam pemrograman berbasis objek. Dalam bidang AI (*Artificial Intelligence*) ontologi memiliki pengertian, yaitu merupakan kosakata yang merepresentasikan domain

atau subyek pembahasan tertentu, serta sebagai suatu *body of knowledge* untuk menjelaskan suatu bahasan tertentu. Model dari ontologi web semantik antara lain RDF, RDF *Schema*, dan *Ontology Web Language*. Secara sintak ketiganya mirip dengan bahasa XML, dimana menggunakan sintak XML dan konsep *namespace*.

5. Logic dan Proof

Lapisan ini berisi rule dan system untuk melakukan reasoning pada ontologi sehingga dapat disimpulkan apakah suatu resource memenuhi syarat tertentu.

6. Trust

Lapisan yang memungkinkan pengguna web untuk mempercayai suatu informasi pada web atau mengevaluasi apakah hasil dari lapisan proof dapat dipercaya.

7. Digital Signature

Layer ini mendefinisikan blok dari data yang akan dimanfaatkan oleh komputer dan agen untuk memastikan apakah suatu informasi disediakan oleh sumber yang terpercaya serta mendekteksi adanya perubahan pada dokumen.

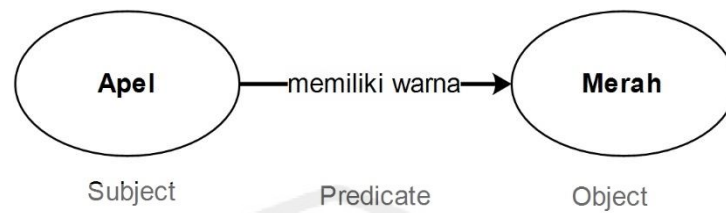
2.8. *Resource Description Framework*

RDF merupakan suatu kerangka kerja umum yang digunakan untuk menggambarkan setiap sumber daya internet seperti situs *web* dan isinya. Deskripsi dari RDF atau deskripsi yang sering disebut sebagai metadata atau “data tentang data” ataupun juga “informasi tentang informasi”. Dalam deskripsi ini biasanya

mengandung penulisan tanggal, sumber daya penciptaan atau pembaharuan, informasi yang menggambarkan isi atau rating konten, kata kunci untuk pengumpulan data mesin pencari, kategori, subjek, dan sebagainya. RDF dapat memudahkan setiap orang untuk berbagi di dalam situs web dan deskripsi lain yang lebih mudah serta para pengembang perangkat lunak untuk membangun produk yang dapat menggunakan metadata untuk menyediakan mesin pencari dan direktori yang lebih baik, untuk bertindak sebagai agen cerdas, dan memberikan para pengguna web lebih mengendalikan apa yang mereka lihat.

RDF terdiri dari tiga komposisi yaitu *subject*, *predicate*, dan *object*. *Predicate* merupakan komposisi yang menerangkan tentang sudut pandang dari *subject* yang dijelaskan *object*, sementara *subject* dan *object* merupakan entitas. *Object* didalam RDF dapat menjadi *subject* yang diterangkan oleh *object* yang lainnya. Dengan ini *object* dapat berupa masukan yang dapat diterangkan secara jelas dan detail., sesuai keinginan pengguna yang memberikan masukan.

Cara kerja RDF dapat diterangkan dengan contoh sederhana berikut, untuk mendefinisikan “apel memiliki warna merah”, maka “apel” direpresentasikan sebagai *subject*, “merah” merupakan *object*, dan “memiliki warna” adalah *predicate*. Untuk lebih mudahnya direpresentasikan pada Gambar 2.15



Gambar 2.15 Penggambaran RDF

2.9. RDF API for PHP

RDF API for PHP (RAP) adalah alat untuk mengembangkan *semantic web*. RAP memiliki fitur yaitu untuk parsing, memanipulasi, penyimpanan, *query*, pelayanan, dan serialisasi *graph* RDF. RAP berawal dari proyek *open source* oleh Freie Universitat Berlin pada 2002 dan telah mengalami perkembangan dengan kontribusi kode dari para komunitas *semantic web*. Inti dari RAP meliputi dua implementasi atas penyimpanan *statement*, yaitu apakah menyimpan *graph* RDF dalam *memory* sistem maupun dalam *database relational*. Penyimpanan RAP menyediakan *interface* yang kaya untuk memanipulasi *graph* RDF yang berbeda lapisan abstraksinya. RAP juga dibekali kemampuan untuk melakukan *query Simple Protocol and RDF Query Language* (SPARQL) pada RDF. Klausula yang digunakan dalam query SPARQL antara lain:

a. PREFIX

PREFIX digunakan untuk menyingkat sebuah *resource*, dalam hal ini dapat diwakili oleh URI (*Uniform Resource Identifier*).

b. SELECT

Statement SELECT didefinisikan sebuah daftar variabel yang akan dikembalikan sebagai hasil dari eksekusi *query*. Setiap variabel diawali dengan notasi (?).

c. WHERE

Statement WHERE didefinisikan sebagai sederetan *triple pattern* yang harus dimiliki oleh setiap hasil *query* yang valid. Seluruh pola yang merepresentasikan suatu kalimat RDF harus sesuai dengan RDF *triples*, yaitu *subject*, *predicate*, dan *object*. Ketiga RDF *triples* tersebut dapat direpresentasikan oleh URI atau sebuah *variable* dan literal.

BAB 3

METODOLOGI PENELITIAN

3.1. Desain Penelitian

Desain penelitian memaparkan tentang alur dan mekanisme penelitian yang akan dilakukan meliputi, tipe penelitian, prosedur penelitian, dan metode pengolahan data. Berikut penjelasan dari masing-masing mekanisme penelitian.

3.1.1 Gambaran Umum Sistem

Sistem yang dibangun adalah sistem yang berfungsi untuk menyimpan *workflow* yang ada pada ERP di pondok pesantren. *Repository* untuk *workflow* ini memiliki dua lapis penyimpanan, yakni data dan metadata. *Repository* data disimpan pada database MySQL dan *query* SQL. Sedangkan *repository* metadata disimpan dalam database RDF dan *query* SPARQL. Keduanya akan dibandingkan tingkat kemudahan ditemukannya dalam *repository*.

3.1.2 Sumber Data

Data yang digunakan dalam penelitian ini berupa model proses bisnis yang terdapat pada ERP pondok pesantren dan penambahan yang dibuat *by design*. Proses bisnis yang dijadikan objek dan merupakan data primer berekstensi *file* XPDL. Data proses bisnis yang didapatkan selanjutnya akan diolah sesuai prosedur penelitian. Adapun tempat pengambilan data berupa proses bisnis akademik, kesarifan, sarana prasarana, dan penerimaan santri baru berada pada lokasi yang berbeda, yakni

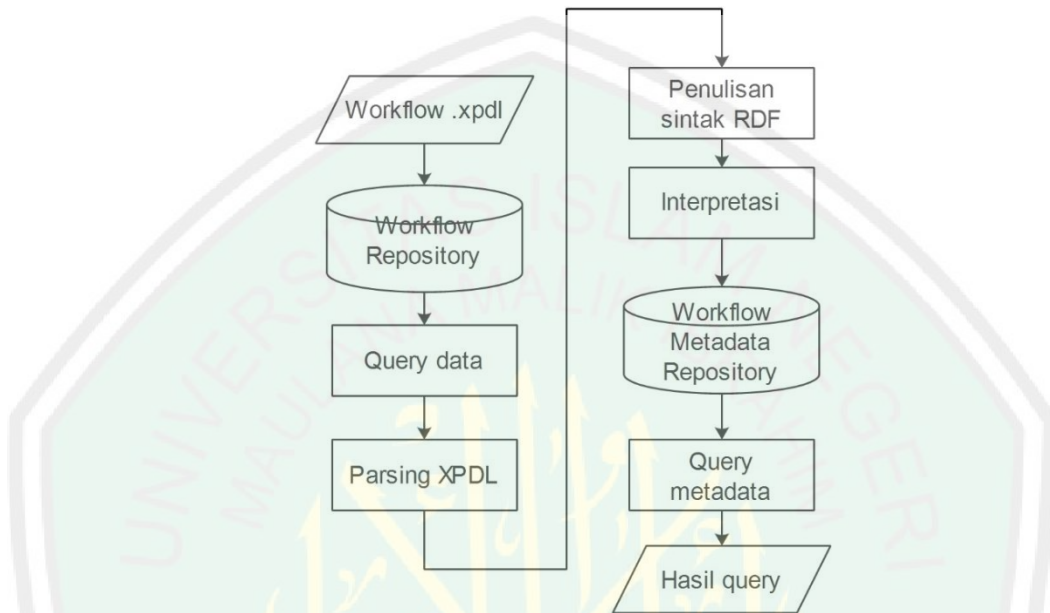
1. Pondok Pesantren Annur 1 Bululawang Kab. Malang
2. Pondok Pesantren Tahfidz Ar Rohmah 2 Batu
3. Pondok Pesantren Modern Ar Rifai Malang
4. Pondok Pesantren Mahasiswa Luhur Malang
5. Pondok Pesantren Salaf Anwarul Huda Malang

Penelitian sebelumnya telah berhasil membuat sistem informasi untuk ERP pondok pesantren. Dalam pembuatannya perlu menggunakan BPMN untuk memodelkan proses bisnis. Karena proses bisnis yang dibuat belum memiliki penyimpanan agar dapat digunakan lagi nantinya, maka ditawarkan solusi untuk membuat penyimpanan proses bisnis ini ke dalam sistem.

3.2. Rancangan Sistem

Pada proses rancangan sistem ini akan dilakukan tahapan-tahapan yang nantinya digunakan untuk pembangunan sistem. Rancangan sistem dimulai dari pengumpulan data *workflow* yang berasal dari beberapa pondok pesantren dan model proses by design. Kedua yakni menyimpan data *workflow* tersebut kedalam tabel data beserta menginputkan nama *workflow* dan nama pondok pesantren. Ketiga yakni *query* data tersebut menggunakan SQL untuk diolah dalam proses parsing. Keempat yaitu parsing XPDL dengan mengambil element yang diperlukan untuk menyusun metadata. Kelima menyusun metadata yang telah terambil dalam sebuah dokumen RDF. Keenam yakni menginterpretasikan metadata serta validasi kebenaran RDF. Ketujuh menyimpan RDF dalam *repository* dan menampilkannya pada tabel. Kedelapan yakni *query* metadata menggunakan SPARQL untuk

mengambil *workflow* yang diinginkan user. hingga penyimpanan metadata pada *workflow metadata repository* dan akan ditunjukkan pada Gambar 3.1 berikut,



Gambar 3.1 Rancangan Sistem

3.2.1 Input Workflow

Pada penelitian ini dokumen yang akan diolah berupa *workflow* berekstensi *.xpdL. Ekstensi ini dipilih karena paling sesuai untuk pengambilan metadata.

Contoh masukan berupa XPDL ditunjukkan pada Gambar 3.2

```

<?xml version="1.0" encoding="utf-8"?>
<Package xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
Id="75825e66-ba88-4360-a071-83658958bcf4" Name="Diagram 1"
xmlns="http://www.wfmc.org/2009/XPDL2.2">
<PackageHeader>
<XPDLVersion>2.2</XPDLVersion>
<Vendor>Bizagi Process Modeler.</Vendor>
<Created>2016-11-05T00:39:01.8247066+07:00</Created>
<ModificationDate>2017-03-
02T19:12:05.0499293+07:00</ModificationDate>
<Description>Diagram 1</Description>
<Documentation />
</PackageHeader>
<RedefinableHeader>
<Author>navIras</Author>
<Version>1.0</Version>
  
```

```
<Countrykey>CO</Countrykey>
</RedefinableHeader>
```

Gambar 3.2 Contoh dokumen *.xpdl

3.2.2 Penyimpanan workflow pada workflow repository

Database Workflow Repository terdiri dari beberapa tabel berikut:

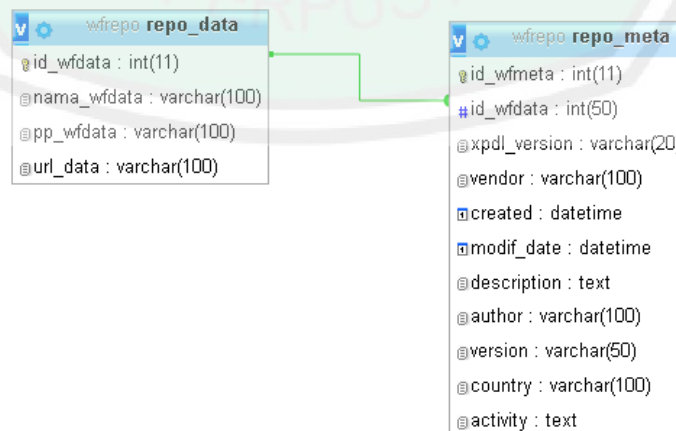
a. Tabel *repo_data*

Kolom pada tabel *repo_data* dapat dilihat pada Tabel 3.1

Tabel 3.1 Tabel *repo_data*

Nama Kolom	Type Data	Lebar	Key	Keterangan
id_wfdata	INT	11	PRIMARY KEY	Menjadi ID unik dari workflow data yang disimpan
nama_wfdata	VARVCHAR	100	-	Berisi input nama workflow yang disimpan
pp_wfdata	VARCHAR	100	-	Berisi asal pondok pesantren dari workflow yang disimpan
url_data	VARCHAR	100	-	Berisi nama file workflow yang disimpan

Skema ERD *Workflow Repository* dapat dilihat pada Gambar 3.3 berikut:



Gambar 3.3 Entity Relational Diagram

Penyimpanan *workflow* data dilakukan dengan mengisi atribut data yang akan disimpan dalam *database* MySQL. Setelah nama *workflow* dan nama pondok pesantren diisikan dalam *field*, maka terdapat aturan saat *upload file workflow* yakni data yang diupload harus berekstensi *.xpdL dan ukuran *file* tidak lebih dari 1044070Kb.

Gambar 3.4 Form input data *workflow*

```

$nama_wfdata=$_POST['nama_wfdata'];
$pp_wfdata=$_POST['pp_wfdata'];
if($_POST['proses']){
    $ekstensi_boleh = array('xpdL');
    $nama = $_FILES['file']['name'];
    $x = explode('.', $nama);
    $ekstensi = strtolower(end($x));
    $ukuran = $_FILES['file']['size'];
    $file_temp=$_FILES['file']['tmp_name'];

    if(in_array($ekstensi, $ekstensi_boleh) === true){
        if ($ukuran < 1044070){
            move_uploaded_file($file_temp, 'file/.'.$nama);
            $query = mysql_query("INSERT INTO repo_data (nama_wfdata,
            pp_wfdata, url_data) VALUES ('$nama_wfdata', '$pp_wfdata',
            '.'.$nama.'')");
        }
    }
}

```

Gambar 3.5 Kode Sumber Simpan Data *Workflow*

Penjelasan untuk kode sumber pada Gambar 3.5 yakni; `$ekstensi_boleh` untuk membuat *array* yang berisikan file berekstensi XPD L saja yang bisa

diupload; \$nama, \$x, \$ekstensi, \$ukuran, \$file_temp digunakan untuk mendeklarasikan properti dari inputan file, seperti nama file, file ekstensi, ukuran, dan file temp; *in array* digunakan untuk mengecek apakah file ekstensi sesuai dengan yang diizinkan, jika *true* maka berhasil; \$ukuran < 1044070 untuk mengecek apakah ukuran file tidak lebih besar dari 1044070 (1Mb), jika lebih kecil maka berhasil; *move_uploaded_file* untuk melakukan proses upload dan lokasi penyimpanan; \$query untuk melakukan query INSERT ke database.
















