

**KOMPOSISI *WEB SERVICE* MENGGUNAKAN *COSINE SIMILARITY* UNTUK MENYUSUN *BUSINESS PROCESS EXECUTING LANGUAGE* (BPEL)**

**SKRIPSI**

**OLEH:  
YULIANI NINGSIH  
NIM. 14650021**



**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2019**

**KOMPOSISI *WEB SERVICE* MENGGUNAKAN *COSINE*  
*SIMILARITY* UNTUK MENYUSUN *BUSINESS*  
*PROCESS EXECUTING LANGUAGE*  
(BPEL)**

**SKRIPSI**

**Diajukan kepada :  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang  
Untuk memenuhi Salah Satu Persyaratan Dalam  
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh :  
YULIANI NINGSIH  
NIM. 14650021**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2019**

# HALAMAN PERSETUJUAN

## HALAMAN PERSETUJUAN


**KOMPOSISI *WEB SERVICE* MENGGUNAKAN *COSINE SIMILARITY* UNTUK MENYUSUN *BUSINESS PROCESS EXECUTING LANGUAGE* (BPEL)**

### SKRIPSI

Oleh :  
**YULIANI NINGSIH**  
NIM. 14650021

Telah Diperiksa dan Disetujui  
Tanggal 17 Juni 2019

Dosen Pembimbing I

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

Dosen Pembimbing II

  
**Ajib Hanani, M.T**  
NIDT. 19840731 20160801 1 076

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

  
**Yoyo Crystian**  
NIP. 19740424 200901 1 008

## HALAMAN PENGESAHAN



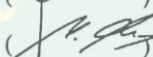

### HALAMAN PENGESAHAN

#### KOMPOSISI *WEB SERVICE* MENGGUNAKAN *COSINE SIMILARITY* UNTUK MENYUSUN *BUSINESS PROCESS EXECUTING LANGUAGE (BPEL)*

#### SKRIPSI

Oleh :  
**YULIANI NINGSIH**  
NIM. 14650021

Telah Dipertahankan di Depan Dewan Penguji Skripsi dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan Untuk Memperoleh Gelar Sarjana Komputer (S. Kom)  
Tanggal : 17 Juni 2019

Susunan Dewan Penguji		Tanda tangan
Penguji utama	: <u>Syahiduz Zaman, M.Kom</u> NIP. 19700502 200501 1 005	(  )
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>Ajib Hanani, M.T</u> NIDT. 19840731 20160801 1 076	(  )

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



  
Cahyo Crysdiyan  
NIP. 19740424 200901 1 008

**HALAMAN MOTTO**

“ life is a struggle

There is no life without a struggle “





## HALAMAN PERSEMBAHAN

الحمد لله رب العالمين

**Puji syukur kehadiran Allah SWT, shalawat dan salam bagi Rasul-Nya**

**Saya persembahkan sebuah karya ini kepada :**

Kedua orangtuaku, Bapak Rohmad, S.Pd.I dan Ibu Sukarsi tercinta, yang tiada henti memberi restu, semangat, motivasi, nasihat, serta do'a yang tak pernah putus disetiap sujudnya.

Adik saya Rosi Oktavian Purnama Safitri, kakak Farid, kakek saya Bapak Djimun dan Nenek saya Ibu Kamini, paman-bibi serta sepupu-sepupu yang menjadi salah satu motivasi untuk menyelesaikan skripsi ini.

Orang-orang yang saya sayangi, sahabat-sahabat seperjuangan serta keluarga Biner (Teknik Informatika angkatan 2014), serta seluruh keluarga besar Teknik Informatika UIN Maulana Malik Ibrahim Malang, yang tidak bisa saya sebutkan satu persatu.

Saya ucapkan terimakasih yang luar biasa, yang telah membantu menyelesaikan skripsi. Semoga Allah melimpahkan rahmat dan hidayah-Nya kepada kita semua.

## PERNYATAAN KEASLIAN TULISAN

### PERNYATAAN KEASLIAN TULISAN

Saya yang bertandatangan di bawah ini :

Nama : Yuliani Ningsih

NIM : 14650021

Jurusan : Teknik Informatika

Fakultas : Sains dan Teknologi

Judul skripsi : **KOMPOSISI *WEB SERVICE* MENGGUNAKAN  
*COSINE SIMILARITY* UNTUK MENYUSUN  
*BUSINESS PROCESS EXECUTING LANGUAGE*  
(BPEL)**

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

Malang, 17 Juni 2019

Yang membuat pernyataan



**Yuliani Ningsih**  
NIM. 14650021

## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji bagi Allah SWT, karena atas rahmat, hidayah serta karunia-Nya. Penulis dapat menyelesaikan skripsi yang berjudul “Komposisi *Web Service* menggunakan *Cosine Similarity* untuk menyusun *Business Process Executing Language* (BPEL)” sebagai salah satu syarat untuk memperoleh gelar sarjana pada Program Studi Teknik Informatika jenjang Strata-1 Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Shalawat serta salam senantiasa terlumpahkan kepada Nabi Muhammad SAW, keluarga dan para sahabat yang telah membimbing umat dari gelapnya alam jahiliyah menuju cahaya islam yang diridoi Allah SWT.

Penulis menyadari adanya banyak keterbatasan yang penulis miliki, sehingga ada banyak pihak yang telah memberikan bantuan baik moril maupun materil dalam menyelesaikan penelitian ini. Maka dari itu dengan segenap kerendahan hati penulis mengucapkan terimakasih kepada :

1. Prof. Dr. H. Abd. Haris, M.Ag selaku rektor UIN Maulana Malik Ibrahim Malang
2. Dr. Sri Harini, M.Si selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Cahyo Crysdiyan, selaku Ketua Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. 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.
5. Ajib Hanani, M.T selaku dosen pembimbing II yang senantiasa memberi masukan dan nasihat serta petunjuk dalam penyusunan skripsi ini.
6. Seluruh dosen dan staff jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang yang telah memberikan bimbingan keilmuan kepada penulis selama masa studi.



7. Bapak dan Mama, serta seluruh keluarga besar yang senantiasa memberikan do'a dan motivasi kepada penulis dalam menuntut ilmu serta do'a yang senantiasa mengiringi setiap langkah penulis.
8. Teman yang telah terlibat dan memberi bantuan dalam menyelesaikan penulisan Makhfud, Rantu, Nindy, Fajrul.
9. Sahabat-sahabat saya Herdian, Yuliana, Aida, Amalia, Sifa, Fitri, Dinda, Putri, Mia, Rofiq dan Tim Skripsi sukses.
10. Teman-teman seperjuangan Teknik Informatika angkatan 2014
11. Semua pihak yang ikut dalam berkontribusi dalam membantu menyelesaikan skripsi.

Penulis menyadari dalam karya ini masih banyak kekurangan. Oleh karena itu penulis menerima segala kritik dan saran dari pembaca. Semoga karya ini bermanfaat bagi seluruh pihak.

Malang, 24 Juni 2019

Penulis

## DAFTAR ISI

HALAMAN COVER.....	i
HALAMAN JUDUL.....	ii
HALAMAN PERSETUJUAN.....	iii
HALAMAN PENGESAHAN.....	iii
HALAMAN MOTTO .....	v
HALAMAN PERSEMBAHAN .....	vi
PERNYATAAN KEASLIAN TULISAN .....	vii
KATA PENGANTAR .....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR .....	xii
DAFTAR TABEL.....	xiv
ABSTRAK.....	xv
ABSTRACT.....	xvi
المخلص .....	xvii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Pernyataan Masalah .....	4
1.3 Tujuan Penelitian .....	4
1.4 Manfaat Penelitian .....	4
1.5 Batasan Penelitian.....	5
1.6 Sistematika Penulisan .....	5
BAB II KAJIAN PUSTAKA.....	7
2.1 Proses Bisnis .....	7
2.2 Pemodelan Proses Bisnis .....	8
2.2.1 BPMN .....	8
2.2.2 XML Process Definition Language (XPDL).....	12
2.3 Web Service .....	13
2.3.1 Arsitektur Web Service.....	14
2.4 Web Services Composition.....	17
2.4.1 Business Process Execution Language (BPEL).....	18

2.5	Text Mining .....	22
2.5.1	Text Preprocessing .....	23
2.5.2	Text Transformation .....	24
2.6	TF (Term Frequency).....	26
2.7	IDF (Inverse Document Frequency) .....	27
2.8	Vector Space Model (VSM) .....	27
2.9	Cosine Similarity .....	28
BAB III METODE PENELITIAN.....		30
3.1	Desain Penelitian .....	30
3.1.1	Gambaran Umum Sistem.....	30
3.1.2	Sumber Data .....	30
3.2	Perancangan Sistem .....	31
3.2.1	Preprocessing .....	32
3.2.2	TF-IDF .....	43
3.2.3	Cosine Similarity .....	48
3.2.4	Generate BPEL .....	52
3.3	Pengujian Sistem.....	53
BAB IV HASIL DAN PEMBAHASAN.....		54
4.1	Prosedur Pengujian .....	54
4.1.1	Pemanggilan Dokumen Pemodelan proses bisnis dari Repository.....	55
4.1.2	Preprocessing .....	56
4.1.3	Pembobotan TF IDF .....	57
4.1.4	Implementasi Cosine Similarity.....	58
4.2	Langkah Uji Coba.....	60
4.3	Pengujian BPEL.....	64
4.4	Pengujian Skenario .....	66
4.5	Integrasi Komposisi <i>Web Service</i> dengan Islam.....	70
BAB V PENUTUP.....		73
5.1	Kesimpulan .....	73
5.2	Saran .....	74
DAFTAR PUSTAKA .....		75

## DAFTAR GAMBAR

Gambar 2. 1	Elemen start, intermediate dan end event .....	10
Gambar 2. 2	Elemen-elemen Activity.....	10
Gambar 2. 3	Elemen sequence flow, message flow, dan association .....	11
Gambar 2. 4	Pool dan Lane.....	11
Gambar 2. 5	Elemen data object, group, dan annotation .....	12
Gambar 2. 6	Komponen-komponen <i>Web Service</i> .....	15
Gambar 2. 7	Case Folding .....	24
Gambar 2. 8	Proses Tokenizing .....	24
Gambar 2. 9	Proses Filtering .....	25
Gambar 2. 10	Proses Stemming.....	26
Gambar 3. 1	Alur Perancangan Sistem .....	31
Gambar 3. 2	Flowchart case folding .....	33
Gambar 3. 3	Source code case folding.....	34
Gambar 3. 4	Flowchart tokenizing.....	35
Gambar 3. 5	Source code tokenizing .....	36
Gambar 3. 6	Flowchart Filtering.....	38
Gambar 3. 7	Source code filtering .....	39
Gambar 3. 8	Flowchart stemming.....	40
Gambar 3. 9	Source code stemming .....	42
Gambar 3. 10	Flowchart TF-IDF .....	44
Gambar 3. 11	Source code TF-IDF.....	47
Gambar 3. 12	Source code TF-IDF.....	48
Gambar 3. 13	Flowchart cosine similarity .....	49
Gambar 3. 14	Source code cosine similarity.....	52
Gambar 3. 15	Struktur BPEL.....	53
Gambar 4. 1	Daftar term hasil preprocessing .....	57
Gambar 4. 2	Daftar nilai term frequency .....	58
Gambar 4. 3	Hasil <i>similarity</i> kata kunci dengan dokumen .....	59
Gambar 4. 4	Tampilan utama aplikasi <i>metadata</i> dan WSDL <i>webservice</i> .....	60
Gambar 4. 5	Hasil pencarian.....	61

Gambar 4. 6	Tampilan pencarian <i>web service</i> .....	62
Gambar 4. 7	Hasil pencarian <i>webservice</i> .....	62
Gambar 4. 8	Tampilan hasil <i>webservice</i> .....	63
Gambar 4. 9	Tampilan Info Hasil <i>Generate BPEL</i> .....	63
Gambar 4. 10	Dokumen hasil <i>generate BPEL</i> .....	64
Gambar 4. 11	Hasil request file BPEL.....	65
Gambar 4. 12	Detail data dari BPEL .....	65





## DAFTAR TABEL

Tabel 3.1	Contoh dokumen untuk pencarian .....	32
Tabel 3.2	Contoh kata kunci untuk pencarian .....	33
Tabel 3.3	Hasil <i>case folding</i> dokumen .....	34
Tabel 3.4	Hasil <i>case folding</i> kata kunci .....	35
Tabel 3.5	Hasil <i>Tokenizing</i> dokumen .....	36
Tabel 3.6	Hasil <i>tokenizing</i> kata kunci .....	37
Tabel 3.7	Hasil <i>filtering</i> dokumen .....	39
Tabel 3.8	Hasil <i>filtering</i> kata kunci .....	40
Tabel 3.9	Hasil <i>stemming</i> dokumen .....	42
Tabel 3.10	Hasil <i>stemming</i> kata kunci .....	43
Tabel 3.11	Menghitung TF IDF .....	48
Tabel 3.12	Menghitung bobot .....	48
Tabel 3.13	Pengujian sistem .....	53
Tabel 4. 1	Skenario pengujian aplikasi .....	66
Tabel 4. 2	Hasil nilai persamaan .....	67
Tabel 4. 3	Hasil nilai persamaan .....	68
Tabel 4. 4	Hasil nilai persamaan .....	69

## ABSTRAK

Ningsih, Yuliani. 2019. **Komposisi Web Service Menggunakan Cosine Similarity Untuk Menyusun Business Process Executing Language.** Skripsi. Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.  
Pembimbing: (I) M. Ainul Yaqin, M.Kom (II) Ajib Hanani, M.T.

---

**Kata Kunci :** BPMN, BPEL, *Web Service*, Komposisi *Web Service*, *Cosine Similarity*

Pemodelan proses bisnis digunakan untuk menggambarkan langkah-langkah yang harus diambil untuk mencapai suatu tujuan. BPMN merupakan standar untuk pemodelan proses bisnis yang menyediakan notasi grafis untuk menentukan proses bisnis dalam bentuk diagram. Secara umum BPMN digunakan untuk menentukan pemodelan proses bisnis dari pandangan seorang analisis. BPMN merupakan suatu notasi grafis yang berbentuk diagram dan bersifat mati sehingga tidak dapat dieksekusi. Sementara itu *web service* merupakan sebuah sistem *software* yang menggunakan standard XML, sehingga dapat diakses oleh sistem atau aplikasi perangkat lunak yang ditulis dengan berbagai bahasa pemrograman, dan dapat berjalan diberbagai *platform*, maupun sistem operasi. Namun *web service* memiliki susunan *variabel* yang berbeda. Maka untuk mengatasi kekurangan tersebut diperlukan adanya proses *mapping* menggunakan model proses bisnis, setelah itu dilakukan komposisi. Komposisi *web service* merupakan suatu proses penyusunan ulang dari beberapa *web service* yang telah dipetakan dengan proses bisnis. TF-IDF (*Term frequency-inverse document frequency*), diterapkan untuk mendapatkan nilai atau bobot dari *activity* yang relevan dengan kata kunci yang telah diinputkan. Sementara itu metode *cosine similarity* digunakan untuk menghitung kemiripan antar *activity*, sehingga dapat menghasilkan komposisi dari rangkaian *web service* yang sesuai dengan proses bisnis. Hasil dari komposisi tersebut kemudian di-generate menjadi BPEL. Berdasarkan pengujian yang dilakukan menggunakan Postman, data dapat berjalan dengan baik yang ditunjukkan dengan *response* berhasil saat melakukan *request*.

## ABSTRACT

Ningsih, Yuliani. 2019. **Web Service Composition Using Cosine Similarity to Arrange Business Process Executing Language**. Thesis. Informatics Engineering Department, Faculty of Science and Technology, State Islamic University of Maulana Malik Ibrahim Malang.  
Supervisor: (I) M. Ainul Yaqin, M.Kom (II) Ajib Hanani, M.T.

---

**Keywords:** BPMN, BPEL, Web Service, Web Service Composition, Cosine Similarity

Business process modeling is used to describe the steps that must be taken to achieve a goal. BPMN is a standard for business process modeling that provides graphical notation to determine business processes in the form of diagrams. In general, BPMN is used to determine business process modeling from the viewpoint of an analysis. BPN is a graphical notation in the form of a diagram and is dead so it cannot be executed. Meanwhile the web service is a software system that uses XML standards, so that it can be accessed by systems or software applications written in various programming languages, and can run in various platforms, as well as operating systems. But web services have a different arrangement of variables. So to overcome these shortcomings a mapping process is needed using a business process model, after which a composition is carried out. The composition of the web service is a process of rearranging from several web services that have been mapped with business processes. TF-IDF (Term frequency-inverse document frequency), is applied to obtain the value or weight of the activity that is relevant to the inputted keyword. Meanwhile the cosine similarity method is used to calculate the similarity between activities, so that it can produce compositions from a series of web services that are in accordance with business processes. The results of the composition are then generated into BPEL. Based on the tests carried out using Postman, the data can run well which is indicated by a successful response when making a request.

## الملخص

تنفيذ العملية التجارية. أطروحة. قسم هندسة المعلوماتية ، كلية العلوم والتكنولوجيا ، الجامعة الإسلامية الحكومية في مولانا مالك إبراهيم مالانج. المستشار: (1) محمد عين اليقين ، محمد كوم (2) أجيب حناني ، إم.

الكلمات الرئيسية: BPMN ، BPEL ، خدمة الويب ، تكوين خدمة الويب ، تشابه جيب التمام

BPMN معيارًا لنمذجة العمليات التجارية التي توفر تدوينًا بيانيًا لتحديد العمليات التجارية في شكل مخططات. بشكل عام ، يتم استخدام BPMN لتحديد نمذجة العمليات التجارية من وجهة نظر التحليل ، BPMN عبارة عن تدوين رسومي في شكل رسم تخطيطي ومات حتى لا يمكن تنفيذه. وفي الوقت نفسه ، فإن خدمة الويب هي نظام برمجي يستخدم معايير XML ، بحيث يمكن الوصول إليها بواسطة أنظمة أو تطبيقات برمجية مكتوبة بلغات برمجة مختلفة ، ويمكن تشغيلها في منصات مختلفة ، وكذلك أنظمة التشغيل. لكن خدمات الويب لها ترتيب مختلف للمتغيرات. لذلك للتغلب على أوجه القصور هذه ، هناك حاجة إلى عملية رسم الخرائط باستخدام نموذج عملية الأعمال ، وبعد ذلك يتم إجراء التكوين. تكوين خدمة الويب هو عملية إعادة ترتيب من العديد من خدمات الويب التي تم تعيينها مع العمليات التجارية. يتم تطبيق TF-IDF (تردد وثيقة معكوس المدى) ، للحصول على قيمة أو وزن النشاط ذي الصلة بالكلمة الرئيسية المدخلة. وفي الوقت نفسه ، يتم استخدام طريقة تشابه جيب التمام لحساب التشابه بين الأنشطة ، بحيث يمكنها إنتاج تراكيب من سلسلة من خدمات الويب التي تتوافق مع العمليات التجارية. ثم يتم إنشاء نتائج التكوين في BPEL. بناءً على الاختبارات التي تم إجراؤها باستخدام Postman ، يمكن تشغيل البيانات بشكل جيد وهو ما يشير إليه استجابة ناجحة عند تقديم طلب.



