TESIS – TE142599

TEMU KEMBALI INFORMASI BERBASIS PEMODELAN TOPIK MENGGUNAKAN KOMBINASI LSI DAN VSM PADA SISTEM TANYA-JAWAB

SYAMSUL BAHRI
NRP 07111650067002

DOSEN PEMBIMBING
Dr. Surya Sumpeno, S.T., M.Sc.
Dr. Supeno Mardi Susiki Nugroho, S.T., M.T.

PROGRAM MAGISTER
BIDANG KEAHLIAN TELEMATIKA – PENGELOLA TIK PEMERINTAHAN
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI ELEKTRO
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2018

TESIS – TE142599

TEMU KEMBALI INFORMASI BERBASIS PEMODELAN TOPIK MENGGUNAKAN KOMBINASI LSI DAN VSM PADA SISTEM TANYA-JAWAB

SYAMSUL BAHRI
NRP 07111650067002

DOSEN PEMBIMBING
Dr. Surya Sumpeno, S.T., M.Sc.
Dr. Supeno Mardi Susiki Nugroho, S.T., M.T.

PROGRAM MAGISTER
BIDANG KEAHLIAN TELEMATIKA – PENGELOLA TIK PEMERINTAHAN
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI ELEKTRO
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2018

LEMBAR PENGESAHAN

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Teknik (M.T)
di
Institut Teknologi Sepuluh Nopember

oleh:

Syamsul Bahri
NRP. 07111650067002

Tanggal Ujian : 6 Juli 2018
Periode Wisuda : September 2018

Disetujui oleh:

1. Dr. Surya Sumpeno, S.T., M.Sc. (Pembimbing I)
   NIP. 196906131997021003

2. Dr. Supeno Mardi S. Nugroho, S.T., M.T. (Pembimbing II)
   NIP. 197003131995121001

3. Dr. I Ketut Eddy Purnama, ST., MT. (Penguji)
   NIP. 196907301995121001

4. Istas Pratomo, ST. MT. PhD (Penguji)
   NIP. 197903252003121001

Dekan Fakultas Teknologi Elektro

Dr. Tri Arief Sardjono, S.T., M.T.
NIP. 197002121995121001

Halaman ini sengaja dikosongkan

PERNYATAAN KEASLIAN TESIS

Dengan ini saya menyatakan bahwa isi keseluruhan Tesis saya dengan judul "**TEMU KEMBALI INFORMASI BERBASIS PEMODELAN TOPIK MENGGUNAKAN KOMBINASI LSI DAN VSM PADA SISTEM TANYA-JAWAB**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 25 Mei 2018

Syamsul Bahri
NRP. 07111650067002

Halaman ini sengaja dikosongkan

TEMU KEMBALI INFORMASI BERBASIS PEMODELAN TOPIK MENGGUNAKAN KOMBINASI LSI DAN VSM PADA SISTEM TANYA-JAWAB

Nama mahasiswa : Syamsul Bahri
NRP : 07111650067002
Pembimbing : 1. Dr. Surya Sumpeno, S.T., M.Sc.
2. Dr. Supeno Mardi Susiki Nugroho, S.T., M.T.

ABSTRAK

Dalam Penerapan *e-government* untuk menuju tata pemerintahan yang baik (*good governance*), pemerintah pusat maupun daerah menyediakan layanan tanya-jawab pada sistem *online*. Layanan tanya-jawab ini sangat penting karena dapat memfasilitasi permintaan informasi secara lebih mudah serta dapat diakses kapan saja, tanpa harus menunggu jam layanan kantor buka. Dalam pelaksanaan layanan tersebut masih dilakukan secara manual, sehingga perlu dikembangkan suatu sistem tanya-jawab yang dikerjakan oleh komputer. Suatu sistem tanya-jawab dibentuk oleh beberapa elemen/modul. Salah satu elemen penting dalam sistem tanya-jawab tersebut elemen temu kembali informasi yang bertanggung jawab dalam pengambilan dokumen-dokumen yang relevan dengan pertanyaan (*query*) pengguna. Metode yang banyak digunakan dalam membangun temu kembali informasi adalah menggunakan adalah metode *Vector Space Model* (VSM) dan *Latent Semantic Indexing* (LSI), dimana keduanya merepresentasikan dokumen ke dalam vektor ruang. Namun kedua metode tersebut memiliki keterbatasan masing-masing. Untuk itu dalam penelitian ini diusulkan model kombinasi antara metode VSM dan LSI untuk memperbaiki beberapa batasan pada keduanya. Dalam mencari dokumen yang relevan dengan *query,* model kombinasi ini bekerja dengan cara mengambil terlebih dahulu dokumen yang memiliki kesamaan topik dengan *query* menggunakan pemodelan topik dalam hal ini metode LSI. Kemudian setelah itu mengurutkannya berdasarkan kesamaan *term* menggunakan metode VSM untuk diambil beberapa dokumen dengan nilai kemiripan tertinggi. Untuk menguji kinerja dari model kombinasi tersebut dalam mencari dokumen relevan pada sistem tanya-jawab, maka pada penelitian ini akan menggunakan data layanan tanya-jawab pada sistem Pengadaan Secara Elektronik (SPSE) sebagai data eksperimen. Dari hasil eksperimen yang dilakukan ditemukan bahwa model yang diusulkan mampu meningkatkan presisi metode dasarnya yakni LSI dan VSM yang berdiri sendiri. Model kombinasi (LSI+VSM) memperoleh *precision at 1* (P@1)=**0,7** dengan *Mean Average Precision* (MAP)=**0,579** sedangkan pada model dasarnya diperoleh P@1=**0,5** dengan MAP=**0,237** untuk LSI, P@1=**0,38** dengan MAP=**0,247** untuk VSM biasa serta P@1=**0,44** dengan MAP=**0,258** untuk VSM dengan pembobotan profesional (VSM+PP).

Kata kunci: LSI, pemodelan topik, sistem tanya-jawab, temu kembali informasi, VSM.

Halaman ini sengaja dikosongkan

INFORMATION RETRIEVAL BASED ON TOPIC-MODELING USING COMBINATION OF LSI AND VSM IN QUESTION-ANSWERING SYSTEM

By : Syamsul Bahri
Student Identity Number : 07111650067002
Supervisor(s) : 1. Dr. Surya Sumpeno, S.T., M.Sc.
2. Dr. Supeno Mardi S. Nugroho, S.T., M.T.

ABSTRACT

In order to achieve good governance through implementation of e-government, the central and local governments provide a question-answering services for online system. This question-answering services are essential to facilitate information requests to make it easier and accessible at any time. In the implementation of the services are still done manually, so it is necessary to develop a computerized question-answering system (QAS). A QAS is formed by several elements/modules. One of important element in QAS is the information retrieval (IR) that is responsible for retrieving relevant documents to the user requests. A widely used methods for developing the information retrieval system are using Vector Space Model (VSM) and Latent Semantic Indexing (LSI), where they represent documents into space vectors. However, both models have their respective limitation. For this reason, in this research proposed a combination model between VSM and LSI to fix some limitations on both. In searching for documents relevant to the query, this combination model works by retrieving documents that have the same topic as the query first using the topic modeling in this case the LSI method and then sort it based on the term similarity using the VSM method to retrieve some documents with the highest similarity value. To evaluate the performance of that combination model in searching relevant documents on the question-answering system, hence in this research will be use question-answer data on the Electronic Procurement System (SPSE) as experimental data. From the experimental results, it was found that the proposed model was able to improve the precision of its basic method i.e. the stand-alone LSI and VSM. The combination model (LSI + VSM) obtained precision at 1 (P@1)=**0.7** with Mean Average Precision (MAP)=**0.579** whereas in the basic methods obtained P@1=**0.5** with MAP=**0.237** for the LSI, P@1=**0.38** with MAP=**0.247** for the traditional VSM and P@1=**0.44** with MAP=**0.258** for the VSM with professional weight concept.

Key words: information retrieval, LSI, QAS, topic modeling, VSM

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Bismillah, segala puji kita panjatkan kepada Allah, Robb semesta alam, atas berbagai macam nikmat yang telah dianugerahkan kepada kita. Dan berkat kehendak serta rahmat-Nya penulis akhirnya dapat menyelesaikan tesis ini. Sholawat dan salam semoga senantiasa terlimpahkan kepada Nabi Muhammad ﷺ, keluarganya, para sahabatnya, serta orang-orang yang selalu istiqomah dalam memegang teguh ajarannya yang murni hingga akhir zaman. Ucapan terima kasih yang tak terhingga kepada ibu, mertua, istriku Ade Ayu Warni, anak-anakku (Syaima, Yasmin, Shofiyya, Athiya), dan semua saudaraku yang tidak pernah lelah mendoakan serta memberikan dukungan moril maupun materiil demi selesainya tesis ini.

Ucapan terima kasih dan penghargaan penulis sampaikan kepada Dr. Surya Sumpeno, S.T., M.Sc. selaku pembimbing pertama dan Dr. Supeno Mardi Susiki Nugroho, S.T., M.T. selaku pembimbing kedua, yang dengan penuh kesabaran selalu meluangkan waktu, memberikan pengarahan dan motivasi serta semangat dalam penulisan tesis ini.

Penyelesaian tesis ini tidak terlepas dari bantuan, kerjasama dan dukungan dari berbagai pihak. Untuk itu penulis menyampaikan terima kasih kepada:

1. Kementerian Komunikasi dan Informasi Republik Indonesia yang telah memberikan kesempatan dan beasiswa Program Magister (S2) Bidang Keahlian Telematika Konsentrasi Pengelola TIK Pemerintahan (PeTIK) pada Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember Surabaya;
2. Prof. Ir. Joni Hermana, M.Sc.Es., Ph.D. selaku Rektor Institut Teknologi Sepuluh Nopember Surabaya;
3. Dr. Tri Arief Sardjono, S.T., M.T. selaku Dekan Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember Surabaya;
4. Dr. Eng. Ardyono Priyadi, S.T., M.Eng. selaku Kepala Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember Surabaya;

5. Dr. Ir. Wirawan, DEA selaku Kepala Program Studi Pascasarjana Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember Surabaya;
6. Dr. Adhi Dharma Wibawa, S.T., M.T. selaku Koordinator Bidang Keahlian Telematika/Pengelola TIK Pemerintahan (PeTIK) pada Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember Surabaya;
7. Eko Setijadi, S.T., M.T., Ph.D. selaku Dosen Wali Akademik untuk mahasiswa Pengelola TIK Pemerintahan (PeTIK) 2016;
8. Seluruh Dosen dan staf Program Studi Magister (S2) Departemen Teknik Elektro, khususnya Bidang Keahlian Telematika/ Pengelola TIK Pemerintahan (PeTIK), atas jasa dan pengabdiannya dalam mendidik, membimbing dan mendewasakan kami;
9. Rekan-Rekan PeTIK 2016 serta Telematika 2016 atas kekompakan selama ini dan saling mendukung, membantu serta mendoakan selama perkuliahan;
10. Serta semua pihak yang tidak dapat penulis sebutkan satu persatu.

Semoga Allah SWT membalas kebaikan semua pihak yang telah memberi kesempatan, dukungan, doa dan bantuan dalam menyelesaikan tesis ini. Penulis menyadari bahwa tesis ini masih jauh dari sempurna, oleh karena itu saran dan kritik yang membangun sangat diharapkan untuk perbaikan dimasa mendatang. Semoga tesis ini memberikan manfaat yang baik bagi para pembacanya.

Surabaya, 25 Mei 2018

Penulis

DAFTAR ISI

DAFTAR GAMBAR

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Halaman ini sengaja dikosongkan

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Dewasa ini kemajuan teknologi khususnya Teknologi Informasi dan Komunikasi (TIK) berkembang dengan begitu cepat seiring dengan kemajuan ilmu pengetahuan. Setiap inovasi dalam TIK diciptakan untuk memberikan manfaat yang begitu besar dalam berbagai bidang terutama dalam bidang pemerintahan. Penerapan TIK dalam pemerintahan dapat memberikan informasi dan pelayanan bagi warganya, urusan bisnis, serta hal-hal lain yang berkenaan dengan pemerintahan dalam hal ini sering disebut dengan *e-government*. Atau dengan kata lain *e-government* dapat diartikan sebagai kumpulan konsep untuk semua tindakan dalam sektor publik (baik di tingkat Pemerintah Pusat maupun Pemerintah Daerah) yang melibatkan teknologi informasi dan komunikasi dalam rangka mengoptimalisasi proses pelayanan publik yang efisien, transparan dan efektif [1].

Penerapan *e-government* dapat membantu mewujudkan tata pemerintahan yang baik (*good governance*) yakni dapat mempercepat proses kerja serta modernisasi administrasi melalui otomatisasi di bidang administrasi perkantoran, modernisasi penyelenggaraan pelayanan kepada masyarakat. Khususnya bagi pemerintah daerah penerapan *e-government* itu sangat membantu permasalahan terbatasnya sumber daya manusia, sarana dan prasarana dalam menjalankan proses pelayanan, proses administrasi dan proses pemerintahan lainnya.

Beberapa keuntungan dalam penerapan *e-government* antara lain [2]:

1) Meningkatkan efisiensi lembaga pemerintah dalam pemrosesan data.
2) Meningkatkan layanan melalui pemahaman yang lebih baik tentang kebutuhan pengguna yang bertujuan untuk layanan *online* tanpa batas. Sehingga masyarakat dapat dilayani kapan saja tanpa harus menunggu jam pelayanan kantor buka.
3) Meningkatkan transparansi, akurasi dan memfasilitasi transformasi informasi antara pemerintah dan masyarakat/publik.

Tujuan *e-government* adalah untuk meningkatkan hubungan pemerintah dengan para *stakeholder*. Bentuk-bentuk hubungan pemerintahan dengan *stakeholder* dalam pemanfaatan TIK antara lain [3]:

1) G2C (*government to citizen*), adalah pemanfaatan TIK untuk melayani kebutuhan masyarakat luas, misalnya melayani kependudukan, kesehatan, pendidikan, bantuan sosial, dan sebagainya.
2) G2B (*government to business*), adalah pemanfaatan TIK untuk melayani kebutuhan dunia usaha, misalnya pengurusan izin usaha, permintaan data statistik yang dibutuhkan pengusaha, dan sebagainya.
3) G2G (*government to governments*), adalah pemanfaatan TIK untuk melayani kebutuhan lembaga pemerintah lain, departemen lain, pemerintah tingkat di atasnya atau di bawahnya.
4) G2E (*government to employees*), pemanfaatan TIK yang berfokus pada hubungan antara pemerintah dan pegawai untuk mengkoordinasikan operasi internal dan memperbaiki efisiensi birokrasi.

Dalam rangka meningkatkan kualitas pelayanan publik khususnya kepada masyarakat (G2C) maupun untuk dunia usaha (G2B), pemerintah pusat maupun daerah menyediakan beberapa layanan tanya-jawab pada sistem *online* yang ada. Layanan tanya-jawab ini sangat penting karena dapat memfasilitasi permintaan informasi secara lebih mudah serta dapat diakses 24 jam, tanpa harus menunggu jam kantor buka. Beberapa contoh layanan tanya-jawab pada beberapa instansi pemerintahan antara lain:

1) Layanan tanya-jawab pada aplikasi Sistem Pengadaan Secara Elektronik (SPSE). Layanan tanya-jawab ini terdapat pada setiap situs *e-procurement* setiap unit Layanan Pengadaan Secara Elektronik (LPSE).
2) Layanan tanya-jawab untuk domain kesehatan, contohnya seperti layanan tanya-jawab pada Dinas Kesehatan Kabupaten Gresik[1].
3) Layanan tanya-jawab melalui SMS Center atau media sosial lainnya tentang penyelenggaraan pemerintahan.

[1] Dapat diakses melalui http://dinkes.gresikkab.go.id/pertanyaan

Pelayanan tanya-jawab tersebut masih belum optimal karena masih dilakukan secara manual, untuk itu diperlukan sebuah sistem tanya-jawab (*Question Answering System,* QAS) yang akan dijalankan oleh sistem komputer. QAS adalah suatu sistem yang menerima *query* dalam bentuk pertanyaan dengan bahasa alami, mencari jawaban dalam sekumpulan dokumen atau pada sebuah domain basis pengetahuan, mengekstraksnya dan kemudian memformulasikan jawaban. Arsitektur umum suatu QAS yang dibagi menjadi empat modul yakni [4]: *question analysis, document retrieval, answer extraction* dan *answer evaluation*

Salah satu modul pada QAS yang berperan penting disini adalah modul *document retrieval*. Salah satu pendekatan yang digunakan dalam modul *document retrieval* menggunakan pendekatan *Information Retrieval* (IR) atau temu kembali informasi [5]. Temu kembali informasi adalah suatu sistem yang digunakan untuk mengambil semua dokumen yang relevan dengan permintaan (*query*) pengguna dengan hanya mengambil beberapa dokumen yang tidak relevan sesedikit mungkin [6]. Dengan kata lain modul ini berperan dalam menentukan informasi yang dibutuhkan atau relevan dengan pertanyaan yang akan diajukan oleh pengguna. IR digunakan oleh beberapa peneliti seperti yang dilakukan oleh [7], [8] dan [9] dalam membangun suatu QAS.

Untuk membatasi pembahasan agar tidak terlalu luas, maka dalam penelitian ini akan difokuskan pada modul *document retrieval* menggunakan IR sebagai studi awal yang nantinya dapat dipadukan dengan modul lainnya untuk digunakan dalam mengembangkan suatu QAS. Metode yang secara luas digunakan dalam IR adalah metode *Vector Space Model* (VSM). Penggunakan VSM dalam mencari dokumen yang relevan masih banyak digunakan oleh beberapa peneliti seperti yang dilakukan oleh [10] [11] [12] [13] dan [14]. Proses temu kembali (*retrieval*) pada VSM dilakukan dengan merepresentasikan setiap dokumen ke dalam bentuk vektor, kemudian tingkat kemiripan (*similarity*) antar dokumen dilakukan dengan menghitung penyimpangan sudut antar vektor. VSM bekerja dengan melihat kecocokan *term* (*term similarity*) antara *query* dan korpus dimana setiap *term* memiliki dependensi yang tinggi, oleh karena itu VSM bekerja sangat baik dalam kasus pencocokan kata kunci [14]. Akan tetapi di lain sisi, dikarenakan setiap *term* memiliki dependensi yang tinggi maka VSM mempunyai permasalahan

yakni tidak mampu menangani masalah sinonim, selain itu VSM juga merepresentasikan dokumen-dokumen ke dalam dimensi ruang (*space*) yang besar dan jarang (*sparse*) sehingga membuat akurasi penghitungan *similarity* menjadi rendah [15].

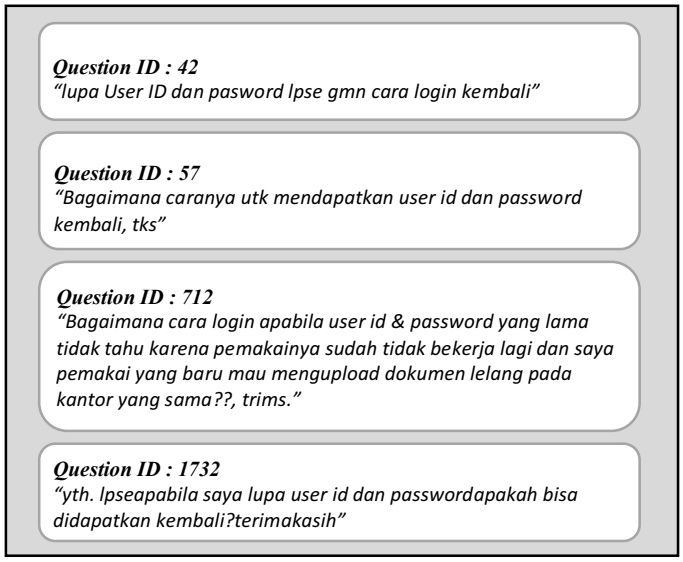
Untuk meningkatkan performansi dari VSM perlu dilakukan filter terhadap dokumen-dokumen yang tidak relevan dengan *query* sehingga semakin sedikit dokumen yang tidak relevan maka akan semakin sedikit *noise* yang ada. Hal ini bisa dilakukan dengan mengambil dokumen-dokumen yang membicarakan topik yang sama dengan *query* terlebih dahulu. Untuk menemukan topik-topik yang ada dalam kumpulan dokumen (korpus) digunakan teknik pemodelan topik. Pemodelan topik dapat memberikan keuntungan yakni dapat meningkatkan performansi proses pencarian pada IR dengan mengungkapkan dokumen yang mungkin menggunakan campuran kata kunci yang berbeda tetapi membicarakan tentang ide atau topik yang sama [16].

Latent Semantic Indexing (LSI) atau dikenal juga dengan nama *Latent Semantic Analysis* (LSA) adalah pengembangan dari metode VSM. LSI juga merupakan salah satu metode untuk melakukan pemodelan topik dengan merepresentasikan dokumen ke dalam ruang topik (*topic space*). LSI mengambil kata-kata penting dari informasi yang diberikan oleh dokumen dan menangkap kesamaan semantik (*semantic similarity*) antara kata-kata sehingga mampu mengatasi masalah sinonim [17] selain itu model LSI mereduksi dimensi ruang vektor sehingga mempunyai akurasi yang lebih baik daripada VSM [15].

Oleh karena itu, proses pencarian pertanyaan yang relevan dengan *query* akan digunakan pendekatan topik dengan mengkombinasikan metode LSI dan VSM. Dari penggunaan kombinasi tersebut diharapkan dapat digunakan untuk mengukur kesamaan topik (*topic similarity*), semantik (*semantic similarity*) dan kata kunci (*keywords)/term* (*term similarity*) antara *query* (pertanyaan dari pengguna) dengan korpus tanya-jawab yang ada pada *database*, sehingga mampu meningkatkan performansi IR jika menggunakan model LSI atau VSM yang berdiri sendiri.