3.2.3 Query data

Setelah data berupa *workflow* berformat *.xpdL telah disimpan dalam *workflow data repository*, selanjutnya *workflow* dipanggil lagi menggunakan query SQL untuk selanjutnya *diparsing*. Query yang digunakan seperti pada Gambar 3.6 Sebagai contoh kita akan mencari data yang memiliki unsur kata 'aka' maka hasil yang akan muncul ditampilkan seperti pada Gambar 3.7.

```
$kata = $_POST['kata'];
$query = mysql_query("SELECT *FROM TABEL WHERE nama_kolom1
like '%" . $kata . "%' or nama_kolom2 like '%" . $kata . "%'");
```

Gambar 3.6 Kode sumber Query Data

aka

Table Data Workflow					
Hasil pencarian dari : aka					
No.	Nama Workflow	Pondok Pesantren	URL	Lihat	Aksi
1	Akademik	Anwarul Huda	Akademik Pondok Pesantren Anwarul Huda.xpdl	View	  
2	Akademik	Al-Rifai	Akademik_Pondok_Pesantren_Al-Rifai.xpdl	View	  
3	Akademik	Anwarul Huda	Akademik_Pondok_Pesantren_Anwarul_Huda.xpdl	View	  
4	Akademik	Ar-Rohmah	Akademik_Pondok_Pesantren_Ar_Rohmah.xpdl	View	  
5	Akademik	Z	Akademik_Pondok_Pesantren_Z.xpdl	View	  

Gambar 3.7 Hasil query data

3.2.4 Parsing

Proses *parsing* merupakan tahap pemilahan *element* yang terdapat pada *file* XPDL untuk dilakukan pengolahan data. Proses ini bertujuan untuk memudahkan pengolahan data dengan cara memfilter beberapa atribut dan dicari bagian yang memang dibutuhkan dalam proses parsing. XPDL ialah salah satu pemodelan berbasis XML, sehingga dapat diolah dengan melakukan konversi ke dalam *multi-programming language*, antara lain bahasa PHP. Untuk membaca dokumen XPDL ke dalam pemrograman PHP dapat dibantu oleh XML DOM *Parser*.

Parsing data dapat diawali dengan menentukan atribut mana nantinya akan difilter data sesuai kebutuhan. Pengambilan data dapat digambarkan dalam *worksheet*. Data sumber diambil untuk kemudian diolah dan disusun menurut struktur maupun kategori yang ada dalam *worksheet* untuk melihat skemanya. Contoh *worksheet* metadata yang akan diolah ditunjukkan pada berikut

Tabel 3.2 Contoh worksheet metadata

No	Element	Objek
1.	URI	http://localhost/wf/file/Akademik_Pondok_Pesantren_Anwarul_Huda.xpdl
2.	<XPDLVersion>...</XPDLVersion>	2.2
3.	<Vendor>...</Vendor>	Bizagi Process Modeler
4.	<Created>...</Created>	2017-12-03 09:53:53
5.	<ModificationDate>...</ModificationDate>	0000-00-00 00:00:00
6.	<Description>...</Description>	Akademik Pondok Pesantren Anwarul Huda
7.	<Author>...</Author>	Digbys
8.	<Version>...</Version>	1.0
9.	<Countrykey>...</Countrykey>	CO
10.	<Activity>...</Activity>	Daftar ulang santri lama Daftar Ulang santri baru Pembuatan Kegiatan Pondok Pendataan Santri pondok Rekap pendataan kegiatan Diniyah Rekap data KBM Kalkulasi nilai ujian dan proses kelulusan Pendataan Alumni

a. Tabel repo_meta

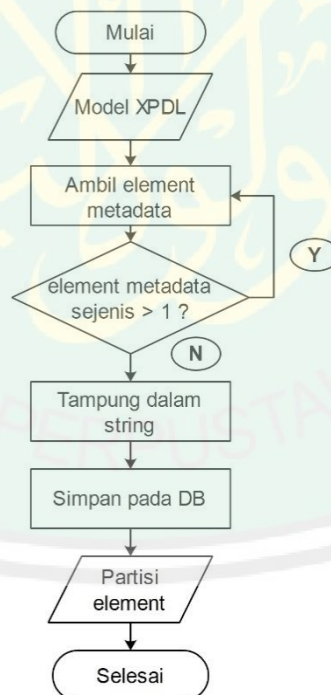
Setelah mengacu pada *worksheet* metadata yang akan diambil dalam *workflow* maka dibuatlah kolom pada tabel repo_meta dapat dilihat pada Tabel 3.3

Tabel 3.3 Tabel repo_meta

Nama Kolom	Tipe Data	Lebar	Key	Keterangan
id_wfmeta	INT	11	PRIMARY KEY	Id unik dari metadata yang disimpan
id_wfdata	INT	11	FOREIGN KEY	Id relasi yang diambil dari tabel repo_data
xpdl_version	VARCHAR	20	-	Berisi dari versi xpdl yang dipakai dalam workflow
Vendor	VARCHAR	100	-	Berisi vendor yang digunakan untuk membangun workflow
created	DATETIME	-	-	Berisi waktu pembuatan awal workflow
modif_date	DATETIME	-	-	Berisi waktu modifikasi terakhir workflow

Description	TEXT	-	-	Berisi deskripsi workflow
Author	VARCHAR	100	-	Berisi nama author dari workflow
version	VARCHAR	50	-	Berisi versi workflow
Country	VARCHAR	100	-	Berisi country key workflow
Activity	TEXT	-	-	Berisi activity workflow yang nantinya akan dijadikan keyword pencarian utama.

Tabel yang telah dibuat dalam *database* berfungsi menampung hasil *parsing*, sumber kode *parsing* dapat dilihat pada Gambar 3.8 menggunakan parser *simplexml*.



Gambar 3.8 Alur parsing element

```

$id_wfdata = $_GET['id_wfdata'];
$query = mysql_query("SELECT * FROM repo_data WHERE
id_wfdata='".$id_wfdata."'");
    $data = mysql_fetch_array($query);
    $xml = simplexml_load_file("file/".$data['url_data']) ;
    $nama_activity = array();
        foreach ($xml->WorkflowProcesses as
$WorkflowProcesses) {
            foreach ($WorkflowProcesses->WorkflowProcess as
$WorkflowProcess) {
                foreach ($WorkflowProcess->Activities as
$Activities) {
                    foreach ($Activities->Activity as $Activity) {
                        $nama_activity [] = $Activity['Name'];
                    }
                }
            }
        }
        foreach($xml->PackageHeader as $ph){
            foreach ($xml->RedefinableHeader as $rh) {
                $xpdl_version= $ph->XPDLVersion;
                $vendor      = $ph->Vendor;
                $created     = $ph->Created;
                $modif_date  = $ph->ModificationDate;
                $description = $ph->Description;
                $author      = $rh->Author;
                $version     = $rh->Version;
                $country     = $rh->Countrykey;
                activity     = implode(" ", $nama_activity);
                $query ="INSERT INTO repo_meta (id_wfdata, xpdl_version, vendor,
created, modif_date, description, author, version, country,
activity) VALUES ('$id_wfdata', '$xpdl_version', '$vendor',
'$created', '$modif_date', '$description', '$author', '$version',
'$country', '$activity')";
                $q = mysql_query($query);
            }
        }

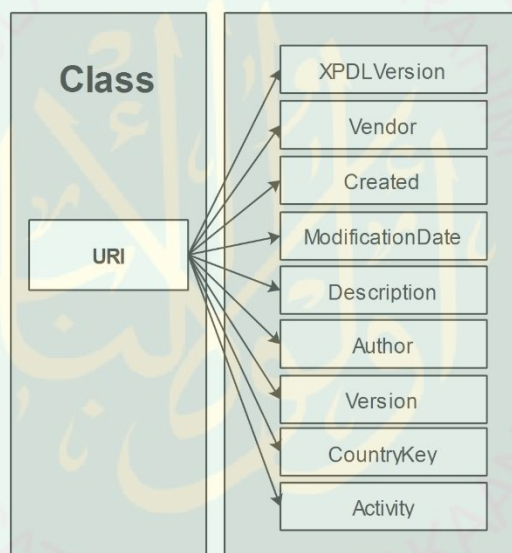
```

Gambar 3.9 Kode Sumber Parsing Metadata Workflow

Setiap satu dokumen XPDL digali semua informasi metadata yang dibutuhkan dengan menggunakan perulangan *foreach*. Langkah yang dilakukan yakni pertama mengimpor dokumen XPDL yang telah tersimpan dalam *repository* data. Kemudian mencari *tag* yang mengandung target elemen *parsing*. *Child element XPDLVersion, Vendor, Created, ModificationDate* dan *Description* terdapat pada *root PackageHeader* serta menggunakan *value \$ph*. Untuk *child element Author, Version, dan CountryKey* berada pada *root RedefinableHeader*

dengan menggunakan *value* \$rh. Sedangkan dalam atribut *Activity* yang akan diparsing letaknya berada dalam *child element* dari *Activities* → *Workflow Process* → *Workflow Processes* dengan menggunakan *value* \$Activity. Letak dari tiap *activity* yang banyak membuat nilai yang diambil tersebut disatukan kembali menjadi satu kesatuan *activity* menggunakan fungsi *implode*.

Selanjutnya metadata yang ada akan disusun dalam skema menurut strukturnya seperti contoh pada Gambar 3.10,



Gambar 3.10 Penyusunan skema

3.2.5 Penulisan sintak RDF

Pembuatan skema seperti Gambar 3.10, tahapan berikutnya yakni penulisan sintak RDF dengan dibantu parser penulisan dokumen XML yaitu *DOMDocument* dengan ekstensi *.rdf. Contoh penulisan format RDF diawali sebagai format pada Gambar 3.11 berikut:

```

<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
ns#" xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description
rdf:about="http://localhost/wf/file/production_planning.xpdl">
    <dc:XPDLVersion>2.2</dc:XPDLVersion>
    <dc:Vendor>Bizagi Process Modeler.</dc:Vendor>
    <dc:Created>2016-11-05 00:39:01</dc:Created>
    <dc:ModificationDate>2016-11-07
18:15:09</dc:ModificationDate>
    <dc:Description>Diagram 1</dc:Description>
    <dc:Author>navIras</dc:Author>
    <dc:Version>1.0</dc:Version>
    <dc:CountryKey>CO</dc:CountryKey>
  </rdf:Description>
</rdf:RDF>

```

Gambar 3.11 Contoh format RDF

Format RDF yang telah ditentukan akan kemudian dibuat susunannya dengan cara menyimpan data yang ada pada tabel `repo_meta` menjadi *array*. Kemudian dengan menggunakan *DOMDocument* mulai dibangun file RDF menurut susunan *root* dan *childnode*. RDF yang telah disusun disimpan dalam sebuah file berekstensi `*.rdf`.

```

$query = "SELECT * FROM repo_meta";
$hasil = mysql_query($query);
$numField = mysql_num_fields($hasil);
$sites = array();
while ( $data = mysql_fetch_array($hasil) ) {
    $sites[]=array('xpdl_version'=>$data['xpdl_version'],
        'vendor' => $data['vendor'],
        'created'=> $data['created'],
        'modif_date' => $data['modif_date'],
        'description'=> $data['description'],
        'author' => $data['author'],
        'version' => $data['version'],
        'country' => $data['country']);
}

```

Gambar 3.12 Sumber Kode Mengambil Data dari MySQL ke Array

```

$dokumen = new DOMDocument();
    $dokumen->formatOutput = true;
    $root = $dokumen->createElementNS
("http://purl.org/dc/elements/1.1/", 'rdf:RDF');
    $root = $dokumen->appendChild($root);
    $root->setAttributeNS("http://www.w3.org/2000/xmlns/",
"xmlns:rdf", "http://www.w3.org/1999/02/22-rdf-syntax-ns#");
    $root-
>setAttributeNS("http://www.w3.org/2000/xmlns/", "xmlns:dc", "http
://purl.org/dc/elements/1.1/");

    foreach ($sites as $meta) {
$loc = "http://localhost/wf/file/".$meta['url_data']."";
$block = $dokumen->createElement("rdf:Description");

$block = $dokumen->createElementNS($loc, "rdf:Description");
$block->setAttributeNS("http://www.w3.org/2000/xmlns/",
"xmlns:rdf" , "http://www.w3.org/1999/02/22-rdf-syntax-ns#");
$block->setAttributeNS("http://www.w3.org/1999/02/22-rdf-syntax-
ns#", "rdf:about" , $loc);

$xml_version = $dokumen->createElement("dc:XPDLVersion");
$xml_version->appendChild($dokumen-
>createTextNode($meta['xml_version']));
$block->appendChild($xml_version);

$url_data= $dokumen->createElement("dc:URLData");
$url_data->appendChild($dokumen-
>createTextNode($meta['url_data']));
$block->appendChild($url_data);

$vendor = $dokumen->createElement("dc:Vendor");
$vendor->appendChild($dokumen->createTextNode($meta['vendor']));
$block->appendChild($vendor);

$created = $dokumen->createElement("dc:Created");
$created->appendChild($dokumen-
>createTextNode($meta['created']));
$block->appendChild($created);

$modif_date = $dokumen->createElement("dc:ModificationDate");
$modif_date->appendChild($dokumen-
>createTextNode($meta['modif_date']));
$block->appendChild($modif_date);

$description = $dokumen->createElement("dc:Description");
$description->appendChild($dokumen-
>createTextNode($meta['description']));
$block->appendChild($description);

$author = $dokumen->createElement("dc:Author");
$author->appendChild($dokumen->createTextNode($meta['author']));
$block->appendChild($author);

$xml_version = $dokumen->createElement("dc:Version");

```

```

$version->appendChild($dokumen-
>createTextNode($meta['version']));
$block->appendChild($version);

$country = $dokumen->createElement("dc:CountryKey");
$country->appendChild($dokumen-
>createTextNode($meta['country']));
$block->appendChild($country);

$activity= $dokumen->createElement("dc:Activity");
$activity->appendChild($dokumen-
>createTextNode($meta['activity']));
$block->appendChild($activity);

$root->appendChild($block);
}

```

Gambar 3.13 Sumber Kode Konversi Array ke XML

```

$dokumen->save("metadata.rdf");

```

Gambar 3.14 Sumber Kode Menyimpan Data ke Sebuah File RDF

RDF yang dibangun menggunakan *DOMDocument* disusun sesuai dengan rancangan RDF sebelumnya. Penamaan *root*, *namespace*, dan *nodes* diatur agar memenuhi kaidah RDF yang *valid*. Root node melingkupi keseluruhan dokumen, *root node* ditunjukkan dengan tag `<rdf:RDF>`. *Element node* yang merupakan bagian yang ditandai dengan tag pembuka dan tag penutup ditunjukkan dengan tag seperti `<dc:XPDLVersion></dc:XPDLVersion>`. Untuk memastikan kevaliditasan dokumen RDF yang telah dibuat dapat diuji dalam RDF Validator pada web <https://www.w3.org/RDF/Validator/>.