# BAB I PENDAHULUAN

## 1.1 Latar Belakang

Proses bisnis adalah suatu kumpulan aktivitas yang terstruktur dan memiliki keterkaitan satu sama lain untuk memperoleh suatu tujuan tertentu, baik yang dikerjakan oleh manusia maupun sistem pada suatu organisasi (Sarno, Fauzan, Nurlaili, Rahmawati, Sungkono, & Effendi, 2017). Dalam organisasi, proses bisnis merupakan aktivitas utama untuk mencapai tujuan organisasi. Saat ini proses bisnis telah banyak digunakan oleh organisasi baik dalam lingkup yang kecil maupun besar. Meskipun banyak organisasi yang menyadari pentingnya pengelolaan proses bisnis pada tingkat kepuasan konsumen, namun banyak organisasi yang masih belum mengerti dalam menjalankan perbaikan tersebut, baik secara bertahap maupun menyeluruh. Oleh sebab itu diperlukannya analisis terhadap proses bisnis, yaitu aktivitas yang dilakukan untuk mengkaji proses bisnis yang sudah ada dan menerapkan berbagai ilmu praktis yang dapat membantu mengubah dan meningkatkan proses. Tahapan pertama dalam melakukan analisis proses bisnis adalah memahami semua aktivitas di dalam proses. Pemahaman ini dapat dibantu dengan pemodelan proses bisnis (Sari & Asniar, 2015).

Pemodelan proses bisnis digunakan untuk menggambarkan langkah-langkah yang harus diambil untuk mencapai suatu tujuan. Dalam menggambarkan pemodelan proses bisnis terdapat sebuah notasi pemodelan untuk membantu perusahaan dalam menggambarkan proses-prosesnya. Notasi pemodelan tersebut adalah dengan *business process modelling notation* (BPMN). BPMN merupakan



standar untuk pemodelan proses bisnis yang menyediakan notasi grafis untuk menentukan proses bisnis dalam bentuk diagram. Secara umum BPMN digunakan untuk menentukan pemodelan proses bisnis dari pandangan seorang analisis. Pemodelan proses bisnis menggunakan BPMN merupakan suatu notasi grafis yang berbentuk diagram dan bersifat mati sehingga tidak dapat dieksekusi. Sementara itu *web service* merupakan sebuah sistem software yang menggunakan standard XML (*Extensible Markup Language*), sehingga dapat diakses oleh sistem atau aplikasi perangkat lunak yang ditulis dengan berbagai bahasa pemrograman, dan dapat berjalan di berbagai platform, maupun sistem operasi. Sehingga BPMN perlu diubah menjadi BPEL (*Business Process Executing Language*), sehingga dapat dijadikan sebagai referensi dalam eksekusi.

Beberapa *web service* memiliki susunan *variabel* yang berbeda. Untuk mengatasi kekurangan tersebut, maka diperlukan adanya proses *mapping* menggunakan pemodelan proses bisnis, sehingga dapat digunakan sebagai acuan. Hasil dari proses *mapping* kemudian dilanjutkan dengan proses komposisi *web service* atau penyusunan *web service* menggunakan *cosine similarity*.

Berbicara mengenai ilmu pengetahuan dan teknologi, konsep untuk mencari ilmu pengetahuan (wawasan) tersebut telah ada dalam kitab suci Al-Qur'an. Al-Qur'an bukanlah buku yang berisi ilmu pengetahuan hasil dari penelitian manusia, namun Al-Qur'an berisi wahyu-wahyu Allah yang diturunkan sebagai pedoman hidup bagi manusia. Akan tetapi Al-Qur'an dapat dikaji kebenarannya setelah manusia membuktikan isi Al-Qur'an secara ilmiah, Seperti ayat Al-Qur'an surat Yunus : 101 ;

فَلْ اَنْظُرُوا مَاذَا فِي السَّمَاوَاتِ وَالْاَرْضِ وَمَا تُغْنِي الْاَيَاتُ وَالنُّذُرُ  
عَنْ قَوْمٍ لَا يُؤْمِنُونَ

Artinya : Katakanlah, “Perhatikanlah apa yang ada di langit dan di bumi!”  
Tidaklah bermanfaat tanda-tanda (kebesaran Allah) dan rasul-rasul yang memberi  
peringatan bagi orang yang tidak beriman.” (QS. Yunus : 101)

Dalam surat Yunus pada ayat ke 101, Allah SWT memerintahkan kepada  
manusia untuk memperhatikan fenomena alam yang ada di langit dan di bumi  
yang merupakan salah satu tanda kebesaran Allah. Fenomena itu tidak hanya  
dilihat, akan tetapi juga harus dikaji dan diteliti serta dipelajari untuk  
dikembangkan menjadi ilmu pengetahuan dan teknologi yang mempunyai  
manfaat untuk kehidupan manusia.

Berdasarkan tafsir Ibnu Katsir QS. Yunus ayat ke-101, Allah SWT  
memerintahkan kepada hamba-hamba-Nya untuk memikirkan tentang nikmat-  
nikmat yang telah diberi Allah dan dalam apa yang Allah ciptakan di langit dan  
di bumi dari ayat-ayat yang agung untuk orang-orang yang mempunyai akal.  
Yang berada di langit berupa bintang-bintang yang bersinar, yang tetap dan yang  
bergerak, matahari, bulan, malam dan siang, serta pergantian keduanya dan  
memasukkan yang satu ke dalam yang lain, hingga yang ini panjang dan yang ini  
pendek, kemudian memendekkan yang ini dan memanjangkan yang itu,  
meninggikan langit, membuatnya luas, indah, dan penuh hiasan (Al-Sheikh,  
1994).

Komposisi *web service* merupakan suatu proses penggabungan/  
penyusunan beberapa *web service* yang telah dipetakan menggunakan proses

bisnis. Komposisi web service menggunakan *Term frequency-inverse document frequency* (TF-IDF) dan *cosine similarity*. TF-IDF diterapkan untuk mendapatkan nilai atau bobot dari *activity* yang relevan dengan kata kunci yang telah diinputkan. Sementara itu metode *cosine similarity* digunakan dalam perhitungan *similarity* antara *activity*, sehingga dapat menghasilkan komposisi dari rangkaian *web service* yang sesuai dengan proses bisnis. Hasil dari komposisi tersebut kemudian digenerate menjadi BPEL. Sehingga dapat digunakan sebagai referensi eksekusi dalam pemrograman.

## 1.2 Pernyataan Masalah

Berdasarkan latar belakang yang telah dipaparkan, maka dapat dirumuskan permasalahan yang akan diangkat dalam penelitian ini sebagai berikut :

1. Bagaimana memetakan pemodelan proses bisnis terhadap *web service*?
2. Bagaimana komposisi web services berdasarkan pemodelan proses bisnis dengan *cosine similarity*?

## 1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut :

1. Memetakan pemodelan proses bisnis dengan web service
2. Komposisi web service berdasarkan pemodelan proses bisnis menggunakan *cosine similarity* dan menghasilkan BPEL (*Business Process Executing Language*)

## 1.4 Manfaat Penelitian

Manfaat yang diperoleh dari penelitian ini adalah :

1. Mempermudah pengembangan *web service*, sehingga organisasi yang membutuhkan *web service* tidak harus membuat dari awal, dengan cara menggunakan *web service* yang sudah ada yang terdapat pada organisasi lain.
2. Dapat mengurangi biaya dalam memulai pengembangan *web service*.

### **1.5 Batasan Penelitian**

Batasan masalah dalam penelitian ini yaitu, data yang digunakan hanya meliputi pemodelan proses bisnis bagian akademik pondok pesantren.

### **1.6 Sistematika Penulisan**

Sistematika penulisan yang digunakan pada penelitian ini adalah sebagai berikut :

#### **BAB 1 PENDAHULUAN**

Bab ini berisi latar belakang penelitian, rumusan masalah, tujuan penelitian, dan sistematika penulisan

#### **BAB 2 KAJIAN PUSTAKA**

Bab ini berisi kajian pustaka yang berkaitan dengan penelitian, yaitu mengenai Proses Bisnis, Pemodelan Proses Bisnis, *Web Service*, *Text Mining*, TF-IDF, serta *Cosine Similarity*.

#### **BAB 3 METODOLOGI PENELITIAN**

Bab ini menjelaskan tentang metode yang digunakan dalam penelitian, desain penelitian, perancangan sistem, serta pengujian sistem.

#### **BAB 4 HASIL DAN PEMBAHASAN**

Bab ini menjelaskan hasil dari perancangan sistem serta pembahasan dari penelitian.

## BAB 5 PENUTUP

Bab ini merupakan penutup yang berisi kesimpulan dari penelitian yang telah dilakukan, serta saran yang diharapkan dapat bermanfaat untuk pengembangan aplikasi komposisi *web service* pada penelitian selanjutnya.





## BAB II KAJIAN PUSTAKA

Pada bagian ini membahas tentang teori-teori yang digunakan dalam melaksanakan penelitian ini.

### 2.1 Proses Bisnis

Proses Bisnis merupakan serangkaian aktifitas yang saling terkait untuk mencapai tujuan bisnis tertentu yang diselesaikan secara berurutan, oleh manusia atau sistem, baik di dalam maupun di luar organisasi (Ramadhani, 2015). Suatu proses bisnis yang baik harus memiliki tujuan efektifitas, efisiensi dan meningkatkan produktifitas dari suatu organisasi. Proses bisnis mempunyai ciri-ciri sebagai berikut (Prasetya, 2017):

1. Mempunyai tujuan tertentu.
2. Mempunyai masukan yang spesifik.
3. Mempunyai keluaran yang spesifik.
4. Memanfaatkan *resource*.
5. Memiliki aktivitas yang dapat dieksekusi dengan urutan tertentu.
6. Dapat melibatkan lebih dari satu organisasi.

Dalam memodelkan suatu proses bisnis, perlu adanya suatu *Standard Operating Procedure* (SOP) yang harus dimiliki oleh setiap organisasi. SOP merupakan dokumen yang berkaitan dengan prosedur yang dilaksanakan secara kronologis untuk menyelesaikan suatu pekerjaan. SOP pada suatu organisasi/perusahaan memiliki tujuan memperoleh hasil kerja yang paling efektif dari pekerja dengan biaya yang serendah-rendahnya sekaligus mengurangi miskomunikasi dan kegagalan dalam mematuhi peraturan perusahaan.

## 2.2 Pemodelan Proses Bisnis

*Business Process Modelling* atau yang sering disebut BPM atau PPB (Pemodelan Proses Bisnis) bisa diartikan sebagai kegiatan yang menggambarkan proses bisnis sebuah perusahaan sehingga prosesnya bisa dipahami, dianalisis, serta ditingkatkan (Katarina & et al, 2013).

Dengan memanfaatkan pemodelan proses bisnis, sebuah organisasi maupun perusahaan bisa melakukan lintas fungsional. Artinya perusahaan maupun organisasi mampu melakukan pengelompokan pekerjaan dan berkas lebih dari satu lembaga dalam sebuah institusi, perusahaan maupun organisasi. Lebih kompleks lagi, pemodelan proses bisnis mampu masuk ke dalam kegiatan eksternal pada organisasi serta system yang terdapat dalam proses inti. Selain itu Pemodelan Proses Bisnis juga memiliki tujuan untuk mendefinisikan tahap-tahap yang dikerjakan dalam rangka memenuhi suatu tujuan (Ramadhani, 2015). Saat ini pemodelan proses bisnis dilakukan dengan standar-standar tertentu, mulai dari UML (*Unified Modeling Language*), BPEL (*Business Process Executing Language*), BPMN (*Business Process Modeling Nnotation*), YAWL (*Yet Another Workflow Language*), dan masih banyak lagi..

### 2.2.1 BPMN

*Business Process Modeling Notation* (BPMN) sebuah teknik yang diperkenalkan oleh *Business Process Modelling Initiative* (BPMI) sebagai sarana untuk mendesain proses bisnis (Dwi Rahmawati, 2017). Proses bisnis diagram dalam BPMN diilustrasikan sebagai diagram alur, disusun dengan tujuan menghasilkan bentuk-bentuk grafis dari operasi bisnis. Dalam operasi bisnis tersebut bisa ditemukan aktivitas-aktivitas serta control langkah kerja

(Ramadhani, 2015). Sasaran inti dari BPMN yaitu untuk menyajikan notasi yang bisa dimanfaatkan dan dipahami pengguna yang mendukung jalannya bisnis, yang terdiri dari bisnis analis yang mendesain proses bisnis, hingga tingkat manajemen yang harus mempunyai kemampuan membaca diagram dengan cepat.

Berikut merupakan aturan yang dapat digunakan sebagai acuan dalam memberikan penotasian proses bisnis (Astrininditiya, 2017) :

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 rendah.
5. Sebuah tanda '+' pada simbol proses menunjukkan bahwa proses ini didekomposisi, jika tidak ada tanda '+', maka proses ini disebut sebuah *task*.

Terdapat empat kategori dasar yang ada pada BPMN, yaitu *Flow Object*, *Connecting Object*, *Swimlanes*, dan *Artifact*. Berikut penjelasan dari keempat kategori tersebut (Ramadhani, 2015);

#### 1. *Flow Object*

- a. *Event* digambarkan dalam bentuk oval serta mendeskripsikan kejadian atau kegiatan yang berlangsung. Secara umum event dibagi menjadi 3, start,

intermediet dan end. Event memberikan dampak jalannya proses serta mengakibatkan kejadian bisa berlangsung (trigger) atau sebuah dampak (result). Setiap event mewakili kejadian permulaan proses bisnis, interupsi proses bisnis dan akhir proses bisnis. Selain 3 event yang disebutkan, terdapat jenis event lain seperti message start yang diilustrasikan seperti start event. Perbedaan mendasar antara kedua event ini adalah adanya tambahan symbol amplop didalam symbol oval.



Gambar 2. 1 Elemen start, intermediate dan end event

- b. *Activity* dalam BPMN digunakan untuk menggambarkan pekerjaan (task) yang harus diselesaikan. Activity secara umum dibagi menjadi empat macam yaitu *task*, *looping task*, *sub process*, dan *looping subprocess*.



Gambar 2. 2 Elemen-elemen Activity

## 2. *Connecting Object*

Antar kejadian yang saling terhubung, aliran pesan bisa diterima menggunakan *connecting object*. Secara umum *connecting object* ada 3 jenis, yaitu :

- a. *Sequence flow*, merupakan opsi *default* sebagai eksekutor proses
- b. *Message Flow*, digunakan sebagai penyalur pesan setiap proses

c. *Association*, merupakan penghubung antara elemen dan *artifact*



Gambar 2.3 Elemen sequence flow, message flow, dan association

### 3. Swimlanes

Elemen dalam diagram BPMN dikelompokkan menggunakan swimlanes. Swimlane secara umum dibagi menjadi dua jenis, yaitu *pool* dan *lane*. Lane merupakan sub bagian dari pool yang mengelompokkan komponen dalam pool menjadi lebih detail.



Gambar 2.4 Pool dan Lane

### 4. Artifact

Penjelasan diagram BPMN bisa digambarkan dengan element artifact. Secara umum artifact dibedakan ke dalam tiga jenis.

- Data yang dibutuhkan dalam proses dijelaskan menggunakan *data Object*
- Untuk menggolongkan beberapa aktivitas dalam proses yang berjalan menggunakan *group*
- Untuk memberikan catatan supaya diagram bisa dimengerti, menggunakan *Annotation*.





Gambar 2. 5 Elemen data object, group, dan annotation

Banyak kelebihan yang dimiliki diagram BPMN, salah satu kelebihan yang dimiliki yaitu mampu memisahkan aliran proses secara visual menurut organisasi atau lembaga yang bekerja.

### 2.2.2 XML Process Definition Language (XPDL)

Standar bahasa berbasis petri-nets bisa didefinisikan menggunakan *XML Process Definition Language* (XPDL), XML digunakan sebagai standar bahasa untuk bertukar definisi proses bisnis antar produk *workflow*. XPDL merupakan standar bahasa yang sesuai dan bisa digunakan untuk menggambarkan proses bisnis yang tidak memiliki interaksi layanan antar aplikasi (Handayani, 2009). XPDL didesain sebagai definisi dalam prosedur, baik berupa semantik serta grafis dari proses bisnis suatu *workflow*. Format terbaik yang telah dirancang khusus agar dapat menyimpan semua aspek dari BPMN serta yang dapat digunakan dalam pergantian diagram BPMN yaitu format XPDL. Dalam artian XPDL merupakan serialisasi XML dari BPMN (Astrininditiya, 2017).

*XML Process Definition Language* atau yang sering disebut dengan XPDL memiliki enam elemen utama, yakni :

1. Elemen <Package> digunakan untuk mengidentifikasi proses bisnis. Model pada komponen digunakan untuk mereduksi detail *redundancy* sehingga bisa dimanfaatkan untuk beberapa jenis proses.

2. Elemen <WorkflowProcess> : yaitu komponen yang dimanfaatkan sebagai identifikasi sub proses dari suatu proses. <WorkflowProcess> dibedakan menjadi dua jenis elemen, <Activity> sebagai dasar untuk identifikasi proses bisnis, dan <Transition> sebagai interaksi antar elemen dari jenis<Activity>.
3. Elemen <ActivitySet> : digunakan sebagai transisi yang menghubungkan antar aktivitas
4. Elemen <BlockActivity> : digunakan sebagai perangkat yang menjalankan perintah dari elemen<ActivitySet>.
5. Elemen <Participant> : menjelaskan detail dari partisipan *workflow*.
6. Elemen <DataType> dan Elemen <DataField> : dipakai sebagai identifikasi detail data relevan *workflow*. Data ini dipakai untuk memilih opsi atau untuk merefrensikan data selain dari *workflow* dan dikirim pada setiap aktivitas serta *subflows*.