Untuk mengetahui performansi dari kombinasi metode LSI dan VSM pada modul temu kembali informasi suatu sistem tanya-jawab, pada penelitian ini akan

digunakan data arsip tanya-jawab pada layanan tanya-jawab SPSE sebagai eksperimen. Karakteristik dari data arsip tanya-jawab pada SPSE adalah sebagian besar pertanyaan yang diajukan oleh pengguna adalah pertanyaan-pertanyaan yang sudah ditanyakan terlebih dahulu oleh pengguna lainnya atau membicarakan tentang topik yang sama. Sebagai contoh dari data yang diambil pada layanan tanya jawab LPSE Provinsi Jawa Barat, sebanyak 1562 dari 2218 atau 70% pertanyaan yang diajukan memiliki topik yang relevan atau kemiripan dengan pertanyaan yang telah ada pada *database* (arsip). Pertanyaan-pertanyaan yang mirip tersebut ditulis dengan kalimat yang hampir sama atau redaksi yang berbeda tetapi mempunyai maksud atau makna yang sama seperti contoh yang ditunjukkan pada Gambar 1.1. Sehingga disini tugas utama dari sistem temu kembali informasi adalah bagaimana mengambil pertanyaan-pertanyaan pada arsip tanya-jawab yang relevan dengan pertanyaan baru dari pengguna. Untuk ke depannya, sistem temu kembali informasi yang dihasilkan nantinya yakni kombinasi LSI dan VSM diharapkan dapat diadopsi untuk sistem tanya-jawab pada layanan *online* pemerintahan lainnya.



Gambar 1.1 Contoh Pertanyaan-Pertanyaan yang Serupa pada Layanan Tanya Jawab LPSE

1.2 Rumusan Masalah

Dari uraian di atas dapat dirumuskan masalah yang dihadapi yakni metode temu kembali informasi menggunakan VSM mempunyai kelemahan yakni tidak mampu menangani sinonim dan hanya menggunakan pencocokan *terms/keywords*, sehingga perlu dikombinasikan dengan metode LSI sebagai pemodelan topik yang

mampu menangani beberapa masalah semantik seperti sinonim. Dengan penggabungan tersebut diharapkan proses IR mampu menemukan pertanyaan yang mempunyai kesamaan topik (*topic similarity*), semantik (*semantic similarity*) dan *keywords/term* (*term similarity*) dengan *query* dari pengguna.

1.3 Tujuan

Tujuan dari penelitian ini adalah untuk membuat suatu model temu kembali informasi menggunakan metode VSM dan menggabungkannya dengan metode LSI sebagai pemodelan topik agar mampu meningkatkan kinerja dari metode VSM dan metode LSI murni sehingga dapat digunakan untuk mencari pertanyaan yang relevan pada data layanan tanya-jawab berdasarkan kesamaan kata kunci dan topik.

1.4 Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut :

1. *Dataset* yang digunakan pada penelitian ini hanya menggunakan arsip pertanyaan pada layanan tanya-jawab LPSE Provinsi Jawa Barat ditambah dengan *Frequently Asked Questions* (FAQ) pada portal *e-procurement* LKPP.
2. Penelitian difokuskan pada teknik IR, tidak sampai membuat suatu QAS secara keseluruhan.
3. Dokumen yang diproses terbatas pada dokumen teks saja, belum mendukung dokumen dalam bentuk audio atau video.

1.5 Kontribusi

Kontribusi yang diharapkan adalah bahwa penelitian ini adalah sebagai studi awal dalam membangun suatu sistem QAS dengan menyediakan model temu kembali (IR) berbasis pemodelan topik sebagai pondasi untuk penelitian selanjutnya. Sehingga ke depannya dapat dibangun suatu sistem tanya-jawab yang dapat mengoptimalkan layanan permintaan informasi dari masyarakat/publik.

1.6 Metodologi Penelitian

Tahapan metodologi penelitian yang dilakukan pada penelitian ini ditunjukkan oleh Gambar 1.2, sedangkan rincian metode dan langkah penelitian

akan dipaparkan pada Bab 3. Secara garis besar proses penelitian dibagi menjadi 3 proses yakni:

1. Persiapan; dalam tahapan ini akan dilakukan analisis permasalahan, studi literatur serta pengumpulan jurnal maupun penelitian-penelitian yang berhubungan dengan temu kembali informasi dan pemodelan topik. Pada tahap ini juga dilakukan pengumpulan *dataset* yang diperlukan untuk kebutuhan penelitian.
2. Pengembangan Model IR; dari hasil analisis permasalahan dan melakukan studi literatur terkait, selanjutnya dilakukan pengembangan model IR berbasis pendekatan topik yang akan digunakan untuk memecahkan permasalahan yang dihadapi.
3. Hasil penelitian; tahapan terakhir ini dilakukan evaluasi terhadap hasil penelitian untuk mengetahui sejauh mana model yang dikembangkan mampu memecahkan permasalahan serta mendefinisikan kekurangan-kekurangan yang ditemukan sebagai dasar melakukan perbaikan penelitian untuk penelitian selanjutnya.

Gambar 1.2 Metodologi Penelitian

Halaman ini sengaja dikosongkan

BAB 2
KAJIAN PUSTAKA

2.1 Kajian Penelitian Terkait

Pada penelitian sebelumnya banyak ditemukan penelitian tentang pencarian dokumen yang relevan menggunakan pendekatan IR baik yang menggunakan metode VSM, LSI ataupun metode lainnya. Walaupun penelitian tentang IR sudah lama dilakukan namun topik penelitian tentang IR masih menjadi topik yang menarik untuk diteliti. Berikut adalah beberapa penelitian sebelumnya yang mempunyai tema terkait dan sebagai pendukung penelitian ini:

2.1.1 Sistem Deteksi Website Negatif Berbahasa Indonesia Menggunakan TF-IDF & VSM [10]

Teguh Bharata Adji et al. dalam penelitiannya menggunakan VSM dalam mendeteksi teks website berbahasa Indonesia yang mengandung konten pornografi. Sistem pendeteksian dilakukan dengan cara mengambil data teks pada suatu website, kemudian dilakukan pengukuran kemiripan (*similarity*) antara teks input dengan data training. Teks input dan data training ditransformasikan ke dalam VSM untuk diukur tingkat kemiripan menggunakan *cosine similarity*. Hasil pengukuran kemiripan diklasifikasikan menjadi 2 kelompok yakni positif (non pornografi) dan negatif (pornografi).

Dataset yang digunakan sebanyak 200 website yang dibagi menjadi 2 yakni 100 website untuk proses *training* kemudian sisanya untuk keperluan proses *testing.* Kemudian training data dibagi menjadi dua kelompok yakni kelompok positif dan negatif seperti yang ditunjukkan oleh Tabel 2.1 berikut:

Tabel 2.1 Data Training Pada Penelitian Teguh Bharata Adji et al.

Negatif (Pornografi)	Positif (Non Pornografi)
Situs web yang menyediakan video pornografi	Situs web yang berisi berita tak bermoral seperti kasus kekerasan, kasus video kotor
Artikel atau cerita yang mengandung materi vulgar	Situs web yang berisi pendidikan seks dan seks yang sehat
Website yang berisi foto vulgar	Situs web menyediakan tanya jawab atau konsultasi tentang seks

Dari hasil penelitian tersebut diperoleh hasil akurasi yang cukup baik yakni 82,8%. Akan tetapi sistem tersebut masih memiliki keterbatasan yakni jika pada dokumen terdapat kamuflase frasa atau kesalahan pengetikan teks. Contoh seperti kata 'bokep' dan 'b0kep' dianggap tidak sama padahal maksudnya sama sehingga menyebabkan kesalahan dalam pengklasifikasian.

2.1.2 Peningkatan Algoritma Kemiripan Kalimat Berbasis VSM dan Penerapannya dalam Sistem Tanya Jawab [11]

Pada penelitian yang dilakukan Xu Liang et al. menggunakan metode VSM dalam menemukan *similarity* antara pertanyaan pengguna dengan pertanyaan yang ada pada FAQ pada suatu sistem QAS. Fokus pada penelitian ini adalah bagaimana untuk meningkatkan akurasi dari VSM dengan melakukan modifikasi pada pembobotan TF-IDF. Modifikasi dilakukan dengan menggunakan pembobotan profesional dan pembobotan umum berbasis konsep (VSM dengan pembobotan profesional). Pembobotan profesional (PP) yakni mengalikan hasil pembobotan TF-IDF dengan 1 (satu) untuk *term-term* yang dianggap penting sedangkan pembobotan umum yakni mengalikan hasil pembobotan TF-IDF dengan 0,8 untuk *term-term* selainnya (di luar konsep profesional).

Data *training* yang digunakan adalah 1400 pasangan pertanyaan-jawaban pada FAQ serta untuk testing digunakan 100. Dari hasil penelitian diperoleh bahwa dengan menggunakan pembobotan profesional sistem yang diajukan dapat meningkatkan hasil akurasi dari VSM tradisional yang semula memperoleh 63% meningkat menjadi 88%.

2.1.3 Temu Kembali Pertanyaan Komunitas Pada Forum Kesehatan [12]

Hamman Samuel et al. dalam penelitiannya menggunakan teknik IR berbasis pembobotan TF-IDF yang dikombinasikan dengan *relevansi heuristik* dan *term expansion*. Tujuan penelitian tersebut adalah untuk membuat model IR yang dapat mencari pertanyaan pada arsip tanya-jawab pada komunitas forum kesehatan yang relevan dengan pertanyaan baru dari pengguna. Hasil penelitian yang diperoleh kemudian dibandingkan dengan beberapa metode IR lain sebagai *baseline* seperti BM25, LDA, *Language Model*, LSI, *Vector Space*, Word2Vec dan

NLTK *Similarity*. Dari hasil tersebut diketahui bahwa sistem yang diajukan mampu mengungguli metode lainnya dengan hasil *recall* mencapai 0,818.

2.1.4 Algoritma Kemiripan Teks Berbasis VSM Semantik [13]

Li Hong Xu et al. dalam penelitiannya menggunakan VSM yang telah dimodifikasi (diberinama VSM-Cilin) untuk mencari audio yang diinginkan pada *library*. Hal tersebut dilakukan dengan bantuan teknologi *speech recognition* terlebih dahulu untuk mengkonversi audio menjadi teks. Sistem kemudian akan mencari lima teks audio yang paling mirip dengan menggunakan algoritma VSM-Cilin untuk merekomendasikannya kepada editor. Disini VSM tradisional telah dikombinasikan dengan algoritma Cilin sehingga mampu menangani masalah semantik seperti sinonim dan polisemi.

Hasil penelitian yang diperoleh kemudian dibandingkan dengan beberapa metode lain sebagai *baseline* yakni VSM dan *Bidirectional Mapping* - Cilin (BM-Cilin). Dari hasil tersebut diketahui bahwa akurasi VSM-Cilin mampu mengungguli metode lainnya seperti yang ditunjukkan oleh Tabel 2.2 berikut.

Tabel 2.2 Hasil Penelitian Li Hong Xu et al.

Algoritma	**Kategori**			
	Hiburan	Militer	Olahraga	Politik
BM-Cilin	56%	64	50%	52,5%
VSM	70%	68	72,5%	64%
VSM-Cilin	84%	86	92%	80%

2.1.5 Evaluasi Kinerja Model VSM dan LSI untuk Menentukan Kemiripan Pada *Bug Reports* [14]

Indu Chawla et al. pada penelitiannya melakukan evaluasi terhadap algoritma IR yakni VSM dan LSI dalam melakukan pencarian kemiripan *bug* pada *bug reports*. Pada penelitian ini menggunakan dataset dari Google *chrome bug repository* dengan nomor ID *bug* dari 8000 sampai dengan 112782. Dari dataset tersebut diambil 106 *bug reports* yang terdiri dari 3 komponen yakni 27 dari konten, 29 dari CEEE dan 50 dari chromeframe.

Output dari LSI dan VSM yang diambil adalah 10 *bug reports* dengan nilai *similarity* tertinggi yang diambil sebagai hasil dari setiap kueri untuk perbandingan antara dua model. Kinerja dari LSI/VSM kemudian dibandingkan

berdasarkan jumlah total *bug reports* yang relevan yang diambil oleh mereka untuk *query* dari komponen yang diberikan. Dari hasil penelitian diperoleh bahwa LSI mampu mengambil lebih banyak jumlah dokumen yang relevan dibandingkan dengan VSM. Hal ini disebabkan karena VSM bekerja sangat baik dalam kasus pencocokan kata kunci, sedangkan LSI baik digunakan dalam pencarian dokumen yang relevan karena bekerja berbasis kemiripan semantik.

2.2 Teori Dasar

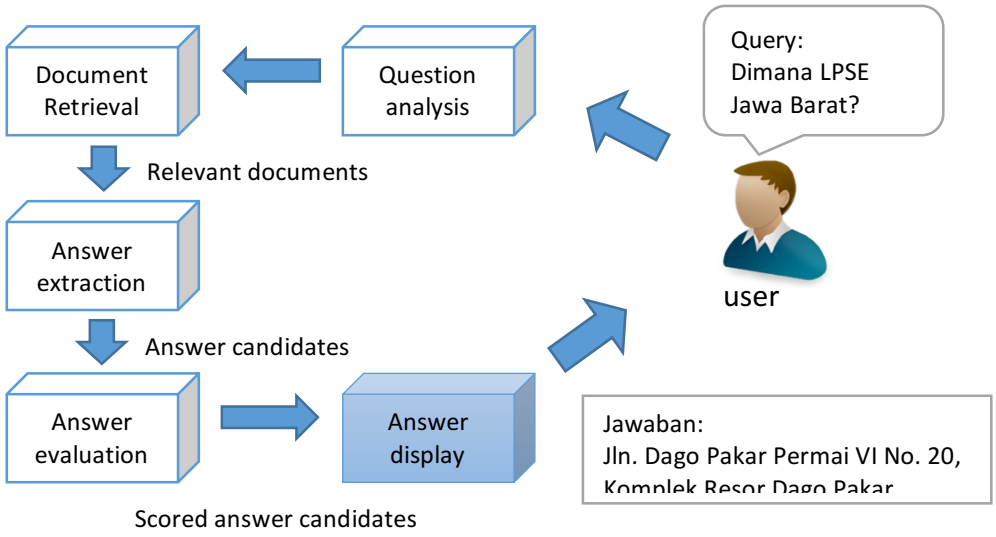
2.2.1 Question Answering System (QAS)

Sebuah QAS, menerima *query* dalam bentuk pertanyaan dengan bahasa alami, mencari jawaban pada sekumpulan dokumen atau pada sebuah domain basis pengetahuan, mengekstraknya dan kemudian memformulasikan jawaban yang ringkas. Kebanyakan sistem QA mengelompokkan pertanyaan berdasarkan jenis pertanyaannya [18]. Jika jenis pertanyaan dapat ditentukan maka jenis jawabannya dapat ditentukan pula. Misalnya pertanyaan "dimana?" membutuhkan alamat alamat/lokasi suatu objek, kemudian pertanyaan "siapa?" membutuhkan jawaban berupa orang/kelompok/ organisasi.

Arsitektur umum suatu QAS (lihat Gambar 2.1) yang dibagi menjadi empat modul yakni [4]: *question analysis, document retrieval, answer extraction* dan *answer evaluation*. Tahapan yang dilakukan adalah sebagai berikut:

1) Modul *question analysis* akan menganalisis sebuah pertanyaan dan menentukan jenis jawabannya. Jenis jawaban mewakili jenis informasi yang diminta oleh sebuah pertanyaan: nama seseorang, nama tempat, dan ungkapan numerik adalah beberapa jenis jawaban yang mungkin. Sebagai contoh "Siapa presiden pertama Amerika Serikat?" Jenis jawaban adalah orang (nama orang).
2) Modul *document retrieval* akan menggunakan mesin pencari untuk mengambil dokumen yang relevan berdasarkan kata kunci dalam pertanyaan. Karena sistem penjawab pertanyaan mencari jawaban hanya pada dokumen yang diambil, keakuratan pengambilan dokumen sangat penting.
3) Modul *answer extraction* akan mengekstrak dari dokumen yang diambil semua menjawab kandidat yang sesuai dengan jenis jawaban. Bila tipe jawaban adalah orang, semua nama orang dalam dokumen yang diambil akan diekstrak

4) Modul *answer evaluation* akan mengevaluasi kesesuaian kandidat jawaban dengan menggunakan informasi seperti bagaimana penampilan mereka dalam dokumen, dan memberikan skor kepada kandidat jawaban. Akhirnya, kandidat jawaban dengan skor tertinggi diajukan untuk digunakan sebagai jawaban.



Gambar 2.1 Bentuk Arsitektur Umum QAS [4]

Berdasarkan domain pengetahuan (*knowledge*), QAS dibagi menjadi dua jenis yakni *open* domain QA dan *restricted* domain QA. *Open* domain QA berbasis pada sejumlah besar dokumen pada *web restricted domain* [19]. Pada *Restricted* domain pemanfaatan pengetahuan formal yang dimiliki dapat meningkatkan keakuratan sistem QA, karena baik pertanyaan maupun jawabannya dianalisis berdasarkan basis pengetahuan tersebut. *Restricted* domain juga biasanya digunakan jika sebuah institusi memiliki dan mengelola basis pengetahuan yang sifatnya terbatas dan hanya dipergunakan dalam lingkup institusi tersebut.

Ada dua paradigma dalam pengembangan suatu QAS yakni [5]:

1) *Information Retrieval (IR)-based question answering* atau disebut juga *text-based question answering* adalah QAS yang bergantung pada sejumlah besar informasi yang tersedia dalam bentuk teks dalam Web atau koleksi dokumen khusus. Dari pertanyaan pengguna, teknik pengambilan informasi dilakukan dengan mengekstrak langsung bagian-bagian dari dokumen-dokumen yang dipandu model/jenis pertanyaannya. Metode ini memproses pertanyaan untuk

menentukan jenis jawaban yang mungkin (seringkali merupakan entitas bernama seperti seseorang, lokasi, atau waktu) dan merumuskan permintaan untuk dikirim ke mesin pencari. Mesin pencari kemudian memberikan dokumen yang telah diranking dan dipecah menjadi bagian-bagian yang sesuai. Akhirnya kandidat jawaban akan diekstrak dari bagian-bagian yang diranking tersebut.

2) *knowledge-based question answering*, disini QAS akan membangun representasi semantik dari *query* pertanyaan. *Knowledge-based* QAS akan menjawab pertanyaan dalam bahasa alami yang akan memetakannya dalam sebuah *query* melalui *database* terstruktur. Sistem akan memetakan teks pertanyaan ke dalam bentuk *logic* yang disebut dengan *semantic parsers*. *Semantic parser* akan memetakan teks pertanyaan dalam bentuk *predicate calculus* penuh atau bahasa *query* seperti SQL/SPARQL.

Berdasarkan jenis data yang ada dalam sumber informasi, QAS dibagi menjadi 3 jenis yakni [20]:

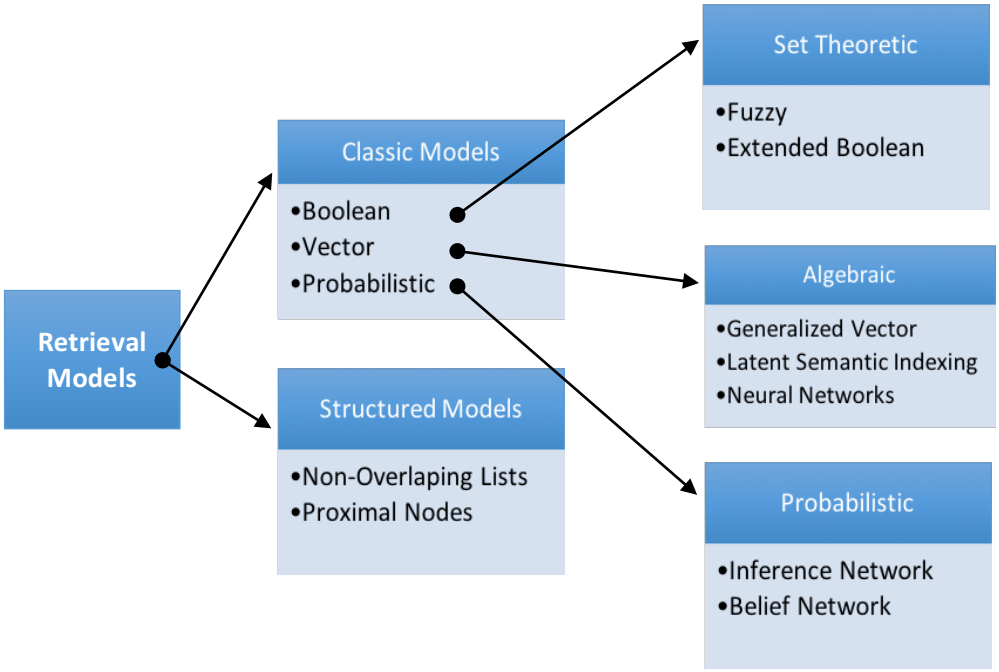
1) Sumber data terstruktur, dalam dokumen terstruktur data disusun dalam set semantik (entitas) yang dikumpulkan dalam bentuk relasi. Entitas dalam relasi yang sama memiliki atribut yang sama. Deskripsi semua entitas dalam satuan disebut skema. Pengaturan data sesuai dengan format yang telah ditentukan.
2) Sumber data semi terstruktur, adalah bentuk data terstruktur yang tidak sesuai dengan struktur formal model data pada *database* relasional atau bentuk tabel data lainnya. Contoh dari data semi terstruktur adalah *Extensible Markup Language* (XML) dan *JavaScript Object Notation* (JSON).
3) Sumber data tidak terstruktur, yakni data bisa dari jenis apapun. Data tidak terstruktur dalam set semantik apapun. Tidak ada aturan ketat untuk pengaturan data dalam sumber data jenis ini.

2.2.2 Temu Kembali Informasi

Temu kembali informasi atau *Information Retrieval* (IR) adalah suatu sistem yang digunakan untuk mengambil semua dokumen yang relevan dengan permintaan (*query*) pengguna dengan hanya mengambil beberapa dokumen yang tidak relevan sesedikit mungkin [6]. Atau IR dapat didefinisikan sebagai sekumpulan algoritma dan teknologi untuk melakukan pemrosesan, penyimpanan dan menemukan kembali informasi yang ada. Proses dari IR terdiri dari beberapa langkah, dimulai dari penginputan *query* untuk menentukan dokumen mana yang sesuai dengan *query* yang di-*input* hingga memprioritaskan dokumen mana yang paling relevan dengan *query* yang di-*input*. Dalam proses pengambilan dokumen ada beberapa model yang digunakan dalam sistem IR seperti yang ditunjukkan oleh Gambar 2.2. Pada model klasik dibagi menjadi 3 jenis yakni *boolean*, vektor dan probabilistik.