3.2.6 SPARQL

Untuk mendapatkan informasi dalam dokumen metadataRDF menggunakan query SPARQL yang telah ditetapkan standardnya oleh *World Wide Web Consortium (W3C)*. *Query* SPARQL memiliki tiga bagian utama yakni *select*, *from*

dan *where*. Bagian *select* berisi variabel yang digunakan untuk mengambil nilai yang diinginkan. Bagian *from* berfungsi untuk menunjuk *resource* yang akan diambil. Untuk bagian ketiga yaitu *where*, digunakan untuk menunjuk ke *subgraph* yang ingin diambil informasinya. RDF yang dibuat sebelumnya telah mengandung subjek, predikat, dan objek. Dibantu dengan *framework* RDF API for PHP (RAP), metadata pada dokumen RDF dapat diambil dengan menggunakan *query* seperti dibawah ini

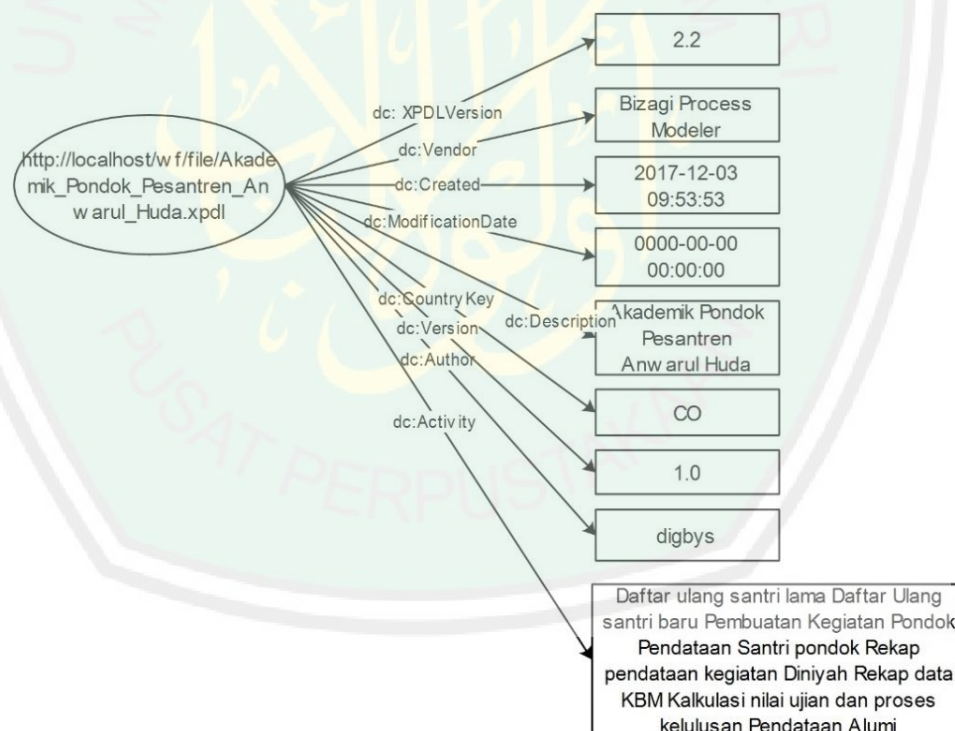
```
SELECT DISTINCT * WHERE {?x dc:XPDLVersion ?XPDLVersion .
                          ?x dc:Vendor ?Vendor .
                          ?x dc:Created ?Created .
                          ?x dc:ModificationDate ?ModificationDate .
                          ?x dc:Description ?Description .
                          ?x dc:Author ?Author .
                          ?x dc:Version ?Version .
                          ?x dc:CountryKey ?CountryKey .
                          ?x dc:URLData ?URLData .
                          ?x dc:Activity ?Activity .
                          FILTER regex (?XPDLVersion, "kata", "i")
                          || regex (?Vendor, "kata", "i")
                          || regex (?Created, "kata", "i")
                          || regex (?ModificationDate, "kata", "i")
                          || regex (?Description, "kata", "i")
                          || regex (?Author, "kata", "i")
                          || regex (?Version, "kata", "i")
                          || regex (?CountryKey, "kata", "i")
                          || regex (?Activity, "kata", "i")}';
```

Gambar 3.15 Kode sumber SPARQL

Query pada Gambar 3.15 dimaksudkan untuk memanggil semua *element* yang telah ditentukan dan *filtering* sesuai dengan masukan *user* pada *string* kata. Selain itu tanda “i” artinya adalah *regular expression* yang dipakai mencocokkan secara *case-insensitive* yakni tidak membedakan huruf kecil dan huruf besar.

3.2.7 Interpretasi

Interpretasi adalah upaya peneliti untuk menemukan makna dari kata yang dikumpulkan untuk menjawab pertanyaan penelitian. Hasil analisis ini dapat disajikan dalam bentuk uraian, tabel, atau grafik. Analisis data yang dilakukan dengan cara memilih, mengelompokkan data yang ada, merangkumnya, kemudian menyajikan dalam bentuk yang mudah dibaca atau dipahami. Dengan menggunakan representasi graph seperti ini kita dapat lebih mudah mengatur ekspresi hanya dengan melihat pada simbol [18]. Interpretasi merupakan inti dari RDF, berikut adalah contoh interpretasi dengan memakai konsep dublin core metadata:



Gambar 3.16 Contoh interpretasi ontologi

3.3. Pengujian Sistem

Pengukuran efektivitas suatu sistem temu kembali dapat dilakukan dengan perhitungan terhadap nilai *recall* (perolehan) dan nilai *precision* (ketepatan). Pada

Workflow Repository ini pengujian dilakukan menggunakan nilai *recall* dan *precision* untuk menguji perbandingan *query* sistem data dan metadata yang telah diinputkan oleh *user*. Dalam Lestari [19], dijelaskan bahwa *recall* adalah proporsi jumlah dokumen yang dapat ditemukan kembali oleh sebuah pencarian di sistem *information retrieval*. Sedangkan *precision* diartikan sebagai kecocokan antara permintaan informasi dengan jawaban terhadap permintaan itu. Dalam pencarian informasi pada sebuah sistem, kepersisan dokumen yang ditemukan adalah relevansi, dengan kata lain seberapa persis atau cocok dokumen tersebut untuk keperluan pencari informasi, bergantung pada seberapa relevan dokumen tersebut bagi si pencari. Dari kedua nilai *recall* dan *precision* akan menentukan nilai *accuracy* yang kemudian didefinisikan sebagai tingkat kedekatan antara nilai prediksi dengan nilai sesungguhnya.

Setelah sistem selesai dirancang dan siap untuk digunakan, maka akan dilakukan uji coba dan evaluasi sistem tersebut. Sehingga dapat diketahui tingkat efektivitas sistem dari hasil evaluasi menggunakan kombinasi *precision* dan *recall*. Nilai *precision* dan *recall* bernilai antara 0 sampai 1. Uji coba pada sistem dilakukan oleh *user* dengan memasukkan beberapa *query* baik pada *repository* data maupun *repository* metadata. Persamaan perhitungan *precision* dan *recall* yang akan digunakan dirumuskan dengan:

		Nilai Sebenarnya	
		TRUE	FALSE
Nilai Prediksi	TRUE	TP (True Positive) Hasil relevan	FP (False Positive) Hasil tidak relevan
	FALSE	FN (False Negative) Hasil relevan tidak ditemukan	TN (True Negative) Hasil tidak relevan yang tidak ditemukan

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

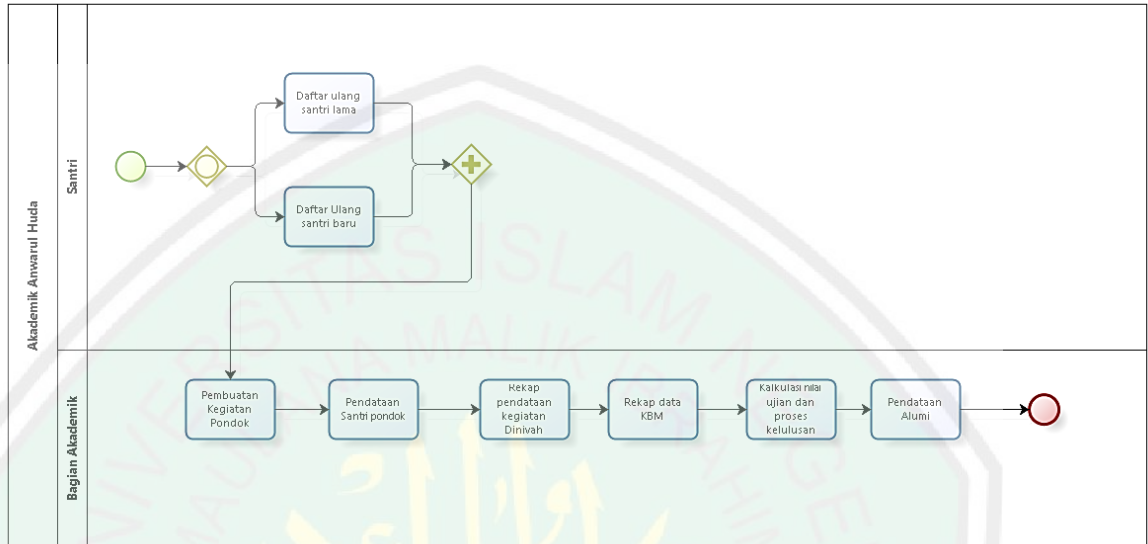
BAB 4

HASIL DAN PEMBAHASAN

Bab ini menjelaskan tentang hasil dan pembahasan dari sistem yang telah dibangun berdasar dari skenario pengujian. Pengujian sistem dilakukan dengan memasukkan model proses bisnis ERP Pondok Pesantren ke dalam *repository data* sebagai data uji, kemudian melakukan parsing, lalu menyimpannya ke dalam *repository metadata*, mencari proses bisnis yang diinginkan dengan *query* hingga dapat diperoleh informasinya dan diunduh. Selanjutnya akan dipaparkan hasil pengujian serta pembahasan dan evaluasi dari kinerja sistem pada penelitian ini.

4.1. Workflow sebagai data uji

Data uji yang digunakan adalah beberapa *workflow* yang terdapat pada ERP Pondok Pesantren dengan ditambahkan beberapa variasi data. *Workflow* yang terdapat pada ERP Pondok Pesantren tersebut diantaranya *workflow* akademik, kesiantrian, penerimaan siswa baru, keuangan, *production planning*, dan sarana prasarana. Data yang telah didapat telah dibentuk dalam format BPMN. Setelah model proses bisnis yang diinginkan telah didapat maka format yang tadinya *.bpmn dikonversi menjadi format *.xpdl menggunakan *tool* Bizagi Process Modeler. Contoh dari file yang masih ekstensi *.bpmn dan *.xpdl dapat dilihat pada Gambar 4.1 dan Gambar 4.2



Gambar 4.1 Pemodelan *workflow* BPMN

```

<?xml version="1.0" encoding="utf-8"?>
<Package xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" Id="65e3dbd1-5ff8-
4a63-87f7-adf2201656ff" Name="Akademik Pondok Pesantren Anwarul
Huda" xmlns="http://www.wfmc.org/2009/XPDL2.2">
  <PackageHeader>
    <XPDLVersion>2.2</XPDLVersion>
    <Vendor>Bizagi Process Modeler.</Vendor>
    <Created>2017-12-03T09:53:53.5263042+07:00</Created>
    <Description>Akademik Pondok Pesantren Anwarul
Huda</Description>
    <Documentation />
  </PackageHeader>
  <RedefinableHeader>
    <Author>digbys</Author>
    <Version>1.0</Version>
    <Countrykey>CO</Countrykey>
  </RedefinableHeader>
  <ExternalPackages />
  <Pools>
    <Pool Id="b87daf56-59c3-4f64-b01e-724dc9cf6e40" Name="Main
Process" Process="b4931480-e6d1-4da5-8968-828b49c1cc32"
BoundaryVisible="false">
      <Lanes />
      <NodeGraphicsInfos>
        <NodeGraphicsInfo ToolId="BizAgi_Process_Modeler"
Height="0" Width="0" BorderColor="-16777216" FillColor="-1">
          <Coordinates XCoordinate="30" YCoordinate="30" />
        </NodeGraphicsInfo>
      </NodeGraphicsInfos>
    </Pool>
  </Pools>

```

```

</Pool>
  <Pool Id="76b3239d-e42f-404c-a2dc-3ca1439ca93d"
Name="Akademik Anwarul Huda" Process="a188c37d-e471-4ccd-90a3-
c5de6cd503e6" BoundaryVisible="true">
  <Lanes>
    <Lane Id="1b324843-a770-4f24-bda9-88e493fedd67"
Name="Santri" ParentPool="76b3239d-e42f-404c-a2dc-3ca1439ca93d">
      <NodeGraphicsInfos>
        <NodeGraphicsInfo ToolId="BizAgi_Process_Modeler"
Height="350" Width="1103" BorderColor="-11513776" FillColor="-
1">
          <Coordinates XCoordinate="50" YCoordinate="0" />
        </NodeGraphicsInfo>
      </NodeGraphicsInfos>
      <ExtendedAttributes />
    </Lane>
    <Lane Id="509269c5-1a6f-433f-bbd2-490c4339732a"
Name="Bagian Akademik" ParentPool="76b3239d-e42f-404c-a2dc-
3ca1439ca93d">
      <NodeGraphicsInfos>
        <NodeGraphicsInfo ToolId="BizAgi_Process_Modeler"
Height="200" Width="1103" BorderColor="-11513776" FillColor="-
1">
          <Coordinates XCoordinate="50" YCoordinate="350" />
        </NodeGraphicsInfo>
      </NodeGraphicsInfos>
      <ExtendedAttributes />
    </Lane>
  </Lanes>
  <NodeGraphicsInfos>
    <NodeGraphicsInfo ToolId="BizAgi_Process_Modeler"
Height="550" Width="1153" BorderColor="-16777216" FillColor="-
1">
      <Coordinates XCoordinate="30" YCoordinate="30" />
    </NodeGraphicsInfo>
  </NodeGraphicsInfos>
</Pool>
</Pools>
<Associations />
<Artifacts />
<WorkflowProcesses>
  <WorkflowProcess Id="b4931480-e6d1-4da5-8968-828b49c1cc32"
Name="Main Process">
    <ProcessHeader>
      <Created>2017-12-03T20:57:15.4168434+07:00</Created>
      <Description />
    </ProcessHeader>
    <RedefinableHeader>
      <Author />
      <Version />
      <Countrykey>CO</Countrykey>
    </RedefinableHeader>
  </WorkflowProcess>
</WorkflowProcesses>