Banyak manfaat yang dimiliki dalam pemanfaatan XPDL, salah satu keuntungan yang didapat yaitu alur proses bisnis organisasi hasil transformasi XML bisa lebih mudah dipahami menggunakan deskripsi XML (Putu Wuri Handayani, 2009).

### 2.3 Web Service

*Web service* adalah software yang akan menyimpan data dengan standard XML, sehingga semua komunikasi *web service* dikodekan menggunakan XML. Data yang tersimpan dengan standard XML tersebut, dapat diakses oleh sistem atau aplikasi perangkat lunak yang ditulis dengan berbagai bahasa pemrograman, dan dapat berjalan diberbagai platform, maupun sistem operasi.

David et.al (Booth, 2004) mendefinisikan *Web Service* sebagai sebuah sistem *software* yang dirancang untuk mendukung interaksi interoperabilitas antara mesin ke mesin pada sebuah jaringan. *Web Service* memiliki *interface* yang dijelaskan dalam format *machine-processable* (khususnya WSDL). Sistem lain berinteraksi dengan *web service* melalui cara yang telah ditentukan dan dideskripsikan menggunakan pesan SOAP, biasanya disampaikan menggunakan HTTP dengan serialisasi XML dalam hubungannya dengan web standar yang saling berhubungan.

Kelebihan yang dimiliki oleh Web Service adalah : Loosely Coupled, Ease of Integration, Service Reuse, Standardized Protocol, XML-Based.

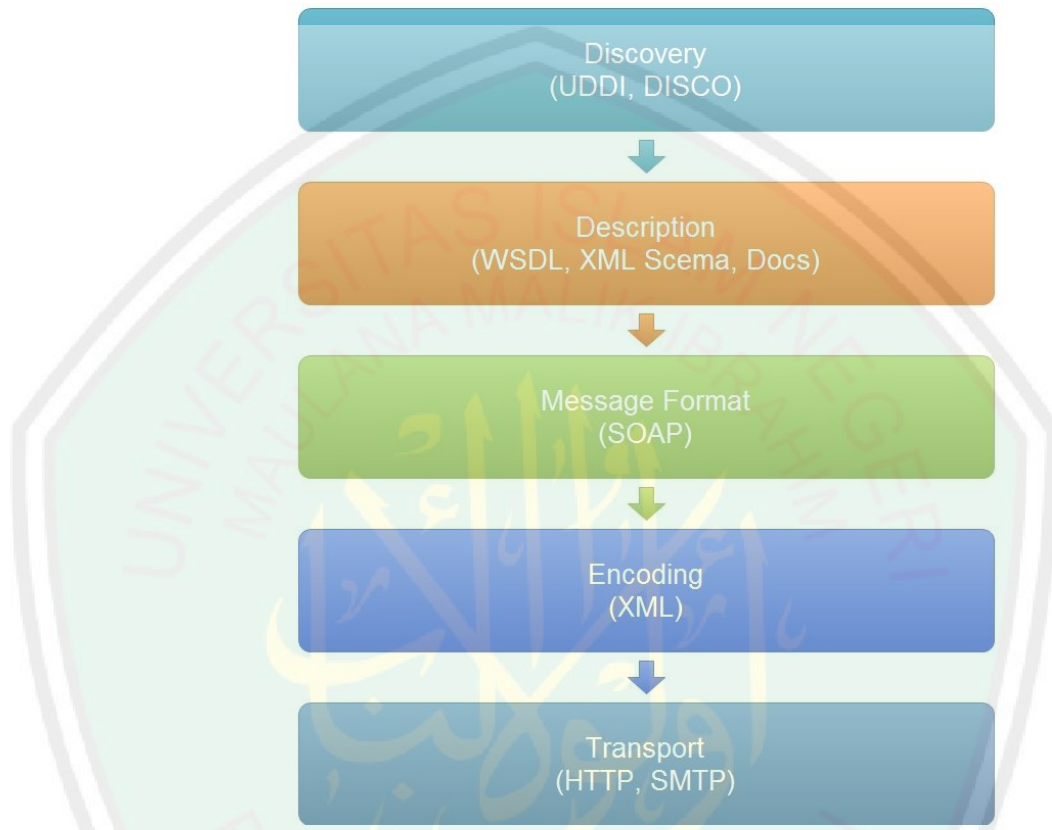
1. *Loosely Coupled* : yaitu tidak memiliki ketergantungan pada *service* yang lain sehingga tiap-tiap *service* dapat berdiri sendiri.
2. *Ease of Integration* : integrasi data lebih mudah..
3. *Service Reuse* : dapat digunakan berkali-kali dengan hanya sekali mendefinisika.
4. *Standardized Protocol* : memanfaatkan protokol standar dalam melakukan komunikasi.
5. *XML-Based* : untuk lapisan transportasi data serta representasi data, *web service* menggunakan XML.

### 2.3.1 Arsitektur Web Service

*Oreinted Architecture* (SOA) merupakan pondasi arsitektur teknologi web service. Dalam SOA setiap komponen mempunyai peran masing-masing. Adapun peran setiap komponen dalam SOA adalah *Service Provider*, *Service Requestor*, *Service Registry* (Wulan, 2017).

1. *Service Provider* : yaitu yang menyediakan *web service*.
2. *Service Requestor* : merupakan pengguna *web service*.
3. *Service Registry* : sebagai penyedia tempat untuk pengembang.

*Web service* memiliki elemen-elemen penyusun sebagai berikut:



Gambar 2. 6 Komponen-komponen *Web Service*

#### 1. *Discovery*

Dalam internet, *Web Service* yang berjalan ditempatkan dalam sebuah penyimpanan. *Universal Description, Discovery and Integration* (UDDI). Menjadi standart untuk melakukan pencarian dan pendaftaran *web services*. Melalui bantuan UDDI *web services* mampu ditemukan dengan parameter *keyword* sesuai dengan kriteria tertentu. Tujuan dari UDDI yaitu menjadi jembatan setiap

perusahaan dengan harapan bisa berkembang dan saling menggunakan *web Services* secara bersama-sama.

## 2. *Description*

Setelah *web services* ditemukan menggunakan UDDI, client perlu mengetahui cara untuk dapat berinteraksi dengan *web service* tersebut dan *service* apa saja yang disediakan. Fungsionalitas *service* yang disediakan oleh *web services* dipaparkan dalam bentuk programmatic interface menggunakan standar description language *Web Service Definition Language* (WSDL). WSDL merupakan suatu dokumen XML yang menjelaskan method yang tersedia dalam suatu *web services*, parameter yang diperlukan untuk memanggil suatu method dan hasil dari method yang akan dipanggil.

## 3. *Message Format*

*Simple Object Access Protocol* (SOAP) merupakan protokol untuk pertukaran informasi terstruktur pada sistem yang terdistribusi / terdesentralisasi. SOAP menggunakan teknologi XML untuk mendefinisikan kerangka struktur *message*. SOAP didesain untuk bersifat *extensible* (user mampu mengembangkan kemampuannya untuk kepentingan di masa mendatang) dan *simple*. Dalam konsep *web service*, SOAP merupakan mekanisme yang digunakan untuk mengirimkan message diantara client dengan *web services*. Soap tidak bergantung pada model pemrograman ataupun sistem operasi.

## 4. *Encoding*

*Extensible Markup Language* (XML) merupakan standar W3C yang mengijinkan data untuk bersifat portabel dan bebas mendiskripsikan dirinya, sehingga dapat dipertukarkan dengan mudah antar aplikasi maupun *devices* pada



berbagai platform. Jadi XML merupakan ide untuk membuat format data standar yang *universal*.

## 5. Transport

*Hypertext Transfer Protocol* (HTTP) mendefinisikan bagaimana suatu pesan disusun dan dikirimkan serta bagaimana web server menanganinya. HTTP merupakan serangkaian aturan untuk pertukaran file (teks, grafik, gambar dan lain-lain) di dalam www. HTTP merupakan protokol aplikasi yang berhubungan dengan protokol TCP/IP yang merupakan protokol utama dalam pertukaran informasi di internet.

### 2.4 Web Services Composition

*Web Service Composition* adalah proses menggabungkan beberapa layanan menjadi satu layanan agar bisa melakukan fungsi yang lebih kompleks. Berdasarkan waktu disusunnya *web services*, *services composition* dikategorikan menjadi dua model, yaitu *semantic (dynamic)* dan *syntactic (static)*. (Quan Z. Sheng, 2014).

#### 1. *Semantic Web Service composition model*

*Semantic* model atau model semantik adalah *web service composition* berbasis *ontology*, yang diusulkan untuk mendukung *service composition* di dalam web semantik. Model semantik memiliki tujuan untuk memungkinkan aplikasi menggunakan anotasi dan mesin inferensi yang tepat untuk mengotomatisasi semua tahap siklus *web services* (*discovery*, *invocation*, *composition* dan *interoperation*) tanpa campur tangan manusia.

## 2. *Syntactic Web service composition model*

*Syntactic model* atau model sintaksis adalah *web service composition* berbasis XML yang menyediakan model “atom” pertukaran pesan *synchronous* atau *asynchronous* untuk memungkinkan mitra bisnis dan prosesnya saling berinteraksi satu sama lain. Interaksi antara *web service* dapat dicapai melalui dua standar XML (*eXtensible Markup Language*) standar : *Web Service Description Language* (WSDL) untuk *service description* dan *Simple Object Access Protocol* (SOAP) untuk pertukaran pesan. Standar WSDL mendefinisikan antarmuka untuk *web service* termasuk metode, pesan, dan standar SOAP untuk memanggil *web service* tertentu.

### 2.4.1 **Business Process Execution Language (BPEL)**

BPEL (*Business Process Executing Language*) merupakan sebuah bahasa yang berbasis XML yang digunakan untuk mendefinisikan proses bisnis dengan *web service*. Penerapan proses bisnis memerlukan standard dan bahasa khusus untuk komposisi ke dalam proses bisnis, dengan menggunakan bahasa yang dapat diterima umum. BPEL merupakan bahasa yang dapat diterima umum dan cepat menjadi standard yang dominan. Tujuan BPEL adalah untuk menyediakan suatu *Web Services* yang berinteraksi dengan *web service* *web service* yang telah tersedia lebih dahulu. Suatu contoh, *web service* yang dapat dikembangkan dengan BPEL adalah layanan pemesanan jasa wisata yang merupakan hasil interaksi dengan layanan pemesanan tiket perjalanan, pemesanan hotel dan layanan pembayaran. (Utomo, 2012).

BPEL berbasis pada *web services*, sehingga diasumsikan bahwa masing-masing proses bisnis yang terlibat diimplementasikan dengan menggunakan *web*

*services*, sehingga BPEL dapat mengatur interaksi yang terjadi antar *web services* dengan menggunakan dokumen XML. BPEL digunakan untuk mendeskripsikan suatu bisnis proses dengan dua cara yang berbeda, yaitu *executable* dan *abstract processes*. *Executable process* adalah menentukan urutan eksekusi dari sejumlah aktivitas, mitra yang terlibat, mempertukarkan pesan antar mitra, dan mekanisme *exception handling*. *Abstract processes* adalah perilaku pertukaran pesan dengan pihak yang berbeda tanpa memberikan informasi mengenai perilaku internal mereka. BPEL mempunyai empat elemen yang terdiri dari : *Process*, *Partner Links*, *Variables*, dan *Sequence* (Widodo, 2016)

1. *Process*, merupakan elemen utama dari BPEL. Nama proses didefinisikan pada *tag* proses dengan atribut *name*. Selain itu *tag* proses juga digunakan untuk memasukkan informasi yang terkait mengenai definisi proses.
2. *Partner Links*, mendefinisikan pihak-pihak yang bekerjasama untuk memenuhi proses bisnis.
3. *Variables*, mendefinisikan struktur data yang akan digunakan program. Definisi mengacu pada jenis pesan WSDL dan elemen dan jenis XSD (XML Schema Definition).
4. *Sequence*, mendefinisikan prosedur proses yang dikomposisi dari BPEL *activities*.

Sesuai dengan konsepnya, proses BPEL terdiri dari beberapa langkah. Setiap langkah disebut *activity*. BPEL mendukung aktivitas dasar dan terstruktur. Aktivitas dasar merepresentasikan konstruksi dasar dan dipakai untuk tugas-tugas yang bersifat umum, di antaranya adalah sebagai berikut (Satrio, 2008) :

### 1. <invoke>

Dipakai untuk memanggil *web services*. Operasi invoke mendukung dua atribut yang lain. Ketika proses bisnis memanggil sebuah operasi *web service*, maka invoke akan mengirim sejumlah parameter seperti parameter pesan masukan input Variable dan output Variable sebagai keluaran. Berikut adalah contoh penggunaan *tag* invoke :

```
<invoke partnerLinks="insuranceA">
  portType="ins:ComputeInsurancePremiumPT"
  operation="ComputerinsurancePremium"
  inputVariable="InsuranceRequest"
  outputVariable="InsuranceResonse"
</invoke>
```

### 2. <receive>

Dipakai untuk menunggu klien saat membangkitkan proses bisnis melalui pengiriman pesan. Contoh atribut atau aktivitas <receive> adalah createInstance, yang berguna untuk memerintahkan aplikasi pengolah BPEL untuk membuat isntan baru dari sebuah proses. Berikut adalah contoh dari aktivitas <receive>

```

<receive partnerLinks="clients">
  portType="com:InsuranceSelectPT"
  operation="SelectInsurance"
  variable="InsuranceRequesnt"
  createInstance="yes"
</receive>

```

### 3. <reply>

Dipakai untuk menghasilkan respon dari operasi. Reply selalu berhubungan dengan <receive> saat proses BPEL dimulai. Contoh penggunaan reply :

```

<reply partnerLink="client">
  <portType="com:InsuranceSelectionPT"/>
  <operation="SelectInsurance"/>
  <variable="InsuranceSelectionResponse"/>
</reply>

```

### 4. <assign>

Dipakai untuk memanipulasi data dalam situasi variabel. Variabel dipakai oleh <invoke>, <receive>, dan <reply> untuk menspesifikasikan pesan input dan output serta untuk membangkitkan operasi pada *web service* partner. Contoh penggunaan <assign>



```

<assign>

  <copy>

    <form variable="InsuranceAResponse"/>

    <to variable="InsuranceSelectionResponse"/>

  <copy>

</assign>

```

#### 5. <throw>

Dipakai untuk mengindikasikan kesalahan dan *exception*

#### 6. <wait>

Dipakai untuk menunggu waktu pengekseskuan *web service*.

#### 7. <terminate>

Dipakai untuk mengakhiri suatu proses.

### 2.5 Text Mining

*Text Mining* memiliki definisi penambangan data yang berupa teks dimana sumber data biasanya didapatkan dari dokumen. Tujuan dari *text mining* adalah untuk mendapatkan informasi yang berguna dari sekumpulan dokumen. Sumber data yang digunakan pada *text mining* adalah data yang memiliki format tidak atau kurang terstruktur, seperti dokumen Word, PDF, kutipan teks, dll. *Text Mining* dapat memberikan solusi dari permasalahan seperti pemrosesan, pengorganisasian dan menganalisa *unstructured text* dalam jumlah besar. Dalam memberi solusi, *text mining* mengadopsi dan mengembangkan banyak teknik dari bidang lain, seperti data mining, *information retrieval*, statistic dan matematik, *machine learning*, *visualization*, *linguistic*, dan *natural language processing*.

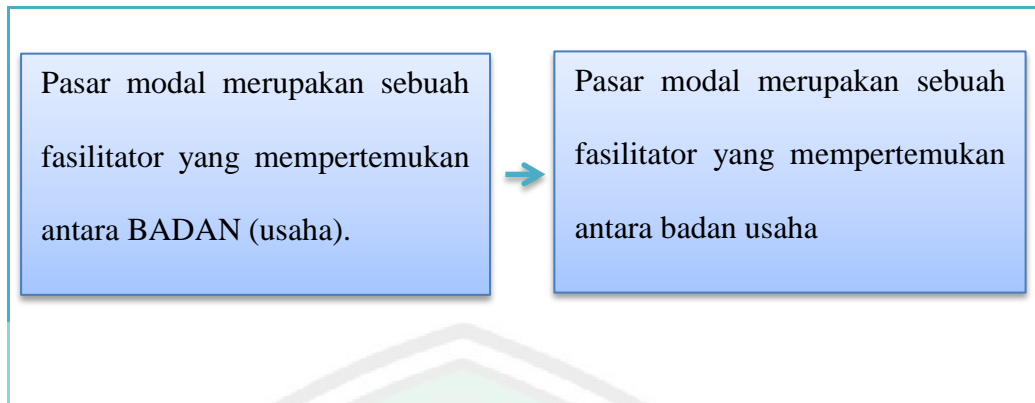
*Text mining* memiliki permasalahan yang sama dengan permasalahan yang dihadapi oleh data mining, yaitu jumlah data yang besar, dimensi yang tinggi, data dan struktur yang terus berubah, dan data noise. Namun keduanya memiliki perbedaan pada data yang digunakan, pada data mining data yang digunakan adalah *structured data*, sedangkan pada *text mining* data yang digunakan pada umumnya adalah *unstructured data*, atau minimal *semistructured*. Hal ini menyebabkan adanya tantangan tambahan pada *text mining* yaitu struktur teks yang kompleks dan tidak lengkap, arti yang tidak jelas dan tidak standard, serta bahasa yang berbeda ditambah translasi yang tidak akurat. Tahapan yang dilakukan secara umum yaitu ekstraksi dokumen (D, 2013).

### 2.5.1 Text Preprocessing

*Text Preprocessing* merupakan tahapan dari proses awal terhadap teks untuk mempersiapkan teks menjadi data yang akan diolah lebih lanjut. Pada tahap ini suatu dokumen dapat dipecah menjadi bab, sub-bab, paragraf, kalimat dan pada akhirnya menjadi potongan kata/*token*. Selain itu angka, huruf kapital, atau karakter-karakter yang lainnya dihilangkan dan dirubah. Berikut tahap *text preprocessing* yang dilakukan secara umum dalam *text mining* :

#### 1. Case Folding

*Case Folding* adalah mengubah semua huruf dalam dokumen menjadi huruf kecil, hanya huruf 'a' sampai 'z' yang diterima. Karakter selain huruf dihilangkan dan dianggap delimiter. Proses *case folding* seperti Gambar 2.7



Gambar 2.7 Case Folding

## 2. Tokenizing

Tahap *tokenizing* adalah tahap pemotongan string input berdasarkan tiap kata yang menyusunnya.