Model yang pertama yakni model *boolean,* model ini adalah model pertama dalam IR yang bekerja berbasis logika *boolean*. Dalam model ini dokumen-dokumen dianggap sebagai sekumpulan *term*. *Query* dan dokumen yang terkait digabungkan dengan menggunakan operator logika matematika George Boole, seperti operator konjungtif (AND), disjungtif (OR), atau negasi (NOT). Model pengambilan *Boolean* adalah model dengan pencocokan tepat, sehingga model mengambil dokumen yang sama persis dengan *query* pengguna. Misalnya, untuk permintaan "*information* AND *retrieval* AND *system*", sistem akan mengembalikan semua dokumen yang memiliki tiga *term query* tersebut [21].

Pada model kedua yakni menggunakan vektor, model tersebut berbasis pada ide kemiripan (*similarity*). Diasumsikan bahwa jika dokumen pertama lebih mirip dengan *query* daripada dokumen yang lain, maka dapat dikatakan bahwa dokumen pertama lebih relevan daripada dokumen kedua atau yang lainnya. Jadi dalam hal ini, dilakukan pemeringkatan untuk mendefinisikan tingkat kemiripan antara *query* dan dokumen [22]. Contoh dari model ini adalah *Vector Space Model* (VSM) [21].

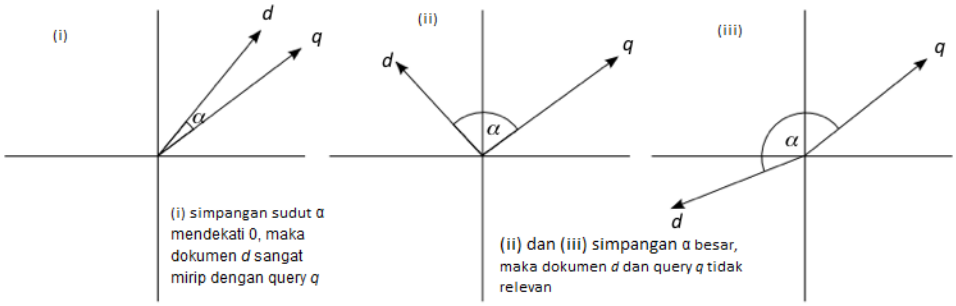


Gambar 2.2 Pembagian Model Temu Kembali Informasi [6]

Model ketiga bekerja dengan strategi yang berbeda. Diasumsikan bahwa *query* dan dokumen adalah semua pengamatan dari variabel acak, dan diasumsikan pula ada variabel acak biner bernama *R* (dengan nilai 1 atau 0) untuk menunjukkan apakah dokumen relevan dengan *query*. Kemudian menentukan skor dokumen yang berhubungan dengan *query* sebagai probabilitas bahwa variabel acak R ini sama dengan 1 diberikan dokumen tertentu dan *query* [22]. Contoh dari model ini adalah *Probabilistic Ranking Principle* (PRP), *Binary Indep.endence Retrieval* (BIR) dan *The Probabilistic Indexing Model* [21].

2.2.3 Vector Space Model (VSM)

Dalam VSM dokumen teks direpresentasikan ke dalam bentuk vektor sehingga dimungkinkan untuk membandingkan dokumen dengan *query* untuk mengetahui seberapa mirip satu dengan yang lain. Kemudian dihitung koefisien kemiripan (*similarity coefficient*, SC) untuk mengukur tingkat kemiripan antara suatu dokumen dengan *query* [23]. Umumnya rumus yang digunakan dalam perhitungan SC adalah *cosine similarity*. *Cosine similarity* mengukur kosinus sudut antara vektor *query q* dan vektor dokumen *d* dengan nilai maksimal 1, semakin tinggi nilai *cosine similarity* maka semakin mirip dokumen tersebut dengan *query*. (Sebagai ilustrasi dapat dilihat pada Gambar 2.3).



Gambar 2.3 Contoh Nilai *Cosine Similarity* pada Dua Dokumen Berbeda [24]

Jika dokumen d_j dan *query q* direpresentasikan dalam bentuk vektor :

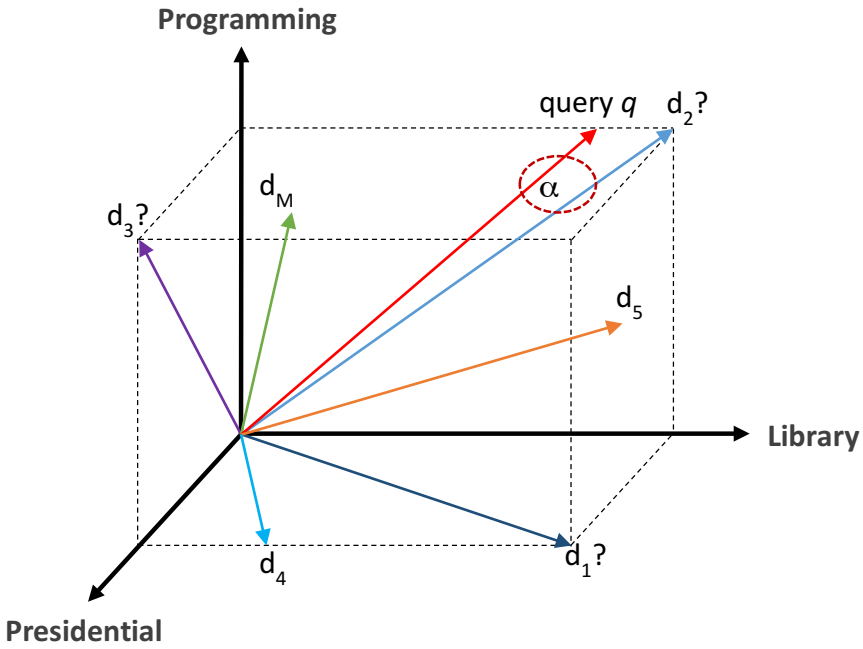
dj = ($w_{1,j}$, $w_{2,j}$, … ,$w_{n,j}$)

q = ($w_{1,q}$, $w_{2,q}$, … ,$w_{n,q}$)

maka *cosine similarity* antara dokumen d_j and *query q* bisa dihitung dengan menggunakan rumus:

$$sim(d_j, q) = \cos\alpha = \frac{d_j \cdot q}{\|d_j\| \times \|q\|} = \frac{\sum_{i=1}^{N} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{N} w_{i,j}^2} \times \sqrt{\sum_{i=1}^{N} w_{i,q}^2}} \quad (2.1)$$

Pada Gambar 2.4 menunjukkan representasi 3 *term* (*programming, library, presidential*) ke vektor dalam ruang 3 dimensi. Semua vektor dokumen dan *query* diletakkan pada ruang vektor tersebut. Sebagai contoh vektor dokumen d_1 yang berisi *term 'library'* dan *'presidential'* tetapi tidak mengandung *term 'programming'*. Vektor dokumen d_2 yang berisi *term 'library'* dan *'programming'* tetapi tidak mengandung *term 'presidential'*. Dan vektor dokumen d_3 yang berisi *term 'programming'* dan *'presidential'* tetapi tidak mengandung *term 'library'*. Untuk menghitung *similarity*, vektor *query q* diletakkan juga pada ruang vektor tersebut, kemudian setelah itu dapat diukur *similarity* antara vektor *query* dan setiap vektor dokumen. Dari gambar tersebut kita dapat dengan mudah menyimpulkan bahwa *query q* sangat dekat dengan dokumen d_2 (karena mempunyai simpangan sudut paling kecil dengan dokumen lainnya) sehingga dari semua dokumen yang ada maka dokumen d_2 adalah yang paling mirip/relevan dengan *query q*.



Gambar 2.4 Representasi Dokumen dan *Query* Dalam Ruang Vektor [25]

Untuk meningkatkan performansi dari VSM maka dilakukan pembobotan setiap *term*. Apabila pembobotan tiap *term* dapat dilakukan dengan baik maka hasil penghitungan *similarity* akan menghasilkan pemeringkatan dokumen yang lebih baik. Pembobotan yang secara luas digunakan dalam VSM adalah menggunakan *Term Frequency* × *Inverse Document Frequency* (TF-IDF). TF-IDF adalah hasil kali antara pembobotan *term frequency* (TF) dan pembobotan *inverse document frequency* (IDF) [21].

Term frequency ($tf_{t,d}$) dari *term t* dan dokumen *d* didefinisikan sebagai jumlah berapa kali *term t* muncul pada dokumen *d*. Rumus TF tanpa normalisasi (*raw-term frequency*):

$$tf_{t,d} = f_{t,d} \qquad (2.2)$$

Jika menggunakan normalisasi, TF dari *term t* pada dokumen *d* diberikan oleh rasio antara frekuensi kemunculan *term* spesifik dan *term* dengan frekuensi maksimum yang muncul pada korpus. Sehingga rumus untuk TF menjadi:

$$tf_{t,d} = 0.5 + \frac{0.5 \times f_{t,d}}{\max_t \left(f_{t,d} \right)} \quad (2.3)$$

Document Frequency (DF) adalah jumlah dokumen yang mengandung *term* tertentu. Sedangkan IDF adalah kebalikan dari DF, IDF digunakan untuk menghargai *term-term* penting yang tidak terjadi di banyak dokumen. Dengan kata lain, sebuah *term* yang jarang muncul pada korpus atau bisa dikatakan sebagai *term* khusus akan memiliki nilai IDF yang tinggi. Rumus IDF adalah sebagai berikut:

$$idf = \log \frac{N}{n_i} \quad (2.4)$$

dimana N adalah jumlah total dokumen pada sistem, dan n_i adalah jumlah dokumen-dokumen yang mengandung *term t* dengan indeks *i* (t_i).

Sehingga rumus untuk TF-IDF menjadi:

$$tf \cdot idf(t,d,N) = tf(t,d) \times idf(t,N) \quad (2.5)$$

Kelebihan dari VSM adalah [15]:

1. Model yang sederhana karena merupakan aljabar linear.
2. Memungkinkan menghitung berkelanjutan secara bersamaan antara *query* dan dokumen.

Sedangkan batasan atau kekurangan dari VSM adalah [15]:

1. Tidak mampu menangani sinonim dan polisemi.
2. Direpresentasikan dalam dimensi yang tinggi.
3. Vektor dokumen-dokumen biasanya sangat *sparse* (lebih banyak nilai nol), sehingga menyebabkan penghitungan *cosine similarity* menjadi tidak akurat.

2.2.4 Pemodelan Topik (*Topic Modeling*)

Pemodelan topik, dalam konteks *Natural Language Processing* (NLP), digambarkan sebagai metode untuk mengungkap struktur tersembunyi dalam kumpulan teks. Bahkan lebih dari itu, pemodelan topik itu bisa didefinisikan sebagai metode untuk [26]:

1. Pengurangan dimensi, dimensi ruang pada VSM dapat direduksi menggunakan pemodelan topik dengan hanya mengambil *term* yang dianggap penting. Pada VSM dokumen direpresentasikan ke dalam *term space* kemudian oleh

pemodelan topik, *term space* tersebut akan ditransformasi ke dalam *topic space* dengan dimensi yang lebih kecil.

2. Unsupervised Learning, Pemodelan topik dapat dengan mudah dibandingkan dengan teknik *clustering*. Seperti dalam kasus *clustering*, jumlah topik itu seperti jumlah *cluster*. Pemodelan topik dapat membangun kelompok kata-kata dan bukan kumpulan teks. Sebuah teks merupakan campuran dari semua topik, masing-masing memiliki bobot tertentu.
3. Sebagai pemberi tanda *(tagging*), jika pada teknik klasifikasi dokumen menetapkan kategori tunggal untuk teks, pemodelan topik mampu menetapkan beberapa *tag* ke teks. Seorang pakar manusia dapat memberi label pada topik yang dihasilkan dengan label yang dapat dibaca manusia dan menggunakan heuristik yang berbeda untuk mengonversi topik yang telah diberi bobot untuk kumpulan *tag*.

Beberapa tujuan dari penerapan pemodelan topik adalah sebagai berikut [16]:

1. Pemodelan topik dapat membantu pengorganisasian dokumen, misalnya untuk mengelompokkan beberapa artikel berita yang saling berhubungan.
2. Pemodelan topik dapat membantu pembuatan sistem rekomendasi tentang apa yang harus dibaca berikutnya dengan mencari bahan yang memiliki daftar topik yang sama
3. Pemodelan topik dapat meningkatkan hasil pencarian dengan mengungkapkan dokumen yang mungkin menggunakan campuran kata kunci yang berbeda tetapi memiliki gagasan atau ide yang sama.

Beberapa contoh algoritma yang digunakan oleh para peneliti dalam membuat pemodelan topik yakni:

1. *Latent Semantic Indexing* (LSI) atau *Latent Semantic Analysis* (LSA) [15]
2. *Probabilistic Latent Semantic Analysis* (PLSA) [15]
3. *Latent Dirichlet Allocation* (LDA) [15] [16]
4. *Hierarchical Dirichlet Processes* (HDP) [27]

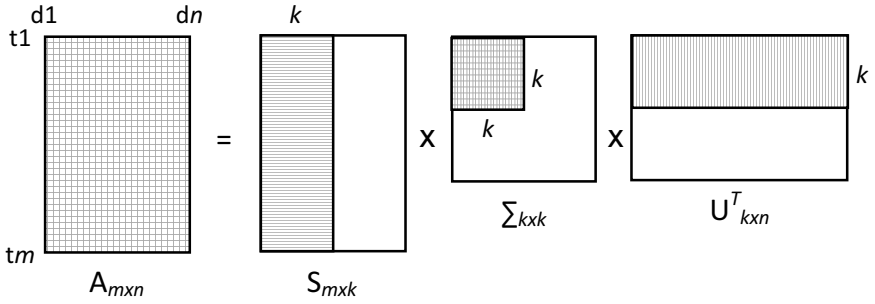
2.2.5 Latent Semantic Indexing (LSI)

LSI atau dikenal juga dengan nama *Latent Semantic Analysis* (LSA) adalah metode pengindeksan secara otomatis yang menggunakan metode pengindeksan hasil pengembangan dari model ruang vektor (VSM). LSI dikembangkan untuk mengatasi keterbatasan VSM dalam menangani masalah sinonim. Pada VSM, dokumen dan term dinyatakan sebagai sebuah vektor, sebaliknya LSI mencoba untuk merepresentasikan dokumen dengan mengambil relasi yang ada di antara *term*. LSI melakukannya dengan merepresentasikan permintaan dan dokumen berdasarkan topik bukan term. Dengan menentukan jumlah topik, LSI mewakili *query* dan dokumen-dokumen sebagai vektor dan setiap entri dari vektor sesuai dengan suatu topik. Tujuannya adalah untuk mengurangi *noise* yang disebabkan oleh sinonim dan polisemi. Dengan merepresentasikan dokumen dan *query* ke dalam topik, kata-kata serupa (misalnya, sinonim) ditetapkan pada topik yang sama, dan polisemi ditetapkan untuk berbagai topik sesuai dengan artinya yang berbeda-beda. [28].

LSI adalah pemodelan topik yang pertama kali dikembangkan. Gagasan utama di balik LSI adalah untuk memanfaatkan istilah *co-occurrence* untuk memperoleh serangkaian konsep tersembunyi, *term-term* yang sering muncul bersama diasumsikan lebih terkait secara makna (semantik). Metode LSI mampu menemukan konsep tersembunyi dalam data dokumen. Setiap dokumen dan istilah (kata) kemudian dinyatakan sebagai vektor dengan elemen yang sesuai dengan konsep ini. Setiap elemen dalam vektor memberi tingkat partisipasi dokumen atau istilah dalam konsep yang sesuai. Tujuannya bukan untuk menggambarkan konsep secara lisan, tapi untuk dapat mewakili dokumen dan persyaratan secara terpadu untuk mengekspos dokumen-dokumen, dokumen-istilah, dan istilah-istilah kesamaan atau hubungan semantik yang tersembunyi [29].

Berikut akan ditunjukkan bagaimana LSI merepresentasikan dokumen sebagai topik (yaitu bagaimana entri dari vektor, yang merupakan bobot dari topik untuk dokumen tersebut dihitung). Misalkan k adalah jumlah topik yang ingin didefinisikan (dengan $k < n$ dan $k < m$, dimana n adalah jumlah dokumen dan m adalah jumlah *term*). Maka vektor dengan dimensi $m \times n$ tersebut diurai menjadi tiga matriks menggunakan *Singular Value Decomposition* (SVD). Sehingga

diperoleh tiga matriks yakni matriks *S* (matriks dengan ukuran $m \times k$), $\sum$ (matriks dengan ukuran $k \times k$) dan U^T (matriks dengan ukuran $k \times n$). Dalam LSI matriks *S,* $\sum$ dan V^T dipotong menjadi dimensi *k* seperti yang ditunjukkan oleh Gambar 2.5. Area yang diarsir dalam matriks S dan U disimpan, seperti juga nilai pada *k* pada matriks $\sum$, area yang tidak diarsir akan dihapus [30]. Penentuan nilai *k* terbaik harus diakukan melalui percobaan [23].



Gambar 2.5 Pemotongan Matriks Menggunakan SVD [30]

Sebagai contoh, dimisalkan kita mempunyai 5 buah dokumen berikut [29]:

d_1 : *Romeo and Juliet.*
d_2 : *Juliet: O happy dagger!*
d_3 : *Romeo died by dagger.*
d_4 : *"Live free or die", that's the New-Hampshire's motto*.
d_5 : *Did you know, New-Hampshire is in New-England.*
Kemudian *query*-nya adalah : *dies, dagger*.

Jika dilakukan pencarian dokumen yang relevan, jelas bahwa d_3 berada pada peringkat teratas karena dokumen tersebut mengandung kata '*dies'* dan '*dagger'*. Dokumen d_2 dan d_4 berada dibawah d_3 karena masing–masing dokumen mengandung satu *query*. Lalu bagaimana dengan dokumen d_1 dan d_5? Jika yang menganalisis dokumen-dokumen di atas adalah seorang manusia, akan disimpulkan bahwa dokumen d_1 sebenarnya berhubungan dengan *query* diatas, sementara itu dokumen d_5 tidak terlalu berkaitan dengan *query* yang dimasukkan Dengan kata lain dokumen d_1 seharusnya berada di posisi yang lebih tinggi daripada dokumen d_5.

Dengan menggunakan LSI, sistem bisa mengetahui bahwa *term 'dagger'* sebenarnya berhubungan dengan dokumen d_1 karena *term 'dagger'* muncul bersamaan dengan *term* 'Romeo' dan 'Juliet' pada dokumen d_1, yaitu pada dokumen d_2 dan dokumen d_3. *Term 'dies'* juga berhubungan dengan dokumen d_1

dan dokumen d_5 karena muncul bersamaan dengan *term* 'Romeo' pada dokumen d_1 di dalam dokumen d_3 dan *term 'New-Hampshire'* pada dokumen d_5 di dalam dokumen d_4. Dari hubungan antar dokumen di atas, LSI menyimpulkan bahwa dokumen d_1 lebih berhubungan dengan *query* daripada dokumen d_5 karena dokumen d_1 mempunyai hubungan dengan *term 'dagger'* melalui 'Romeo' dan 'Juliet' dan juga mempunyai hubungan dengan *term 'die'* melalui *term* 'Romeo', sementara dokumen d_5 hanya mempunyai satu hubungan dengan *term 'die'* melalui *term 'New-Hamspire'*.