```

Gambar 4.2 Dokumen XPDL

Data yang disimpan tersaji seperti pada Tabel 4.1

Tabel 4.1 Data Workflow

No.	Nama Workflow	Pondok Pesantren	URL
1	Akademik	An Nur	SIA.xpdl
2	Akademik	Al-Rifai	Akademik_Pondok_Pesantren_Al-Rifai.xpdl
3	Akademik	Anwarul Huda	Akademik_Pondok_Pesantren_Anwarul_Huda.xpdl
4	Akademik	Ar-Rohmah	Akademik_Pondok_Pesantren_Ar_Rohmah.xpdl
5	Akademik	Z	Akademik_Pondok_Pesantren_Z.xpdl
6	Kesantrian	Z	Kesantrian_Z.xpdl
7	Kesantrian	Y	Kesantrian_Y.xpd
8	Kesantrian	Al-Rifai	Kesantrian_Pondok_Pesantren Al-Rifai.xpdl
9	Kesantrian	Luhur	Kesantrian_Luhur.xpdl
10	Keuangan	An-Nur	Keuangan.xpdl
12	Penerimaan Santri Baru	Luhur	PSB_Pondok_Pesantren_Luhur.xpdl
13	Penerimaan Santri Baru	Anwarul Huda	PSB_Pondok_Pesantren_Anwarul_Huda.xpdl
14	Penerimaan Santri Baru	An Nur 1	PSB_Pondok_Pesantren_An_Nur_1.xpdl
15	Penerimaan Siswa Baru	SMPN 13 Malang	PSB_SMPN_13_Malang.xpdl
16	Sarana Prasarana	Anwarul Huda	Sarana_Prasarana_Pondok_Pesantren_Anwarul_Huda.xpdl
17	Sarana Prasarana	Al-Rifai	Sarana_Prasarana_Pondok_Pesantren_Al_Rifai.xpdl
18	Sarana Prasarana	Luhur	Sarana_Prasarana_Pondok_Pesantren_Luhur.xpdl
19	Sarana Prasarana	Z	Sarana_Prasarana_Z.xpdl
20.	Kegiatan Pengurus I	An-Nur	Kegiatan_ar_rahman.xpdl
21	Production Planning	An-Nur	Production_planning.xpdl

4.2. Langkah Uji Coba

Pengujian pada Workflow Repository yang dibangun pada penelitian dilakukan untuk mengetahui tingkat keefektifan sistem. Sebelum masuk pada pengujian sistem, terlebih dahulu dilakukan uji coba antarmuka (*interface*) sistem dan uji coba *query* data & *query* metadata. Pada Workflow Repository, terdapat tiga buah menu pada sebelah kiri tampilan yaitu Form dan Tabel. Menu Form memiliki submenu Form Data Workflow. Sedangkan menu Tabel memiliki submenu Tabel Data dan Tabel Metadata. Tampilan dari halaman awal Workflow Repository adalah sebagai berikut:



Gambar 4.3 Halaman awal *Workflow Repository*

Skenario awal yaitu memasukkan data *workflow* dan mengupload file XPDL. Data *workflow* yang diinputkan berupa Nama *Workflow* dan nama Pondok Pesantren. Penyimpanan data tidak bisa dilanjutkan jika kedua *text field* belum diisi. Pada tahap ini dicoba memasukkan Akademik sebagai Nama *Workflow* , Anwarul Huda sebagai nama Pondok Pesantren, dan

Akademik_Pondok_Pesantren_Anwarul_Huda.xpdl sebagai file yang diunggah.

Gambar 4.4 berikut merupakan inputan data *workflow*.

Gambar 4.4 Form Data *Workflow*

Data yang telah tersimpan dapat dilihat pada submenu Tabel Data Workflow. Pada tabel ini disajikan data *workflow* yang telah tersimpan pada database MySQL. Terdapat beberapa kolom yakni Nama *Workflow*, Pondok Pesantren, URL sebagai nama *file*, Lihat sebagai *link* untuk melihat XPDLL melalui *browser*, dan Aksi yang meliputi hapus data, edit data, dan parsing. *Interface* Tabel Data *Workflow* ditunjukkan pada Gambar 4.5

No.	Nama Workflow	Pondok Pesantren	URL	Lihat	Aksi
1	Akademik	An-Nur	SIA.xpdl	View	[Delete] [Edit] [Parsing]
2	Akademik	Z	Akademik_Pondok_Pesantren_Z.xpdl	View	[Delete] [Edit] [Parsing]
3	Akademik	Al-Rifai	Akademik_Pondok_Pesantren_Al-Rifai.xpdl	View	[Delete] [Edit] [Parsing]
4	Akademik	Anwarul Huda	Akademik_Pondok_Pesantren_Anwarul_Huda.xpdl	View	[Delete] [Edit] [Parsing]
5	Akademik	Ar-Rohmah	Akademik_Pondok_Pesantren_Ar_Rohmah.xpdl	View	[Delete] [Edit] [Parsing]