Gambar 2.8 Proses Tokenizing

### 2.5.2 Text Transformation

*Text Transformation* adalah tahapan yang dipergunakan untuk mengubah kata-kata ke dalam bentuk dasar, sekaligus untuk mengurangi jumlah kata-kata

tersebut. Pendekatan yang dapat dilakukan dengan penghapusan *stopwords* dan *stemming*.

### 1. Stopword Removal / Filtering

*Filtering* adalah tahap mengambil kata-kata penting dari hasil term. Dapat menggunakan algoritma *stoplist* (membuang kata yang kurang penting) atau *wordlist* (menyimpan kata penting). *Stoplist/stopword* adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan bag-of-words. Contoh *stopwords* adalah “di”, “dan”, “dari”, “yang”, dan sebagainya. Proses *filtering* seperti pada Gambar 2.10.

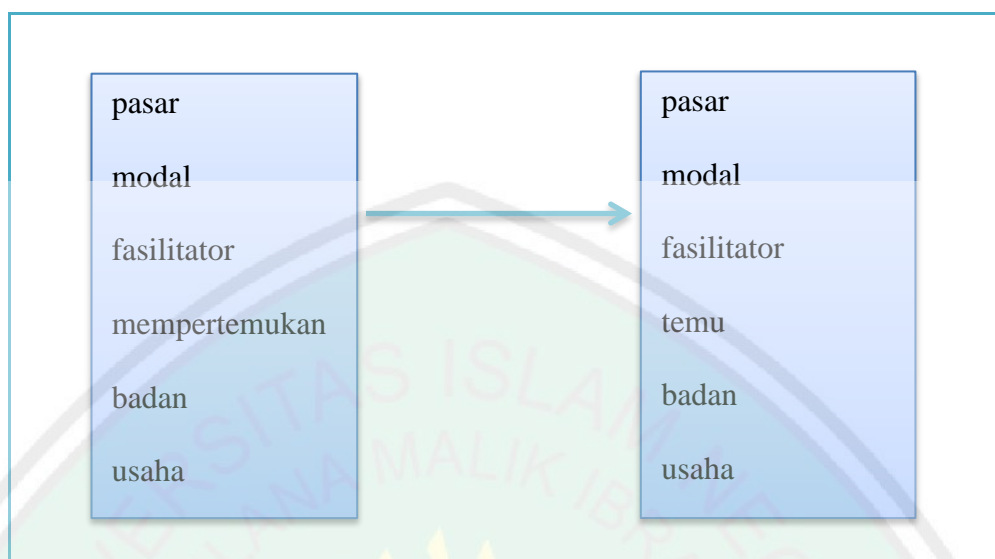


Gambar 2. 9 Proses Filtering

### 2. Stemming

Tahap *stemming* adalah tahap mencari root kata dari tiap kata hasil filtering. Pada tahap ini dilakukan proses pengembalian berbagai bentukan kata ke

dalam suatu representasi yang sama. Proses tahapan *stemming* seperti pada Gambar 2.11 sebagai berikut :



Gambar 2. 10 Proses Stemming

## 2.6 TF (Term Frequency)

*Term Frequency* merupakan salah satu metode untuk menghitung bobot term dalam teks. Dalam metode ini, tiap *term* diasumsikan memiliki kepentingan yang sebanding dengan jumlah kemunculan *term* tersebut pada teks. Bobot sebuah *term*  $t$  pada sebuah teks dirumuskan dalam persamaan berikut :

$$W(d,t)=TF(d,t)$$

Dimana  $TF(d,t)$  adalah *term frequency* dari *term*  $t$  di teks  $d$ . *Term frequency* dapat memperbaiki nilai *recall* pada *information retrieval*, tetapi tidak selalu memperbaiki nilai *precision*. Hal ini disebabkan *term* yang *frequent* cenderung muncul di banyak teks, sehingga *term-term* tersebut memiliki kekuatan keunikan yang kecil. Untuk memperbaiki permasalahan ini, *term* dengan nilai frekuensi yang tinggi sebaiknya dibuang dari *set term*. Menemukan *threshold* yang optimal merupakan fokus dari metode ini (Larasati, 2015).



## 2.7 IDF (Inverse Document Frequency)

IDF (*Inverse Document Frequency*) adalah jumlah dokumen yang mengandung suatu term tertentu. Tiap term akan dihitung nilai document frequency-nya (DF). Lalu term tersebut diseleksi berdasarkan jumlah nilai DF. Jika nilai DF berada di bawah threshold yang telah ditentukan, maka term akan dibuang. Asumsi awalnya adalah bahwa term yang lebih jarang muncul tidak memiliki pengaruh besar dalam proses pengelompokan dokumen. Pembuangan term yang jarang ini dapat mengurangi dimensi fitur yang besar pada text mining. Perbaikan dalam pengelompokan dokumen ini juga dapat terjadi jika term yang dibuang tersebut juga merupakan noise term. Document frequency merupakan metode feature selection yang paling sederhana dengan waktu komputasi yang rendah (Larasati, 2015).

## 2.8 Vector Space Model (VSM)

Metode *Vector Space Model* atau *Term Vector Model* adalah sebuah model aljabar untuk menggambarkan dokumentasi dari beberapa objek sebagai suatu vektor pengenal, contohnya indeks kata. *Vector Space Model* biasanya digunakan dalam penyaringan informasi, temu balik informasi, pengindeksan, dan perangkaan relevansi. Dalam *Vector Space Model*, setiap dokumen terdiri dari *term* ( $T_1, T_2, \dots, T_n$ ) dan setiap *term*  $T_i$  memiliki bobot  $W_i$ . *Term* ( $T_1, T_2, \dots, T_n$ ) dianggap sebagai salah satu elemen vector dalam sistem koordinat N-dimensi.

Proses dari perhitungan metode ini adalah *indexing* dokumen, pembobotan *term* dan perhitungan kesamaan. Proses *indexing* dokumen adalah melalui tahapan-tahapan dalam *text mining*. Proses selanjutnya yaitu pembobotan *term* dengan menggunakan algoritma TF-IDF. Proses yang terakhir yaitu

perhitungan kesamaan dengan pendekatan *Cosine Similarity*. Pada algoritma *vector space model* digunakan rumus untuk mencari nilai kosinus sudut antara dua vektor dari setiap bobot dokumen (WD) dan bobot dari kata kunci (WK). Setelah mendapat kosinus setiap dokumen, maka hasil bobot dari kata kunci diurutkan. Bobot yang besar menjadi prioritas sebagai dokumen yang memiliki hubungan dengan kata kunci. Rumus yang digunakan dalam *vector space model* adalah sebagai berikut :

$$\text{similarity}(d_j, q) = \frac{d_j \cdot q}{|d_j| \cdot |q|} = \frac{\sum_i W_{ij} \cdot W_q}{\sqrt{\sum_i W_{ij}^2} \cdot \sqrt{\sum_i W_q^2}}$$

Keterangan :

- $W_{ij}$  : bobot kata  $i$  pada dokumen  $j$
- $W_q$  : bobot query

## 2.9 Cosine Similarity

Metode *Cosine Similarity* merupakan metode yang digunakan untuk menghitung *similarity* (tingkat kesamaan) antar dua buah objek. Pada umumnya, nilai yang dihasilkan oleh fungsi *similarity* berkisar pada interval  $[0...1]$ . Namun ada juga beberapa fungsi *similarity* yang menghasilkan nilai yang berada di luar interval tersebut. Namun juga terdapat beberapa fungsi *similarity* yang menghasilkan nilai yang berada di luar interval tersebut. Normalisasi dapat dilakukan untuk memetakan hasil fungsi tersebut pada interval  $[0...1]$ .

*Cosine Similarity* adalah perhitungan tingkat kesamaan antara dua vektor  $n$  dimensi dengan mencari kosinus dari sudut diantara keduanya dan sering digunakan untuk membandingkan dokumen dalam preprocessing. Rumus *cosine similarity* adalah sebagai berikut :

$$\text{Cosine } (x, y) = \cos (\theta) = \frac{x \cdot y}{\|x\| \cdot \|y\|}$$

Dimana :

$x \cdot y$  : vector dot product dari x dan y, dihitung dengan  $\sum_{k=1}^n x_k y_k$

$\|x\|$  : panjang vektor x, dihitung dengan  $\sum_{k=1}^n x_k^2$

$\|y\|$  : panjang vektor y, dihitung dengan  $\sum_{k=1}^n y_k^2$

Semakin besar nilai dari hasil fungsi *cosine similarity*, maka kedua objek yang dievaluasi dianggap semakin mirip. Sebaliknya, maka semakin kecil hasil fungsi *similarity*, maka kedua objek tersebut dianggap semakin berbeda. Pada fungsi yang menghasilkan nilai pada jangkauan [0...1], nilai 1 melambangkan kedua objek persis sama, sedangkan nilai 0 melambangkan kedua objek sama sekali berbeda (Triana, Saptono, & Sulistyono, 2014).

## **BAB III**

### **METODE PENELITIAN**

Pada bab ini akan dibahas mengenai beberapa hal, yaitu tahapan penelitian yang akan dilakukan, kebutuhan sistem yang akan dibuat dan penyelesaian masalah pencocokan pemodelan proses bisnis dengan *web service* menggunakan pembobotan metode TF IDF dan *cosinus similarity*.

#### **3.1 Desain Penelitian**

##### **3.1.1 Gambaran Umum Sistem**

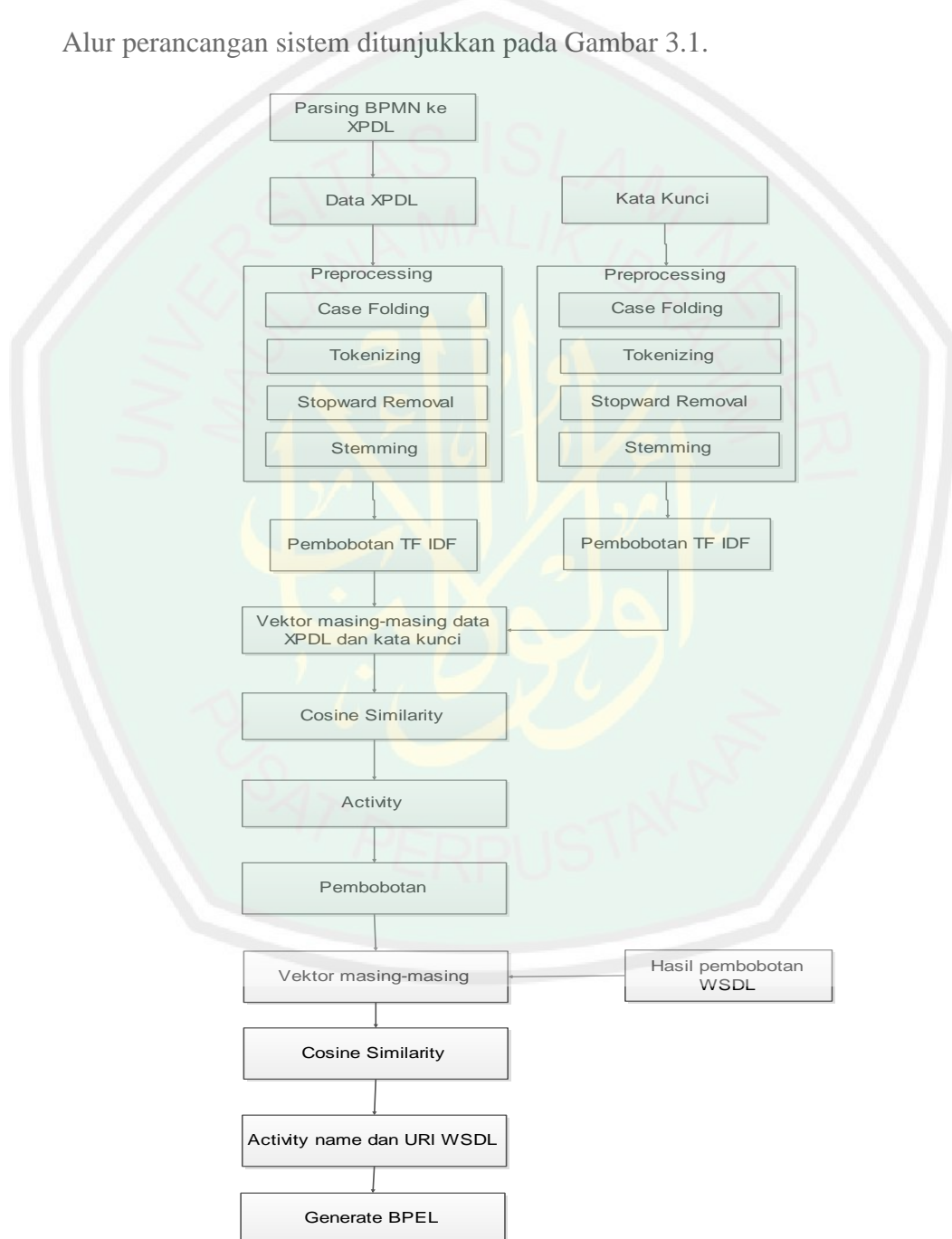
Sistem yang dibangun merupakan sistem yang berfungsi melakukan komposisi *web service*. Komposisi dilakukan dengan melakukan pencarian dokumen XPDL untuk mengambil *activity*. *Activity* dijadikan dasar untuk melakukan pemetaan dengan *web service* menggunakan metode TF-IDF dan *cosine similarity*. Hasil pemetaan antara *web service* dengan proses bisnis digunakan untuk komposisi (penyusunan ulang) dari beberapa *web service* yang telah dipetakan. Hasil dari komposisi kemudian di *generate* menjadi BPEL.

##### **3.1.2 Sumber Data**

Sumber data yang digunakan dalam penelitian ini merupakan data sekunder. Data sekunder merupakan data yang diperoleh dari sumber yang sudah ada. Pada penelitian ini data yang digunakan diperoleh dari penelitian sebelumnya mengenai *web service discovery* (Dewi, 2018) dan *workflow repository* (Astrininditiya, 2017).

### 3.2 Perancangan Sistem

Proses perancangan sistem digunakan untuk menentukan tahapan-tahapan yang akan dilakukan untuk menyelesaikan masalah dalam penelitian ini. Tahapan perancangan sistem dimulai dari *parsing file* BPMN ke XPD, selanjutnya *preprocessing* data XPD dan kata kunci, hingga dapat di-*generate* menjadi BPEL. Alur perancangan sistem ditunjukkan pada Gambar 3.1.



Gambar 3.1 Alur Perancangan Sistem



### 3.2.1 Preprocessing

Tahapan pertama sebelum data dapat dianalisa yaitu melakukan *preprocessing*. Data XPDL dari *file* BPMN yang telah diparsing diolah menggunakan *preprocessing*. *Preprocessing* merupakan tahapan awal dalam *text mining* yang bertujuan untuk melakukan seleksi data yang akan diproses pada setiap dokumen. Proses *preprocessing* terdiri dari *case folding*, *tokenizing*, *filtering*, dan *stemming*. Sebagai contoh terdapat empat dokumen data XPDL yang diberi kode D1, D2, D3, dan D4. Seluruh dokumen berhubungan dengan penerimaan santri baru saja, dan difokuskan dalam perhitungan. Contoh dokumen tersebut seperti pada tabel 3.1 di bawah ini :

Tabel 3.1 Contoh dokumen untuk pencarian

Dokumen	Isi Dokumen
D1	Daftar ulang santri baru daftar ulang santri lama pendataan santri pondok pembuatan kalender akademik rekap data kbm rekap hasil ujian kalkulasi nilai ujian dan proses kelulusan pendataan alumni daftar wisuda
D2	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
D3	Pendataan sarpras perbaikan sarana pengecekan barang rekap data sarpras pembelian sarana prasana baru
D4	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

Dilakukan pencarian terhadap empat dokumen tersebut dengan kata kunci/  
*query* : "Daftar ulang santri baru".

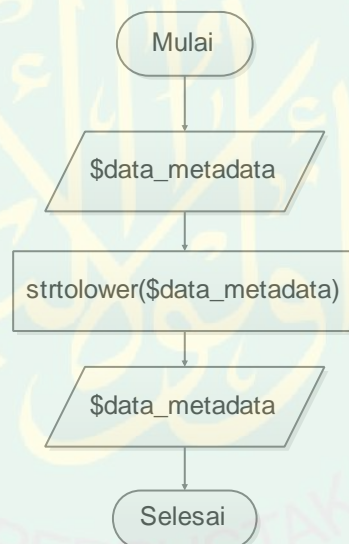
Tabel 3.2 Contoh kata kunci untuk pencarian

Kata Kunci	Isi Kata Kunci
K	Daftar ulang santri baru

### *Case Folding*

*Case Folding* merupakan tahapan yang mengubah huruf kapital dalam dokumen menjadi huruf kecil. Hanya huruf 'a' hingga huruf 'z' yang diterima. Karakter selain huruf akan dihilangkan dan dianggap delimiter (pembatas).

Berikut merupakan *flowchart* dari tahap *case folding* :



Gambar 3. 2 Flowchart case folding

Tahapan *preprocessing* yang pertama yaitu *case folding*, skrip program *case folding* dituliskan seperti pada gambar 3.3 berikut ini:

```

function preprocessing($data_metadata){
    $listtanda = array(".",":",";","?","!","(",")");
    foreach ($listtanda as $i => $value) {
        $data_metadata = str_replace($i, "", $data_metadata);
    }
    $data_metadata = strtolower($data_metadata);
}

```

Gambar 3.3 Skrip program *case folding*

Berikut merupakan hasil dari *preprocessing* pada tahap *case folding* yang disajikan pada tabel , dimana secara keseluruhan dokumen diubah menjadi huruf kecil.