Dengan menggunakan LSI koleksi dokumen di atas direpresentasikan dalam *term-document* matriks A dengan dimensi $m \times n$. Jika *term i* muncul sebanyak *a* kali pada dokumen *j,* maka $A[i,j] = a$. Sehingga dimensi dari *A, m* dan *n* berhubungan dengan jumlah kata dan koleksi dokumen. Dengan contoh di atas maka matriks A adalah :

	d_1	d_2	d_3	d_4	d_5
romeo	1	0	1	0	0
juliet	1	1	0	0	0
happy	0	1	0	0	0
dagger	0	1	1	0	0
live	0	0	0	1	0
die	0	0	1	1	0
free	0	0	0	1	0
new-hampshire	0	0	0	1	1

Diasumsikan $B = A^T A$ adalah matriks dokumen-dokumen, jika dokumen *i* dan *j* mempunyai *b* kata maka $B[i,j] = b$. Di lain sisi $C = AA^T$ adalah matriks *term-term*. Jika *term i* muncul bersama dalam dokumen *c,* maka $C[i,j] = c$. Kedua matriks B adalah simetris, B adalah matriks $m \times m$ dan C adalah matriks $n \times n$. Dengan melakukan *Singular Value Decomposition* (SVD) matriks A menggunakan matriks B dan C didapatkan :

$$A = S\Sigma U^T \qquad (2.6)$$

Dimana S adalah matriks *eigenvector* dari B, U adalah matriks *eigenvector* dari C, $\sum$ adalah matriks diagonal nilai tunggal diperoleh sebagai akar kuadrat dari nilai eigen B. Matriks $\sum$ dari contoh kasus diatas adalah:

$$\Sigma = \begin{bmatrix} 2.285 & 0 & 0 & 0 & 0 \\ 0 & 2.010 & 0 & 0 & 0 \\ 0 & 0 & 1.361 & 0 & 0 \\ 0 & 0 & 0 & 1.118 & 0 \\ 0 & 0 & 0 & 0 & 0.797 \end{bmatrix}$$

kemudian mereduksi dimensi maktriks Σ menjadi Σ_k dengan ukuran matriks $k \times k$ yang hanya berisi *k* nilai tunggal yang kita simpan, *k* disini juga merupakan jumlah topik yang akan dihasilkan oleh LSI. Sehingga matriks A menjadi:

$$A_K = S_K \Sigma_K U_K^T \qquad (2.7)$$

pada contoh kasus diasumsikan jumlah topik *k* = 2, kita hanya akan mempertimbangkan dua nilai tunggal yang pertama. Sehingga diperoleh:

$$\Sigma_2 = \begin{bmatrix} 2.285 & 0 \\ 0 & 2.010 \end{bmatrix}$$

$$\begin{matrix} romeo \\ juliet \\ happy \\ dagger \\ live \\ die \\ free \\ new-hampshire \end{matrix} \rightarrow S_2 = \begin{bmatrix} -0.396 & 0.280 \\ -0.394 & 0.450 \\ -0.178 & 0.269 \\ -0.438 & 0.369 \\ -0.264 & -0.346 \\ -0.524 & -0.246 \\ -0.264 & -0.346 \\ -0.326 & -0.460 \end{bmatrix}$$

$$U_2^T = \begin{bmatrix} -0.311 & -0.407 & -0.594 & -0.603 & -0.143 \\ 0.363 & 0.541 & 0.200 & -0.695 & -0.229 \end{bmatrix}$$

Term dalam ruang konsep diwakili oleh vektor baris S_2 sedangkan dokumennya oleh vektor kolom dariU_2^T. Sebenarnya kita menghitung koordinat (dua) vektor ini dengan cara mengalikan dengan nilai tunggal yang sesuai dari Σ_2 dan mewakili *term* oleh baris vektor $S_2\Sigma_2$ dan dokumen oleh kolom vektor $\Sigma_2 U_2^T$, akhirnya didapatkan :

$$romeo = \begin{bmatrix} -0.905 \\ 0.563 \end{bmatrix}, juliet = \begin{bmatrix} -0.717 \\ 0.905 \end{bmatrix}, happy = \begin{bmatrix} -0.407 \\ 0.541 \end{bmatrix}, dagger = \begin{bmatrix} -1.001 \\ 0.742 \end{bmatrix}$$

$$live = \begin{bmatrix} -0.603 \\ 0.695 \end{bmatrix}, die = \begin{bmatrix} -1.197 \\ -0.494 \end{bmatrix}, free = \begin{bmatrix} -0.603 \\ -0.695 \end{bmatrix}, new - hampshire = \begin{bmatrix} -0.745 \\ -0.925 \end{bmatrix}$$

$$d_1 = \begin{bmatrix} -0.711 \\ 0.730 \end{bmatrix}, d_2 = \begin{bmatrix} -0.930 \\ 1.087 \end{bmatrix}, d_3 = \begin{bmatrix} -1.357 \\ 0.402 \end{bmatrix}, d_4 = \begin{bmatrix} -1.378 \\ -1.397 \end{bmatrix}, d_5 = \begin{bmatrix} -0.327 \\ -0.460 \end{bmatrix}$$

Sekarang *query* diwakili oleh vektor yang dihitung sebagai pusat massa vektor untuk *term*-nya. Dalam contoh kita, *query*-nya adalah *die, dagger* dan sehingga vektornya adalah

$$q = \frac{\begin{bmatrix} -1.197 \\ -0.494 \end{bmatrix} + \begin{bmatrix} -1.001 \\ 0.742 \end{bmatrix}}{2} = \begin{bmatrix} -1.099 \\ 0.124 \end{bmatrix}$$

Untuk menentukan peringkat dokumen dalam kaitannya dengan *query* q kita akan menggunakan *cosine similarity* dengan menggunakan persamaan (2.1). Dari contoh di atas dapat disimpulkan hasil akhirnya adalah :

1. Dokumen d_1 lebih dekat dengan *query q* daripada d_5. Akibatnya d_1 berada di peringkat lebih tinggi dari d_5. Ini sesuai dengan preferensi manusia (*Romeo and Juliet died by a dagger*).
2. Dokumen d_1 sedikit lebih dekat dengan *q* daripada d_2. Dengan demikian sistem ini cukup cerdas untuk mengetahui bahwa d_1, yang berisi Romeo dan Juliet, lebih relevan dengan *query* daripada d_2 meskipun d_2 berisi secara eksplisit salah satu kata dalam *query* sementara d_1 tidak. Seorang pengguna manusia mungkin akan melakukan hal yang sama.

Dari contoh di atas, jumlah topik *k* yang ditentukan adalah 2 dan bobot masing-masing *term* terhadap setiap topik dihasilkan oleh matriks S_2 seperti yang ditunjukkan oleh Tabel 2.3.

Tabel 2.3 Contoh Bobot Setiap *Term* Pada Setiap Topik Menggunakan LSI

Term	**Bobot**	
	Topik 1	**Topik 2**
Romeo	-0.396	0.280
Juliet	-0.394	0.450
Happy	-0.178	0.269
Dagger	-0.438	0.369
Live	-0.264	-0.346
Die	-0.524	-0.246
Free	-0.264	-0.346
New-hampshire	-0.326	-0.460

Kelebihan dari metode LSI adalah sebagai berikut:

1) Mampu mengambil kata-kata bernilai penting dari informasi yang diberikan oleh korpus [17].
2) Mampu menangkap hubungan semantik antara kata-kata yang mempunyai makna sama, sehingga dapat menyelesaikan masalah sinonim [31].
3) Merepresentasikan dokumen ke dalam ruang vektor dengan dimensi yang jauh lebih kecil dibandingkan dengan metode VSM [15].

Keterbatasan dari metode LSI adalah sebagai berikut:

1) Penggunaannya terbatas pada informasi statistik dari *term* pada ruang topik LSI [17]. Dengan kata lain dengan menggunakan metode LSI akan mengabaikan *term-term* yang tidak mempunyai bobot pada ruang topik.
2) Tidak ada cara tertentu dalam menentukan nilai jumlah topik *k* terbaik, nilai tersebut harus didapatkan melalui percobaan [23].

2.2.6 Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) adalah model probabilistik generatif dari korpus. Ide dasarnya adalah dokumen tersebut direpresentasikan sebagai campuran acak topik yang tersembunyi di dalamnya, dimana masing-masing topik dicirikan dengan distribusi atas kata-kata. Menerapkan model LDA dapat kumpulan dokumen teks dapat mempelajari probabilitas kemunculan topik pada sebuah dokumen dan probabilitas kata-kata pada suatu topik [32].

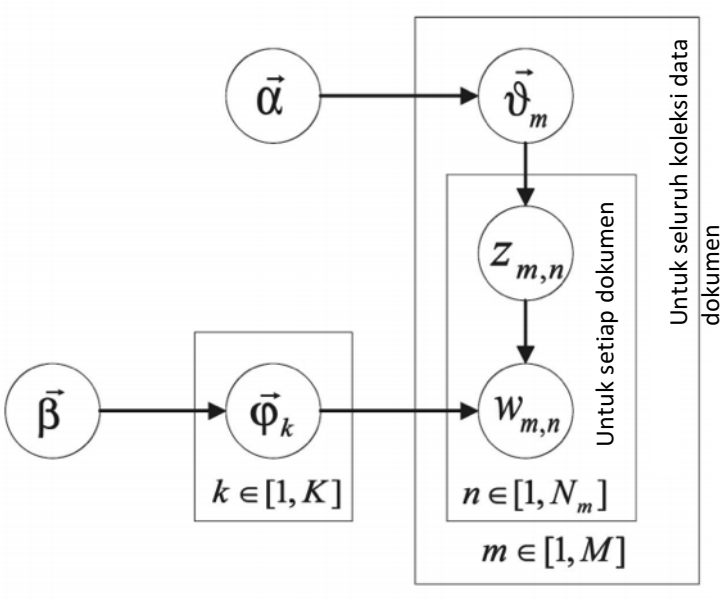
Dalam pemodelan topik, diasumsikan bahwa dalam koleksi dokumen terkait, setiap dokumen mencakup beberapa kombinasi topik. Kumpulan dokumen bisa berupa makalah akademis, e-mail, postingan facebook atau twitter, dan sebagainya. Dengan pemodelan topik, tujuan utamanya adalah menemukan struktur topik tersembunyi untuk koleksi dokumen. Struktur topik mencakup tiga hal: topik itu sendiri, distribusi statistik dari topik-topik di antara dokumen-dokumen, dan kata-kata dalam dokumen yang terdiri dari topik.

Teknik pemodelan topik probabilistik seperti LDA dapat bekerja dengan cara *unsupervised* (yang tidak diawasi) karena teknik tersebut menggunakan probabilitas bersyarat untuk mendapatkan struktur topik tersembunyi ini untuk koleksi dokumen tertentu. Untuk melakukan ini, LDA mengasumsikan bahwa setiap dokumen dalam koleksi adalah tentang beberapa topik, tetapi bahwa topik ini didistribusikan secara tidak merata di seluruh dokumen. Struktur topik itu sendiri adalah variabel tersembunyi yang perlu diturunkan berdasarkan variabel yang diamati, yang merupakan kata-kata dalam dokumen. Tantangan komputasi untuk LDA adalah menghitung probabilitas masing-masing struktur topik yang mungkin diberikan kata-kata, atau pengamatan. Jika jumlah topik dan kata-kata keduanya besar, maka akan menjadi masalah komputasi yang sulit dipecahkan. Untuk mengatasi masalah ini untuk kumpulan data besar, algoritma pemodelan topik akan berusaha mengurangi jumlah kemungkinan untuk topik atau kata-kata [16].

LDA dikembangkan berdasarkan asumsi, proses pembuatan dokumen oleh LDA digambarkan oleh Gambar 2.6. Dalam LDA, sebuah dokumen $\vec{w}_m = \{w_{m,n}\}_{n=1}^{N_m}$ dihasilkan dengan pengambilan pertama dari distribusi pada topik $\vec{v}_m$ dari suatu distribusi *Dirichlet* $\left(Dir(\vec{\propto})\right)$, yang bertugas untuk menentukan topik untuk kata-kata dalam dokumen. Kemudian penentuan topik pada setiap *placeholder* kata $[m,n]$ dilakukan dengan melakukan *sampling* sebuah topik tertentu $z_{m,n}$ dari distribusi multinominal $mult(\vec{v}_m)$. Terakhir, suatu kata tertentu $w_{m,n}$ dihasilkan untuk *placeholder* kata $[m,n]$ dengan melakukan sampling dari distribusi multinominal $Mult(\vec{\varphi}_{z_{m,n}})$ [33]. Dari Gambar 2.6 tersebut kita bisa

menulis gabungan distribusi semua variabel yang diketahui dan yang tersembunyi yang diberikan oleh parameter *dirichlet* sebagai berikut:

$$p\left(\vec{w}_m, \vec{z}_m, \vec{\vartheta}_m, \Phi \mid \vec{\alpha}, \vec{\beta}\right) = p\left(\Phi \mid \vec{\beta}\right) \prod_{n=1}^{N_m} p\left(w_{m,n} \mid \vec{\varphi}_{z_{m,n}}\right) p\left(z_{m,n} \mid \vec{\vartheta}_m\right) p\left(\vec{\vartheta}_m \mid \vec{\alpha}\right) \tag{2.8}$$



Gambar 2.6 Model Grafis LDA [32]

2.2.7 Sistem Pengadaan Secara Elektronik (SPSE)

Di beberapa negara maju seperti di Amerika dan negara yang tergabung dalam Komunitas Eropa, tidak kurang dari 20% GDP mereka dialokasikan untuk pengadaan barang/jasa, sedangkan di Indonesia tiap tahunnya tidak kurang dari 30% APBN dialokasikan untuk pengadaan barang/jasa. Karena alokasi untuk pengadaan barang/jasa cukup besar, maka sistem pengadaan publik yang transparan, non diskriminasi, berkeadilan, efektif dan efisien sangat penting dalam penyelenggaraan pemerintahan yang baik. Salah satu isu dan permasalahan pokok dalam penyelenggaraan pengadaan publik yang diakui oleh berbagai kalangan baik dari masyarakat bahkan dari pemerintah adalah praktek diskriminatif, kecurangan, dan korupsi yang terjadi tidak hanya di negara berkembang seperti di dalam pengadaan pemerintah di Indonesia, tetapi juga diberbagai negara maju [34].

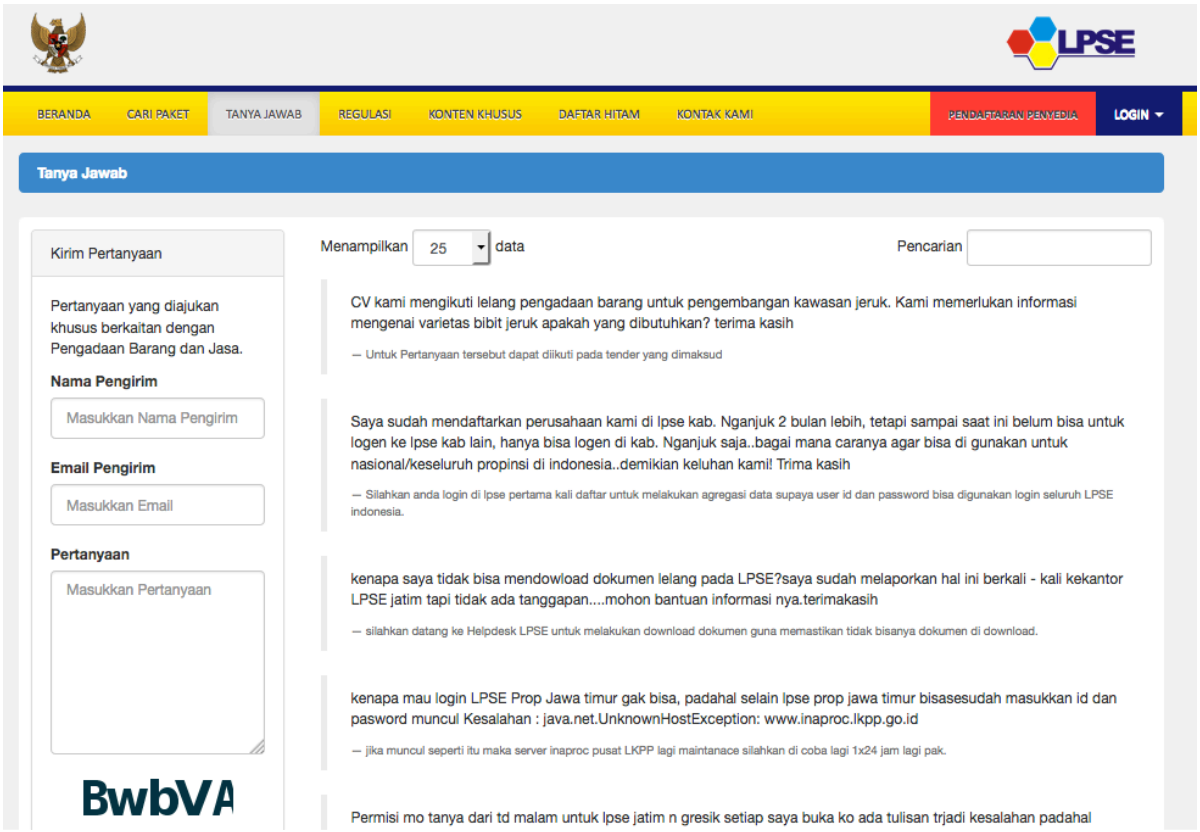
Untuk mendukung hal tersebut Pemerintah Republik Indonesia membentuk unit kerja Layanan Pengadaan Secara Elektronik (LPSE) yang menangani pengadaan barang/jasa. Hal tersebut diatur melalui Peraturan Presiden Republik Indonesia Nomor 54 Tahun 2010 pasal 111 ayat 2 berbunyi "K/L/I dapat membentuk LPSE untuk memfasilitasi ULP/Pejabat Pengadaan dalam melaksanakan Pengadaan Barang/Jasa secara elektronik" [35], yang operasionalnya secara teknis diatur melalui Peraturan Kepala LKPP Nomor 2 Tahun 2010 tentang Layanan pengadaan Secara Elektronik [36].

Untuk merealisasikan amanat tersebut, maka Lembaga Kebijakan Pengadaan Barang/Jasa Pemerintah (LKPP) mengembangkan aplikasi pengadaan elektronik (*e-procurement*) berbasis web yang diberi nama Sistem Pengadaan Secara Elektronik (SPSE). Aplikasi tersebut dipasang pada setiap LPSE K/L/D/I (Kementerian/Lembaga/Satuan Kerja Perangkat Daerah/Institusi) yang saat ini telah berjumlah 679 LPSE [37], dengan jumlah penyedia (perusahaan) terverifikasi sebanyak 307.960 perusahaan [38].

Saat ini versi terakhir SPSE adalah versi 4.2, pengembangan versi terbaru SPSE ini bertujuan untuk meningkatkan kualitas Pengadaan Barang/Jasa Pemerintah agar lebih transparan, akuntabel, dan kredibel. Aplikasi SPSE v.4.2 menyediakan beberapa fitur mayor baru, yaitu Lelang Konsolidasi, Lelang *Itemized*, Pengadaan Langsung (Pencatatan dan Transaksional), Penunjukan Langsung (Pencatatan dan Transaksional), Kontes (Pencatatan dan Transaksional), Sayembara (Pencatatan dan Transaksional), Swakelola (Pencatatan), serta Integrasi dengan Sistem Informasi Kinerja Penyedia (SIKaP). Selain itu, Aplikasi SPSE v.4.2 juga menyediakan beberapa fitur minor baru (perubahan tata letak *user interface*, penambahan/perubahan pesan *error*, dan penambahan/perubahan notifikasi untuk pengguna) serta perbaikan atas *error* (*bug*) yang muncul di versi sebelumnya [39].

Untuk mengakses aplikasi SPSE dapat menggunakan *browser* dengan format alamat URL: 'lpse.*nama-domain.*go.id' dimana *nama-domain* tersebut adalah nama domain resmi dari K/L/D/I, sebagai contoh: lpse.kemendagri.go.id, lpse.ntbprov.go.id, lpse.badungkab.go.id. Pengguna dari aplikasi SPSE dibagi menjadi 2 yakni penyedia (untuk peserta lelang) dan non-penyedia (seperti panitia, *helpdesk*, auditor, admin PPE).

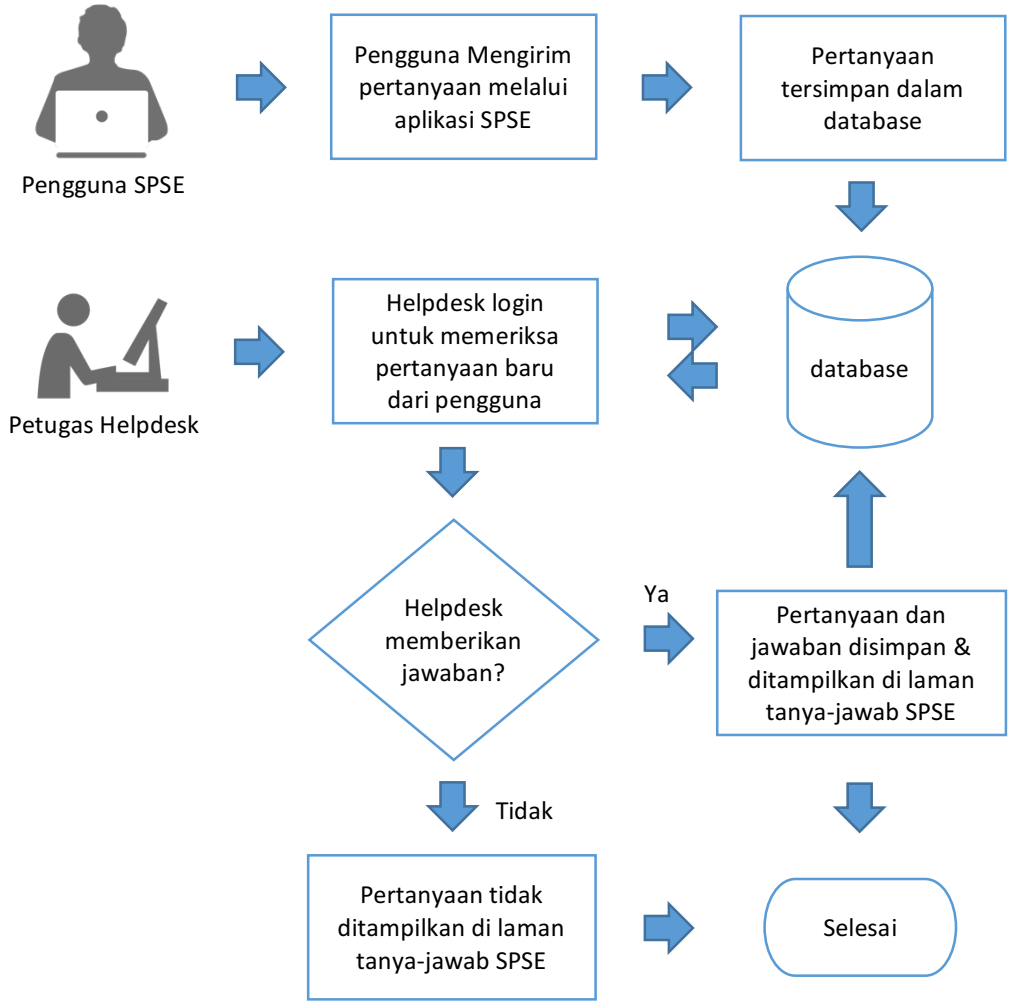
Kemudian untuk memudahkan penyedia dalam menyampaikan permasalahan terkait dengan sistem dan tata cara pengadaan melalui SPSE, pada aplikasi SPSE disediakan fasilitas layanan tanya jawab. Layanan tanya jawab pada aplikasi SPSE masih dikelola secara manual, dimana setiap LPSE akan menunjuk beberapa orang *helpdesk* yang bertugas untuk memeriksa dan menjawab pertanyaan yang diajukan oleh pengguna setiap hari. Menu layanan tanya jawab terdapat pada menu 'TANYA JAWAB' (lihat Gambar 2.7). Sedangkan alur penanganan pertanyaan dapat dilihat di Gambar 2.8.