Gambar 4.5 Tabel Data *Workflow*

Proses *parsing* dapat dilakukan jika *user* memilih tombol *Parsing*. Tidak semua data workflow yang disimpan akan diambil metadatanya agar meminimalisir *memory* yang terpakai dengan percuma, maka disediakan tombol *parsing* pada masing-masing data. *Parsing* data workflow menggunakan XML DOM Parser dengan mengambil *element* yang telah ditentukan kemudian menyimpan metadata dari *file workflow* dan otomatis tersusun dalam *file *.rdf* pada *repository* metadata.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description
rdf:about="http://localhost/wf/file/Akademik_Pondok_Pesantren_Anwarul_Huda.xpdl">
<dc:URLData>Akademik_Pondok_Pesantren_Anwarul_Huda.xpdl</dc:URLData>
  <dc:XPDLVersion>2.2</dc:XPDLVersion>
  <dc:Vendor>Bizagi Process Modeler.</dc:Vendor>
  <dc:Created>2017-12-03 09:53:53</dc:Created>
  <dc:ModificationDate>0000-00-00
00:00:00</dc:ModificationDate>
  <dc:Description>Akademik Pondok Pesantren Anwarul Huda</dc:Description>
  <dc:Author>digbys</dc:Author>
  <dc:Version>1.0</dc:Version>
  <dc:CountryKey>CO</dc:CountryKey>
  <dc:Activity> Daftar ulang santri lama Daftar Ulang santri baru Pembuatan Kegiatan Pondok Pendataan Santri pondok Rekap pendataan kegiatan Diniyah Rekap data KBM Kalkulasi nilai ujian dan proses kelulusan Pendataan Alumni </dc:Activity>
  </rdf:Description>
</rdf:RDF>
```

Gambar 4.6 Dokumen RDF yang dihasilkan

Semua metadata dijadikan satu dalam metadata.rdf agar saat dilakukan proses *query* maka semua parsingan bisa terambil. Satu dokumen metadata.rdf akan selalu dibangun ulang setiap ada data yang diparsing, ini berarti RDF ditulis berulang secara dinamis sesuai dengan kebutuhan penyimpanan. Penulisan RDF juga tidak dapat dilakukan secara asal seperti dokumen XML yang lain. Penamaan *namespace*

penting untuk membuat dokumen RDF benar terpanggil. Untuk mengecek validitas dokumen RDF dapat menggunakan RDF Validator yang ada pada web <https://www.w3.org/RDF/Validator/>. Hasil dari validasi dokumen RDF jika benar akan seperti pada Gambar 4.7 berikut

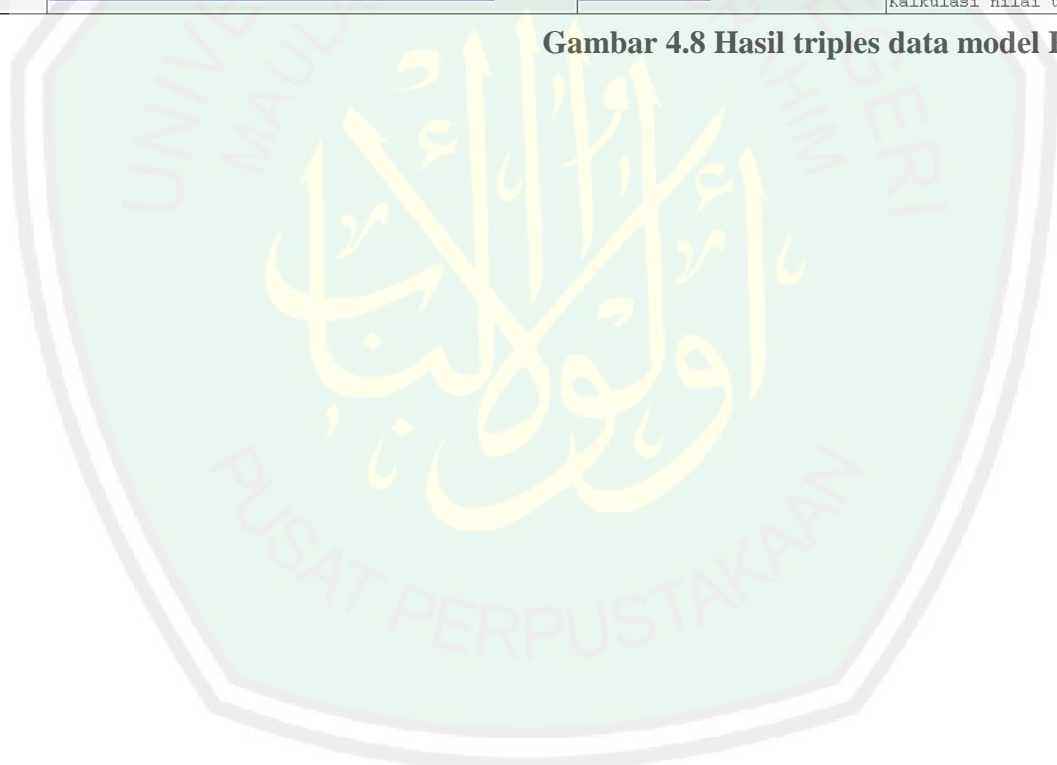


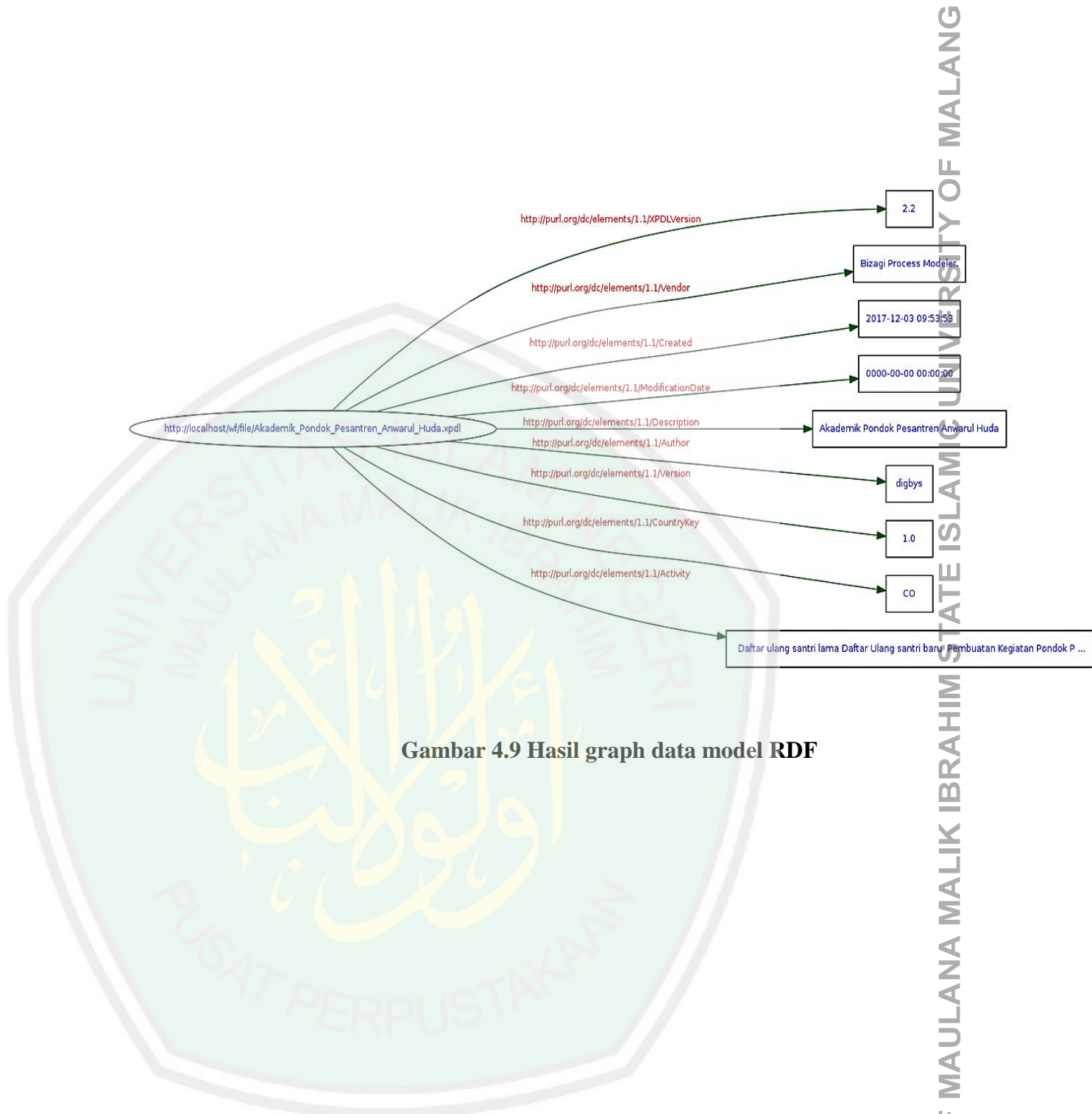
Gambar 4.7 Validasi RDF Sukses

Triples of the Data Model

Number	Subject	Predicate	Object
1	http://localhost/wf/file/Akademik Pondok Pesantren Anwarul Huda.xpdl	http://purl.org/dc/elements/1.1/XPDLVersion	"2.2"
2	http://localhost/wf/file/Akademik Pondok Pesantren Anwarul Huda.xpdl	http://purl.org/dc/elements/1.1/Vendor	"Bizagi Process Modeler."
3	http://localhost/wf/file/Akademik Pondok Pesantren Anwarul Huda.xpdl	http://purl.org/dc/elements/1.1/Created	"2017-12-03 09:53:53"
4	http://localhost/wf/file/Akademik Pondok Pesantren Anwarul Huda.xpdl	http://purl.org/dc/elements/1.1/ModificationDate	"0000-00-00 00:00:00"
5	http://localhost/wf/file/Akademik Pondok Pesantren Anwarul Huda.xpdl	http://purl.org/dc/elements/1.1/Description	"Akademik Pondok Pesantren Anwarul Huda"
6	http://localhost/wf/file/Akademik Pondok Pesantren Anwarul Huda.xpdl	http://purl.org/dc/elements/1.1/Author	"digbys"
7	http://localhost/wf/file/Akademik Pondok Pesantren Anwarul Huda.xpdl	http://purl.org/dc/elements/1.1/Version	"1.0"
8	http://localhost/wf/file/Akademik Pondok Pesantren Anwarul Huda.xpdl	http://purl.org/dc/elements/1.1/CountryKey	"cc"
9	http://localhost/wf/file/Akademik Pondok Pesantren Anwarul Huda.xpdl	http://purl.org/dc/elements/1.1/Activity	"Daftar ulang santri lama Daftar Ulang Santri baru Pembuatan Kegiatan Pondok Pendataan Santri pondok Rekap pendataan Kegiatan Diniyah Rekap data KBM Kalkulasi nilai ujian dan proses kelulusan Pendataan Alumni"

Gambar 4.8 Hasil triples data model RDF





Gambar 4.9 Hasil graph data model RDF

Gambar Interpretasi hasil validasi RDF menunjukkan bahwa suatu *subject* http://localhost/wf/file/Akademik_Pondok_Pesantren_Anwarul_Huda.xpdl memiliki *predicate* dan *object* seperti pada penjelasan Tabel 4.2

Tabel 4.2 Penjelasan interpretasi

<i>Predicate</i>	<i>Menerangkan</i>	<i>Object</i>
http://purl.org/dc/elements/1.1/XPDLVersion	Menggunakan versi XPDL	2.2
http://purl.org/dc/elements/1.1/Vendor	Dibangun menggunakan vendor	Bizagi Process Modeler
http://purl.org/dc/elements/1.1/Created	Dibuat pada tanggal dan waktu	2017-12-03 09:53:53
http://purl.org/dc/elements/1.1/ModificationDate	Dimodifikasi terakhir pada tanggal dan waktu	0000-00-00 00:00:00
http://purl.org/dc/elements/1.1/Description	Memiliki deskripsi	Akademik Pondok Pesantren Anwarul Huda
http://purl.org/dc/elements/1.1/Author	Dibuat oleh	Digbys
http://purl.org/dc/elements/1.1/Version	Versi XPDL yang digunakan	1.0
http://purl.org/dc/elements/1.1/CountryKey	Menggunakan <i>country key</i>	CO
http://purl.org/dc/elements/1.1/Activity	Memiliki <i>activity</i>	Daftar ulang santri lama Daftar Ulang santri baru Pembuatan Kegiatan Pondok Pendataan Santri pondok Rekap pendataan kegiatan Diniyah Rekap data KBM Kalkulasi nilai ujian dan proses kelulusan Pendataan Alumni

Dokumen metadata.rdf yang berhasil divalidasi dapat ditunjukkan dengan *Triples* Data Model pada dan *Graph* Data Model pada Gambar 4.6. Jika diambil contoh dalam satu buah element dapat dijelaskan seperti URI sebagai subject memiliki *object* digbys dan berpredicate *Author*. Dalam kesatuan kalimat dibaca dengan

http://localhost/wf/file/Akademik_Pondok_Pesantren_Anwarul_Huda.xpdl dibuat oleh *author* bernama digbys. Selanjutnya RDF ditampilkan dalam tabel metadata *Workflow Repository* seperti pada gambar

No.	XPDL Version	Vendor	Created	Modification Date	Description	Author	Version	Country Key	Aksi
1	2.2	Bizagi Process Modeler.	2016-11-05 00:39:01	2016-11-07 18:15:09	Diagram 1	navlras	1.0	CO	Download
2	2.2	Bizagi Process Modeler.	2016-11-13 19:42:02	2016-11-14 01:44:22	Diagram 1	Ahmad	2.0	CO	Download
3	2.2	Bizagi Process Modeler.	2016-10-22 16:16:21	2016-11-14 12:37:34	Diagram 1	PIPIIT	1.0	CO	Download
4	2.2	Bizagi Process Modeler.	2017-12-03 20:46:07	0000-00-00 00:00:00	Sarana Prasarana Pondok Pesantren Anwarul Huda	digbys	1.0	CO	Download
5	2.2	Bizagi Process Modeler.	2017-12-03 14:43:29	0000-00-00 00:00:00	PSB Pondok Pesantren An Nur 1	digbys	1.0	CO	Download
6	2.2	Bizagi Process Modeler.	2017-12-03 11:08:41	0000-00-00 00:00:00	Kesantrian Pondok Pesantren Al-Rifai	digbys	1.0	CO	Download
















Gambar 4.10 Tabel metadata workflow

Perbandingan *query* pada *repository* data dan *repository* metadata akan menghasilkan perbedaan hasil keluaran. Proses *query* dalam pencocokan menggunakan *exact string-matching* yaitu pencocokan *string* secara tepat dengan susunan karakter dalam *string* yang dicocokkan memiliki jumlah maupun urutan karakter dalam *string* yang sama. Bagian ini bermanfaat jika *user* ingin mencari

string dalam dokumen yang sama persis dengan *string* masukan. *Interface* proses *query* masukan “akademik” pada repository data ditunjukkan pada Gambar 4.11 dan *repository* metadata pada Gambar 4.12

Table Data Workflow

Hasil pencarian dari : akademik

No.	Nama Workflow	Pondok Pesantren	URL	Lihat	Aksi
1	Akademik	Al-Rifai	Akademik_Pondok_Pesantren_Al-Rifai.xpdl	View	  
2	Akademik	Anwarul Huda	Akademik_Pondok_Pesantren_Anwarul_Huda.xpdl	View	  
3	Akademik	Ar-Rohmah	Akademik_Pondok_Pesantren_Ar-Rohmah.xpdl	View	  
4	Akademik	Z	Akademik_Pondok_Pesantren_Z.xpdl	View	  
5	Akademik	An-Nur	SIA.xpdl	View	  

Gambar 4.11 Query kata “akademik” pada repository data

Tabel Metadata Workflow

Hasil pencarian dari : akademik

No.	XPDL Version	Vendor	Created	Modification Date	Description	Author	Version	Country Key	Aksi
1	2.2	Bizagi Process Modeler.	2017-12-03 09:53:53	0000-00-00 00:00:00	Akademik Pondok Pesantren Anwarul Huda	digbys	1.0	CO	Download
2	2.2	Bizagi Process Modeler.	2016-10-12 12:52:18	2016-10-26 14:02:24	Diagram 1	Nurika	1.0	CO	Download
3	2.2	Bizagi Process Modeler.	2017-12-03 10:56:53	0000-00-00 00:00:00	Akademik Pondok Pesantren Z	digbys	1.0	CO	Download
4	2.2	Bizagi Process Modeler.	2017-12-03 09:19:32	0000-00-00 00:00:00	Akademik Pondok Pesantren Al-Rifai	digbys	1.0	CO	Download
5	2.2	Bizagi Process Modeler.	2017-12-03 10:03:25	0000-00-00 00:00:00	Akademik Pondok Pesantren Ar Rohmah	digbys	1.0	CO	Download
6	2.2	Bizagi Process Modeler.	2016-10-22 16:16:21	2016-11-14 12:37:34	Diagram 1	PIPIT	1.0	CO	Download

Gambar 4.12 Query kata “akademik” pada repository data

4.3. Hasil Uji Coba

Mengacu pada skenario uji coba yang telah dijelaskan, selanjutnya akan ditampilkan hasil pengujian sistem. Uji coba sistem dalam *workflow repository* ini meliputi *query* pada *repository* data dan *repository* metadata. Nilai *precision* dan *recall* akan menunjukkan nilai relevansi dokumen yang ditemukan melalui pencarian. Sebagai contoh dari hasil pada penelitian ini diambil *keyword* “akademik” dengan melakukan *query* pencarian pada Tabel Data Workflow dan Tabel Metadata Workflow. Hasil pengujian didapat tabel yang keluar seperti pada Tabel 4.3

Tabel 4.3 Pencarian katakunci dalam repository

No.	Tabel DataWorkflow	Tabel Metadata Workflow	Match
1.	Akademik Pondok Pesantren Al-Rifai	Akademik Pondok Pesantren Al-Rifai	✓
2.	Akademik Pondok Pesantren Anwarul Huda	Akademik Pondok Pesantren Anwarul Huda	✓
3.	Akademik Pondok Pesantren Ar-Rahman	Akademik Pondok Pesantren Ar-Rahman	✓
4.	Akademik Pondok Pesantren Z	Akademik Pondok Pesantren Z	✓
5.	Akademik Pondok Pesantren An-Nur	Akademik Pondok Pesantren An-Nur	✓
6.	-	Production Planning An-Nur	✗

Tabel 4.4 Perhitungan manual pada katakunci “akademik”

	Ret	TP	FP	FN	TN	Precision	Recall	Accuracy
Tabel Data Workflow	5	5	0	0	15	$\frac{5}{5+0} = 1$	$\frac{5}{5+0} = 1$	$\frac{5+15}{5+15+0+0} = 1$
Tabel Metadata Workflow	6	5	1	0	14	$\frac{5}{5+1} = 0,833$	$\frac{5}{5+0} = 1$	$\frac{5+14}{5+14+1+0} = 0,95$

Tabel data *workflow* menunjukkan hasil yang keluar hanya yang mengandung kata “akademik”, sedangkan pada tabel metadata *workflow* hasil yang keluar lima *workflow* yang sesuai dan satu *workflow* yang tidak sesuai yakni Production Planning An-Nur. Penyebabnya adalah ada unsur kata “akademik” dalam *workflow*

tersebut namun tidak relevan dengan pencarian. Hasil ini hanya berdasar pada satu kata kunci, masih ada 20 kata kunci yang akan diujikan. Data uji *query* beserta perhitungan nilai *recall*, *precision*, dan *accuracy* disajikan dalam Tabel 4.5 dan

Tabel 4.6

Tabel 4.5 Hasil uji *query data repository*

No.	Keyword	Ret	TP	FP	FN	TN	P	R	A
1.	Sarana Prasarana	4	4	0	0	16	1	1	1
2.	Akademik	5	5	0	0	15	1	1	1
3.	Penerimaan santri baru	3	3	0	0	17	1	1	1
4.	Kesantrian	4	4	0	0	16	1	1	1
5.	Santri	7	7	0	0	13	1	1	1
6.	Pembayaran	0	0	0	0	20	0	0	1
7.	Wisuda	0	0	0	0	20	0	0	1
8.	An Nur	2	2	0	0	18	1	1	1
9.	Anwarul Huda	3	3	0	0	17	1	1	1
10.	Daftar ulang	0	0	0	0	20	0	0	1
11.	Validasi	0	0	0	0	20	0	0	1
12.	Rekap data	0	0	0	0	20	0	0	1
13.	Nurika	0	0	0	0	20	0	0	1
14.	2016	0	0	0	0	20	0	0	1
15.	Ujian	0	0	0	0	20	0	0	1
16.	Diniyah	0	0	0	0	20	0	0	1
17.	Pegawai	0	0	0	0	20	0	0	1
18.	Tahun takwim	0	0	0	0	20	0	0	1
19.	Akuntansi	0	0	0	0	20	0	0	1
20.	Absensi Santri	0	0	0	0	20	0	0	0
Rata-rata							0,35	0,35	1

Ret = Retrieval, TP = True Positive, FP = False Positive, FN = False Negative, TN = True Negative, P = Precision, R = Recall, A = Accuracy

Tabel 4.6 Hasil uji query metadata repository

No.	Keyword	Ret	TP	FP	FN	TN	P	R	A
1.	Sarana Prasarana	4	4	0	0	16	1	1	1
2.	Akademik	6	5	1	0	14	0,833	1	0,95
3.	Penerimaan santri baru	0	0	0	4	16	0	0	0,8
5.	Kesantrian	5	4	1	0	15	0,8	1	0,95
5.	Santri	13	13	0	0	7	1	1	1
6.	Pembayaran	1	1	0	0	19	1	1	1
7.	Wisuda	3	3	0	0	17	1	1	1
8.	An Nur	1	1	0	1	18	0,5	0,666	0,95
9.	Anwarul Huda	3	3	0	0	17	1	1	1
10.	Daftar ulang	4	4	0	0	16	1	1	1
11.	Validasi	4	4	0	0	16	1	1	1
12.	Rekap data	6	6	0	0	14	1	1	1
13.	Nurika	1	1	0	0	19	1	1	1
14.	2016	5	5	0	0	15	1	1	1
15.	Ujian	5	5	0	0	15	1	1	1
16.	Diniyah	6	5	1	0	14	0,833	1	0,95
17.	Pegawai	5	1	4	0	15	0,2	1	0,8
18.	Tahun Takwim	1	1	0	0	19	1	1	1
19.	Akuntansi	1	1	0	0	19	1	1	1
20.	Absensi Santri	2	2	0	1	17	1	0,66	0,95
Rata-rata							0,877	0,921	0,968

Ret = Retrieval, TP = True Positive, FP = False Positive, FN = False Negative, TN = True Negative, P = Precision, R = Recall, A = Accuracy

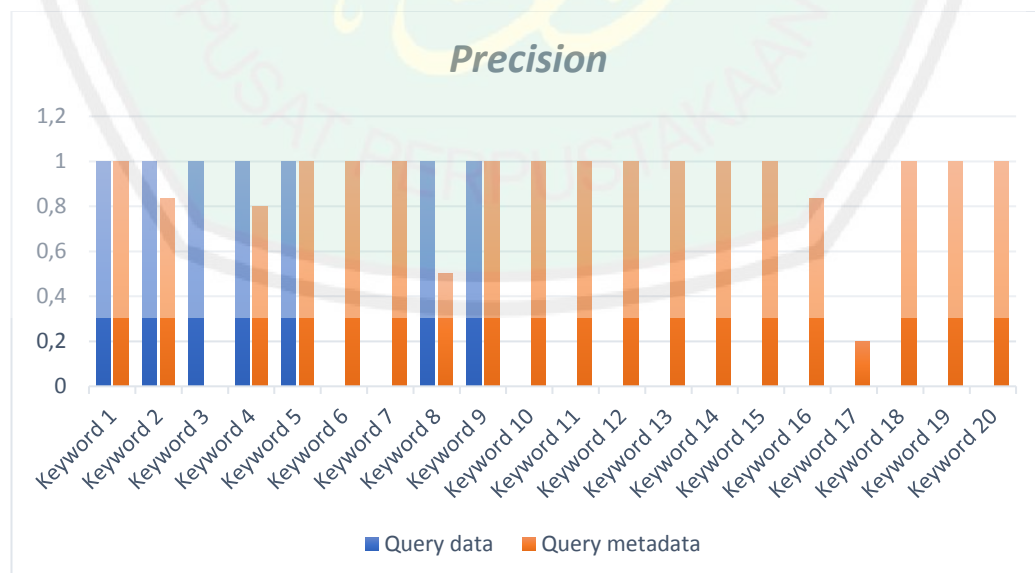
Tabel 4.5 menunjukkan hasil perhitungan *precision* dan *recall* dari 20 *query* yang telah dilakukan uji coba pada *repository* data. Berdasar hasil tabel perhitungan tersebut maka diketahui nilai rata-rata *precision* adalah 0,35, nilai rata-rata *recall* adalah 0,35, dan nilai rata-rata *accuracy* adalah 1. Jika digambarkan dalam nilai presentase didapat nilai rata-rata *precision* sebesar 35%, nilai rata-rata *recall* sebesar 35%, dan nilai rata-rata *accuracy* sebesar 100% juga.

Sedangkan pada Tabel 4.6 menunjukkan hasil perhitungan *precision* dan *recall* dari 20 *query* yang sama dan telah dilakukan uji coba pada *repository*

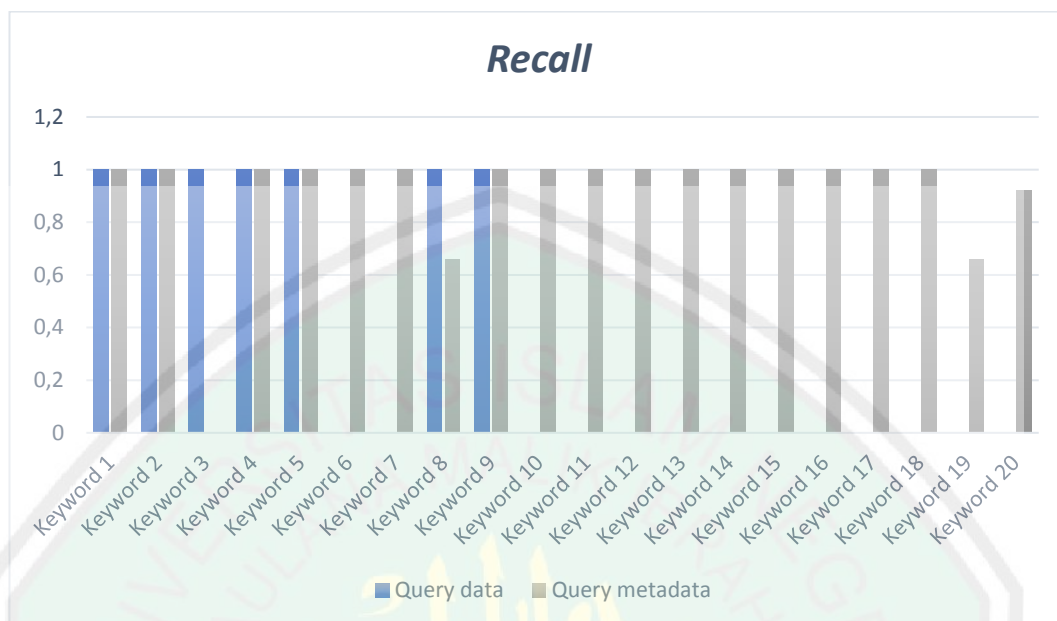
metadata. Berdasar hasil tabel perhitungan tersebut maka diketahui nilai rata-rata *precision* adalah 0,877, nilai rata-rata *recall* adalah 0,921, dan nilai rata-rata *accuracy* adalah 0,968. Jika digambarkan dalam nilai presentase didapat nilai rata-rata *precision* sebesar 87,7%, nilai rata-rata *recall* sebesar 92,1%, dan nilai rata-rata *accuracy* sebesar 96,8%.

4.4. Pembahasan

Merujuk pada hasil uji coba yang telah dilakukan sebelumnya dapat diketahui bahwa terdapat perbedaan yang cukup signifikan antara pencarian kata *exact string-matching* yang dimasukkan oleh *user* pada *repository* data dan *repository* metadata. Dengan menggunakan 20 *keyword testing* rata-rata nilai *precision* dan *recall* pada *repository* data hanya mampu mengembalikan nilai rata-rata titik temu sebesar 35%. Berbeda dengan nilai rata-rata titik temu kembali pada *repository* metadata yang mampu menunjukkan nilai diatas 50%.



Gambar 4.13 Grafik perbandingan nilai *precision*



Gambar 4.14 Grafik perbandingan nilai *recall*

Gambar 4.13 menunjukkan grafik perbandingan antara nilai *precision* terhadap 20 *keyword* uji coba dengan dilakukan pada *repository* data dan *repository* metadata. Sedangkan Gambar 4.14 menunjukkan perbandingan antara nilai *recall* terhadap 20 *keyword* uji coba pada *repository* data dan *repository* metadata. Sistem *workflow repository* metadata dinilai lebih mumpuni daripada *repository* data jika digunakan dalam proses *discovery* atau temu kembali informasi. Berdasarkan teori yang diungkapkan oleh Lancaster dalam Pendit [20] bahwa efektivitas sistem temu kembali informasi dikategorikan menjadi dua yakni efektif jika nilainya diatas 50% dan tidak efektif jika nilainya dibawah 50%.

Terdapat beberapa analisa terhadap *query* pada *repository* data yaitu menggunakan *exact string matching* dengan bahasa *query* SQL dan fungsi *regex* LIKE ditemukan dokumen bila masukan sama persis dengan data yang disajikan. Pada tipe data CHAR, VARCHAR, atau TEXT maka LIKE akan bersifat *case*

insensitif yakni huruf besar dan kecil dianggap sama dan tidak dibedakan. Pencarian menggunakan *query* data juga terbatas pada *keyword* yang disediakan yaitu masukan identitas data berupa nama *workflow* dan nama pondok pesantren. Ini menyebabkan banyak *keyword* uji coba bernilai 0 karena tidak ditemukan kata masukan yang sama persis dengan data yang ada.