Tabel 3.3 Hasil *case folding* dokumen

Dokumen	Isi Dokumen
D1	daftar ulang santri baru daftar ulang santri lama pendataan santri pondok pembuatan kalender akademik rekap data kbm rekap hasil ujian kalkulasi nilai ujian dan proses kelulusan pendataan alumni daftar wisuda
D2	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
D3	pendataan sarpras perbaikan sarana pengecekan barang rekap data sarpras pembelian sarana prasana baru
D4	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

Kemudian hasil dari *preprocessing* pada tahap *case folding* kata kunci semua kata dirubah menjadi huruf kecil, dapat dilihat pada tabel 3.4.

Tabel 3.4 Hasil *case folding* kata kunci

Kata Kunci	Isi Kata Kunci
K	daftar ulang santri baru

### 1. Tokenizing

*Tokenizing* adalah tahap pemotongan *string input* berdasarkan tiap kata yang menyusunnya. Input berupa kalimat/kata hasil proses *case folding*, dan memiliki output berupa kumpulan kata. Berikut merupakan *flowchart tokenizing* yang disajikan pada gambar 3.4 :



Gambar 3.4 Flowchart tokenizing

Langkah selanjutnya pada *preprocessing* adalah *tokenizing*, dimana implementasi dalam program dapat dilihat pada gambar 3.5.

```

function preprocessing($data_metadata) {
    $listtanda = array(".", ",", ":", ";", "?", "!", "(", ")");
    foreach ($listtanda as $i => $value) {
        $data_metadata = str_replace($listtanda, " ",
$data_metadata);
    }
    //memisah per kata
    $kata = explode(" ", $data_metadata);
    // Menghitung jumlah kata
    $jml_kata = count($kata)-1;
    //menghapus kata yang sama dengan stopwords
    for ($i=0; $i<=$jml_kata; $i++){
        if (in_array($kata[$i], $stoplist)) {
            unset($kata[$i]);
        }
    }
}

```

Gambar 3.5 Source code tokenizing

Setelah melakukan *case folding* pada proses sebelumnya, pada tahap ini dokumen akan melalui proses pemotongan menjadi perkata serta menghilangkan tanda baca yang ada sehingga tidak menghambat prosedur pencarian. Berikut merupakan hasil dari proses *tokenizing* dokumen dan kata kunci yang telah disajikan pada tabel 3.5 dan tabel 3.6.

Tabel 3.5 Hasil *Tokenizing* dokumen

No	Hasil	No	Hasil	No	Hasil	No	Hasil
1	Daftar	11	Rekap	21	penyesuaian	31	Perencanaan
2	Ulang	12	Kbm	22	Guru	32	Kegiatan
3	Santri	13	Hasil	23	Wali	33	Hari
4	Baru	14	Ujian	24	Beli	34	Absen



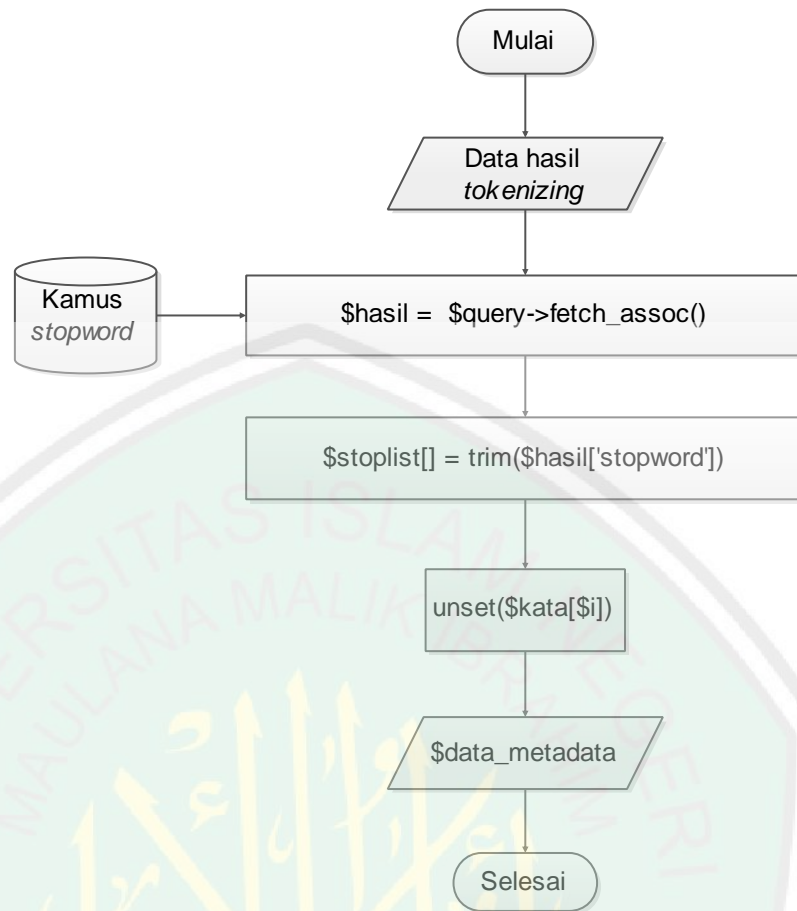
No	Hasil	No	Hasil	No	Hasil	No	Hasil
5	lama	15	Proses	25	Cek	35	Agama
6	Pendataan	16	Lulus	26	Barang	36	Pagi
7	Pondok	17	Alumni	27	Sarana	37	Rutin
8	pembuatan	18	Wisuda	28	Sarpras	38	Sore
9	Kalender	19	Kelas	29	Nilai	39	Sekolah
10	Akademik	20	pengelompokan	30	Kalkulasi	40	Diniyah

Tabel 3.6 Perolehan *tokenizing* kata kunci

No	Hasil
1	Daftar
2	Ulang
3	Santri
4	Baru

## 2. Filtering

Tahap *filtering* merupakan tahapan pengambilan kata-kata yang penting yang diperoleh dari tahap *tokenizing*. Tahapan *filtering* menggunakan algoritma *wordlist* (menyimpan kata yang dianggap penting) atau *stoplist/stopword* (membuang kata yang kurang penting). Contoh *stopword* yaitu “di”, “dari”, “dan”, “yang” dan lain-lain. Input berupa kumpulan kata hasil dari proses *tokenizing*, dan mempunyai output berupa kumpulan term yang siap untuk diolah dengan *stemming*, seperti pada gambar 3.6.



Gambar 3. 6 Flowchart Filtering

Skrip program *filtering* dituliskan seperti pada gambar 3.7 berikut ini:

```

$sql = "SELECT * FROM tb_stopwords";
$query = $conn->query($sql);
//memanggil stopwords
while($hasil = $query->fetch_assoc()){
    $stoplist[] = trim($hasil['stopword']);
}

//memisah perkata
$kata = explode(" ", $data_metadada);
//Menghitung jumlah kata
$jml_kata = count($kata)-1;
//menghapus kata yang sama dengan stopwords
  
```

```

for ($i=0; $i<=$jml_kata; $i++){
    if (in_array($kata[$i], $stoplist)) {
        unset($kata[$i]);
    }
}

//menggabungkan perkata
$data_metadata = implode(" ", $kata);
$data_metadata = strtolower(trim($data_metadata));
return $data_metadata;

```

Gambar 3. 7 Source code filtering

Selanjutnya, pada proses *filtering* yaitu tahap penghilangan kata-kata yang tidak penting dan hasilnya ditunjukkan pada tabel 3.7. Proses ini memiliki tujuan supaya tidak merusak nilai bobot yang ada pada tiap dokumen. Karena, yang diperhitungkan kata-kata yang terpenting.

Tabel 3.7 Hasil *filtering* dokumen

No	Hasil	No	Hasil	No	Hasil	No	Hasil
1	Daftar	11	Rekap	21	penyesuaian	31	Perencanaan
2	Ulang	12	Kbm	22	Guru	32	Kegiatan
3	Santri	13	Hasil	23	Wali	33	Hari
4	Baru	14	Ujian	24	Beli	34	Absen
5	lama	15	Proses	25	Cek	35	Agama
6	pendataan	16	Lulus	26	Barang	36	Pagi
7	Pondok	17	Alumni	27	Sarana	37	Rutin
8	pembuatan	18	Wisuda	28	Sarpras	38	Sore
9	Kalender	19	Kelas	29	Nilai	39	Sekolah
10	akademik	20	pengelompokan	30	Kalkulasi	40	Diniyah

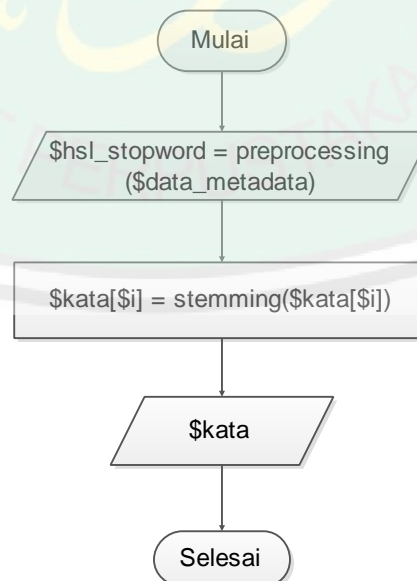
Sedangkan untuk hasil *filtering* pada kata kunci ditunjukkan pada tabel 3.8, dimana *filtering* pada kata kunci tidak memiliki perubahan dari langkah sebelumnya.

Tabel 3.8 Hasil *filtering* kata kunci

No	Hasil
1	Daftar
2	Ulang
3	Santri
4	Baru

### 3. Stemming

*Stemming* merupakan suatu proses yang dikerjakan setelah tahap penghilangan kata-kata serta simbol yang kurang penting. Tahap *stemming* merupakan suatu proses mengubah kata-kata yang ada menjadi akar kata, sehingga mudah untuk diproses. Hasil dari *stemming* dapat diekspansi untuk mendapatkan kata yang sama atau sinonim. Proses *stemming* dilakukan seperti pada gambar 3.8.



Gambar 3. 8 Flowchart stemming

Berikut merupakan implementasi proses *stemming* ke dalam *source code* :

```

function cari($data_metadata) {
    include "konnnect.php";
    $sql = "SELECT * FROM tb_katadasar WHERE
katadasar='$data_metadata'";
    $query = $conn->query($sql);
    // $hasil = mysql_num_rows(mysql_query("SELECT * FROM
tb_katadasar WHERE katadasar='$data_metadata'"));
    $hasil = $query->num_rows;
    return $hasil;
}

//langkah 1 - hapus partikel
function hapuspartikel($data_metadata) {
    if(cari($data_metadata)!=1) {
        if((substr($data_metadata, -3) == 'kah' )||
            ( substr($data_metadata, -3) == 'lah' )||
            ( substr($data_metadata, -3) == 'pun' )){
            $data_metadata = substr($data_metadata, 0, -3);
        }
    }
    return $data_metadata;
}

//langkah 2 - hapus possesive pronoun
function hapuspp($data_metadata) {
    if(cari($data_metadata)!=1) {
        if(strlen($data_metadata) > 4) {
            if((substr($data_metadata,-2)==

```



```

'ku') || (substr($data_metadata, -2) == 'mu')) {

    $data_metadata = substr($data_metadata, 0, -2);
} else if ((substr($data_metadata, -3) == 'nya')) {

    $data_metadata = substr($data_metadata, 0, -3);

}

}

}

return $data_metadata;

}

```

Gambar 3.9 Source code stemming

Tahap *stemming* mengubah *term* menjadi akar kata sehingga memudahkan proses pengambilan sinonimnya. Hasil dari tahapan *stemming* dokumen dan kata kunci disajikan pada tabel berikut :

Tabel 3.9 Hasil *stemming* dokumen

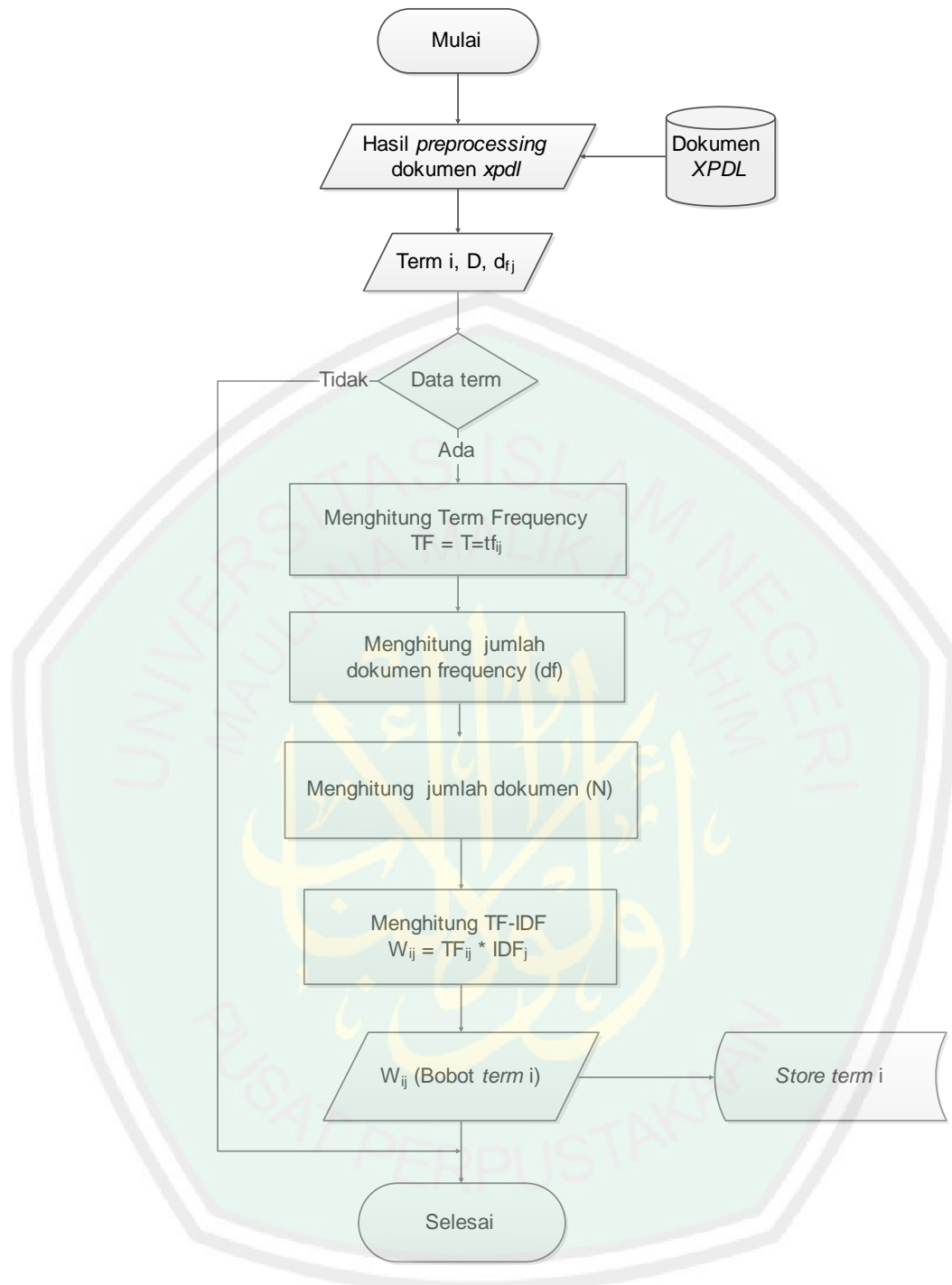
No	Hasil	No	Hasil	No	Hasil	No	Hasil
1	Daftar	11	Rekap	21	Suai	31	Rencana
2	Ulang	12	Kbm	22	Guru	32	Giat
3	Santri	13	Hasil	23	Wali	33	Hari
4	Baru	14	Ujian	24	Beli	34	Absen
5	lama	15	Proses	25	Cek	35	Agama
6	Data	16	Lulus	26	Barang	36	Pagi
7	Pondok	17	Alumni	27	Sarana	37	Rutin
8	Buat	18	Wisuda	28	Sarpras	38	Sore
9	kalender	19	Kelas	29	Nilai	39	Sekolah
10	Akademik	20	Kelompok	30	Kalkulasi	40	Diniyah

Tabel 3.10 Perolehan *stemming* kata kunci

No	Hasil
1	Daftar
2	Ulang
3	Santri
4	Baru

### 3.2.2 TF-IDF

Metode TF-IDF (*Term frequency – Inverse Document Frequency*) merupakan suatu metode yang menggunakan dasar nilai statistik untuk memperlihatkan munculnya suatu kata yang terdapat pada dokumen. TF (*Term frequency*) merupakan frekuensi kemunculan *term* (t) pada dokumen. DF (*Document frequency*) merupakan banyaknya dokumen dimana suatu *term* (t) muncul.