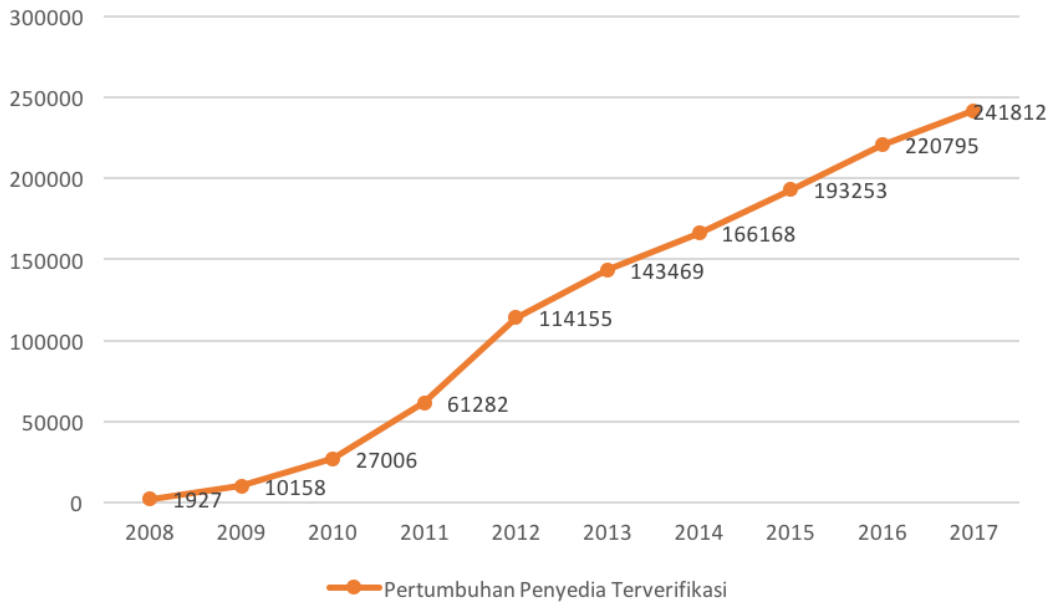
Gambar 2.7 Halaman Tanya-Jawab Pada Aplikasi SPSE

Pada sisi pengguna, pertumbuhan pengguna terverifikasi secara nasional terus mengalami peningkatan dengan rata-rata peningkatan 24181 pengguna setiap tahunnya (lihat Gambar 2.9). Seiring dengan pertumbuhan jumlah penyedia baru tersebut, maka penanganan pertanyaan dari pengguna ini menjadi penting. Hal ini dikarenakan penyedia baru tersebut kemungkinan belum terlalu mengenal sistem pengadaan elektronik melalui SPSE ataupun akan menjumpai permasalahan-permasalahan dalam mengoperasikannya. Sehingga akan mencari informasi

tentang SPSE secara langsung ke petugas *helpdesk* atau menanyakan melalui layanan tanya jawab ini. Akan tetapi tidak semua pertanyaan dari pengguna tersebut dijawab oleh *helpdesk.* Hal ini disebabkan oleh beberapa faktor yakni *helpdesk* tidak menguasai permasalahan yang ditanyakan atau *hepdesk* malas untuk memberikan jawaban karena pertanyaan yang disampaikan sering ditanyakan oleh pengguna-pengguna sebelumnya.



Gambar 2.8 Alur Penanganan Pertanyaan Pada Layanan Tanya-Jawab SPSE



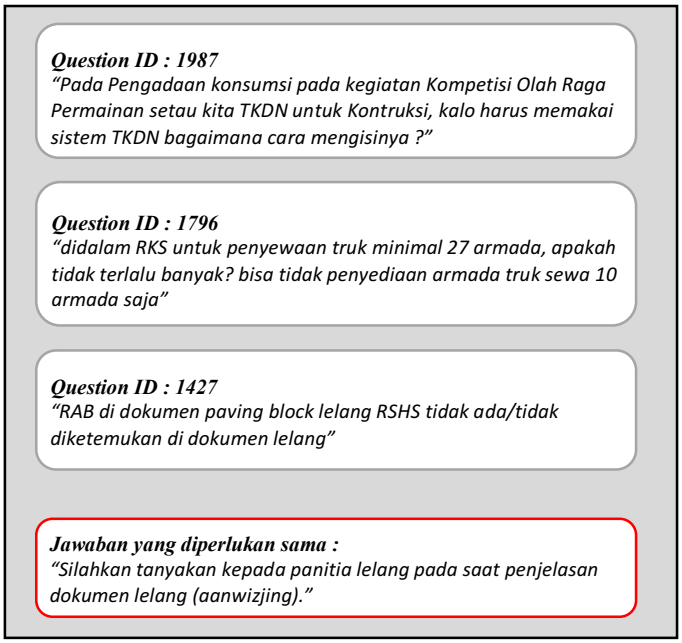
Gambar 2.9 Pertumbuhan Penyedia Terverifikasi pada LPSE Secara Nasional [38].

Sebagian besar pertanyaan yang diajukan pengguna adalah pertanyaan-pertanyaan yang sudah ditanyakan terlebih dahulu oleh pengguna lainnya atau membicarakan tentang topik yang sama. Sebagai contoh dari data yang diambil pada layanan tanya jawab LPSE Provinsi Jawa Barat, persentase pertanyaan yang relevan antara satu dengan lainnya cukup tinggi yakni mencapai 70%. Pertanyaan-pertanyaan yang relevan tersebut mempunyai tiga keadaan yakni:

1) Ditulis dengan kalimat yang hampir sama
2) Redaksi yang berbeda tetapi mempunyai maksud atau makna yang sama seperti contoh yang ditunjukkan pada Gambar 1.1.
3) Membicarakan topik yang sama dimana pertanyaan tersebut membutuhkan jawaban yang sama seperti dicontohkan pada Gambar 2.10. Pada contoh tersebut pengguna menanyakan pertanyaan dengan topik yang sama yakni menanyakan perihal pekerjaan lelang yang merupakan wewenang dari panitia lelang atau panitia pokja (kelompok kerja). Sehingga membutuhkan jawaban yang sama yakni: “Silahkan tanyakan kepada panitia lelang pada saat penjelasan dokumen lelang (aanwidzing)”.

Oleh karena itu hal tersebut sangat tidak efisien karena helpdesk akan disibukkan untuk menjawab pertanyaan-pertanyaan yang serupa dengan

pertanyaan-pertanyaan sebelumnya sehingga akan membuat bosan. Untuk itu diperlukan solusi yang dapat mempercepat proses kerja melalui otomatisasi dengan memanfaatkan arsip tanya-jawab yang ada melalui penyediaan fasilitas bantu seperti sistem rekomendasi (*recommender system*) atau sistem penjawab pertanyaan otomatis (*Question Answering System*-QAS) sehingga dapat meminimalkan campur tangan *helpdesk* dalam memberikan tanggapan.

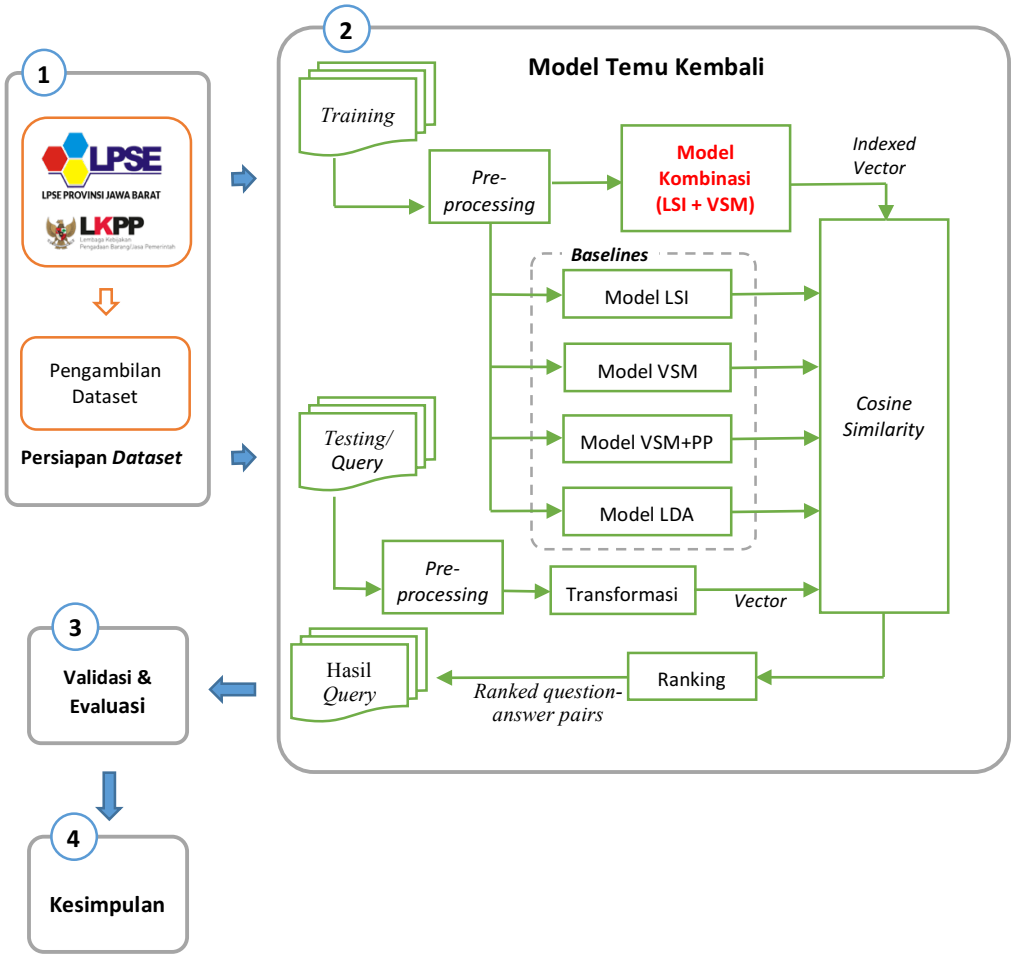


Gambar 2.10 Contoh Pertanyaan Dengan Topik Yang Sama

Halaman ini sengaja dikosongkan

BAB 3
METODOLOGI PENELITIAN

Alur penelitian dalam penelitian ini secara detail ditunjukkan oleh Gambar 3.1. Secara global penelitian ini dibagi menjadi empat tahap yakni: (1) tahap persiapan; (2) pengembangan model IR; (3) validasi & evaluasi hasil; (4) penarikan kesimpulan. Hasil penelitian pada model temu kembali informasi (IR) yang diajukan pada penelitian ini yakni kombinasi LSI+VSM akan dibandingkan dengan 4 *baselines* yakni LSI, VSM, VSM dengan pembobotan profesional (VSM+PP) dan LDA yang berdiri sendiri. Hal ini dilakukan untuk mengetahui sejauh mana model mampu meningkatkan kinerja dari model dasarnya yakni LSI dan VSM, selain itu dibandingkan pula dengan algoritma pemodelan topik lainnya yakni LDA.



Gambar 3.1 Metodologi Penelitian

3.1 Persiapan *Dataset*

Dataset yang digunakan dalam penelitian ini menggunakan arsip pertanyaan-jawaban pada layanan tanya-jawab aplikasi SPSE Provinsi Jawa Barat ditambah dengan data FAQ pada portal *e-procurement* LKPP. Semua proses pengambilan data dilakukan secara manual. Data dari LPSE Provinsi Jawa Barat dipilih karena LPSE tersebut adalah LPSE yang paling aktif dalam melayani pertanyaan pengguna sehingga mempunyai data yang cukup banyak yakni sebanyak 2.515 pasangan pertanyaan-jawaban[2]. Dari FAQ portal *e-procurement* LKPP diperoleh 95 pasangan pertanyaan-jawaban[3]. Setelah itu setiap pasangan pertanyaan dan jawaban akan diberi nomor ID. Kemudian dari *dataset* yang diperoleh dari LPSE Provinsi Jawa Barat dilakukan penyeleksian untuk memilih pertanyaan yang benar-benar mengandung pertanyaan atau keluhan dari pengguna, karena terdapat pula arsip yang hanya berisi ucapan terima kasih atas pelayanan ataupun saran yang ditujukan kepada pihak LPSE.

Pada penelitian ini *dataset* diacak kemudian dibagi menjadi dua yakni untuk keperluan *training* (*training dataset*) dan keperluan ujicoba (*testing dataset*). Untuk *testing dataset* dokumen yang diambil hanya pertanyaan sebagai *query* (dapat dilihat pada lampiran 1). Pada *dataset* untuk *training* setiap pertanyaan akan digabung dengan pasangan jawabannya menjadi satu dokumen. Hal ini dilakukan karena terkadang pertanyaan baru yang diinputkan oleh pengguna mempunyai kemiripan secara sintaksis dengan jawaban yang telah ada dan mempunyai relevansi terhadap pertanyaan baru tersebut.

Arsip pada layanan tanya-jawab pada aplikasi SPSE hanya menampilkan pertanyaan yang telah diajukan oleh pengguna sebelumnya beserta jawaban yang diberikan oleh *helpdesk*, tidak ada informasi lain seperti kapan pertanyaan diajukan atau dijawab, data perusahaan yang mengirim pertanyaan ataupun informasi tentang siapa yang memberikan jawaban.

---

[2] Sumber : https://lpse.jabarprov.go.id/eproc4/tanyajawab, diambil pada tanggal 3 Desember 2017
[3] Sumber : https://eproc.lkpp.go.id/faq, diambil pada tanggal 22 Februari 2018

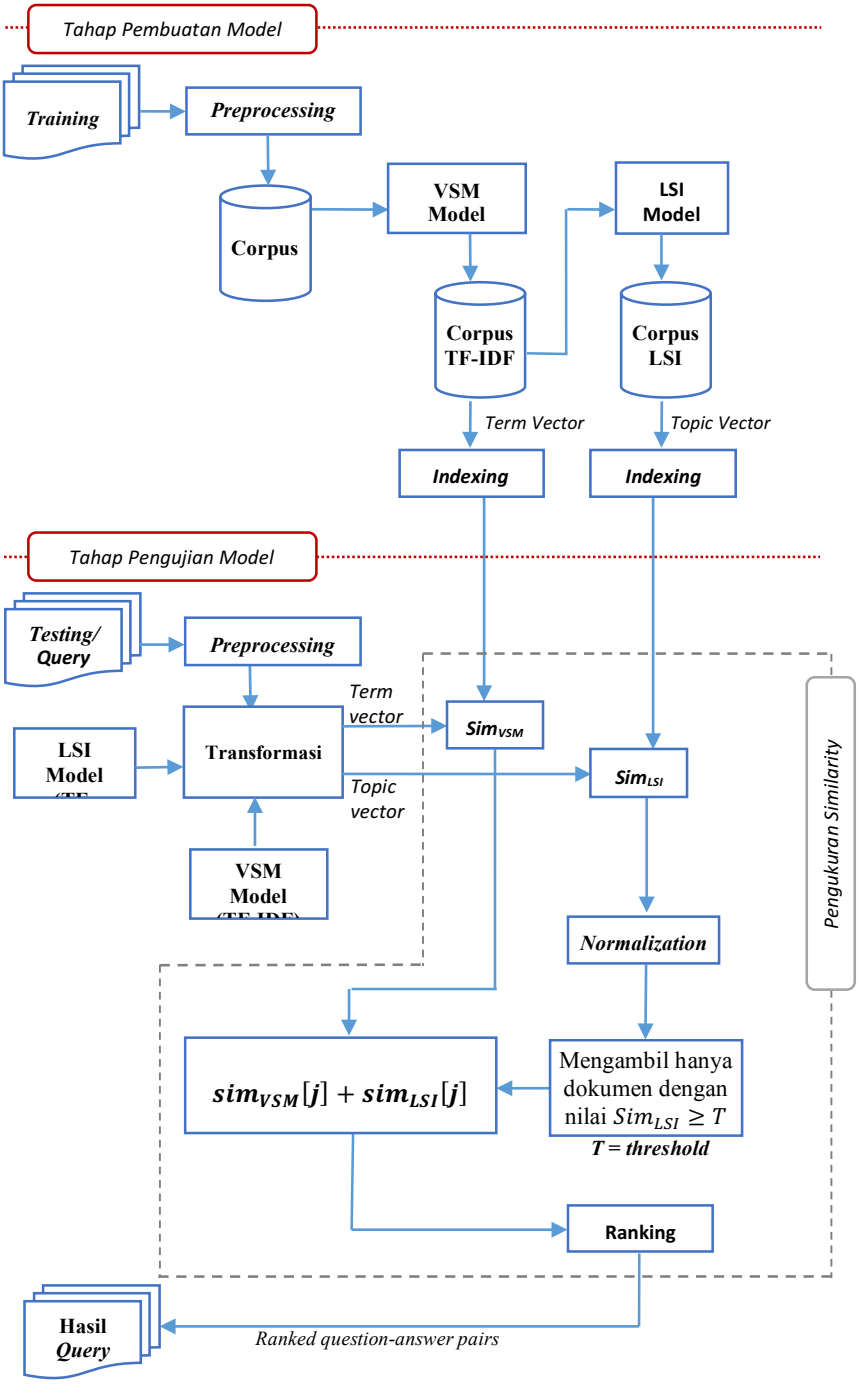
Tabel 3.1 Contoh Pertanyaan-Jawaban Pada LPSE Provinsi Jawa Barat

No.	Pertanyaan	Jawaban
1	Yth. LPSE, Kami dari PT. Inti Teknologi Komputasi ingin mendaftar. Kami sudah isi email kami, tapi sampai sekarang belum ada email konfirmasi dari LPSE, mohon bantuannya, terima kasih.	Cek di Spam.
2	Untuk mengikuti Pelatihan LPSE apa setiap minggu dilaksanakan, Hari dan waktu kapan? trm kasih	Setiap hari Rabu dan Kamis mulai pk. 09.00.
3	syaratuntuk pendaftaran LPSE apa sj ya dan ke mana dokumen hrs dibawa	Lihat Suarat Verifikasi pada Menu Special Content di Home website ini.

3.2 Pengembangan Model IR Kombinasi LSI dan VSM

Dalam model IR yang dikembangkan dalam penelitian ini menggunakan kombinasi dari algoritma pemodelan topik LSI dan algoritma VSM. Tujuannya adalah agar sistem IR mampu digunakan untuk mencari dokumen pada korpus yang memiliki kesamaan topik (*topic similarity*), kesamaan semantik (*semantic similarity*) dan kesamaan *keywords/term* (*term similarity*) dengan *query*. Hal ini dilakukan dengan cara memfilter dengan mengambil dokumen-dokumen yang memiliki kesamaan topik dan kesamaan semantik terlebih dahulu menggunakan LSI dengan nilai *cosine similarity* yang dibatasi oleh nilai ambang *T* (*threshold*). Kemudian setelah itu pada dokumen yang telah difilter tersebut dicari kesamaan *keywords* menggunakan algoritma VSM.

Dari kombinasi tersebut diharapkan model yang diajukan dapat mengambil kelebihan yang dimiliki oleh LSI (yakni kemampuan dalam menangani masalah semantik seperti sinonim serta kemampuan mencari dokumen yang membicarakan masalah topik yang sama) serta kelebihan yang dimiliki oleh VSM (yakni bekerja sangat baik dalam pencocokan *keywords/term*). Hal tersebut tentu mengakibatkan sistem memiliki kinerja yang lebih baik jika dibandingkan dengan algoritma dasarnya yakni LSI dan VSM.



Gambar 3.2 Arsitektur IR Berbasis Pemodelan Topik Menggunakan LSI & VSM

Arsitektur pengembangan model IR yang dikembangkan dalam penelitian ini dapat dilihat pada Gambar 3.2. Pengembangan model dibagi menjadi dua tahap yakni tahap pembuatan model dan tahap pengujian model. Tahap pembuatan model akan menghasilkan dua model yakni model LSI dan VSM.

Tahap pembuatan model:

1. *Preprocessing* untuk *dataset training*
2. Transformasi ke *bag of words* (BOW)
3. Pembuatan model VSM menggunakan pembobotan TF-IDF
4. Pembuatan model Topik menggunakan LSI
5. Melakukan pengindeksan.

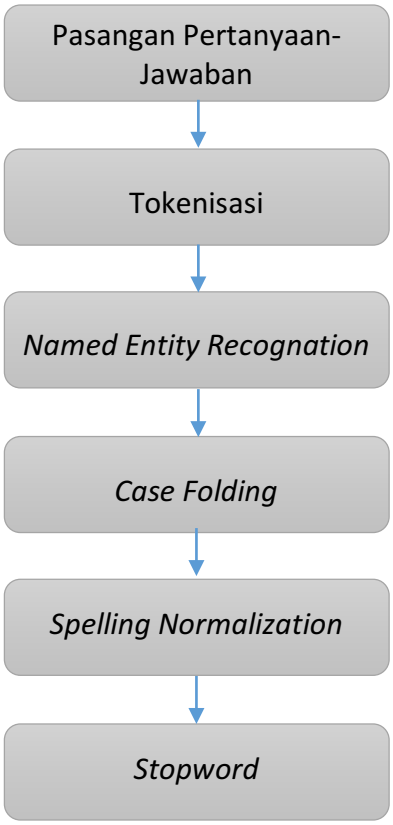
Tahap pengujian model:

1. *Preprocessing* untuk *dataset testing*
2. Transformasi ke dalam vektor masing-masing menggunakan model LSI dan VSM
3. Melakukan pengukuran kemiripan menggunakan *cosine similarity*
4. Meranking hasil pengukuran kemiripan

3.2.1 Pembuatan Model

3.2.1.1 Preprocessing

Sebelum membentuk model IR terlebih dahulu dilakukan *preprocessing* untuk memperbaiki dan menormalkan teks sehingga menjadi lebih terstruktur agar dapat diproses lebih lanjut. Hal ini dikarenakan pertanyaan-pertanyaan pada arsip tanya-jawab banyak begitu kotor, yakni terdapat banyak kata-kata yang penulisannya tidak baku serta terdapat banyak penggunaan kata-kata yang tidak perlu. Tahap *preprocessing* ini sangat penting karena akan mempengaruhi kinerja dari sistem IR. Pada *preprocessing* ini (ditunjukkan oleh Gambar 3.3) akan dilakukan proses tokenisasi, *named entity recognition, case folding, spelling normalization* dan *stopword*. Tahap ini bertugas untuk membuang *noise* yakni kata-kata tidak perlu atau tidak mempunyai peran penting terhadap makna suatu kalimat. Semakin sedikit *noise* yang ada maka kinerja sistem akan semakin baik. Penghapusan *noise* akan dilakukan oleh tahap *Named Entity Recognition* (NER) dan *stopwords*.