Sedangkan pada pencarian dokumen menggunakan *query* metadata dapat menemukan hasil yang cukup baik dengan nilai rata-rata diatas 50%. Sama halnya dengan *query* pada *repository* data, *repository* metadata juga menggunakan pencarian *exact string matching* yang mengharuskan masukan dan hasil keluaran yang mengandung unsur kata sama persis dengan bahasa SPARQL dan fungsi *rege*. Namun pada kenyataannya untuk hasil *query* menggunakan *query* metadata jauh lebih unggul dibanding *repository* data karena mengandung *element-element* yang dapat dijadikan *keyword* pencarian. Umumnya *keyword* pencarian sebuah dokumen XPDL dapat dilihat dari *element description*, *author*, dan *activity*. Tetapi tidak menutup kemungkinan dapat mencari *keyword* berdasarkan metadata XPDL yang lain. Meski begitu tidak semua dokumen XPDL yang ditemukan pada *repository* metadata relevan dengan apa yang diinginkan oleh *user*.

Analisa pada *keyword* “akademik” dalam *repository* metadata menunjukkan bahwa nilai retrieval adalah 6, nilai TP adalah 5, nilai FP adalah 1, nilai FN adalah 0, dan nilai TN adalah 14 memiliki ketimpangan nilai antara apa yang terambil kembali dan apa yang relevan. Penyebabnya adalah terdapat kata “akademik” pada dokumen *production_planning.xpdl* yang memiliki unsur kata yang dicari namun tidak sesuai dengan apa yang diinginkan *user*.

Keyword “penerimaan santri baru” dalam *repository* metadata menunjukkan bahwa nilai *retrieval* adalah 0, nilai TP adalah 0, nilai FP adalah 0, nilai FN adalah 4, dan nilai TN adalah 16 memiliki ketimpangan nilai antara apa yang terambil kembali dan total dokumen yang harusnya ditemukan. Penyebabnya adalah tidak adanya metadata yang tepat sama persis dengan masukan kata “penerimaan santri baru” yang diharapkan oleh *user*. Padahal terdapat 4 dokumen yang relevan dengan masukan tersebut, ini dikarenakan pada metadata XPDL menggunakan kata “PSB” yakni singkatan dari kata masukan “penerimaan santri baru”

Keyword “kesantrian” dalam *repository* metadata menunjukkan bahwa nilai *retrieval* adalah 5, nilai TP adalah 4, nilai FP adalah 1, nilai FN adalah 0, dan nilai TN adalah 15 memiliki ketimpangan nilai antara apa yang terambil kembali dan apa yang relevan. Penyebabnya adalah terdapat kata “kesantrian” pada dokumen keuangan.xpdl yang memiliki unsur kata yang dicari namun tidak sesuai dengan apa yang diinginkan *user*.

Keyword “An Nur” dalam *repository* metadata menunjukkan bahwa nilai *retrieval* adalah 1, nilai TP adalah 1, nilai FP adalah 0, nilai FN adalah 1, dan nilai TN adalah 18 memiliki ketimpangan nilai antara apa yang terambil kembali dan apa yang relevan. Penyebabnya adalah pada dokumen kegiatan_ar_rahman.xpdl yang seharusnya ditemukan pada pencarian terdapat missing kata “An Nur” dalam proses pendeskripsian dokumen XPDL.

Keyword “diniyah” dalam *repository* metadata menunjukkan bahwa nilai *retrieval* adalah 6, nilai TP adalah 5, nilai FP adalah 1, nilai FN adalah 0, dan nilai TN adalah 14 memiliki ketimpangan nilai antara apa yang terambil kembali dan apa yang relevan. Penyebabnya adalah terdapat kata “diniyah” pada dokumen Akademik_Pondok_Pesantren_Anwarul_Huda.xpdl yang memiliki unsur kata yang dicari namun tidak sesuai dengan apa yang diinginkan *user*.

Keyword “pegawai” dalam *repository* metadata menunjukkan bahwa nilai *retrieval* adalah 5, nilai TP adalah 1, nilai FP adalah 4, nilai FN adalah 0, dan nilai TN adalah 15 memiliki ketimpangan nilai antara apa yang terambil kembali dan apa yang relevan. Penyebabnya adalah terdapat kata “pegawai” pada keempat dokumen lain yang ditemukan memiliki unsur kata yang dicari namun tidak sesuai dengan apa yang diinginkan *user*.

Keyword “Absensi santri” dalam *repository* metadata menunjukkan bahwa nilai *retrieval* adalah 2, nilai TP adalah 2, nilai FP adalah 0, nilai FN adalah 1, dan nilai TN adalah 17 memiliki ketimpangan nilai antara apa yang terambil kembali dan apa yang relevan. Penyebabnya adalah pada dokumen yang harusnya ditemukan namun tidak terjaring *query* memiliki makna adanya aktivitas absensi santri didalamnya namun tidak tepat sama persis dengan keyword masukan *user*.

Selain ketimpangan diatas jika rasio yang dihasilkan dari *repository* data dan metadata atas nilai *Retrieve:TP:FP:FN:TN* adalah $\frac{n}{20} : \frac{n}{20} : 0 : 0 : \frac{20-n}{20}$ dengan rasio *precision:recall* sebesar 1:1 maka kondisi ini dapat dikatakan efektif terhadap pencarian masukan dan yang diharapkan *user*.

Sebagaimana *workflow repository* dibangun untuk memenuhi kebutuhan *user* atas penyimpanan model proses bisnis agar suatu saat nanti bisa dipakai kembali dalam proses pembuatan *multitenant* lain pada ERP pondok pesantren. Dengan menggunakan ontologi berupa RDF yang menampung ilmu pengetahuan lebih diharapkan mampu memenuhi keinginan dan harapan dari *user*. Seperti yang dijelaskan pada surat An-Nahl ayat 43:

وَمَا أَرْسَلْنَا مِنْ قَبْلِكَ إِلَّا رِجَالًا نُوْحِي إِلَيْهِمْ فَسَأَلُوا أَهْلَ الذِّكْرِ
 إِنْ كُنْتُمْ لَا تَعْمُونَ ﴿٤٣﴾

Artinya: “Dan Kami tidak mengutus sebelum kamu, kecuali orang-orang lelaki yang Kami beri wahyu kepada mereka; maka bertanyalah kepada orang yang mempunyai pengetahuan jika kamu tidak mengetahui” (An-Nahl:43)

Pengertian dari ayat diatas adalah bahwa (dan Kami tidak mengutus sebelum kamu kecuali orang-orang lelaki yang Kami beri wahyu kepada mereka) bukannya malaikat melainkan (maka bertanyalah kepada orang yang mempunyai pengetahuan) yakni para ulama yang ahli dalam kitab Taurat dan kitab Injil (jika kalian tidak mengetahui) hal tersebut, mereka pasti mengetahuinya karena kepercayaan kalian kepada mereka lebih dekat daripada kepercayaan kalian terhadap Nabi Muhammad SAW.

Kemudian dengan menyediakan penyimpanan dua lapis yakni *repository* data dengan database MySQL dan *repository* metadata dengan *database* RDF akan memudahkan *user* dalam melakukan pencarian kembali dokumen XPDL yang

diinginkan dan sesuai dengan yang diharapkan. Sebagaimana dalam surat Al-Kahfi ayat 84 yang menyatakan bahwa Allah akan memudahkan jalan dalam menghadapi sesuatu dan jalan untuk mencapai yang dikehendaknya,

إِنَّا مَكِّنَّا لَهُ فِي الْأَرْضِ وَءَانِئِنهٗ مِنْ كُلِّ شَيْءٍ سَبَبًا ﴿٨٤﴾

Artinya :”*Sesungguhnya Kami telah memberi kekuasaan kepadanya di(muka) bumi, dan Kami telah memberikan kepadanya jalan (jalan untuk mencapai) segala sesuatu*” (Al-Kahfi:84)

Pengertian dari ayat tersebut adalah bahwa (sesungguhnya Kami telah memberi kekuasaan kepadanya di muka bumi) dengan memudahkan perjalanan baginya di muka bumi ini(dan Kami telah memberikan kepadanya di dalam menghadapi segala sesuatu) yang ia perlukan (jalan untuk mencapainya) jalan yang dapat mengantarkannya kepada yang dikehendaknya.

BAB 5

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Kesimpulan dari penelitian yang telah dilakukan adalah sebagai berikut:

1. Metadata sebuah file dapat dibangun dengan menggunakan *Resource Definition Framework* (RDF). Karena RDF ini merupakan salah satu jenis ontologi sederhana maka penerapan lebih mudah daripada jenis ontologi lain. RDF harus men gandung *subject*, *predicate*, dan *object*.
2. RDF dapat digunakan jika penulisannya benar sesuai dan dapat divalidasi dalam RDF Validator.
3. Perbedaan *repository* data dan metadata terlihat dengan adanya *query*. Data *workflow* hanya dapat dipanggil berdasarkan nama dan pondok pesantren yang diinputkan. Sedangkan dengan metadata *workflow*, kosakata metadata yang terambil dapat memudahkan pengguna untuk mencari file yang diinginkan.
4. Berdasarkan hasil perhitungan uji coba, didapatkan nilai *recall*, *precision*, dan *accuracy* pada *repository* data sebesar *recall* 35%, *precision* 35%, dan *accuracy* 100%. Berbeda halnya dengan *repository* data, *repository* metadata menunjukkan performa lebih baik dengan nilai *recall* 92,1%, *precision* 87,7%, dan *accuracy* 96,8%.
5. Beberapa hal yang dapat mempengaruhi nilai dokumen relevan yang tidak ditemukan pada *repository* metadata adalah dibutuhkan penulisan deskripsi yang lengkap terhadap dokumen, adanya kata yang sama maknanya namun tidak terjaring *query*. Selain itu terdapat dokumen tidak relevan yang ditemukan

dikarenakan ada unsur kata yang dimasukkan namun tidak sesuai dengan keinginan *user*.

5.2. Saran

Berdasarkan penelitian yang dilakukan terdapat beberapa kekurangan. Oleh karena itu diharapkan adanya penelitian lebih lanjut meliputi:

1. Perlu adanya dekomposisi *workflow* agar saat suatu *tenant* menginginkan keterbaruan dari sebuah *workflow*, pengembang hanya perlu menambahkan *fragment* yang dibutuhkan dan mengurangi *fragment* yang tidak dibutuhkan.
2. Melakukan standarisasi saat pembuatan deskripsi *workflow* dan *similarity* kata menggunakan OWL agar dapat *discovery* dengan lebih mudah.
3. Perlu adanya penelitian lebih lanjut tentang pencarian *graph database* yang telah dibuat menggunakan ontologi guna meningkatkan reusabilitas dalam *workflow*.

DAFTAR PUSTAKA

- [1] M. Themistocleous, Z. Irani, O. Robert M and P. Ray, "ERP Problems and Application Integration," in *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on*, IEEE, 2001, pp. 10--pp.
- [2] R. W. Atmagi, "Penggunaan Workflow Repository Dan Kombinasi Metode Temu Kembali Berprioritas Untuk Meningkatkan Performa Configurable-Process Di ERP: Studi Kasus ERP 2011/13," *Jurnal Ilmiah Mikrotek*, vol. 1, pp. 39--46, 2014.
- [3] N. Press, "Understanding Metadata," *National Information Standards*, vol. 20, 2004.
- [4] M. Katsumi and M. Gruninger, "What s Ontology Reuse? I," in *Formal Ontology in Information Systems: Proceedings of the 9th International Conference (FOIS 2016)*, 2016.
- [5] J. F. Chang, *Business process management systems: strategy and implementation*, CRC Press, 2016.
- [6] J. Susilo, "BPMS - Satu Software Untuk Semua Kebutuhan Sistem Informasi Anda," 2 April 2014. [Online]. Available: <http://bpmsindonesia.blogspot.co.id/2014/04/bpms-satu-software-untuk-semua.html>. [Accessed 4 Desember 2017].
- [7] "Catalog and Archive Workflow Management Component," Apache Software Foundation, [Online]. Available: <http://svn.apache.org/repos/asf/oodt/site/components/maven/workflow/development/developer.html>. [Accessed 23 March 2017].
- [8] C. Liu, X. Lin, X. Zhou and M. Orłowska, "Building a repository for workflow systems," in *Technology of Object-Oriented Languages and Systems, 1999. TOOLS 31. Proceedings*, IEEE, 1999, pp. 348--357.
- [9] Z. Yan, R. Dijkman and P. Grefen, "Business Process Model Repositories--Framework and Survey," *Information and Software Technology*, vol. 54, pp. 380--395, 2012.
- [10] E. Deelman, K. Vahi, G. Juve and others, "Pegasus, a Workflow Management System for Science Automation," *Future Generation Computer Systems*, vol. 46, pp. 17--35, 2015.

- [11] S. A. White, "Introduction to BPMN," *IBM Cooperation*, vol. 2, 2004.
- [12] M. La Rosa, H. A. Reijers, W. M. Van Der Aalst, R. M. Dijkman, J. Mendling, M. Dumas and L. Garc{\i}A-Ba{\n}Uelos, "Apromore: An advanced process model repository," *Expert Systems with Applications*, vol. 38, pp. 7029--7040, 2011.
- [13] L. P. Dewi, U. Indahyanti and Y. Hari, "Pemodelan Proses Bisnis Menggunakan Activity Diagram UML dan BPMN (Studi Kasus FRS Online)," 2012.
- [14] S. K. Sari and A. Asniar, "Analisis dan Pemodelan Proses Bisnis Prosedur Pelaksanaan Proyek Akhir Sebagai Alat Bantu Identifikasi Kebutuhan Sistem," *JURNAL INFOTEL*, vol. 7, pp. 143--152, 2015.
- [15] P. W. Handayani, S. Bressan, D. K. Han and O. Boucelma, "Implementasi Sistem Pengelolaan Proses Bisnis Mengacu Pada Mit Process Handbook," *Jurnal Sistem Informasi*, vol. 5, pp. 106--113, 2012.
- [16] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing?," *International journal of human-computer studies*, vol. 43, pp. 907--928, 1995.
- [17] A. Prasetyo and I. Wiryana, "Database XML Pendukung Sistem Ontologi terhadap Informasi Biodiversitas," *Jurnal Ilmiah Program Studi Teknik Informatika*, 2010.
- [18] Suhartono, M. Hariadi and M. Purnomo, "Integration of Fuzzy System into Genetic L-System Programming based plant modeling environment with mathematica," *Australian Journal of Basic and Applied*, vol. 5, pp. 1760--5, 2011.
- [19] N. P. Lestari, "Uji Recall dan Precision Sistem Temu Kembali Informasi OPAC Perpustakaan ITS Surabaya".
- [20] P. Pendit, *Perpustakaan Digital: Perspektif Perpustakaan Perguruan Tinggi*, Jakarta: Sagung Seto, 2007.
- [21] M. Mijać and Z. Stapić, "Reusability Metrics of Software Components: Survey," in *26th Central European Conference on Information and Intelligent Systems (CECIIS 2015)*, 2015.
- [22] A. Khoshkbarforoushha, P. Jamshidi, M. F. Gholami, L. Wang and R. Ranjan, "Metrics for BPEL process reusability analysis in a workflow system," *IEEE Systems Journal*, vol. 10, pp. 36--45, 2016.
- [23] "Chapter 25. Release Notes," jBPM, [Online]. Available: <https://docs.jboss.org/jbpm/v6.2/userguide/ch25.html>. [Accessed 14 March 2017].
- [24] W. Frakes and C. Terry, "Software reuse: metrics and models," *ACM Computing Surveys (CSUR)*, vol. 28, pp. 415--435, 1996.
- [25] D. Gagne and S. Ringuette, "BPMN Quick Guide Second Edition," BPMN, [Online]. Available: <http://www.bpmnquickguide.com/view-bpmn-quick-guide/>. [Accessed 11 March 2017].

- [26] "An Introduction to RDF and the Jena RDF API," Apache Jena, [Online]. Available: https://jena.apache.org/tutorials/rdf_api.html. [Accessed 7 March 2017].
- [27] R. Garc{^e}s, T. de Jesus, J. Cardoso and P. Valente, "Open Source Workflow Management Systems: A Concise Survey," *Chapter in Book*, pp. 179--190, 2009.
- [28] K. Baina and S. Baina, "User experience-based evaluation of open source workflow systems: The cases of Bonita, Activiti, jBPM, and Intalio," in *ISKO-Maghreb, 2013 3rd International Symposium*, IEEE, 2013, pp. 1--8.
- [29] R. M. Dijkman, M. La Rosa and H. A. Reijers, "Managing Large Collections of business process Models-Current Techniques and Challenges," *Computers in Industry*, vol. 63, pp. 91--97, 2012.
- [30] J. Weissenfels, M. Gillmann, O. Roth, G. Shegalov and . W. Wonner, "The Mentor-lite Prototype: a Light-weight Workflow Management System," in *Data Engineering, 2000. Proceedings. 16th International Conference on*, IEEE, 2000, pp. 685--686.
- [31] T. Sutopo, "Resource Description Framework (RDF)," 3 Agustus 2009. [Online]. Available: <https://teguh8581.wordpress.com/2009/08/03/resource-description-framework-rdf/>. [Accessed 28 November 2017].
- [32] Laikaulfa, "Resource Description Framework (RDF)," 14 Desember 2011. [Online]. Available: <https://laikaulfa.wordpress.com/2011/12/14/resource-description-framework-rdf/>. [Accessed 27 November 2017].
- [33] G. Sukwan, "NALISIS PENYAJIAN DAN INTERPRETASI DATA," 24 Mei 2011. [Online]. Available: <https://gurusukwanblog.wordpress.com/2011/05/24/analisis-penyajian-dan-interpretasi-data/>. [Accessed 15 November 2017].

LAMPIRAN

1. Dokumen RDF