Gambar 3. 10 Flowchart *Term frequency – Inverse Document Frequency*

TF-IDF merupakan bobot dari kata (*term*) yang diperoleh dari nilai TF serta nilai IDF, yang didefinisikan dengan

$$TF_i = tf_{ij}$$

Kemudian dilanjutkan dengan mencari IDF (*Inverse Document Frequency*) yang digunakan untuk mencari banyaknya *document* yang mengandung sebuah *term*. Untuk mendapatkan nilai *Inverse DF* dapat menggunakan rumus sebagai berikut :

$$IDF_t = \frac{1}{df} \text{ Atau}$$

$$IDF_t = \left( \log \left( \frac{N}{n_j} \right) \right)$$

Yang mana N adalah jumlah seluruh dokumen yang ada, serta  $n_j$  adalah banyaknya dokumen yang berisikan *term* / kata i. Nilai bobot *term* i ( $W_{ij}$ ) dalam *information retrieval* dapat dihitung menggunakan rumus :

$$W_{ij} = TF_{ij} \times IDF_{fi}$$

Implementasi TF-IDF dalam program dapat dilihat pada gambar berikut :

```
function indexData(){
    include 'Konnnect.php';
    //hapus index sebelumnya
    mysql_query("TRUNCATE TABLE tb_index_meta");

    //ambil semua berita (teks)
    $sql_opt = mysql_query($conn, "SELECT * FROM repo_meta
ORDER BY id_wfmeta");
    $data = mysql_num_rows($sql_opt);
    print("Mengindeks sebanyak " . $data . " berita. <br />");

    while($row = mysql_fetch_array($sql_opt)) {
        $id_opt = $row['id_wfmeta'];
        $data_metadata = $row['activity'];
```

```

//terapkan preprocessing

$data_metadata = kataDasar($data_metadata);

//simpan ke inverted index (tb_index_meta)

$opt = explode(" ", trim($data_metadata));

foreach ($opt as $j => $value) {

//hanya jika Term tidak null atau nil, tidak kosong

if ($opt[$j] != "") {

//berapa baris hasil yang dikembalikan query tersebut?

$hitung_term = mysql_query($conn, "SELECT count_keyword
FROM tb_index_meta WHERE term_keyword = '$opt[$j]' AND id_wfmeta
= $id_opt");

$jml_term = mysql_num_rows($hitung_term);

//jika sudah ada id_wfmeta dan Term tersebut , naikan

if ($jml_term > 0) {

$data_term = mysql_fetch_array($hitung_term);

$count = $data_term['count_keyword'];

$count++;

mysql_query($conn, "UPDATE tb_index_meta SET
count_keyword = $count WHERE term_keyword = '$opt[$j]' AND
id_wfmeta = $id_opt");

}

//jika belum ada, langsung simpan ke tb_index_meta

else {

mysql_query($conn, "INSERT INTO
tb_index_meta(tb_index_meta,id_wfmeta,term_keyword,count_keyword

```



```

) VALUES (NULL, '$id_opt', '$opt[$j]', 1)"); }}}}
}

```

Gambar 3. 11 Skrip program TF-IDF

```

function hitungBobot() {
    include 'Konnnect.php';
    //berapa jumlah id_wfmeta total?, n
    $index = mysql_query($conn, "SELECT DISTINCT id_wfmeta
FROM tb_index_meta");
    $n = mysql_num_rows($index);
    //ambil setiap record dalam tabel tb_index_meta
    //hitung bobot untuk setiap Term dalam setiap id_wfmeta
    $sql_index = mysql_query($conn, "SELECT * FROM
tb_index_meta ORDER BY id_index_meta");
    $hasil = mysql_num_rows($sql_index);
    print("Terdapat " . $hasil . " Term yang diberikan bobot.
<br />");

    while($dt_index = mysql_fetch_array($index)) {
        $term = $dt_index['term_keyword'];
        $tf = $dt_index['count_keyword'];
        $id = $dt_index['tb_index_meta'];

        //berapa jumlah dokumen yang mengandung term itu?, N
        $hitung_term = mysql_query($conn, "SELECT Count(*)
as N FROM tb_index_meta WHERE term_keyword = '$term'");
        $hasil_term = mysql_fetch_array($hitung_term);
        $NTerm = $hasil_term['N'];
        $w = $tf * log($n/$NTerm);
        //update bobot dari term tersebut

```

```

$updateBobot = mysql_query($conn, "UPDATE
tb_index_meta SET bobot_keyword = $w WHERE id_index_meta =
$id");

} //end while $rowbobot

}

```

Gambar 3. 12 Skrip program TF-IDF

Penerapan perhitungan TF-IDF untuk mendapatkan bobot dari perolehan *preprocessing*, disajikan pada tabel 3.11:

Tabel 3.11 Menghitung TF-IDF

Term	Kata Kunci	A1	A2	A3	A4	DF	Idf = $\log(n/df)$
daftar	1	3	3	0	0	7	0,234083
Ulang	1	2	2	0	0	5	0,30103
Santri	1	3	3	0	2	9	0,191886

Selanjutnya, sesudah memperoleh hasil TF-IDF, tahap selanjutnya adalah menghitung bobot seperti pada tabel 3.12 :

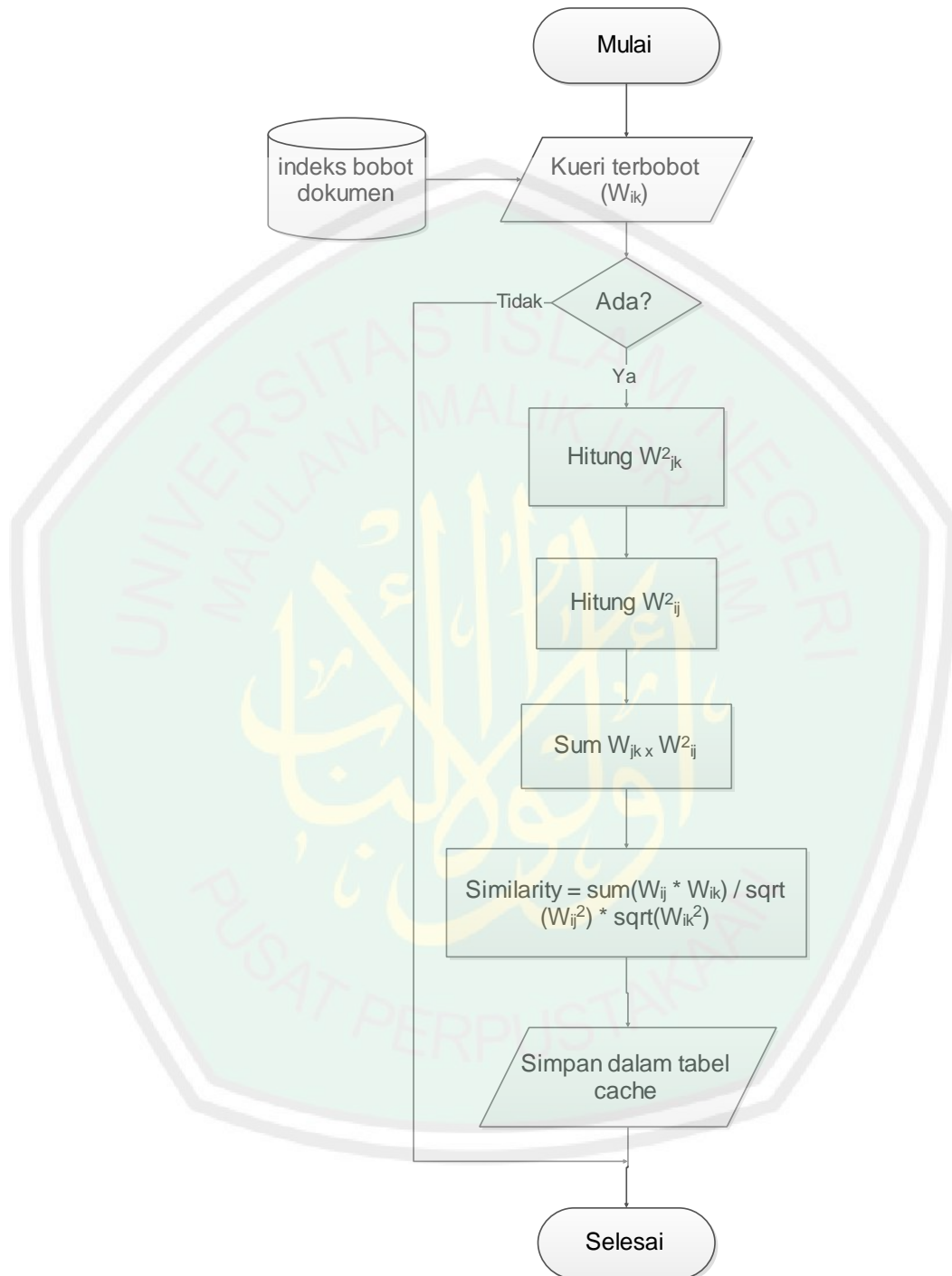
Tabel 3.12 Menghitung bobot

Term	IDF= $\log(n/df)$	W				
		Kata Kunci	A1	A2	A3	A4
Daftar	0,234083	0,234083	0,70225	0,70225	0	0
Ulang	0,30103	0,30103	0,60206	0,60206	0	0
Santri	0,191886	0,191886	0,575657	0,575657	0	0,383771

### 3.2.3 Cosine Similarity

Setelah mendapatkan hasil bobot *term* dengan TF-IDF, maka pengukuran selanjutnya yang dilakukan yaitu menghitung kemiripan menggunakan *cosine similarity*. Kemudiann setelah dihitung, langsung disimpan pada tabel yang telah disediakan agar memudahkan dalam menampilkan hasil pencarian. Setelah

dilakukan pembobotan, tahap selanjutnya melakukan perhitungan kemiripan antara *web service A* dan *web service B*.



Gambar 3. 13 Flowchart cosine similarity

Untuk mendapatkan nilai *cosine similarity* digunakan rumus sebagai berikut :

$$\text{Cosine (Di)} = \frac{\text{Sum } (W_{ij} \times W_{ik})}{\text{sqrt}(W_{ij}^2) \times \text{sqrt}(W_{ik}^2)}$$

Jika nilai pendekatan sama, maka yang ditentukan *web service* yang mendekati kriteria.

```
function cosinesim($query){
    include 'konnnect.php';
    include_once 'fungsi_preprocessing2.php';
    $jmltermn = $termn->fetch_assoc();
    $n = $jmltermn['n'];
    $tquery = explode(" ", $query);
    $panjangvektorquery = 0;
    $bobotquery = array();

    for ($i=0; $i<count($tquery); $i++) {
        $sNTerm = $conn->query("SELECT count(*) as N from
tb_index_meta WHERE term_keyword = '$tquery[$i]'");
        $dataNTerm = $sNTerm->fetch_assoc();
        $NTerm = $dataNTerm['N'];

        if($NTerm > 0){
            $idf = log($n/$NTerm);
        }else{
            $idf = 0;
        }

        $bobotquery[] = $idf;
        $panjangvektorquery = $panjangvektorquery + $idf *
$idf;
    }

    $panjangvektorquery = sqrt($panjangvektorquery);
}
```

```

    $jmlsim = 0;

    $panggildocId = $conn->query("SELECT * FROM tb_vector_meta
ORDER BY id_vektor_meta");

    while ($datadocId = $panggildocId->fetch_assoc()) {

        $dotproduk = 0;

        $id_vector_meta = $datadocId['id_wfmeta'];
        $pvdocId = $datadocId['panjang_vektor'];

        $panggilTerm = $conn->query("SELECT * FROM
tb_index_meta WHERE id_wfmeta = $id_vector_meta");
        while ($dataTerm = $panggilTerm->fetch_assoc()) {
            for ($i=0; $i<count($tquery); $i++) {
                if($dataTerm['term_keyword'] == $tquery[$i]){
                    $dotproduk = $dotproduk +
                    $dataTerm['bobot_keyword'] * $bobotquery[$i]; }
            }
            if($dotproduk > 0){
                $similarity = $dotproduk / ($panjangvektorquery *
                $pvdocId);
                $simpanCache = $conn->query("INSERT INTO
                tb_cache_meta (query,docid,value) VALUES
                ('$query',$id_vector_meta,$similarity)");
                $jmlsim++; }
        }

        if($jmlsim == 0){
            $simpanCache = $conn->query("INSERT INTO
            tb_cache_meta (query,docid,value) VALUES ('$query',0,0)");
        }
    }
}

```



Gambar 3. 14 Source code cosine similarity

Hasil penerapan perhitungan *Cosine Similarity* untuk menghitung persamaan antara kata kunci dan *activity*, dapat dilihat pada tabel 3.13:

Tabel 3.13 Hasil perhitungan *cosine similarity*

Jumlah kuadrat bobot	0,306267	6,652711	8,503442	4,935512	11,26388
Panjang Vektor	0,553414	2,579285	2,916066	2,221601	3,356171
Perkalian Bobot Kata Kunci dengan activity		1,427411	1,613791	1,229464	1,857351
<i>Cosine Similarity</i>		0,406411	0,359474	0,100883	0,039648

### 3.2.4 Generate BPEL

Setelah proses *mapping* dan *composing* pemodelan proses bisnis dan *web services* maka tahap selanjutnya yakni mengenerate BPEL. Berikut merupakan contoh struktur BPEL.

```

<process>
  <!-- Definition and roles of process
  participants -->
  <partnerLinks> ... </partnerLinks>
  <!-- Data/state used within the process -->
  <variables> ... </variables>
  <!-- Properties that enable conversations -->
  <correlationSets> ... </correlationSets>
  <!-- Exception handling -->
  <faultHandlers> ... </faultHandlers>
  <!-- Error recovery - undoing actions -->
  <compensationHandlers> ...

```

```

</compensationHandlers>

  <!-- Concurrent events with process itself -->

  <eventHandlers> ... </eventHandlers>

  <!-- Business process flow -->

  (activities)*

</process>

```

Gambar 3. 15 Struktur BPEL

### 3.3 Pengujian Sistem

Pengujian sistem merupakan teknik atau aturan yang digunakan sebagai uji coba perangkat lunak, mempunyai mekanisme untuk menentukan data uji yang dapat menguji perangkat lunak secara lengkap serta memiliki peluang yang tinggi dalam mendeteksi kesalahan. Pada penelitian ini mengaplikasikan pengujian dengan menggunakan *Test case*.

Tabel 3.14 Pengujian sistem

No	Kata Kunci	Output
1	Pendaftaran santri baru	Activity PSB Operasi PSB Url Web service PSB
2	Pendaftaran pondok pesantren modern	Activity PSB Operasi PSB Url Web service PSB
3	Penerimaan calon peserta baru	Activity PSB Operasi PSB Url Web service PSB

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

Bab ini menjelaskan uji coba dan pembahasan sistem yang telah dibangun. Untuk itu, diperlukan penjelasan dari penerapan tiap tahapan-tahapan yang ada dalam percobaan serta menerapkan perolehan uji coba yang sudah dilaksanakan. Dari penelitian ini diperoleh hasil keluaran sistem pencarian dan pencocokan pemodelan proses bisnis dengan *web service*. Proses yang digunakan pada penelitian ini meliputi preprosesing pada basis data activity pemodelan proses bisnis, pembobotan, serta pencocokan dengan *web service*. Melalui penelitian ini diharapkan dapat memberikan manfaat dan kemudahan bagi ilmuan khususnya dan masyarakat pada umumnya untuk mencocokkan pemodelan proses bisnis dengan *web service* sehingga dapat digenerate menjadi BPEL.

#### **4.1 Prosedur Pengujian**

Prosedur pengujian merupakan proses untuk mendapatkan hasil uji coba yang memiliki tingkat akurasi terukur dengan cara menerapkan metode yang diaplikasikan pada penelitian komposisi *web service* dengan tahapan-tahapan yang terencana dan sistematis. Bagian penting pada pengujian aplikasi yaitu dengan mendeskripsikan tahapan-tahapan yang akan digunakan sebagai bagaian dari pengujian. Untuk mendapatkan nilai atau bobot yang dibutuhkan dalam memperkirakan tingkat keberhasilan riset, maka diperlukan tahapan-tahapan yang harus direncanakan dan dijalankan. Adapun tahapan-tahapan proses pengujian sistem dalam penelitian ini, akan dijabarkan sebagai berikut :

#### 4.1.1 Pemanggilan Dokumen Pemodelan proses bisnis dari Repository

Dokumen pemodelan proses bisnis yang digunakan sebagai data pada penelitian ini tersimpan di dalam *repository* pemodelan proses bisnis. *Repository* tersimpan pada sebuah DBMS (*Database Management System*) MySQL yang terdiri dari beberapa tabel yang diproses yaitu tabel *repo meta*, *index meta*, *vector meta*, *cache*, *stopword*, dan kata dasar. Untuk memanggil data dari *repository* tersebut dibutuhkan koneksi php dengan nama *file* *konnect.php*. *File* *konnect.php* akan di-*include*-kan pada setiap proses ketika menggunakan perintah *query database*, baik berupa fungsi *insert*, *update*, *select* dan *delete*. Semua fungsi menggunakan bahasa pemrograman PHP (*Hypertext Preprocessor*), karena pembuatan program berbasis website.

Di dalam *file* *konnect* terdapat atribut *conn* sebagai sebuah nama variabel yang akan dipanggil ketika menjalankan sebuah *query* pada setiap fungsi manajemen basis data. Terdapat beberapa *method* utama yang sering digunakan untuk memproses *database*, salah satu *method* tersebut yaitu *method new mysqli()*. *Method new mysqli()* digunakan untuk membuat sebuah koneksi. *Method mysqli\_query()* digunakan untuk memproses perintah pada *database* MySQL seperti *insert*, *select*, *update*, dan *delete*. *Method mysqli\_num\_rows()* digunakan untuk menghitung jumlah data yang dipanggil. *Method mysqli\_fetch\_array()* digunakan untuk menampilkan data yang sudah dipanggil dari basis data. Pada *file* *konnect* terdapat beberapa parameter untuk melakukan koneksi pada *database*, yaitu alamat *host*, nama *database*, *username* dan *password*.

### 4.1.2 Preprocessing

Tahapan pertama yang harus dilakukan yaitu *preprocessing* terhadap dokumen pemodelan proses bisnis. *Preprocessing* dokumen pemodelan proses bisnis dilakukan untuk mendapatkan indeks kata yang dapat digunakan sebagai pengganti dokumen. Proses *preprocessing* terdiri dari *case folding*, *tokenizing*, *filtering*, serta *stemming*.

#### a. Case Folding

*Case Folding* dilakukan untuk mengubah semua huruf dalam dokumen dari 'a' hingga 'z' menjadi huruf kecil (*lowercase*). Tujuan dari proses ini yaitu untuk menghilangkan *noise* pada saat pengambilan informasi. Sehingga akan memudahkan saat pengolahan data pada proses selanjutnya.

#### b. Tokenizing

*Tokenizing* dilakukan untuk memotong kalimat-kalimat menjadi kata-kata, serta menghilangkan delimiter seperti tanda koma(,), titik(.), spasi dan karakter lainnya.

#### c. Filtering

*Filtering* merupakan proses penghapusan *term* yang dianggap tidak memiliki makna atau kata yang tidak penting yang diperoleh dari hasil *tokenizing*.

Tahapan ini memiliki tujuan supaya tidak merusak nilai bobot yang ada pada tiap dokumen. Karena, yang diperhitungkan kata-kata yang terpenting.

Tahapan ini memiliki tujuan untuk mengurangi *volume* kata sehingga hanya kata-kata yang penting yang terdapat pada dokumen. *Stopword* dapat berupa kata depan, kata pengganti dan kata penghubung.

#### d. Stemming



*Stemming* dilakukan untuk mengembalikan kata-kata yang diperoleh dari hasil *filtering* menjadi kata dasar, menghilangkan imbuhan awal (*prefix*) dan imbuhan akhir (*sufliz*). Proses *stemming* bertujuan untuk mengurangi variasi kata yang sebenarnya memiliki kata dasar yang sama, sehingga dapat dikembangkan untuk mencari kata yang bersinonim.