Gambar 3.3 Tahapan *Preprocessing*

Selain itu tujuan dari *preprocessing* ini adalah untuk melalukan normalisasi terhadap penulisan (*spelling normalization*) setiap kata pada dokumen teks agar sesuai dengan bentuk baku. Hal ini penting karena setiap *term* mempunyai independensi yang tinggi terhadap *term* lainnya, berbeda satu huruf sudah dianggap sebagai *term* yang berbeda. Sebagai contoh *term* 'password', 'fassword' dan 'paswort' walaupun *term* tersebut maksudnya sama akan tetapi oleh komputer akan dianggap tidak sama sehingga perlu dilakukan pengembalian kedalam bentuk bakunya yakni 'password'.

a) Tokenisasi

Tokenisasi adalah memecah sekumpulan karakter dalam suatu teks ke dalam satuan kata. Di sini yang perlu diperhatikan adalah bagaimana cara untuk membedakan karakter-karakter tertentu yang dapat diperlakukan sebagai pemisah kata atau bukan. Dokumen dipecah menjadi *term-term* serta membuang tanda baca (seperti: !()-[]{};:'"\,<>./?@#$%^&*_~) yang ada di dalamnya. Hal ini berguna untuk memudahkan melakukan manipulasi untuk setiap *term* tersebut agar sesuai

dengan yang diinginkan. Karakter selain huruf seperti simbol dan tanda baca akan dihapus dari teks, tujuannya untuk menghilangkan *noise* pada saat pengambilan informasi. Hasil tokenisasi berupa satuan kata selanjutnya disebut dengan *term.* Sebagai contoh kalimat "Buat kegiatan swakelola & paket Penyedia secara manual." proses tokenisasi akan menghasilkan 8 token yakni: ["Buat", "kegiatan", "swakelola", "paket", "Penyedia", "secara" dan "manual"].

b) Name Entity Recognition

Setelah dokumen dipecah menjadi token-token kemudian dilanjutkan dengan pemeriksaan apakah pada dokumen terdapat nama perusahaan menggunakan NER. *Named Entity Recognition* (NER) adalah subtugas dari ekstraksi informasi yang berusaha untuk mencari dan mengklasifikasikan nama entitas dalam teks ke dalam kategori yang telah ditentukan seperti nama-nama orang, organisasi, lokasi, ekspresi waktu, kuantitas, nilai moneter, persentase. Pada penelitian ini NER digunakan terbatas untuk membuang nama perusahaan (penyedia barang/jasa) karena entitas tersebut tidak mempunyai bobot yang penting dalam proses temu kembali informasi. Sebagai contoh pada kalimat "Saya dari CV. Rizky Utama, saya lupa user dan pasword saya, terima kasih", disini nama perusahaan (CV. Rizky Utama) akan dibuang dari kalimat tersebut pada sistem ini karena nama perusahaan tidak terlalu penting dalam proses IR. Hal yang perlu diperhatikan adalah obyek pertanyaannya yakni "lupa user dan password". Pada penelitian ini digunakan teknik NER berbasis aturan. Aturan yang digunakan masih sangat sederhana yakni seperti ditunjukkan oleh Tabel 3.2.

c) Case Folding

Pada *case folding* dilakukan pengubahan semua huruf dalam pertanyaan menjadi huruf kecil. Contoh kalimat "Yth LPSE PROP JABAR Minimal data kapasitas file yang berbentuk rhs BERAPA MEGABITS", hasil proses *case folding* akan menghasilkan: "yth lpse prop jabar minimal data kapasitas file yang berbentuk rhs berapa megabits".

Tabel 3.2 Aturan dalam Tahap NER

No.	Aturan
1	Memeriksa dokumen apakah mengandung *term* "PT", "CV" atau "UD" karena *term* tersebut digunakan untuk menyebutkan nama awal suatu perusahaan.
2	Memeriksa sampai maksimal tiga *term* setelah *term-term* pada poin 1 tersebut apakah awal katanya ditulis dalam huruf kapital. hal ini dilakukan karena kebanyakan nama perusahaan tidak lebih dari 3 kata/*term* dan secara formal huruf awalnya menggunakan huruf kapital.
3	Apabila *term-term* pada poin 2 (setelah *term* "PT", "CV" atau "UD") awal katanya ditulis dalam huruf kapital maka *term-term* tersebut akan dianggap sebagai nama perusahaan
4	*Term-term* yang dianggap sebagai nama perusahaan akan dihapus.

d) Spelling Normalization

Spelling normalization merupakan suatu cara perbaikan kata-kata yang salah eja atau disingkat dengan bentuk tertentu. Proses ini sangat penting dilakukan karena kata-kata pada *dataset* sangat kotor karena banyak mengandung kata-kata salah eja maupun singkatan. Hal tersebut terjadi karena kesalahan pengetikan maupun kebiasaan pengguna dalam melakukan penyingkatan suatu kata dalam kalimat sehingga perlu dilakukan pengembalian ke dalam bentuk bakunya. Misalnya kata "tidak" memiliki banyak bentuk penulisan seperti tdk, gak, nggak, enggak, kemudian kata "saja" menjadi aja, ja dan lain sebagainya. Selain bentuk penulisan dilakukan normalisasi juga pada kata-kata slang atau bahasa lokal yang sering digunakan. Proses ini dilakukan dengan menyusun kamus kata dan singkatan. Proses *Spelling normalization* pada penelitian ini menggunakan *library* aspell untuk bahasa pemrograman python dengan lisensi *free* dan tersedia dalam kode sumber terbuka (*open source*)[4]. Kamus pada aspell tersedia dalam banyak bahasa termasuk bahasa Indonesia, selain itu kita dapat mengadopsi istilah-istilah bahasa asing yang lumrah digunakan pada bahasa Indonesia. Sebagai contoh kata-kata seperti login, user, password, email, server dan kata-kata lainnya bisa

[4] dapat diambil secara bebas di laman http://aspell.net/

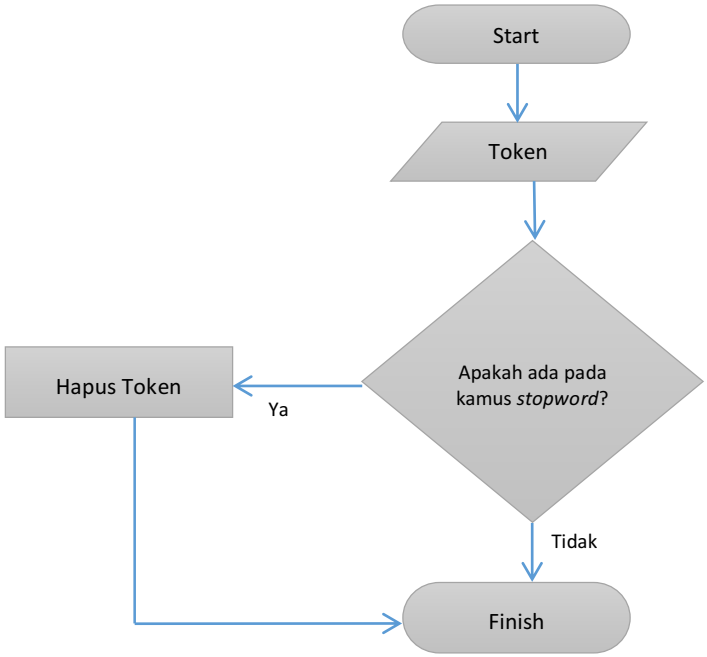
ditambahkan ke dalam kamus sehingga kata-kata tersebut tidak perlu dikoreksi oleh aspell. Contoh dalam kalimat: “gimana cara gnti password” akan dikoreksi menjadi “bagaimana cara ganti password”.

Aspell memiliki fitur penambahan kata ke dalam kamus, fitur ini menambahkan kata-kata ke dalam kamus berguna untuk mengadopsi istilah-istilah bahasa asing yang sering digunakan dalam penggunaan aplikasi SPSE. Contohnya seperti *login*, *logout*, *server*, *email*, *aanwidzing*, *account*, *training* dan sebagainya (untuk daftar lengkapnya dapat dilihat pada lampiran 2). Kata-kata yang akan diadopsi tersebut dapat ditambah atau di-*update* dalam kamus yang disimpan pada folder *home* komputer dengan nama file ‘.aspeel.id.pws’.

Fitur kedua dari aspell yakni mampu melakukan penyesuaian dalam pengoreksian kata berguna untuk memperbaiki kesalahan *library* Aspell dalam mengoreksi suatu kata. Contoh kata ‘utk’ seharusnya diganti menjadi ‘untuk’ tetapi oleh Aspell kata tersebut diganti menjadi ‘utak’. Contoh lainnya yakni kata ‘daptar’ seharusnya diganti menjadi ‘daftar’ tetapi oleh aspell kata tersebut diganti menjadi ‘datar’. Sehingga fitur ini sangat penting untuk digunakan supaya mendapatkan hasil pengoreksian kata yang lebih baik. Kata-kata yang akan disesuaikan tersebut dapat ditambah atau di-*update* dalam kamus yang disimpan pada folder *home* komputer dengan nama file ‘.aspeel.id.prepl’ (untuk daftar lengkapnya dapat dilihat pada lampiran 3).

e) Stopword

Penghapusan *stopword* adalah proses pembuangan *stopword* (kata yang sering muncul dan tidak dipakai, yang bertujuan untuk mengurangi volume kata sehingga hasil *retrieval* dapat lebih akurat karena *noise* akan direduksi. *Stopword* dapat berupa kata depan, kata penghubung, dan kata pengganti. Daftar *stopword* ini disimpan ke dalam file teks. Dokumen yang telah diproses sebelumnya berupa token akan diperiksa, jika sama dengan kata yang pada daftar *stopword* maka token tersebut akan dihapus. Proses ini ditunjukkan oleh Gambar 3.4.



Gambar 3.4 Proses Penghapusan *Stopword* Pada *Token*

3.2.1.2 Transformasi ke bag of words (BOW)

Dokumen yang telah diubah menjadi token pada hasil *preprocessing* akan ditransformasi ke dalam korpus menggunakan format *bag of words* (BOW) kemudian disimpan ke dalam *harddisk* untuk nantinya digunakan untuk membuat model VSM menggunakan algoritma TF-IDF.

Contoh pada kalimat yang telah melalui *preprocessing*:

1. "lpse" "jabar" "bisa" " agregasi " "lpse" "daerah" "tidak" "bisa" " agregasi "
2. "perusahaan" "belum" "bisa" "agregasi" "lpse" "jabar"
3. "cara" "agregasi" "lpse" "jabar"

pada 3 dokumen tersebut terdapat 9 *term* unik, maka matriks BOW-nya menjadi matriks seperti yang ditunjukkan oleh Tabel 3.3.

Tabel 3.3 Contoh Transformasi Dokumen-Dokumen ke BOW

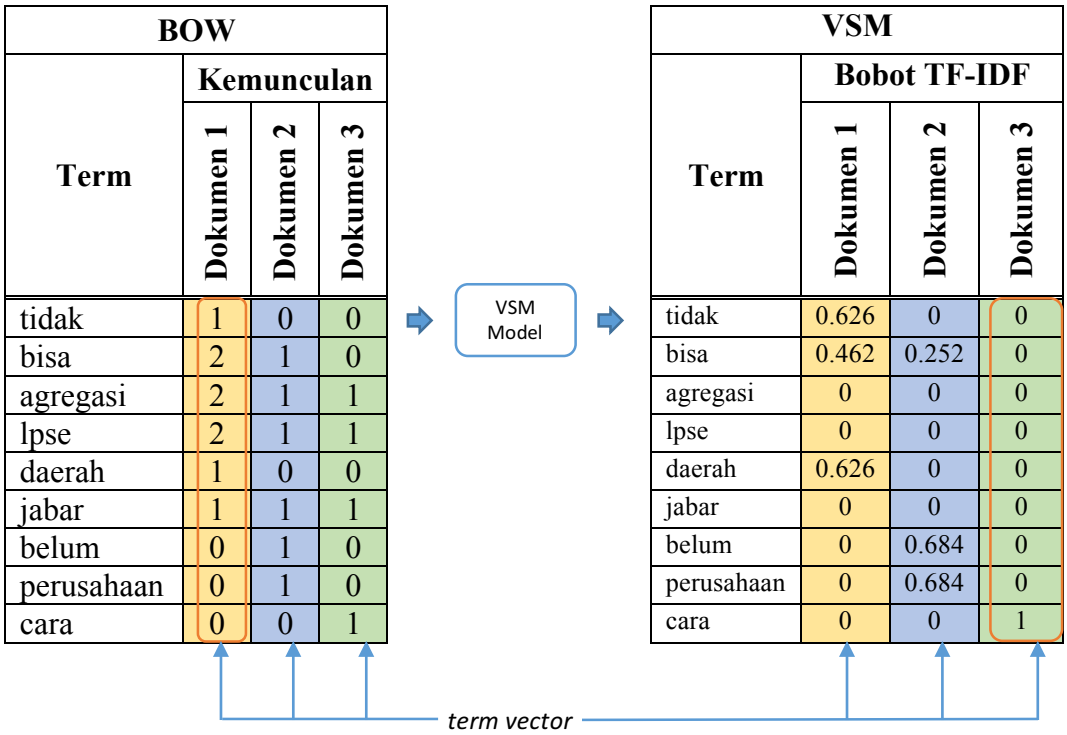
Term	Kemunculan		
	Dokumen 1	Dokumen 2	Dokumen 3
tidak	1	0	0
bisa	2	1	0
agregasi	2	1	1
lpse	2	1	1
daerah	1	0	0
jabar	1	1	1
belum	0	1	0
perusahaan	0	1	0
cara	0	0	1
Keterangan : = *non-zero entries* (bukan nol)			

3.2.1.3 Pembuatan model VSM menggunakan pembobotan TF-IDF

Pada tahap ini dilakukan pembuatan model VSM dari matriks BOW. Pembuatan model dilakukan dengan menggunakan pembobotan TF-IDF pada term-term dalam matriks BOW. Setelah model terbentuk maka model VSM tersebut dapat digunakan untuk mentransformasi korpus BOW menjadi korpus VSM. Dari contoh dokumen sebelumnya, hasil transformasi korpus BOW ke korpus VSM ditunjukkan oleh Gambar 3.5. Bentuk representasi dokumen dari korpus BOW sama dengan VSM yakni dokumen direpresentasikan dalam vektor *term* akan tetapi telah dilakukan pembobotan untuk setiap *term* pada setiap dokumen.

Dalam pembuatan model VSM, pembobotan TF-IDF dilakukan penghitungan frekuensi kemunculan *term* dalam suatu dokumen dan jumlah dokumen yang muncul dengan *term* yang sama. Dari proses penghitungan tersebut akan terbentuk model VSM yang nantinya digunakan untuk mentransformasi korpus hasil *preprocessing* menjadi korpus yang telah diberi bobot yang selanjutnya akan disebut sebagai ‘korpus TF-IDF’. Selain itu model VSM tersebut nantinya akan digunakan untuk mentransformasikan setiap dokumen *query* (pada

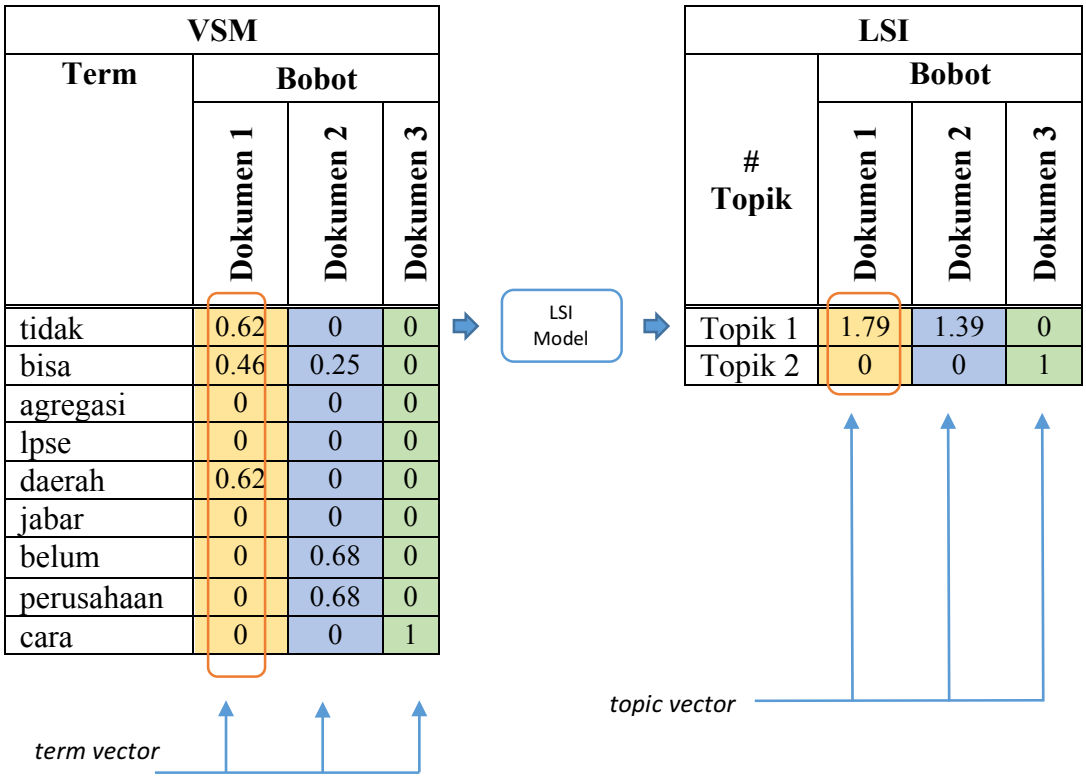
saat pengujian model) menjadi vektor *term* untuk diukur kemiripannya dengan dokumen pada korpus TF-IDF.



Gambar 3.5 Transfrmasi Korpus BOW ke Korpus VSM

3.2.1.4 Pembuatan model Topik menggunakan LSI

Kemudian selanjutnya pada tahap ini dilakukan pembuatan model LSI dari matriks VSM. Dalam pembuatan model LSI disertai dengan penambahan parameter yakni jumlah topik *k*. Untuk mencari nilai *k* terbaik harus dilakukan melalui eksperimen. Dalam penelitian ini model LSI yang akan dibentuk adalah model LSI dengan jumlah topik k dengan dengan nilai dari 5 sampai dengan 50. Setelah model terbentuk maka model LSI tersebut dapat digunakan untuk mentransformasi korpus VSM menjadi korpus LSI. Dari contoh dokumen sebelumnya, hasil transformasi korpus VSM ke korpus LSI dengan nilai $k = 2$ ditunjukkan oleh Gambar 3.6. Bentuk representasi dokumen dari korpus VSM berbeda dengan LSI yakni jika pada VSM direpresentasikan dalam vektor *term* maka pada LSI dokumen direpresentasikan dalam bentuk vektor topik.



Gambar 3.6 Transfrmasi Korpus VSM ke Korpus LSI

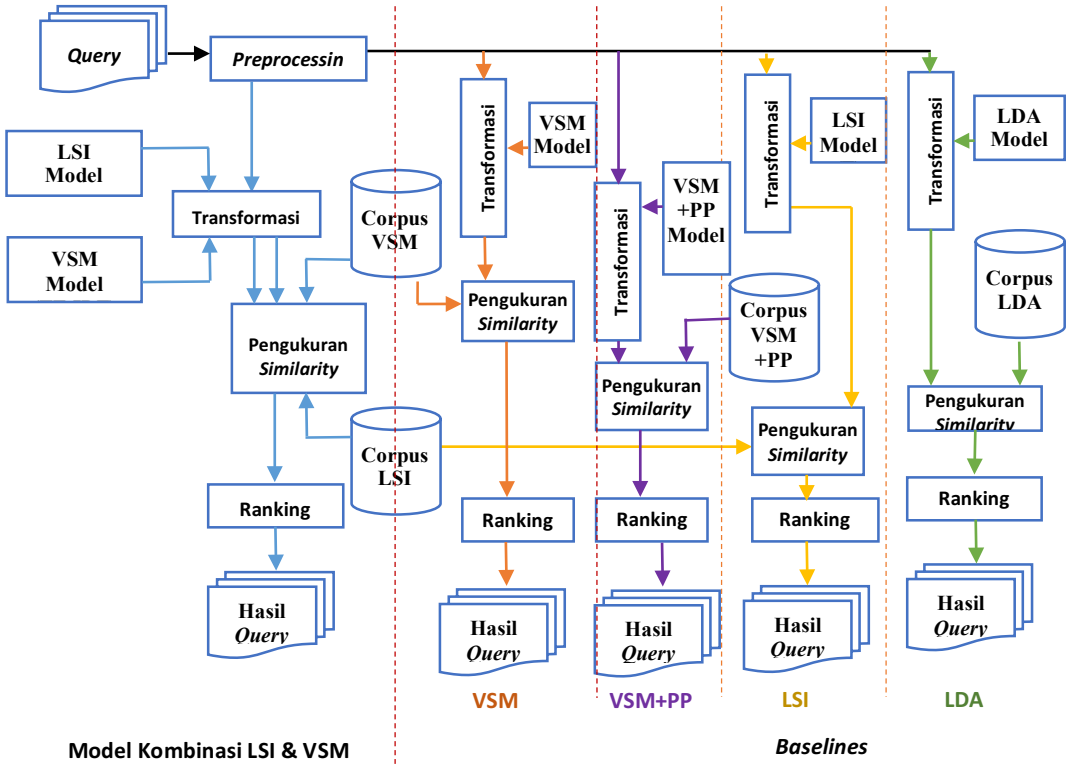
Sedangkan untuk hasil pembobotan setiap term terhadap setiap topik pada model LSI tersebut dapat diperoleh dari matriks *S* dari penghitungan menggunakan SVD pada persamaan (2.7). Hasil pembobotan dapat dilihat pada di bawah ini.

Tabel 3.4 Contoh Hasil Pembobotan Setiap *Term* Pada Setiap Topik dengan Jumlah Topik $k = 2$.

Term	**Bobot**	
	Topik 1	**Topik 2**
tidak	0.419	0
bisa	0.479	0
agregasi	0	0
lpse	0	0
daerah	0.419	0
jabar	0	0
belum	0.458	0
perusahaan	0.458	0
cara	0	1

3.2.1.5 Melakukan pengindeksan

Salah satu faktor yang mempunyai pengaruh besar terhadap sistem temu kembali (IR) ialah proses pengindeksan. Untuk mempersiapkan penghitungan kemiripan dengan *query*, maka perlu dilakukan pengindeksan untuk memasukkan semua dokumen yang ingin dibandingkan dengan *query* berikutnya. Tujuan lain pengindeksan adalah memungkinkan ditemukannya dokumen yang relevan dengan pertanyaan (*query*) dengan cepat dan tepat. Setelah pengindeksan selesai maka proses penghitungan *similarity* korpus terhadap *query* sudah siap untuk dilakukan.