```

<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description
rdf:about="http://localhost/wf/file/Akademik_Pondok_Pesantren_Anwarul_H
uda.xpdl">
    <dc:XPDLVersion>2.2</dc:XPDLVersion>
    <dc:Vendor>Bizagi Process Modeler.</dc:Vendor>
    <dc:Created>2017-12-03 09:53:53</dc:Created>
    <dc:ModificationDate>0000-00-00 00:00:00</dc:ModificationDate>
    <dc:Description>Akademik Pondok Pesantren Anwarul
Huda</dc:Description>
    <dc:Author>digbys</dc:Author>
    <dc:Version>1.0</dc:Version>
    <dc:CountryKey>CO</dc:CountryKey>
    <dc:Activity> Daftar ulang santri lama Daftar Ulang santri baru
Pembuatan Kegiatan Pondok Pendataan Santri pondok Rekap pendataan
kegiatan Diniyah Rekap data KBM Kalkulasi nilai ujian dan proses
kelulusan Pendataan Alumni </dc:Activity>
  </rdf:Description>
  <rdf:Description rdf:about="http://localhost/wf/file/SIA.xpdl">
    <dc:XPDLVersion>2.2</dc:XPDLVersion>
    <dc:Vendor>Bizagi Process Modeler.</dc:Vendor>
    <dc:Created>2016-10-12 12:52:18</dc:Created>
    <dc:ModificationDate>2016-10-26 14:02:24</dc:ModificationDate>
    <dc:Description>Diagram 1</dc:Description>
    <dc:Author>Nurika</dc:Author>
    <dc:Version>1.0</dc:Version>
    <dc:CountryKey>CO</dc:CountryKey>
    <dc:Activity> Penerimaan Data pegawai Menampilkan data pegawai
akademik dan non akademik Menampilkan data tahun ajaran baru Kenaikan
Kelas Jadwal guru Jadwal kelas Penempatan siswa Mutasi Kelulusan
Pendataan alumni Menentukan angkatan pada awal tahun ajaran baru
Menentukan tingkatan pada awal tahun ajaran baru Menentukan semester
pada awal tahun ajaran baru Penentuan semester genap atau ganjil
Penentuan tingkatan I. II. III Menentukan kelas Penentuan kelas
IA, IB, IC dst Penempatan siswa baru Siswa ditempatkan sesuai dengan
angkatan, tingkatan, kelas Menentukan jenis pengujian Menentukan aspek
penilaian Jenis pengujian pada tiap mata pelajaran Aspek penilaian pada
tiap departemen Grading ditentukan pada tiap guru pengampu mata
pelajaran Menentukan grading Penilaian rapor pada tiap guru pengampu
mata pelajaran Menentukan aturan penilaian rapor Menentukan guru tiap
mata pelajaran Menyusun jadwal tiap guru Merangkum jadwal tiap guru
Menyusun jadwal tiap kelas Menyusun jam belajar Menyusun presensi data
kelas Pengisian nilai tiap mata pelajaran Menentukan kriteria kenaikan
dan kelulusan siswa Receive data santri Receive data tahun ajaran
Perpindahan siswa Receive data tahun ajaran Receive data tahun ajaran
Receive data tahun ajaran Receive data tahun ajaran Receive data
ruang Receive data tahun ajaran Receive data tahun ajaran Receive data
bangku Receive data pegawai Receive data santri Receive data tahun
ajaran Receive data tahun ajaran Receive data tahun ajaran Receive data
tahun ajaran Receive data pegawai Receive data kurikulum Receive data
pegawai Receive data kurikulum Receive data tahun ajaran Receive data
pegawai Receive data kurikulum Receive data tahun ajaran Receive data
tahun ajaran Receive data tahun ajaran Receive data tahun ajaran
  
```

```

Penyediaan data departemen Pondok Pesantren Send data departemen BPM
Penentuan tahun ajaran baru Send data tahun ajaran Data tahun ajaran
Send data kurikulum Penentuan mata pelajaran Mata pelajaran
Penyediaan data pegawai Send data pegawai Data pegawai Send data
santri Penyediaan data santri Data Santri Send data bangku Kalkulasi
bangku kelas yang dibutuhkan Penghitungan kapasitas tiap kelas Data
bangku Kalkulasi ruang yang dibuthkan Send data ruang Data ruang
Penghitungan ruang yang ada </dc:Activity>
  </rdf:Description>
  <rdf:Description
rdf:about="http://localhost/wf/file/Akademik_Pondok_Pesantren_Z.xpdl">
  <dc:XPDLVersion>2.2</dc:XPDLVersion>
  <dc:Vendor>Bizagi Process Modeler.</dc:Vendor>
  <dc:Created>2017-12-03 10:56:53</dc:Created>
  <dc:ModificationDate>0000-00-00 00:00:00</dc:ModificationDate>
  <dc:Description>Akademik Pondok Pesantren Z</dc:Description>
  <dc:Author>digbys</dc:Author>
  <dc:Version>1.0</dc:Version>
  <dc:CountryKey>CO</dc:CountryKey>
  <dc:Activity> daftar ulang santri baru daftar ulang santri lama
pendataan santri pondok pendataan jumlah kelas dan pengelompokan
penyesuaian guru dan wali kelas pembuatan kalender akademik rekap data
KBM rekap hasil ujian kalkulasi nilai ujian dan proses kelulusan
pendaftaran wisuda </dc:Activity>
  </rdf:Description>
  <rdf:Description
rdf:about="http://localhost/wf/file/Akademik_Pondok_Pesantren_Al-
Rifai.xpdl">
  <dc:XPDLVersion>2.2</dc:XPDLVersion>
  <dc:Vendor>Bizagi Process Modeler.</dc:Vendor>
  <dc:Created>2017-12-03 09:19:32</dc:Created>
  <dc:ModificationDate>0000-00-00 00:00:00</dc:ModificationDate>
  <dc:Description>Akademik Pondok Pesantren Al-Rifai</dc:Description>
  <dc:Author>digbys</dc:Author>
  <dc:Version>1.0</dc:Version>
  <dc:CountryKey>CO</dc:CountryKey>
  <dc:Activity> Daftar ulang santri baru Daftar ulang santri lama
Pendataan jumlah kelas dan pengelompokan Pendataan Santri Pondok
Penyesuaian Guru dan Wali pembuatan kalender akademik proses KBM Ujian
Kalkulasi nilai ujian dan proses kelulusan Wisuda </dc:Activity>
  </rdf:Description>
  <rdf:Description
rdf:about="http://localhost/wf/file/Akademik_Pondok_Pesantren_Ar_Rohmah
.xpdl">
  <dc:XPDLVersion>2.2</dc:XPDLVersion>
  <dc:Vendor>Bizagi Process Modeler.</dc:Vendor>
  <dc:Created>2017-12-03 10:03:25</dc:Created>
  <dc:ModificationDate>0000-00-00 00:00:00</dc:ModificationDate>
  <dc:Description>Akademik Pondok Pesantren Ar
Rohmah</dc:Description>
  <dc:Author>digbys</dc:Author>
  <dc:Version>1.0</dc:Version>
  <dc:CountryKey>CO</dc:CountryKey>
  <dc:Activity> daftar ulang santri baru daftar ulang santri lama
Pemdataan Santri Pondok Pembuatan Kalender akademik Rekap data KBM
Rekap hasil ujian Kalkulasi nilai ujian dan proses kelulusan Pendataan
Alumni Daftar Wisuda </dc:Activity>
  </rdf:Description>
  <rdf:Description
rdf:about="http://localhost/wf/file/kegiatan_ar_rahman.xpdl">
  <dc:XPDLVersion>2.2</dc:XPDLVersion>
  <dc:Vendor>Bizagi Process Modeler.</dc:Vendor>
  <dc:Created>2016-11-13 19:42:02</dc:Created>

```