Hasil dari seluruh proses *preprocessing* disimpan dalam database. Daftar *term* hasil dari proses *preprocessing* dapat dilihat pada tabel

Tabel 4. 1 Daftar term hasil preprocessing

id_index_meta	id_wfmeta	term_keyword
1	91	receive
2	91	rencana
3	91	anggar
4	91	transaksi
5	91	beranda
6	91	kode

#### 4.1.3 Pembobotan TF IDF

Pembobotan TF IDF merupakan tahapan yang dilakukan setelah *preprocessing*. Pada tahap ini dilakukan pembobotan pada setiap *term* yang telah disimpan dalam *database*. Pembobotan ini dilakukan dengan cara menghitung TF (*term frequency*), kemudian IDF (*Inverse Document Frequency*) dari masing-masing *term* pada seluruh dokumen yang ada. Setelah memperoleh nilai dari perhitungan TF dan IDF, maka selanjutnya dilakukan pembobotan TF IDF. Perhitungan bobot dilakukan menggunakan bahasa PHP dan hasil dari pembobotan disimpan pada *database* MySQL seperti yang ditunjukkan pada tabel berikut.

Tabel 4.2 Daftar nilai term frequency

id_index_meta	id_wfmeta	term_keyword	count_keyword	bobot_keyword
1	91	receive	9	9.26657
2	91	rencana	4	10.5562
3	91	anggar	7	18.4734
4	91	transaksi	8	21.1125
5	91	beranda	1	2.63906

#### 4.1.4 Implementasi Cosine Similarity

Pemetaan dan komposisi dihitung menggunakan *cosine similarity* antara kata kunci dengan seluruh dokumen *activity*, serta *activity* dengan seluruh dokumen *web service*. Perhitungan *cosine similarity* mengacu pada nilai pembobotan TF-IDF. Perhitungan nilai kemiripan dilakukan dengan cara membandingkan nilai kemiripan antara nilai *vector* kata kunci dengan *vector* tiap dokumen pada *repository activity*, serta nilai *vector activity* dengan *vector* tiap dokumen *web service*.

Kata-kata dalam dokumen *web service* direpresentasikan dalam *vector*, sehingga dapat dibandingkan dokumen *activity* dengan kata kunci serta dokumen *web service* dengan *activity* untuk menunjukkan kemiripan isi keduanya. Kata kunci yang paling mirip dengan isi pada *activity* dianggap paling relevan, begitu juga dengan *activity* dengan isi dokumen *web service*. Maka metode *cosine similarity* dipilih sebagai metode yang digunakan untuk mengukur kedekatan *vector activity* dengan *vector* kata kunci serta, *vector web service* dan *vector activity*.

Sebelum menghitung nilai *similarity* menggunakan *cosine similarity* antara *activity* dengan dokumen *web service*, dilakukan tahapan *preprocessing* terhadap

*activity* dan diberikan bobot tiap kata dalam *activity* tersebut, sama seperti tahapan *preprocessing* pada dokumen *web service*.

Hasil *preprocessing* dan perhitungan nilai *similarity* atau kemiripan disimpan pada tabel *tb\_cache\_meta*. Misalkan kata kunci yang dimasukkan oleh pengguna adalah : “Pendaftaran ulang santri”. Selanjutnya kalimat tersebut akan diproses melalui beberapa tahapan dalam *preprocessing* dan kata yang dihasilkan dari proses tersebut yaitu “daftar ulang santri” dengan masing-masing nilai kemiripan.

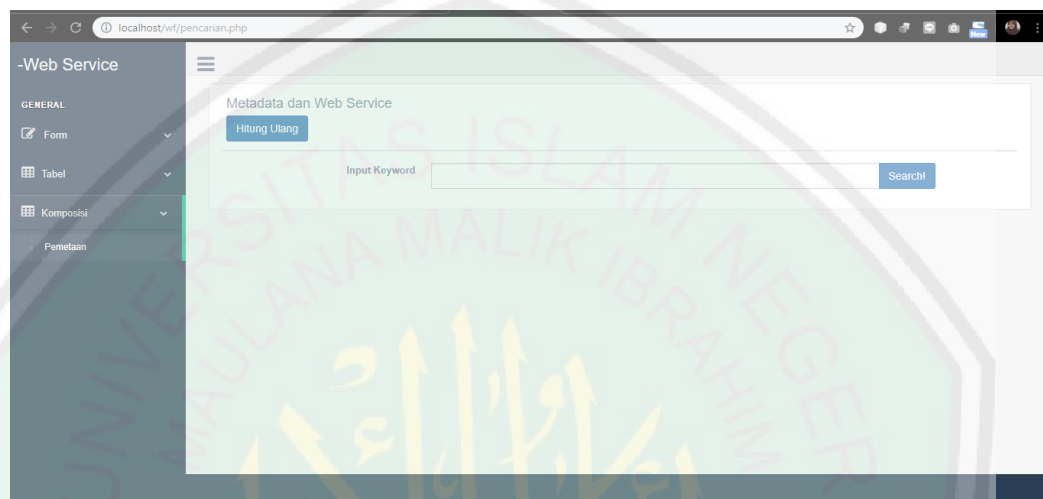
Berikut merupakan hasil perhitungan kemiripan antara kata kunci dengan dokumen *web service* yang tersimpan pada tabel *tb\_cache\_meta*. Kolom *query* merupakan hasil dari kata kunci yang telah melalui tahap *preprocessing*, *id* dokumen (*docid*) merupakan nomor dokumen yang memiliki nilai persamaan dengan kata kunci, *value* merupakan nilai dari persamaan antara kata kunci dengan *id* dokumen.

Tabel 4.3 Hasil *similarity* kata kunci dengan dokumen

query	docid	value
daftar ulang santri	95	0.045673648244946
daftar ulang santri	109	0.041116004734749
daftar ulang santri	114	0.16184511106476
daftar ulang santri	115	0.51997276751119
daftar ulang santri	116	0.045673648244946
daftar ulang santri	117	0.045673648244946
daftar ulang santri	118	0.59945252109372
daftar ulang santri	121	0.069889290635472

## 4.2 Langkah Uji Coba

Pada tahap ini dilakukan uji coba program menggunakan antarmuka (*interface*), untuk mengetahui tingkat keefektifan sistem. Tampilan awal program berisi halaman awal dengan menu pencarian. Tampilan awal program dapat dilihat pada Gambar



Gambar 4.1 Tampilan utama aplikasi *metadata* dan WSDL *webservice*

Skenario berawal dari proses pencarian data *activity* pada menu pencarian. Pada tahap ini dilakukan proses pencarian data *activity* yang proses pencariannya hanya berdasarkan data yang tersimpan pada *repository* yaitu mengenai penerimaan santri/siswa baru. Proses ini mengambil contoh kata kunci yang dimasukkan yaitu “Pendaftaran santri baru” yang digambarkan pada gambar. Kemudian memilih bobot *activity* paling besar yang selanjutnya dijadikan sebagai kata kunci untuk mencari *web service* yang cocok dengan *activity*.

Metadatan dan Web Service

Hitung Ulang

Input Keyword: pendaftaran santri baru Search Discovery Success

Discovery Result pendaftaran santri baru

Result metadata

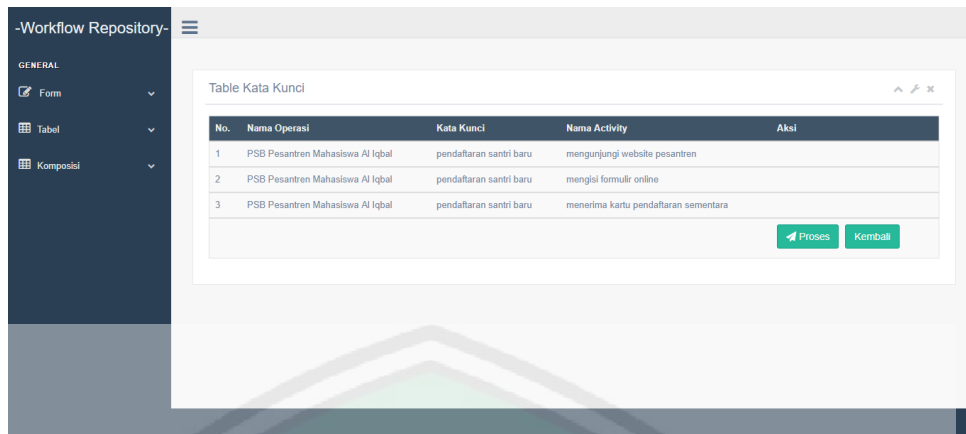
Hasil preprocessing meta data

No.	Nama Model Proses	Activity	Bobot	Aksi
1	penerimaan siswa baru MA al karimi	mengunjungi sekolah  melakukan pendaftaran diri  menerima siswa baru  memberikan formulir pendaftaran  memberikan rincian persyaratan  menerima formulir pendaftaran  menerima rincian persyaratan  mengisi form pendaftaran  menyerahkan form isi  menerima form pendaftaran  membuat kartu ujian  menyerahkan kartu ujian  menerima kartu ujian  melakukan ujian masuk  menerima data peserta ujian  melakukan ujian baca quran  melakukan ujian tulis  melakukan rekap ujian  membuat pengumuman ujian  melihat hasil ujian  dinyatakan lulus  melakukan pembayaran administrasi  menerima biaya administrasi  membuat kwitansi pembayaran  memberikan seragam  memberikan informasi orientasi  siswa dinyatakan tidak diterima	0.150582674862	Lihat
2	penerimaan santri baru pondok pesantren cokro kertopati	psb santri baru  mengunjungi pondok pesantren  menemui panitia PSB  menerima santri baru  memberikan rincian pendaftaran  menjelaskan peraturan pondok  menerima rincian pendaftaran  melakukan sowan dengan wali  menerima surat izin pengasuh  melakukan pembayaran  menerima uang pembayaran  mengantarkan ke kamar mukim	0.082577099138446	Lihat
3	Penerimaan santri baru pondok pesantren miftahu nurul huda	mengunjungi pondok pesantren  menemui panitia PSB  menerima santri baru  memberikan rincian persyaratan  memberikan rincian pembayaran  menerima persyaratan pendaftaran  menerima rincian pendaftaran  mengisi formulir pendaftaran  melakukan pembayaran  menerima form pendaftaran  menerima biaya administrasi  memberikan informasi ujian madin  menerima data peserta ujian  melakukan ujian baca quran  melakukan ujian baca kitab kuning  mengisi kelas madin berdasar hasil ujian  mulai aktivitas diniyah	0.032128138744038	Lihat
4	PSB Pondok Pesantren Tahfidz Quran Oemah Quran	penerimaan santri baru  mendaftarkan diri  menerima calon santri  memberikan rincian administrasi  membuat daftar ujian placement test  mengikuti placement test  menerima data peserta ujian  melakukan ujian baca quran  melakukan ujian baca kitab  membuat rekap hasil ujian  melihat hasil ujian  menerima hasil ujian  melakukan daftar ulang  menunjukkan kamar mukim	0.028355724440788	Lihat
5	penerimaan santri baru pondok pesantren NU tuluagung	mengunjungi pondok pesantren  mendaftarkan diri  menerima santri baru  memberikan form pendaftaran  memberikan surat pernyataan  memberikan rincian pendaftaran  menerima form pendaftaran  menerima surat pernyataan  menerima rincian pendaftaran  mengisi form pendaftaran  mengisi surat pernyataan  melakukan pembayaran administrasi  menerima biaya pendaftaran  mengantarkan ke kamar mukim	0.032508234840794	Lihat
6	PSB Pesantren Mahasiswa Al Iqbal	mengunjungi website pesantren  mengisi formulir online  menerima kartu pendaftaran sementara  mencetak kartu pendaftaran  membawa kartu pendaftaran ke asrama  menerima calon santri baru  menerima bukti pendaftaran  melakukan validasi kartu pendaftaran  melakukan validasi pendftar pada website  data ditemukan ?  memberikan rincian biaya pembayaran  menerima rincian pembayaran  memberikan arahan pendaftaran online  menerima biaya pendaftaran  membuat kwitansi pembayaran  mengantarkan ke kamar asrama	0.3739947322214	Lihat

Gambar 4. 2 Hasil pencarian

Pada tahap berikutnya yaitu proses pencarian *web service*, yaitu proses yang menggunakan kata kunci berupa *activity*.





The screenshot shows a web interface titled '-Workflow Repository-'. On the left, there is a sidebar with 'GENERAL' and three menu items: 'Form', 'Tabel', and 'Komposisi'. The main area displays a table titled 'Table Kata Kunci' with the following data:

No.	Nama Operasi	Kata Kunci	Nama Activity	Aksi
1	PSB Pesantren Mahasiswa Al Iqbal	pendaftaran santri baru	mengunjungi website pesantren	
2	PSB Pesantren Mahasiswa Al Iqbal	pendaftaran santri baru	mengisi formulir online	
3	PSB Pesantren Mahasiswa Al Iqbal	pendaftaran santri baru	menerima kartu pendaftaran sementara	

At the bottom right of the table, there are two buttons: 'Proses' and 'Kembali'.

Gambar 4.3 Tampilan pencarian *web service*

Halaman ini merupakan hasil dari pencarian *web service* berdasarkan kata kunci dari *activity*, yaitu menampilkan *web service* yang memiliki persamaan dengan *activity*.



The screenshot shows a detailed search result page with a 'Kembali' button at the top left. It contains three sections of discovery results:

1. **Discovery Result mengunjungi website pesantren**: This section shows 'Tidak ada data' (No data).

2. **Discovery Result mengisi formulir online**: This section shows a table with one result:

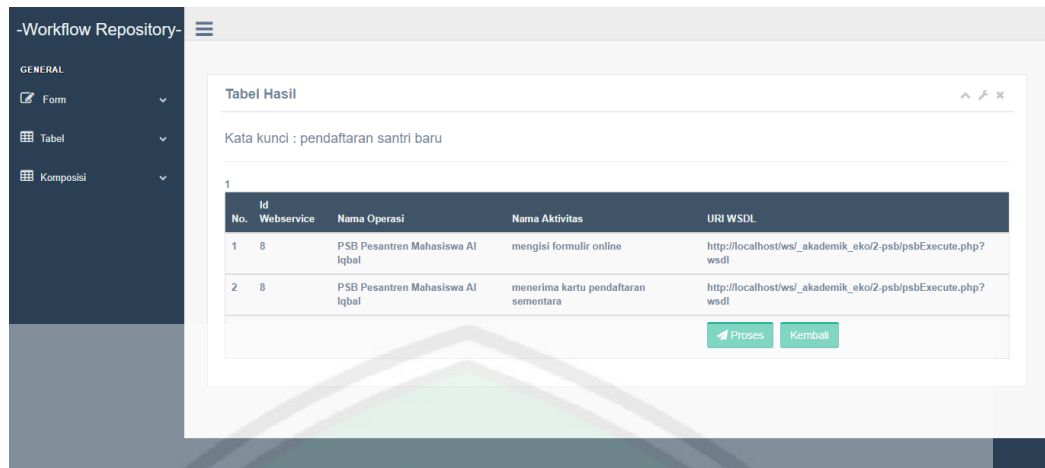
No	Nama Operasi	Input	Output	Meta	URL WSDL	Bobot	Aksi
1	psbExecute.updateagama	tns:psbExecute.updateagamaRequest	tns:psbExecute.updateagamaResponse	psb agama	Lihat	2.3754805948495	+Tambah

3. **Discovery Result menerima kartu pendaftaran sementara**: This section shows a table with one result:

No	Nama Operasi	Input	Output	Meta	URL WSDL	Bobot	Aksi
1	psbExecute.addsuku	tns:psbExecute.addsukuRequest	tns:psbExecute.addsukuResponse	psb suku	Lihat	1.7820753783757	+Tambah

Gambar 4.4 Hasil pencarian *webservice*

Halaman ini merupakan hasil dari pencarian detail yang menampilkan kata kunci yang digunakan, nama operasi dari proses bisnis, *activity*, serta url dari *web service* seperti pada gambar 4.5



-Workflow Repository-

GENERAL

- Form
- Tabel
- Komposisi

Tabel Hasil

Kata kunci : pendaftaran santri baru

1


No.	ID Webservice	Nama Operasi	Nama Aktivitas	URI WSDL
1	8	PSB Pesantren Mahasiswa Al Iqbal	mengisi formulir online	http://localhost/ws/_akademik_eko/2-psb/psbExecute.php?wsdl
2	8	PSB Pesantren Mahasiswa Al Iqbal	menerima kartu pendaftaran sementara	http://localhost/ws/_akademik_eko/2-psb/psbExecute.php?wsdl

Proses Kembali

Gambar 4.5 Tampilan hasil *webservice*

Data yang telah dihasilkan tersebut kemudian di-*generate* menjadi BPEL.

Berikut merupakan hasil dari *generate* BPEL.



-Workflow Repository-

GENERAL

- Form
- Tabel
- Komposisi

Table Kata Kunci

Hasil Komposisi	Lihat	Aksi
BPEL	View	Download

Gambar 4.6 Tampilan Info Hasil *Generate BPEL*

```

<?xml version="1.0"?>
<BPELSuitcase>
  <BPELProcess/>
  <partnerLinkBindings>
    <id_komposisi>1</id_komposisi>
    <id_webservice>8</id_webservice>
    <nama_activity> mengisi formulir online</nama_activity>
    <nama_operasi>PSB Pesantren Mahasiswa Al
Iqbal</nama_operasi>
    <url>http://localhost/ws/_akademik_eko/2-
psb/psbExecute.php?wsdl</url>
  </partnerLinkBindings>
  <BPELProcess/>
  <partnerLinkBindings>
    <id_komposisi>2</id_komposisi>
    <id_webservice>8</id_webservice>
    <nama_activity> menerima kartu pendaftaran
sementara</nama_activity>
    <nama_operasi>PSB Pesantren Mahasiswa Al
Iqbal</nama_operasi>
    <url>http://localhost/ws/_akademik_eko/2-
psb/psbExecute.php?wsdl</url>
  </partnerLinkBindings>
</BPELSuitcase>

```

Gambar 4. 7 Dokumen hasil *generate* BPEL

### 4.3 Pengujian BPEL

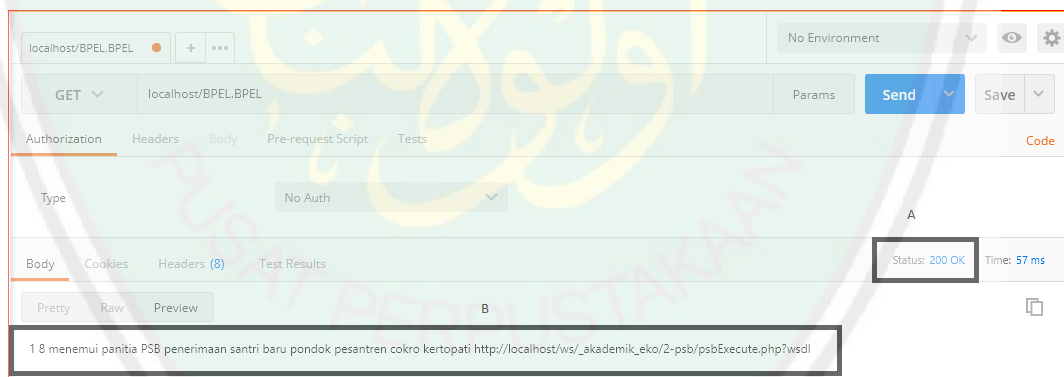
Pengujian BPEL dengan menggunakan *postman*. Pengujian ini dilakukan untuk menunjukkan dimana hasil BPEL tersebut dapat berjalan dengan baik dan

dapat digunakan pada aplikasi lainnya. Pada aplikasi *postman* data menunjukkan *response* berhasil yang didapat saat melakukan *request*. Pada Gambar 4.8 menunjukkan hasil yang di dapat dari pengujian menggunakan *postman*.