Gambar 3.7 Proses Pengujian Model dan *Baselines*

3.2.2 Pengujian Model

Pengujian model akan dibagi menjadi dua tahap, pertama yakni tahap pengujian untuk menentukan parameter untuk menghasilkan kinerja model yang terbaik, kedua yakni pengujian untuk model *baselines* (model LSI, VSM, VSM+PP dan LDA) seperti yang ditunjukkan oleh Gambar 3.7. Pada model kombinasi, parameter yang akan diubah-ubah adalah parameter jumlah topik *k* pada model LSI-nya dan parameter nilai ambang (*threshold*) *T* pada proses penghitungan *cosine similarity*.

3.2.2.1 ***Preprocessing*** **untuk dataset** ***testing***

Tahapan *preprocessing* pada pengujian model sama persis dengan *preprocessing* pada tahap pembuatan model yakni akan dilakukan proses *named entity recognation, case folding,* tokenisasi*, spelling normalization* dan *stopword*. Output dari *preprocessing* ini adalah berupa token-token.

3.2.2.2 **Transformasi ke dalam vektor menggunakan model LSI dan VSM**

Agar bisa dilakukan pengukuran *similarity* maka selanjutnya dilakukan transformasi terhadap *query* ke dalam vektor masing-masing menggunakan model LSI (akan ditransformasi menjadi vektor topik) dan VSM (akan ditransformasi menjadi vektor *term*).

3.2.2.3 **Melakukan pengukuran** ***similarity*** **menggunakan** ***cosine similarity***

Dalam melakukan pengukuran *similarity* ada dua kali pengukuran yang akan dilakukan. Pengukuran *cosine similarity* pertama dilakukan antara *query* dengan korpus LSI dan korpus VSM secara terpisah. Oleh karena nilai *cosine similarity* pada korpus LSI (sim_{LSI}) menghasilkan nilai dari -1 sampai dengan 1, sedangkan pada korpus VSM (sim_{VSM}) bernilai antara 0 sampai dengan 1 maka perlu dilakukan normalisasi pada sim_{LSI}. Menggunakan rumus *min-max* seperti ditunjukkan oleh persamaan (3.1).

$$sim_{LSI}\left(d_j,q\right)\left[j\right]=\frac{sim_{LSI}\left(d_j,q\right)\left[j\right]+1}{2} \qquad (3.1)$$

dimana d_j adalah dokumen dengan indeks j, dan q adalah *query* dari pengguna.

Setelah itu dilakukan pengukuran *cosine similarity* kedua, yakni jika nilai $sim_{LSI}(d_j,q)$ lebih besar daripada nilai ambang (*threshold*) T. Proses penghitungan yang dilakukan adalah dengan menjumlahkan nilai $sim_{LSI}(d_j,q)$ dan $sim_{VSM}(d_j,q)$. Rumus untuk menghitung *cosine similarity* akhir (*sim*) ditunjukkan oleh persamaan (3.2) sedangkan persamaan untuk mencari nilai *threshold* T ditunjukkan oleh persamaan (3.3).

$$sim = \begin{cases} sim_{LSI}(d_j, q) + sim_{VSM}(d_j, q) & ; \text{jika } sim_{LSI}(d_j, q) > T \\ sim_{LSI}(d_j, q) & ; \text{selainnya} \end{cases} \quad (3.2)$$

$$T = \frac{c}{100} \max(sim_{LSI}) \quad (3.3)$$

dimana c konstanta dengan nilai antara 70 dan 100. Nilai c ini mengungkapkan bahwa kita mengambil c% dari nilai maksimum sim_{LSI} sebagai ambang batas T dengan nilai $T \geq 0{,}7$. Algoritma dari penghitungan *sim* ditampilkan dalam Gambar 3.8.

Algorithm 1 Cosine Similarity (*sim*) Calculation

```
corpus_LSI = corpus in the LSI model
corpus_VSM = corpus in the VSM model
q = query
c = choose constant between 50 and 100
for each document j in corpus_LSI do
    calculate sim_LSI(d_j, q)[j]
    #Normalization using Min-Max formula
    sim_LSI(d_j, q)[j] = (sim_LSI(d_j, q)[j] + 1) / 2
    #
end for
for each document j in corpus_VSM do
    calculate sim_VSM(d_j, q)[j]
end for
#Determine threshold T[j]
T[j] = (c/100) max(sim_LSI)
#
for each document j in corpus_LSI do
    if sim_LSI(d_j, q)[j] > T[j] then
        sim[j] = sim_LSI(d_j, q)[j] + sim_VSM(d_j, q)[j]
    end if
end for
sim = sort.descending(sim)
```

Gambar 3.8 Algoritma Penghitungan *Simlimarity* (*sim*)

3.2.2.4 Meranking hasil pengukuran kemiripan

Setelah penghitungan *similarity* dilakukan langkah terakhir adalah mengurutkan nilai *cosine similarity* (*sim*) dokumen-dokumen pada korpus secara *descending* yakni dari nilai tertinggi sampai terendah.

3.3 Evaluasi & Validasi

Untuk memastikan kebenaran hasil temu kembali informasi maka perlu dilakukan proses validasi. proses validasi arsip tanya-jawab yang relevan dengan *query* dilakukan secara manual yaitu dengan cara memasukkan nomor id dari setiap pasangan tanya-jawab pada korpus ke dalam daftar dokumen yang relevan dengan *query*. Proses tersebut dilakukan oleh Pegawai Negeri Sipil yang telah cukup lama bekerja pada unit kerja LPSE sehingga memahami SOP yang berjalan pada LPSE. Sehingga setiap *query* akan diketahui relevan atau mirip dengan pertanyaan yang mana saja pada korpus. Sehingga baru bisa evaluasi kinerja dari model IR yang telah dibentuk.

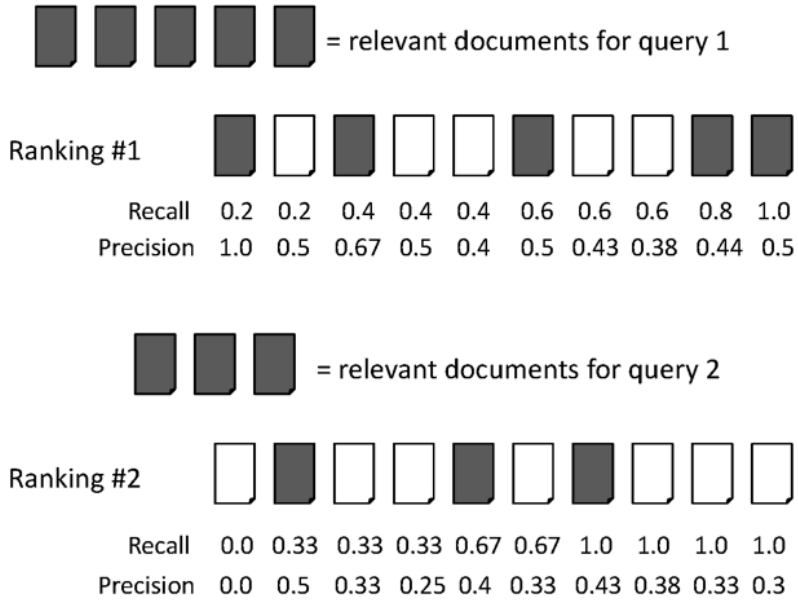
Teknik evaluasi yang digunakan dalam penelitian ini adalah teknik evaluasi untuk temu kembali informasi teranking (*ranked information retrieval*) yakni seperti *Precision at n* (P@n) dan *Mean Average Precision* (MAP) [21]. P@n adalah pengukuran presisi (*precision*) pada poin *cut-off n* teratas, P@n hanya mengambil hasil (*n*) teratas yang dikembalikan oleh sistem. Contoh jika n=1 maka penghitungan presisi dilakukan pada 1 hasil *query* teratas, jika n=3 maka penghitungan presisi dilakukan pada 3 hasil *query* teratas. Rumus untuk penghitungan presisi ditunjukkan oleh persamaan (3.4).

$$presisi = \frac{\left|\text{dokumen relevan} \cap \text{dokumen terambil sistem}\right|}{\text{dokumen terambil sistem}} \qquad (3.4)$$

Sedangkan MAP adalah nilai rata-rata presisi terhadap multi *query* yang dihitung menggunakan persamaan (3.5).

$$MAP = \frac{1}{n(Qu)} \sum_{q=1}^{n(Qu)} AP_q \qquad (3.5)$$

dimana *n*(Qu) adalah jumlah *query,* AP_q adalah rata-rata presisi untuk masing-masing *query*, rata-rata presisi adalah rata-rata P@n untuk setiap dokumen relevan pada posisi *n*.



Gambar 3.9 Contoh Hasil Suatu IR

Contoh penghitungan P@n untuk Gambar 3.9:

1. Untuk *query* 1

$$P@1 = \frac{1}{1} = 1$$

$$P@3 = \frac{2}{3} = 0{,}67$$

$$P@5 = \frac{2}{5} = 0{,}4$$

$$AP_1 = \frac{1{,}0 + 0{,}67 + 0{,}5 + 0{,}44 + 0{,}5}{5} = 0{,}62$$

2. Untuk *query* 2

$$P@1 = \frac{0}{1} = 0$$

$$P@2 = \frac{1}{2} = 0{,}5$$

$$AP_2 = \frac{0{,}5 + 0{,}4 + 0{,}43}{3} = 0{,}44$$

Sehingga dari 2 *query* tersebut dengan menggunakan persamaan (3.6) diperoleh nilai MAP:

$$MAP = \frac{AP_1 + AP_2}{2} = \frac{0{,}62 + 0{,}44}{2} = 0{,}53$$

3.4 Penarikan Kesimpulan

Pada tahap ini dapat diketahui pengaruh kombinasi LSI dan VSM dalam meningkatkan kinerja model dasarnya yakni model LSI dan VSM yang berdiri sendiri. Serta dapat diketahui sejauh mana model yang diusulkan dapat menjawab

rumusan permasalahan penelitian kemudian menganalisis kekurangan-kekurangan yang ada untuk keperluan pengembangan pada penelitian ke depan.

Halaman ini sengaja dikosongkan

BAB 4
HASIL DAN PEMBAHASAN

Bab ini membahas hasil penelitian Temu Kembali Informasi Berbasis Pemodelan Topik Menggunakan Kombinasi LSI dan VSM Pada Sistem Tanya-Jawab sesuai dengan tahapan yang sudah dijelaskan pada bab sebelumnya. Bab ini menyajikan seberapa jauh penggunaan pendekatan pemodelan topik kombinasi LSI & VSM sesuai untuk menyelesaikan permasalahan yang dipaparkan pada BAB 1.

4.1 Hasil Pengambilan Dataset

Statistik hasil pengambilan *dataset* yang dilakukan ditunjukkan oleh Tabel 4.1. Dari penyeleksian yang dilakukan didapatkan sebanyak 21 arsip berupa dokumen non pertanyaan/keluhan, sehingga jumlah *dataset* yang akan digunakan dari LPSE Provinsi Jawa Barat yakni 2.494 arsip/dokumen. Sedangkan untuk FAQ Portal *e-procurement* LKPP digunakan semuanya yakni berjumlah 95 arsip/dokumen. Sehingga jumlah total dokumen pada *dataset* berjumlah 2.589 dokumen. Dari *dataset* tersebut digunakan 2.539 dokumen untuk *training* sedangkan untuk ujicoba sebagai *query* digunakan 50 dokumen.

Tabel 4.1 *Dataset* yang Digunakan pada Penelitian

Sumber	
Layanan Tanya Jawab LPSE Prov. Jabar	2.494
FAQ Portal *e-procurement* LKPP	95
Jumlah Total	2589
Penggunaan	
Training Dataset	2539
Testing Dataset	50

4.2 Hasil *Preprocessing*

4.2.1 Hasil Tokenisasi

Contoh beberapa hasil tokenisasi pada dokumen *training* ditunjukkan oleh Tabel 4.2. Dari hasil tokenisasi dapat dilihat bahwa dokumen dipecah menjadi satuan kata dan semua simbol dan tanda baca sudah dihilangkan. Untuk selanjutnya setiap pecahan kata tersebut akan dinamakan sebagai token.

Tabel 4.2 Contoh Hasil Tokenisasi

No.	Dokumen Asli	Sesudah Tokenisasi
1	assalamu'alauikum wrwb saya mau tau CV mana saja yg sudah masuk di web ini dan sy punya CV bagaimana cara masuk ke sini	['assalamualauikum', 'wrwb', 'saya', 'mau', 'tau', 'CV', 'mana', 'saja', 'yg', 'sudah', 'masuk', 'di', 'web', 'ini', 'dan', 'sy', 'punya', 'CV', 'bagaimana', 'cara', 'masuk', 'ke', 'sini']
2	pa mau tanya apa USERID CV. AWAL dari pangandaran??????	['pa', 'mau', 'tanya', 'apa', 'USERID', 'CV', 'AWAL', 'dari', 'pangandaran']
3	konfirmasi email baru utk CV. Trie Kamulya & CV. Rukun Karya belum kami terima, mohon dikirim segera via email kami	['konfirmasi', 'email', 'baru', 'utk', 'CV', 'Trie', 'Kamulya', 'CV', 'Rukun', 'Karya', 'belum', 'kami', 'terima', 'mohon', 'dikirim', 'segera', 'via', 'email', 'kami']
4	Kami dari CV Tri Mulya Utama, untuk admin TMU_KONSULTAN tetapi password kami lupa, mohon bantuannya. terima kasih	['Kami', 'dari', 'CV', 'Tri', 'Mulya', 'Utama', 'untuk', 'admin', 'TMUKONSULTAN', 'tetapi', 'password', 'kami', 'lupa', 'mohon', 'bantuannya', 'terima', 'kasih']
5	apakah pt lang jaya makmur bersama sudah terdaftar jadi anggota lpse jabar, mohon info kekurangan data perusahaan kami, terima kasih	['apakah', 'pt', 'lang', 'jaya', 'makmur', 'bersama', 'sudah', 'terdaftar', 'jadi', 'anggota', 'lpse', 'jabar', 'mohon', 'info', 'kekurangan', 'data', 'perusahaan', 'kami', 'terima', 'kasih']

4.2.2 Hasil NER dan *Case Folding*

Hasil dari NER dan *case folding* dapat dilihat pada Tabel 4.3. Karena proses NER ini berbasis aturan, maka hasil yang diperoleh sangat ditentukan oleh aturan yang dibuat. Pada contoh dokumen nomor 1 (tidak terdapat nama perusahaan) tiga *term* setelah *term* "CV" menggunakan huruf kecil semua, maka tiga *term* setelah *term* "CV" tidak dihapus. Kemudian pada contoh nomor 2 sampai dengan nomor 4 (terdapat nama perusahaan), maka nama perusahaan tersebut yakni *term* pertama sampai maksimal *term* ketiga setelah *term* "CV" yang huruf awalnya adalah huruf kapital akan dihapus dan hanya menyisakan jenis perusahaannya saja. Tetapi karena aturan yang digunakan pada NER sangat sederhana maka proses NER

mengalami kendala jika nama perusahaan ditulis dengan huruf kecil semua. Jika nama perusahaan ditulis dengan huruf kecil semua maka nama perusahaan tersebut tidak terdeteksi sehingga tidak akan dihapus. Hal ini dicontohkan pada dokumen nomor 5.

Tabel 4.3 Contoh Hasil NER dan *Case Folding*

No.	Dokumen Asli	Sesudah NER + *Case Folding*	Hasil
1	assalamu'alauikum wrwb saya mau tau CV mana saja yg sudah masuk di web ini dan sy punya CV bagaimana cara masuk ke sini	['assalamu', 'alauikum', 'wrwb', 'saya', 'mau', 'tau', 'cv', 'mana', 'saja', 'yg', 'sudah', 'masuk', 'di', 'web', 'ini', 'dan', 'sy', 'punya', 'cv', 'bagaimana', 'cara', 'masuk', 'ke', 'sini']	Sesuai
2	pa mau tanya apa USERID CV. AWAL dari pangandaran??????	['pa', 'mau', 'tanya', 'apa', 'userid', 'cv', 'dari', 'pangandaran']	Sesuai
3	konfirmasi email baru utk CV. Trie Kamulya & CV. Rukun Karya belum kami terima, mohon dikirim segera via email kami	['konfirmasi', 'email', 'baru', 'utk', 'cv', 'belum', 'kami', 'terima', 'mohon', 'dikirim', 'segera', 'via', 'email', 'kami']	Sesuai
4	Kami dari CV Tri Mulya Utama, untuk admin TMU_KONSULTAN tetapi password kami lupa, mohon bantuannya. terima kasih	['kami', 'dari', 'cv', 'untuk', 'admin', 'tmu_konsultan', 'tetapi', 'password', 'kami', 'lupa', 'mohon', 'bantuannya', 'terima', 'kasih']	Sesuai
5	apakah pt lang jaya makmur bersama sudah terdaftar jadi anggota lpse jabar, mohon info kekurangan data perusahaan kami, terima kasih	['apakah', 'pt', 'lang', 'jaya', 'makmur', 'bersama', 'sudah', 'terdaftar', 'jadi', 'anggota', 'lpse', 'jabar', 'mohon', 'info', 'kekurangan', 'data', 'perusahaan', 'kami', 'terima', 'kasih']	Tidak Sesuai

4.2.3 Hasil *Spelling Normalization*

Contoh hasil *spelling normalization* disajikan pada Tabel 4.4, dari tabel tersebut dapat dilihat bahwa sebagian besar kesalahan-kesalahan penulisan dapat dikoreksi dengan benar. Akurasi dari hasil pengoreksian kata sangat bergantung pada lengkap atau tidaknya koleksi kamus pada *library* Aspell, kamus

‘.aspeel.id.pws’ serta kamus pada ‘.aspeel.id.prepl’. Seperti yang ditunjukkan pada dokumen nomor 5, kata ‘poin’ yang seharusnya ‘point’ belum diadopsi ke dalam kamus, sehingga kata tersebut akan dikoreksi menjadi kata yang salah yakni ‘pion’. Contoh lainnya ditunjukkan pada dokumen nomor 6, kata ‘*chating’* belum diadopsi ke dalam kamus, maka kata tersebut akan dikoreksi menjadi ‘canting’. Seharusnya kata tersebut tidak perlu dilakukan pengoreksian karena telah diketik dengan benar.

Selain itu Aspell juga mampu mengoreksi tidak adanya spasi antar kata tetapi terbatas hanya pada dua kata saja. Lebih dari itu *library* tersebut tidak mampu melakukan pengoreksian. Sebagai contoh ‘maumenanyakan’, ‘bisadiaktifkan’ dan ‘atasbantuannya’ berhasil dikoreksi dengan baik masing-masing menjadi ‘mau menanyakan’, ‘bisa diaktifkan’ dan ‘atas bantuannya’. Tetapi jika lebih dari dua seperti ‘dibukaterimakasih’ dan ‘userIDpasswodnya’ tidak dikenali oleh Aspell sehingga tidak menghasilkan output (*null*).

Tabel 4.4 Contoh Hasil *Spelling Normalization*

No.	Dokumen Asli	Sesudah *Spelling Normalization*
1	mohon dikirim passwrd dan user id yg baru kami lupa passwrd dan user id yang lama terima kasih	['mohon', 'dikirim', 'password', 'dan', 'user', 'id', 'baru', 'kami', 'lupa', 'password', 'dan', 'user', 'id', 'yang', 'lama', 'terima', 'kasih']
2	pegisian di porm penyedia nama user di isi p pak/bu.? mohon dibantu.. terimakasih	['pengisian', 'form', 'penyedia', 'nama', 'user', 'isi', 'pak', 'mohon', 'dibantu', 'terima', 'kasih']
3	Maaf saya maumenanyakan kenapa user ID yang abdullatifjamiltidak bisadiaktifkan atau dibukaterimakasih atasbantuannya	['maaf', 'saya', 'mau', 'menanyakan', 'kenapa', 'user', 'id', 'yang', 'bisa', 'diaktifkan', 'atau', 'atas', 'bantuannya']
4	gimana caranya saya lupa userIDpasswodnya waktu daftar lpse	['bagaimana', 'cara', 'saya', 'lupa', 'waktu', 'daftar', 'lpse']
5	pada poin kedua pakta integritas, nama APIP nya apa ? terima kasih	['pada', 'pion', 'kedua', 'pakta', 'integritas', 'nama', 'api', 'nya', 'apa', 'terima', 'kasih']
6	Supaya bisa chating waktu Aanwizjing caranya gimana?	['supaya', 'bisa', 'canting', 'waktu', 'aanwidzing', 'cara', 'bagaimana']
7	Admin Agency, pagi saya endan dari BKPPWIV, jabatan Helpdesk untuk informasi berkaitan dengan id dan fassword saya belum. mks	['admin', 'agency', 'pagi', 'saya', 'edan', 'dari', 'jabatan', 'helpdesk', 'untuk', 'informasi', 'berkaitan', 'dengan', 'id', 'dan', 'password', 'saya', 'belum', 'mks']

Hasil pada tahap *spelling normalization* ini juga dipengaruhi oleh tahap sebelumnya yakni NER. Keberhasilan NER dalam membuang nama entitas seperti nama perusahaan dan nama person akan membuat hasil pada tahap ini semakin baik. Hal ini dikarenakan kebanyakan nama entitas tidak terdapat dalam kamus Aspell sehingga akan menyebabkan nama entitas tersebut dikoreksi menjadi kata lain yang mungkin dapat menyebabkan makna kalimat berubah. Sebagai contoh pada dokumen nomor 7, karena pada penelitian ini NER masih terbatas untuk menghapus nama perusahaan bukan nama person maka nama person 'endan' yang tidak terhapus oleh NER akan dikoreksi menjadi 'edan' yang akan menjadi *noise* baru dalam dokumen. Tentunya hal tersebut dapat mengakibatkan tahap *preprocessing* ini secara keseluruhan tidak mampu membuang *noise* pada dokumen dengan sempurna.