```

<dc:ModificationDate>2016-11-14 01:44:22</dc:ModificationDate>
<dc:Description>Diagram 1</dc:Description>
<dc:Author>Ahmad</dc:Author>
<dc:Version>2.0</dc:Version>
<dc:CountryKey>CO</dc:CountryKey>
<dc:Activity> Tahun Takwim Send Tutup Tahun Perencanaan Kegiatan
Kegiatan Bomaterial Boperson Bomachine Pendataan Rapat
Pendapatan Kegiatan Pengeluaran Kegiatan Daftar Harga Pengelolaan
Kegiata Pengelola Kegiatan Sisa LPJ Send Kegiatan Send Bomaterial Send
Boperson Send Pendataan Kegiatan Send Pengeluaran Kegiatan Send Daftar
Harga Send Pendataan Rapat Send Pengelola Kegiatan Baru Send Sisa
Laporan LPJ Send Bomachine Send Tahun Baru Send Tutup Pengelola
Kegiatan Receive Tahun Takwim Receive Kegiatan Receive Bomaterial
Receive Boperson Receive Bomachine Receive Data Rapat Receive
pengelolaan Receive Daftar Harga Receive Pendapatan Kegiatan Receive
Pengeluaran Sisa Receive Kepegawaian Receive Sarpras Servis Kegiatan
Service Kepegawaian Service Sarpras Service Enterprise</dc:Activity>
</rdf:Description>
<rdf:Description
rdf:about="http://localhost/wf/file/Kesantrian_Z.xpdl">
<dc:XPDLVersion>2.2</dc:XPDLVersion>
<dc:Vendor>Bizagi Process Modeler.</dc:Vendor>
<dc:Created>2017-12-03 14:38:40</dc:Created>
<dc:ModificationDate>0000-00-00 00:00:00</dc:ModificationDate>
<dc:Description>Kesantrian Z</dc:Description>
<dc:Author>digbys</dc:Author>
<dc:Version>1.0</dc:Version>
<dc:CountryKey>CO</dc:CountryKey>
<dc:Activity> rekap hasil kegiatan agama pagi rekap absensi santri
pondok rekap hasil kegiatan rutin harian pondok rekap hasil kegiatan
sore harian rekap hasil aktivitas diniyah </dc:Activity>
</rdf:Description>
<rdf:Description
rdf:about="http://localhost/wf/file/Kesantrian_Luhur.xpdl">
<dc:XPDLVersion>2.2</dc:XPDLVersion>
<dc:Vendor>Bizagi Process Modeler.</dc:Vendor>
<dc:Created>2017-12-03 11:19:37</dc:Created>
<dc:ModificationDate>0000-00-00 00:00:00</dc:ModificationDate>
<dc:Description>Kesantrian Luhur</dc:Description>
<dc:Author>digbys</dc:Author>
<dc:Version>1.0</dc:Version>
<dc:CountryKey>CO</dc:CountryKey>
<dc:Activity> Rekap hasil Kegaitan agama pagi penentuan ustdadz
tahsin dan pendataan santri kuliah pembuatan absensi piket harian
rekap data santri dan data ustadz pondok rekap hasil kegiatan sore
harian penentuan jadwal pengajar diniyah rekap hasil kegiatan diniyah
</dc:Activity>
</rdf:Description>
<rdf:Description
rdf:about="http://localhost/wf/file/Kesantrian_Y.xpdl">
<dc:XPDLVersion>2.2</dc:XPDLVersion>
<dc:Vendor>Bizagi Process Modeler.</dc:Vendor>
<dc:Created>2017-12-03 12:42:23</dc:Created>
<dc:ModificationDate>0000-00-00 00:00:00</dc:ModificationDate>
<dc:Description>Kesantrian Y</dc:Description>
<dc:Author>digbys</dc:Author>
<dc:Version>1.0</dc:Version>
<dc:CountryKey>CO</dc:CountryKey>
<dc:Activity> rekap hasil kegiatan pagi pembuatan jadwal piket
harian santri perencanaan kegiatan santri harian rekap hasil kegiatan
rutin harian Rekap hasil kegiatan sore harian rekap hasil
kegiatan diniyah </dc:Activity>
</rdf:Description>

```

```

<rdf:Description rdf:about="http://localhost/wf/file/Keuangan.xpdl">
  <dc:XPDLVersion>2.2</dc:XPDLVersion>
  <dc:Vendor>Bizagi Process Modeler.</dc:Vendor>
  <dc:Created>2016-11-05 00:39:01</dc:Created>
  <dc:ModificationDate>2016-11-07 18:15:09</dc:ModificationDate>
  <dc>Description>Diagram 1</dc>Description>
  <dc:Author>navIras</dc:Author>
  <dc:Version>1.0</dc:Version>
  <dc:CountryKey>CO</dc:CountryKey>
  <dc:Activity> Receive Tahun Buku Receive Rencana Anggaran Receive
Transaksi Beranda Kode Akun Send Kode Akun Tahun Buku Buku Baru? Head
and Menu Send Buku Baru Send Tutup Buku Rencana Anggaran Anggaran
Masuk? Rencana Anggaran Masuk Rencana Anggaran Keluar Send Anggaran
Masuk Send Anggaran Keluar Transaksi Transaksi Masuk? Transaksi Masuk
Transaksi Keluar Send Transaksi Masuk Send Transaksi Keluar Laporan
Laporan Cash Flow Laporan Anggaran Laporan Transaksi Receive Akuntansi
Receive Pengadaan Receive Kegiatan Receive Kepegawaian Receive
Kesantrian Receive Super Admin service keuanganView service
keuanganExecute service kepegawaian service Kesantrian service kegiatan
service Akuntansi service Company service Enterprise service Factory
service Organization</dc:Activity>
</rdf:Description>
<rdf:Description
rdf:about="http://localhost/wf/file/PSB_Pondok_Pesantren_An_Nur_1.xpdl"
>
  <dc:XPDLVersion>2.2</dc:XPDLVersion>
  <dc:Vendor>Bizagi Process Modeler.</dc:Vendor>
  <dc:Created>2017-12-03 14:43:29</dc:Created>
  <dc:ModificationDate>0000-00-00 00:00:00</dc:ModificationDate>
  <dc>Description>PSB Pondok Pesantren An Nur 1</dc>Description>
  <dc:Author>digbys</dc:Author>
  <dc:Version>1.0</dc:Version>
  <dc:CountryKey>CO</dc:CountryKey>
  <dc:Activity> pengisian form pendaftaran cetak bukti tanda terima
pendaftaran validasi formulir dan bukti tanda terima pendaftaran print
out formulir dan bukti yang telah di validasi melakukan pemabayaran ke
administrasi keuangan penentuan kelompok masa orientasi peserta didik
baru </dc:Activity>
</rdf:Description>
<rdf:Description
rdf:about="http://localhost/wf/file/PSB_Pondok_Pesantren_Luhur.xpdl">
  <dc:XPDLVersion>2.2</dc:XPDLVersion>
  <dc:Vendor>Bizagi Process Modeler.</dc:Vendor>
  <dc:Created>2017-12-03 18:44:37</dc:Created>
  <dc:ModificationDate>0000-00-00 00:00:00</dc:ModificationDate>
  <dc>Description>PSB Pondok Pesantren Luhur</dc>Description>
  <dc:Author>digbys</dc:Author>
  <dc:Version>1.0</dc:Version>
  <dc:CountryKey>CO</dc:CountryKey>
  <dc:Activity> mengisi formulir pendaftaran mencetak tanda bukti
pendaftaran validasi formulir pendaftaran cetak bukti tanda terima
pondok pesantren pendataan kamar santri baru pembagian kelompok masa
orientasi peserta didik baru </dc:Activity>
</rdf:Description>
<rdf:Description
rdf:about="http://localhost/wf/file/PSB_Pondok_Pesantren_Anwarul_Huda.x
pdl">
  <dc:XPDLVersion>2.2</dc:XPDLVersion>
  <dc:Vendor>Bizagi Process Modeler.</dc:Vendor>
  <dc:Created>2017-12-03 17:47:53</dc:Created>
  <dc:ModificationDate>0000-00-00 00:00:00</dc:ModificationDate>
  <dc>Description>PSB Pondok Pesantren Anwarul Huda</dc>Description>
  <dc:Author>digbys</dc:Author>

```

```

    <dc:Version>1.0</dc:Version>
    <dc:CountryKey>CO</dc:CountryKey>
    <dc:Activity> mengisi formulir pendaftaran upload berkas
    pendaftaran lain Validasi formulir dan bekas pendaftaran mencetak
    bukti pendaftaran dan formulir Registrasi ulang pembayaran ke
    administrasi keuangan validasi pembayaran pendataan kamar santri baru
  </dc:Activity>
</rdf:Description>

  <rdf:Description
rdf:about="http://localhost/wf/file/production_planning.xpdl">
    <dc:XPDLVersion>2.2</dc:XPDLVersion>
    <dc:Vendor>Bizagi Process Modeler.</dc:Vendor>
    <dc:Created>2016-10-22 16:16:21</dc:Created>
    <dc:ModificationDate>2016-11-14 12:37:34</dc:ModificationDate>
    <dc:Description>Diagram 1</dc:Description>
    <dc:Author>PIPIT</dc:Author>
    <dc:Version>1.0</dc:Version>
    <dc:CountryKey>CO</dc:CountryKey>
    <dc:Activity> Pilih Menu Home tahun ajaran          kalender akademik
    instansi rombel jadwal pelajaran matapelajaran silabus ruang kelas
    sarana material data guru Penentuan Tahun Ajaran Penentuan Kuota Santri
    hitung jumlah kebutuhan sarana cek ketersediaan pembuatan daftar
    pengadaan sarana penentuan jumlah jam pelajaran hitung jumlah kebutuhan
    guru cek ketersediaan pembuatan daftar recruitment guru penentuan
    jumlah kebutuhan material perkelas perhitungan jumlah kebutuhan
    material peningkatan cek ketersediaan pembuatan daftar pengadaan
    material send kuota send pengadaan sarana send kebutuhan sarana send
    recruitment guru send kebutuhan guru send pengadaan material send
    kebutuhan material send perencanaan aktifitas send bk guru perencanaan
    aktifitas pembelajaran send bk machine hitung jumlah jam tatap muka cek
    jabatan struktural BK TM ditambah BK struktural hitung BK machine
    Service Tahun Ajaran Service Kalender Akademik Service instansi
    Service rombel Service matapelajaran Service jadwal pelajaran Service
    silabus Service data ruang Service data sarana Service data material
    Service data pegawai</dc:Activity>
  </rdf:Description>
  <rdf:Description
rdf:about="http://localhost/wf/file/Sarana_Prasarana_Pondok_Pesantren
_Aнварul_Huda.xpdl">
    <dc:XPDLVersion>2.2</dc:XPDLVersion>
    <dc:Vendor>Bizagi Process Modeler.</dc:Vendor>
    <dc:Created>2017-12-03 20:46:07</dc:Created>
    <dc:ModificationDate>0000-00-00 00:00:00</dc:ModificationDate>
    <dc:Description>Sarana Prasarana Pondok Pesantren
    Anwarul Huda</dc:Description>
    <dc:Author>digbys</dc:Author>
    <dc:Version>1.0</dc:Version>
    <dc:CountryKey>CO</dc:CountryKey>
    <dc:Activity> Pendataan sarpras Perbaikan sarana Pengecekan Barang
    rekap data sarpras pembelian sarana prasana baru </dc:Activity>
  </rdf:Description>
  <rdf:Description
rdf:about="http://localhost/wf/file/Sarana_Prasarana_Pondok_Pesantren_L
uhur.xpdl">
    <dc:XPDLVersion>2.2</dc:XPDLVersion>
    <dc:Vendor>Bizagi Process Modeler.</dc:Vendor>
    <dc:Created>2017-12-03 20:51:43</dc:Created>
    <dc:ModificationDate>0000-00-00 00:00:00</dc:ModificationDate>
    <dc:Description>Sarana Prasarana Pondok Pesantren
    Luhur</dc:Description>
    <dc:Author>digbys</dc:Author>
    <dc:Version>1.0</dc:Version>

```

```

    <dc:CountryKey>CO</dc:CountryKey>
    <dc:Activity> pendataan dan pengecekan barang perbaikan sarana
rencana pengadaan dengan menjual dan iuran bersama pembelian sarpras
baru </dc:Activity>
  </rdf:Description>
</rdf:Description>
rdf:about="http://localhost/wf/file/Sarana_Prasarana_Pondok_Pesantren_A
l_Rifai.xpdl">
  <dc:XPDLVersion>2.2</dc:XPDLVersion>
  <dc:Vendor>Bizagi Process Modeler.</dc:Vendor>
  <dc:Created>2017-12-03 20:18:53</dc:Created>
  <dc:ModificationDate>0000-00-00 00:00:00</dc:ModificationDate>
  <dc:Description>Sarana Prasarana Pondok Pesantren Al
Rifai</dc:Description>
  <dc:Author>digbys</dc:Author>
  <dc:Version>1.0</dc:Version>
  <dc:CountryKey>CO</dc:CountryKey>
  <dc:Activity> pendataan dan pengecekan barang cek kondisi dan
pengadaan distribusi rekap data perbaikan sarana pembelian sarpras baru
</dc:Activity>
  </rdf:Description>
</rdf:Description>
rdf:about="http://localhost/wf/file/Kesantrian_Pondok_Pesantren_Al-
Rifai.xpdl">
  <dc:XPDLVersion>2.2</dc:XPDLVersion>
  <dc:Vendor>Bizagi Process Modeler.</dc:Vendor>
  <dc:Created>2017-12-03 11:08:41</dc:Created>
  <dc:ModificationDate>0000-00-00 00:00:00</dc:ModificationDate>
  <dc:Description>Kesantrian Pondok Pesantren Al-
Rifai</dc:Description>
  <dc:Author>digbys</dc:Author>
  <dc:Version>1.0</dc:Version>
  <dc:CountryKey>CO</dc:CountryKey>
  <dc:Activity> Perencanaan kegiatan santri harian rekap absensi
santri pondok rekap hasil kegiatan agama pagi rekap hasil kegiatan
rutin harian pondok rekap hasil kegiatan agama sore rekap hasil
sekolah diniyah </dc:Activity>
  </rdf:Description>
</rdf:Description>
rdf:about="http://localhost/wf/file/Sarana_Prasarana_Z.xpdl">
  <dc:XPDLVersion>2.2</dc:XPDLVersion>
  <dc:Vendor>Bizagi Process Modeler.</dc:Vendor>
  <dc:Created>2017-12-03 20:55:12</dc:Created>
  <dc:ModificationDate>0000-00-00 00:00:00</dc:ModificationDate>
  <dc:Description>Sarana Prasarana Z</dc:Description>
  <dc:Author>digbys</dc:Author>
  <dc:Version>1.0</dc:Version>
  <dc:CountryKey>CO</dc:CountryKey>
  <dc:Activity> Pendataan dan pengecekan barang perbaikan sarana
pembelian sarpras baru </dc:Activity>
  </rdf:Description>
</rdf:RDF>

```