Gambar 4. 8 Hasil request file BPEL

Pada Gambar 4.9 menunjukkan bahwa *postman* melakukan request ke file BPEL untuk mendapatkan detail data dari BPEL.



Gambar 4. 9 Detail data dari BPEL

Pada Gambar 4.9 teks A menunjukkan bahwa *response* yang diberikan oleh *postman* bernilai 200 yang berarti data berhasil dikirim. Sedangkan pada teks B menunjukkan hasil dari *file* BPEL yang berada pada Gambar 4.8. Berikut merupakan rincian hasil pengujian :

HTTP request :

- HTTP method : GET
- URI : localhost/BPEL.BPEL

HTTP response :

- Response Code : 200
- Response Body : data list berupa *id webservice*, nama operasi, nama aktivitas, URI WSDL

#### 4.4 Pengujian Skenario

Berdasarkan skenario uji coba pada sistem komposisi *web service* ini dilakukan untuk mengetahui hasil *output* dari komposisi *web service* yang menggunakan metode TF-IDF dan *cosine similarity*. Uji coba sistem ini menggunakan beberapa pengujian skenario. Berikut merupakan skenario yang digunakan dalam pengujian sistem :

Tabel 4. 4 Skenario pengujian aplikasi

No	Kata Kunci	Output
1	Pendaftaran santri baru	Activity PSB Operasi PSB Url Web service PSB
2	Pendaftaran pondok pesantren modern	Activity PSB Operasi PSB Url Web service PSB
3	Penerimaan calon peserta baru	Activity PSB Operasi PSB Url Web service PSB

Berikut merupakan hasil dari pengujian masing-masing skenario :



## 1. Pendaftaran santri baru

Pada skenario pengujian diperoleh hasil sebagai berikut :

Activity	Mengunjungi pondok pesantren Menemui panitia PSB Menerima santri baru
Nama operasi	Penerimaan santri baru pondok pesantren cokro kertopati
Url web service	<a href="http://localhost/ws/_akademik_eko/2-psb/tambahAgama.php">http://localhost/ws/_akademik_eko/2-psb/tambahAgama.php</a> <a href="http://localhost/ws/_akademik_eko/2-psb/hapusKondisi.php">http://localhost/ws/_akademik_eko/2-psb/hapusKondisi.php</a> <a href="http://localhost/ws/_akademik_eko/2-psb/hapusSuku.ph">http://localhost/ws/_akademik_eko/2-psb/hapusSuku.ph</a> <a href="http://localhost/ws/_akademik_eko/2-psb/hapusAgama.php">http://localhost/ws/_akademik_eko/2-psb/hapusAgama.php</a>

Tabel 4. 5 Hasil nilai persamaan

No	Activity	Nama Operasi	URL	Similarity
1	Mengunjungi pondok pesantren	Penerimaan santri baru pondok pesantren cokro kertopati	-	-
2	Menemui panitia PSB	Penerimaan santri baru pondok pesantren cokro kertopati	<a href="http://localhost/ws/_akademik_eko/2-psb/tambahAgama.php">http://localhost/ws/_akademik_eko/2-psb/tambahAgama.php</a>	1
			<a href="http://localhost/ws/_akademik_eko/2-psb/hapusKondisi.php">http://localhost/ws/_akademik_eko/2-psb/hapusKondisi.php</a>	1
			<a href="http://localhost/ws/_akademik_eko/2-psb/hapusAgama.php">http://localhost/ws/_akademik_eko/2-psb/hapusAgama.php</a>	1

No	Activity	Nama Operasi	URL	Similarity
			_akademik_eko/2- psb/hapusSuku.php	
			http://localhost/ws/ _akademik_eko/2- psb/hapusAgama.php	1
3	Menerima santri baru	Penerimaan santri baru pondok pesantren cokro kertopati	-	-

## 2. Pendaftaran pondok pesantren modern

Pada skenario pengujian diperoleh hasil sebagai berikut :

Activity	Mengunjungi website pesantren Mengisi formulir pendaftaran Menerima kartu pendaftaran sementara
Nama operasi	PSB pesantren mahasiswa Al Iqbal
Url web service	http://localhost/ws/_akademik_eko/2- psb/tambahSuku.php http://localhost/ws/_akademik_eko/2- psb/tambahSuku.php

Tabel 4. 6 Hasil nilai persamaan

No	Activity	Nama operasi	URL	Similarity
1	Mengunjungi website pesantren	PSB pesantren mahasiswa Al Iqbal	-	-

No	Activity	Nama operasi	URL	Similarity
2	Mengisi formulir pendaftaran	PSB pesantren mahasiswa Al Iqbal	http://localhost/ws/_akademik_eko/2- psb/tambahSuku.php	2.37548059 48495
3	Menerima kartu pendaftaran sementara	PSB pesantren mahasiswa Al Iqbal	http://localhost/ws/_akademik_eko/2- psb/tambahSuku.php	1.78207537 83757

### 3. Penerimaan calon peserta baru

Pada skenario pengujian diperoleh hasil sebagai berikut :

Activity	Mendaftarkan diri
	Menerima calon santri
	Memberikan rincian administrasi
Nama operasi	PSB pondok pesantren tahfidz Oemah Quran
Url web service	Bizagi process modeler

Tabel 4. 7 Hasil nilai persamaan

No	Activity	Nama Operasi	Url	Similarity
1	Mendaftarkan diri	PSB pondok pesantren tahfidz Oemah Quran	http://localhost/ws/_akademik_eko/2- psb/psbExecute.php?w sdl	1.78207537 83757
2	Menerima calon santri	PSB pondok pesantren tahfidz Oemah Quran	-	-
3	Memberikan rincian administrasi	PSB pondok pesantren tahfidz Oemah Quran	-	-

#### 4.5 Integrasi Komposisi *Web Service* dengan Islam

Dalam agama islam, konsep untuk mencari ilmu pengetahuan (wawasan) telah ada pada Al-Qur'an. Kitab Al-Qur'an bukan merupakan kitab yang berisikan ilmu pengetahuan hasil dari penelitian manusia, namun kitab Al-Qur'an merupakan wahyu Allah yang terakhir yang diturunkan sebagai tuntunan hidup manusia. Akan tetapi kitab Al-Qur'an dapat dikaji kebenarannya setelah manusia membuktikan isi Al-Qur'an secara ilmiah, Seperti ayat Al-Qur'an surat Yunus : 101 ;

قُلْ انظُرُوا مَاذَا فِي السَّمَاوَاتِ وَالْأَرْضِ وَمَا تُعْجِبُ الْآيَاتُ  
وَالنُّذُرِ عَنْ قَوْمٍ لَا يُؤْمِنُونَ

Artinya : Katakanlah, “Perhatikanlah apa yang ada di langit dan di bumi!”

Tidaklah bermanfaat tanda-tanda (kebesaran Allah) dan rasul-rasul yang memberi peringatan bagi orang yang tidak beriman.” (QS. Yunus : 101)

Dalam surat Yunus pada ayat ke 101, Allah SWT memerintahkan kepada manusia untuk memperhatikan fenomena alam yang ada di langit dan di bumi yang merupakan salah satu tanda kebesaran Allah. Fenomena itu tidak hanya dilihat, akan tetapi juga harus dikaji dan diteliti serta dipelajari untuk dikembangkan menjadi ilmu pengetahuan dan teknologi yang bermanfaat bagi kehidupan manusia.

Dalam Tafsir Ibnu Katsir menjelaskan bahwa Allah memberikan petunjuk kepada hamba-hamba-Nya untuk merenungkan tanda-tanda kekuasaan-Nya dan semua makhluk yang diciptakan Allah di langit dan di bumi, yang semua itu

mengandung tanda-tanda yang jelas yang menunjukkan akan kekuasaan Allah Yang Mahabesar bagi orang-orang yang berakal.

Makhluk Allah yang ada di langit antara lain bintang-bintang yang bersinar terang –ada yang tetap dan ada yang beredar-, juga matahari serta rembulan, adanya siang dan malam yang keduanya silih berganti. Salah satunya masuk kepada yang lain hingga menjadi panjang waktunya, sedangkan yang lainnya menjadi pendek waktunya; demikian pula sebaliknya.

Langit yang tinggi dan luas serta keindahannya dan semua hiasan yang ada padanya adalah makhluk Allah pula. Allah menurunkan hujan dari langit, dengan hujan itu Allah menghidupkan bumi sesudah matinya; dan dikeluarkan-Nya dari bumi berbagai macam tumbuh-tumbuhan, pohon-pohonan yang menghasilkan biji-bijian dan buah-buahan serta bunga-bunga yang beraneka ragam bentuk, warna dan kegunaannya. Di bumi terdapat gunung-gunung yang menjulang tinggi, dataran-dataran yang luas menghampar, padang-padang sahara, hutan belantara, dan daerah-daerah yang layak untuk dihuni.

Begitu pula di laut dengan ombaknya, di dalamnya terkandung banyak hal yang menakjubkan. Sekalipun demikian, laut ditundukkan oleh Allah dan dimudahkan sehingga dapat ditempuh oleh bahtera. Laut membawa kapal-kapal dan perahu-perahu berlayar dengan lembutnya berkat kekuasaan Tuhan Yang Mahakuasa yang telah menundukkannya. Tidak ada Tuhan Selain Allah, dan tidak ada Rabb selain Dia (Al-Sheikh, 1994).

Al-Qur'an surat Yunus ayat 101 memiliki gambaran yang sama dengan penelitian ini, yaitu untuk mengkaji dan meneliti serta dipelajari untuk dikembangkan menjadi ilmu pengetahuan dan teknologi (iptek) yang memiliki



manfaat bagi kehidupan manusia. Penelitian ini dilakukan untuk mendapatkan komposisi/ susunan *web service*. Tahapan-tahapan yang dilakukan, mulai dari proses *parsing file* BPMN menjadi XPDL yang digunakan sebagai data *input-an*. Pada tahap *preprocessing* terdapat rangkaian tahapan seperti *case folding*, *tokenizing*, *stopward*, serta *stemming*. Kemudian dilakukan proses TF-IDF untuk mendapatkan nilai bobot tiap kata pada dokumen, selanjutnya dilakukan proses perhitungan *similarity* menggunakan metode *cosine similarity*. Hasil komposisi *web service* didapatkan pengguna dengan cara memasukkan kata kunci. Komposisi yang telah didapatkan kemudian diuji oleh peneliti. Hasil dari komposisi tersebut telah dapat menyusun BPEL secara otomatis, sehingga hasil akhir BPEL dapat dimanfaatkan sebagai acuan dalam eksekusi pemrograman.

## **BAB V**

### **PENUTUP**

Pada penelitian ini bab ke-lima merupakan bab penutup yang membahas kesimpulan serta saran dari perolehan penelitian yang dapat digunakan sebagai pengembangan dalam penelitian selanjutnya.

#### **5.1 Kesimpulan**

Setelah melakukan kajian pustaka, perancangan sistem , implementasi serta pengujian dapat disimpulkan bahwa :

1. Berdasarkan hasil pemetaan pemodelan proses bisnis dengan *web service*, melalui proses parsing BPMN, *preprocessing*, TF-IDF hingga *similarirty* didapatkan hasil kecocokan antara pemodelan proses bisnis dengan *web service*. Kecocokan anantara pemodelan proses bisnis dengan *web service* tersebut berdasarkan nilai persamaana tertinggi yang diambil dari tiap *activity* sehingga dapat di petakan dengan *web service* yang sesuai.
2. Dalam melakukan penyusunan atau komposisi *web service* yang menggunakan algoritma *cosine similarirty*, sistem telah dapat dijalankan dengan baik serta dapat melakukan komposisi *web service* secara otomatis. Dari tiga kali percobaan komposisi *web service* menggunakan algoritma *cosine similarity*, menunjukkan nilai persamaan tertinggi yaitu sebesar 1; 1,782; 2,3754. Sehingga nilai persamaan tertinggi tersebut dapat diambil sebagai patokan dalam komposisi *web service*. Hasil dari komposisi *web service* tersebut dapat secara otomatis disusun menjadi BPEL. Sehingga hasil akhir BPEL dapat dimanfaatkan sebagai acuan eksekusi dalam pemrograman.

## 5.2 Saran

Untuk pengembangan selanjutnya, maka saran yang dapat penulis berikan antara lain :

1. Penambahan data pemodelan proses bisnis, dikarenakan pada penelitian ini hanya mencakup pemodelan proses bisnis bagian akademik.
2. Menggunakan algoritma yang lain untuk melakukan proses *similarity* seperti algoritma *Jaccard* dan lain-lain.



## DAFTAR PUSTAKA

- Al-Sheikh, A. b. (1994). *Lubaabut Tafsir Min Ibni Katsir*. Jakarta: Pustaka Imam asy-Syafi'i.
- Astrininditiya, R. (2017). Workflow Repository pada ERP Pondok Pesantren Menggunakan Ontologi untuk meningkatkan Reusability. *Konferensi Nasional Sistem Informas* (hal. 304-309). Pangkalpinang: Underfraduated Thesis.
- D, R. M. (2013). Pembangunan Aplikasi Pencarian Dokumen Menggunakan Text Mining Berbasis Web. Dalam R. M. D. Yogyakarta: UNIVERSITAS ATMA JAYA YOGYAKARTA.
- David Booth, H. H. (2004, February 11). <https://www.w3.org/TR/ws-arch/wsa.pdf>. Dipetik February 21, 2018, dari <https://www.w3.org/>: <https://www.w3.org/TR/ws-arch/wsa.pdf>
- Dewi, R. M. (2018, May). *Web Service Discovery Menggunakan Cosine Similarity Untuk Meningkatkan Akurasi Query Pada Web Service*. Malang, Indonesia: Universitas Islam Negeri Maulana Malik Ibrahim.
- Dwi Rahmawati, R. I. (2017). Analisis dan Pemodelan Proses Bisnis Bidang Pelayanan Perizinan Menggunakan Business Process Model and Nation (BPMN). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 1387-1347.
- Katarina , G., & et al. (2013). Integration of business process modeling and Web Services : a survey. *Springer*, DOI 10.1007/s11761-013-0138-2.
- Larasati, R. (2015). Klasifikasi Teks Dengan Menggunakan Algoritma K-Nearest Neighbor Pada Dokumen Tugas Akhir. Dalam R. Larasati, *Klasifikasi Teks Dengan Menggunakan Algoritma K-Nearest Neighbor Pada Dokumen Tugas Akhir*. BANDUNG: UNIVERSITAS WIDYATAMA BANDUNG.

- Prasetya, A. W. (2017). Aplikasi Manajemen Proses Bisnis Menggunakan Metode Analisis Kemiripan Behavioral, Structural, dan Semantic untuk Meningkatkan Akurasi dalam Penentuan Common Fragment Workflow Pada ERP Pesantren. *Konferensi Nasional Sistem Informas* (hal. 484-489). Pangkalpinang: Universitas Islam Negeri Maulana Malik Ibrahim Malang.
- Putu Wuri Handayani, S. B. (2009). Implementasi System Pengelolaan Proses Bisnis Mengacu Pada MIT Process Handbook. *Journal of Information Systems*, 106-113.
- Quan Z. Sheng, X. Q. (2014). Web service composition : A decade's overview. *Elsivier*, 218-238.
- Ramadhani, M. A. (2015). Pemodelan Proses Bisnis Sistem Akademik Menggunakan Pendekatan Business Process Modelling Notation (BPMN) (Studi Kasus Institusi Perguruan Tinggi XZY). *Jurnal Informasi Volume VII No.2 / November / 2015*, 83.
- Sari, S. K., & Asniar. (2015). Analisis dan Pemodelan Proses Bisnis Prosedur Pelaksanaan Proyek Akhir Sebagai Alat Bantu Identifikasi Kebutuhan Sistem. *Jurnal Infotel Vol.7 No. 2*, 143-152.
- Sarno, R., Fauzan, A. C., Nurlaili, A. N., Rahmawati, D., Sungkono, K. R., & Effendi, Y. A. (2017). *Manajemen Proses Bisnis Model dan Simulasi*. Surabaya: ITS Tecno Sains.
- Satrio, B. (2008). Business Process Reporting Service Subsystem Sms Based Service. Dalam B. Satrio, *Business Process Reporting Service Subsystem Sms Based Service*. Bandung: Institut Teknologi Bandung.
- Triana, A., Saptono, R., & Sulistyono, M. E. (2014). Pemanfaatan Metode Vector Space Model dan Metode Cosine Similarity pada Fitur Deteksi Hama dan Penyakit Tanaman Padi. *Jurnal Teknologi dan Infromasi*, 3, No 2.
- Utomo, W. H. (2012). *Pemrograman Aplikasi SOA*. Salatiga: Fakultas Teknologi Informasi UKSW Press.



Widodo, A. P. (2016). Kinerja Arsitektur Interoperabilitas E-Government Multi Platform. *Jurnal Matematika*, 16-28.

Wulan, F. R. (2017). Dekomposisi Web Service Untuk Meningkatkan Reusabilitas. *Konferensi Nasional Sistem Informas*, (hal. 1-6). Pangkalpinang.