Tabel 4.5 Contoh Hasil *Stopword*

No.	Dokumen Asli	Sesudah *Spelling Normalization*
1	mohon dikirim passwrd dan user id yg baru kami lupa passwrd dan user id yang lama terima kasih	['dikirim', 'password', 'user', 'id', 'baru', 'lupa', 'password', 'user', 'id', 'lama']
2	pegisian di porm penyedia nama user di isi p pak/bu.? mohon dibantu.. terimakasih	['pengisian', 'form', 'penyedia', 'nama', 'user', 'isi', 'dibantu']
3	Maaf saya maumenanyakan kenapa user ID yang abdullatifjamiltidak bisadiaktifkan atau dibukaterimakasih atasbantuannya	['kenapa', 'user', 'id', 'bisa', 'diaktifkan']
4	gimana caranya saya lupa userIDpasswodnya waktu daftar lpse	['cara', 'lupa', 'daftar', 'lpse']
5	pada poin kedua pakta integritas, nama APIP nya apa ? terima kasih	['pion', 'pakta', 'integritas', 'nama', 'api']
6	Supaya bisa chating waktu Aanwizjing caranya gimana?	['bisa', 'canting', 'aanwidzing', 'cara']
7	Admin Agency, pagi saya endan dari BKPPWIV, jabatan Helpdesk untuk informasi berkaitan dengan id dan fassword saya belum. mks	['admin', 'agency', 'edan', 'jabatan', 'helpdesk', 'informasi', 'berkaitan', 'id', 'password', 'belum']

4.2.4 Hasil Stopword

Tahap *stopword* ini adalah tahap akhir dari tahap *preprocessing* yang berperan dalam membuang kata umum yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna. Sehingga dapat mengurangi jumlah teks yang

akan diproses tanpa menghilangkan makna kalimat pada dokumen. Contoh hasil *stopword* disajikan pada Tabel 4.5. Hasil akhir token-token dari tahap *preprocessing* untuk selanjutnya disebut sebagai *term-term.*

Tabel 4.6 Perbandingan Statistik Dokumen Sebelum dan Sesudah *Preprocessing*

Dataset *Training*	
Jumlah Dokumen (D)	2.539
Statistik Sebelum *Preprocessing*	
Jumlah total *term* (t_{total})	102.723
Jumlah *term* unik (t_u)	7.844
Ukuran matriks BOW ($t_u \times D$)	7.844×2.539
Jumlah komponen matriks	19.915.916
Jumlah komponen matriks dengan data bukan nol (*non-zero entries*)	79.031
Kepadatan $\left(\frac{\textit{non-zero entries}}{\text{Jumlah Komponen Matriks}}\right)$	0,397%
Statistik Setelah *Preprocessing*	
Jumlah total *term* (t_{total})	50.916
Jumlah *term* unik (t_u)	2.105
Ukuran matriks BOW ($t_u \times D$)	2.105×2.539
Jumlah komponen matriks	5.344.595
Jumlah komponen matriks dengan data bukan nol (*non-zero entries*)	40.306
Kepadatan $\left(\frac{\textit{non-zero entries}}{\text{Jumlah Komponen Matriks}}\right)$	0,754%

Tabel 4.6 di atas mendeskripsikan statistik penghilangan *noise* pada dataset *training* yang dilakukan oleh tahap *preprocessing*. Tahap *preprocessing* ini mereduksi lebih dari setengah jumlah *term* sebelumnya dari 102.723 *term* menjadi 50.916 *term.* Sehingga *term* yang akan diolah menjadi lebih kecil, tentunya hal ini dapat lebih meringankan proses komputasi yang dilakukan oleh komputer. Hasil dari Tahap *preprocessing* akan disimpan menjadi korpus dalam format matriks *bag-of-words* (BOW). Jumlah *term* unik yang dihasilkan sebanyak 2.105 *term*, angka tersebut akan menjadi jumlah baris matriks BOW sedangkan jumlah dokumen akan menjadi jumlah kolomnya. Sehingga ukuran atau dimensi matriks BOW menjadi 2.105×2.539. Matriks BOW yang dihasilkan masih sangat *sparse* (banyak komponen matriks bernilai nol), hal ini dapat dilihat dari kepadatan matriks hanya mencapai 0,754%. Matriks yang sangat *sparse* akan mengakibatkan proses

penghitungan *cosine similarity* menjadi kurang akurat sehingga hal ini akan mempunyai dampak terhadap kinerja sistem IR yang akan dibuat.

Tabel 4.7 Contoh Hasil Transformasi Dokumen ke Dalam VSM Menggunakan Pembobotan TF-IDF

Contoh Dokumen	
Pertanyaan : LPSE Jabar, saat ini melihat daftar pelelangan yang memakai portal ini sangat banyak. Bisa tidak beberapa ULP tk II di jabar melelangkan di portal lain mis LPSE Depok, LPSE kota bogor, atau LPSE ITB. Jawaban : Bisa saja.	
Kamus *Term* (id:*term*)	
(10:lpse, 16:jabar, 17:ulp, 19:pelelangan, 31:bisa, 41:tidak, 76:daftar, 166:lain, 199:melelahkan, 200:bogor, 201:misa, 202:memakai, 203:portal, 204:depok)	
VSM	
Vektor Dokumen Sebelum Pembobotan TF-IDF (id, kemunculan)	**Vektor Dokumen Setelah Pembobotan TF-IDF (id, bobot)**
[(10, 4.0), (16, 2.0), (17, 1.0), (19, 1.0), (31, 2.0), (41, 1.0), (76, 1.0), (166, 1.0), (199, 1.0), (200, 1.0), (201, 1.0), (202, 1.0), (203, 2.0), (204, 1.0)]	[(10, 0.093), (16, 0.125), (17, 0.156), (19, 0.166), (31, 0.121), (41, 0.051), (76, 0.14), (166, 0.176), (199, 0.337), (200, 0.262), (201, 0.386), (202, 0.285), (203, 0.6), (204, 0.294)]

4.3 Hasil Pembuatan Model IR

4.3.1 Model VSM

Model VSM yang telah dibentuk akan digunakan untuk proses transformasi korpus ke dalam bentuk vector *term* baru yang telah memiliki bobot. Proses transformasi menggunakan model VSM ini tidak melakukan reduksi matriks seperti pada model LSI sehingga ukuran korpus TF-IDF sama seperti korpus aslinya yakni 2.105×2.539. Contoh hasil korpus ke dalam VSM menggunakan TF-IDF pada dataset *training* disajikan dalam Tabel 4.7. Pada tabel tersebut memperlihatkan bahwa *term* ‘lpse’ dan ‘jabar’ mempunyai bobot yang kecil walaupun mempunyai kemunculan yang lebih banyak pada dokumen tersebut yakni masing-masing 4 dan 2. Hal ini disebabkan karena *term* tersebut sering atau banyak muncul pada dokumen lainnya pada dataset *training.* Tetapi berbeda dengan *term* ‘portal’ yang mempunyai bobot tinggi dengan jumlah kemunculan sebanyak 2 kali

dalam dokumen. Hal tersebut dikarenakan *term* 'portal' tidak banyak muncul pada dokumen lainnya.

4.3.2 Model LSI

Model LSI baru bisa dibentuk jika korpus TF-IDF telah tersedia. Dalam membentuk model LSI ada parameter tambahan yakni jumlah topik *k,* dengan nilai *k* lebih kecil dari pada jumlah dokumen dan jumlah *term* unik. Untuk menentukan nilai *k* terbaik, maka dalam penelitian ini akan dibuat model LSI dengan jumlah topik *k* mulai dari 5 sampai dengan 50. Hal ini karena mempertimbangkan jumlah dokumen pada *training*-nya kecil yakni hanya 2.539. Model LSI akan mereduksi dimensi matriks pada VSM sesuai dengan jumlah topiknya. Jika *k*=6 maka dimensi VSM yang semula berukuran 2.105×2.539 akan direduksi menjadi 2.105×5, jika *k*=6 maka dimensi akan direduksi menjadi 2.105×6, demikian juga seterusnya sampai model LSI dengan *k*=50 dimensi VSM akan direduksi menjadi 2.105×50. Hal ini dapat dilustrasikan melalui Gambar 4.1.

Setelah model LSI terbentuk, selanjutnya model tersebut digunakan untuk mentransformasikan korpus hasil *preprocessing* menjadi korpus LSI (direpresentasikan dalam bentuk vektor topik). Selain itu model LSI tersebut nantinya akan digunakan untuk mentransformasikan setiap dokumen *query* (pada saat pengujian model) menjadi vektor topik untuk diukur kemiripannya dengan dokumen pada korpus LSI. Sebagai contoh jika dokumen *query* : "password belum dikirim", jika menggunakan model LSI dengan nilai *k*=5 maka dokumen *query* tersebut dengan menggunakan matriks pada Gambar 4.1 akan ditransformasikan menjadi vektor topik sebagai berikut:

$\overrightarrow{query} = [(\text{nomor topik, bobot topik})_1, (\text{nomor topik, bobot topik})_2,$
$(\text{nomor topik, bobot topik})_3, \dots, (\text{nomor topik, bobot topik})_n]$

$\overrightarrow{query} = \big[\big(0, (0.39 + 0.11 + 0.06)\big), \big(1, (-0{,}28 + 0.01 - 0.04)\big), \big(2, (-0.22 + 0.07 - 0.01)\big), \big(3, (0.3 + 0 + 0.01)\big), \big(4, (-0.12 - 0.18 + 0.02)\big)\big]$

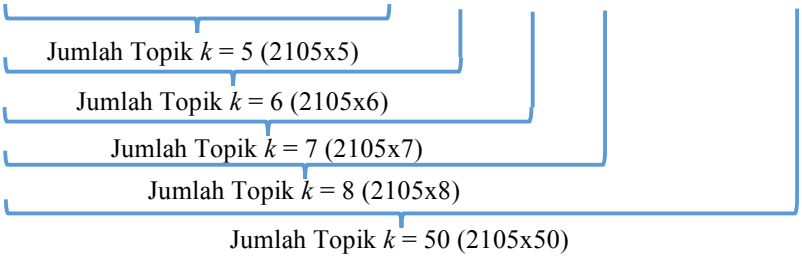
$\overrightarrow{query}_{topik} = [(0, 0.56), (1, -0{,}31), (2, -0{,}16), (3, 0.31), (4, -0.28)]$

dimana $\overrightarrow{query}_{topik}$ adalah vektor topik dari *query*. Untuk penghitungan bobot dilakukan dengan cara sebagai berikut:

bobot topik = $\sum_{t=1}^{n}$ (jumlah kemunculan term × bobot)

dengan t = urutan *term* dan n = jumlah *term* unik dalam dokumen.

id	*Term*	Bobot *Term* Untuk Setiap Nomor Topik									
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**....**	**49**
0	helpdesk	0,12	0.05	0.17	0.20	0.02	0.23	0.01	-0.07	....	0
1	belum	0.11	0.01	0.07	0	-0.18	0.37	0.08	-0.02	....	0
2	informasi	0.03	0.01	0.04	0.02	0.01	-0.02	-0.04	-0.03	....	-0.05
3	email	0.29	-0.08	-0.07	-0.27	0.34	0.16	0.38	0	....	-0.03
4	password	0.39	-0.28	-0.22	0.3	-0.12	-0.02	-0.02	0.07	....	0.05
5	permintaan	0	0	0	0	0	0	0	-3.51	....	0
6	dikirim	0.06	-0.04	-0.01	0.01	0.02	0.09	0.01	0.04	....	0.18
7	agency	0.01	0	0.03	0.01	0.01	-0.01	-0.01	-0.02	....	-0.08
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
2104	allowed	0	0	0	0	0	0	0	0	0	0



Gambar 4.1Ilustrasi Hasil Reduksi Dimensi Matriks VSM Menggunakan Pemodelan Topik LSI

4.4 Hasil Pengujian dan Validasi

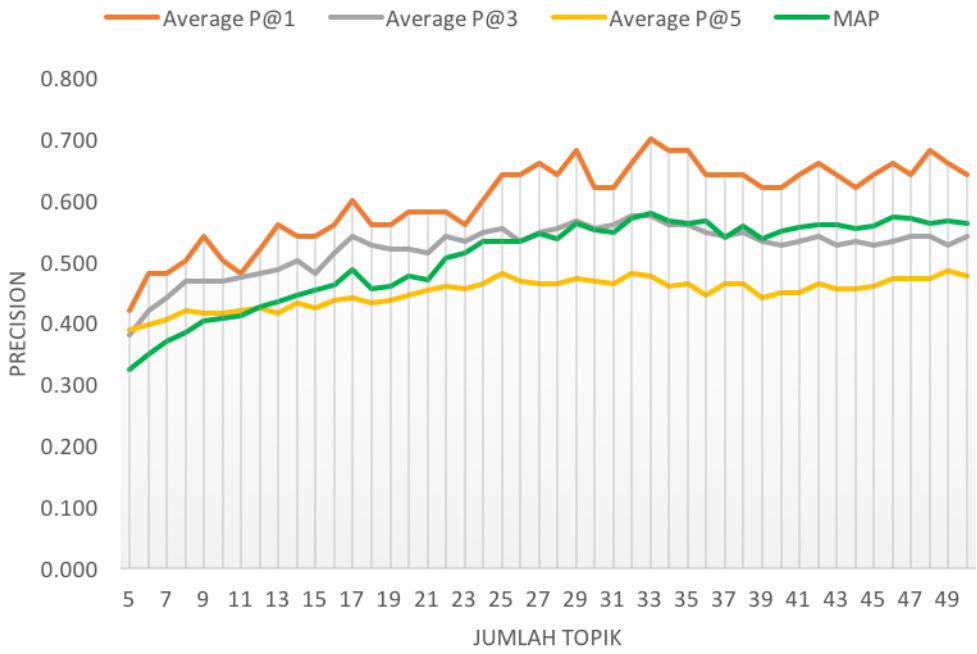
Pada penelitian ini, proses validasi arsip tanya-jawab yang relevan dengan *query* dilakukan secara manual yaitu dengan cara memasukkan nomor id dari setiap pasangan tanya-jawab pada korpus ke dalam daftar dokumen yang relevan dengan *query*. Koleksi dokumen pada korpus dengan *query* disajikan pada Tabel 4.8. Setelah daftar dokumen yang relevan diketahui, selanjutnya dilakukan pengukuran performansi menggunakan metrik *Precision at n* (P@n) dan *Mean Average Precision* (MAP) dengan menggunakan persamaan (3.4) dan (3.5).

Tabel 4.8 Daftar ID dokumen pada korpus yang relevan dengan *query*

No.	*Query*	ID dokumen pada korpus yang relevan dengan *query*
1	Utk memenuhi syarat verifikasi, apakah copy data perusahaan yang diperlukan bisa dikirimkan melalui pos/ kurir? Terima kasih.	813, 816, 1308, 1368, 1545, 1694, 2219, 2334, 2370, 2458
2	Yth LPSE, bagaimana caranya untuk mengikuti tender pengadaan pencetakan dan informasi jadwalnya? Terimakasih	9, 12, 51, 67, 94, 102, 135, 165, 174, 176, 177, 184, 200, 239, 257, 260, 313, 317, 319, 336, 339, 356, 357, 358, 376, 378, 425, 445, 449, 452, 469, 478, 480, 493, 579, 580, 593, 608, 620, 627, 650, 658, 659, 673, 693, 706, 707, 765, 772, 798, 800, 825, 849, 863, 882, 904, 922, 934, 938, 942, 964, 976, 1032, 1060, 1124, 1126, 1133, 1134, 1142, 1184, 1215, 1270, 1313, 1316, 1366, 1407, 1420, 1423, 1453, 1493, 1518, 1519, 1542, 1556, 1564, 1575, 1633, 1697, 1699, 1700, 1709, 1714, 1782, 1800, 1979, 2011, 2053, 2062, 2066, 2227, 2230, 2239, 2260, 2269, 2270, 2279, 2287, 2343, 2344, 2354, 2368, 2369, 2395, 2400, 2405, 2407, 2418, 2461, 2500, 2506
⋮	⋮	⋮
50	Kenapa untuk login ke lpse selalu saja ngga bisa dan ada tulisan kesalahan inaproc, maksudnya apa ya pa	84, 178, 186, 187, 188, 266, 288, 323, 326, 327, 348, 350, 351, 352, 353, 354, 403, 404, 427, 615, 978, 991, 1037, 1038, 1095, 1197, 1292, 1344, 1457, 1645, 2509, 2584

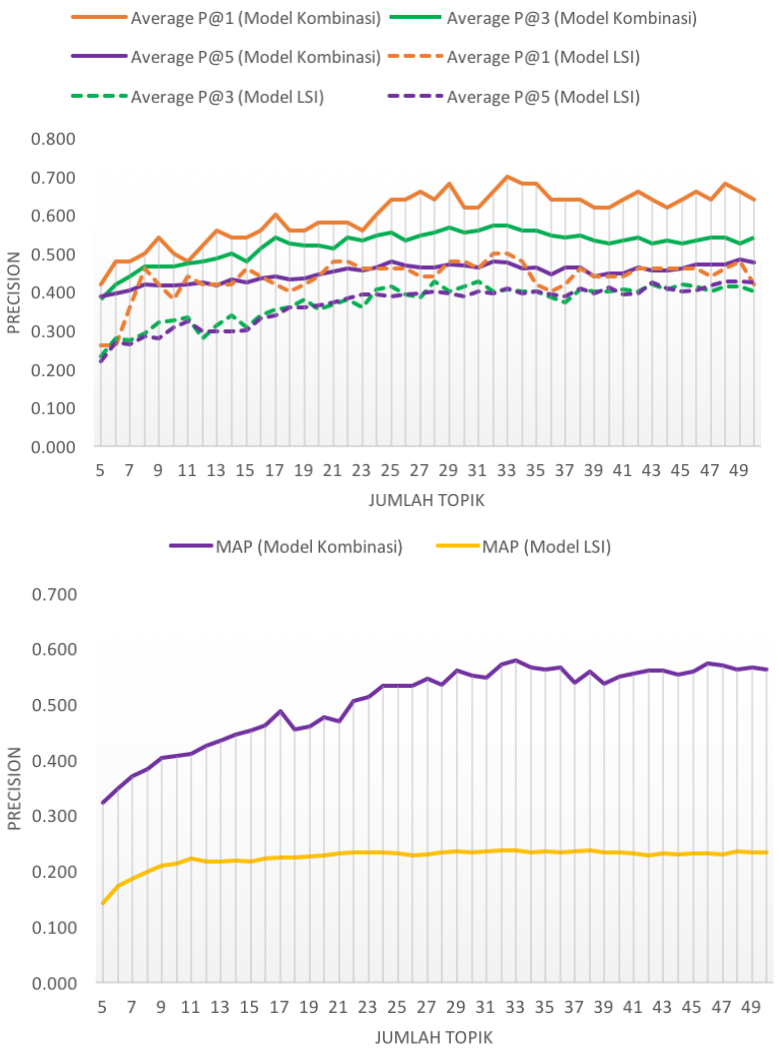
Dari Gambar 4.2 dapat diketahui bahwa dari 50 dokumen *query,* model kombinasi memperoleh hasil terbaik jika menggunakan jumlah topik k = 33 dan nilai konstanta c = 90 (*threshold* $T = \frac{90}{100} \times \max sim_{LSI}$). Hasil penelitian memperoleh rata-rata P@1 = 0,700, rata-rata P@3 = 0,573 rata-rata P@5 = 0,476 dan MAP = 0,579. Hasil dari grafik tersebut juga menunjukkan bahwa hasil eksperimen tidak naik secara linear seiring dengan kenaikan jumlah topik, sehingga dapat disimpulkan bahwa tidak ada cara terbaik untuk menentukan nilai k kecuali

dengan melakukan eksperimen. Kenaikan tinggi di awal kemudian, tetapi kemudian terjadi naik-turun dengan skala yang kecil.



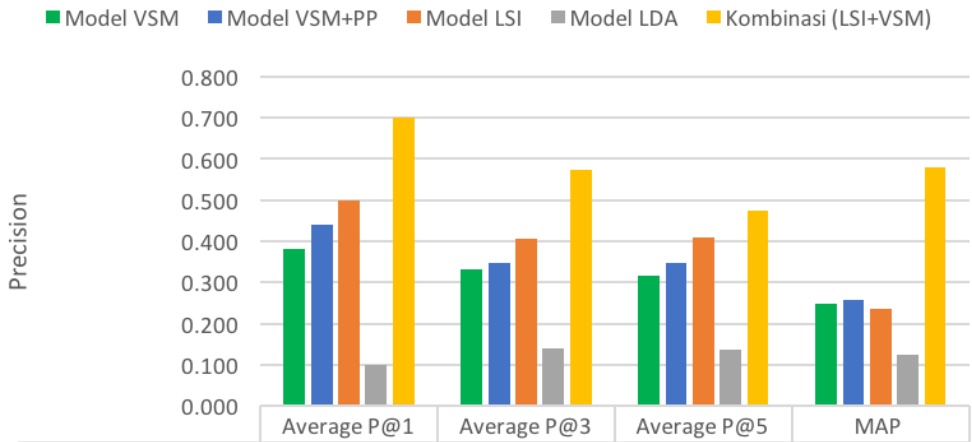
Gambar 4.2 Hasil Eksperimen Menggunakan Jumlah Topik LSI yang Berbeda pada Model Kombinasi LSI & VSM

Untuk mengetahui kemampuan kombinasi LSI+VSM dalam meningkatkan presisi hasil temu kembali informasi pada model LSI, maka dilakukan pengukuran semua metrik pada model LSI dan VSM yang berdiri sendiri. Perbandingan hasil model kombinasi dan model LSI murni pada setiap jumlah topik yang berbeda disajikan pada Gambar 4.3. Model kombinasi LSI+VSM mampu meningkatkan presisi dari model LSI murni memperoleh presisi yang lebih baik pada setiap jumlah topik yang digunakan. Hasil terbaik juga didapatkan ketika nilai k=33 dengan nilai rata-rata P@1, P@3 dan P@5 secara berturut-turut memperoleh 0.500, 0.407 dan 0.408, dengan nilai MAP=0.237. Sedangkan pada model VSM (baik menggunakan VSM biasa atau VSM+PP) diperoleh hasil yang lebih rendah daripada LSI murni (hasil evaluasi disajikan pada Gambar 4.4) yakni memperoleh rata-rata P@1, P@3 dan P@5 secara berturut-turut memperoleh 0.380, 0.333 dan 0.316, dengan nilai MAP=0.247 untuk VSM biasa. Sedangkan model VSM+PP memperoleh rata-rata P@1, P@3 dan P@5 secara berturut-turut memperoleh 0.44, 0.347 dan 0.348, dengan nilai MAP=0.258



Gambar 4.3 Perbandingan Hasil Eksperimen Menggunakan Jumlah Topik yang Berbeda Antara Model Kombinasi (LSI & VSM) dan LSI Murni

Perbandingan hasil presisi model kombinasi LSI+VSM dengan pemodelan topik lainnya dalam hal ini LDA juga disajikan pada Gambar 4.4. presisi pada Model kombinasi tersebut mengungguli jauh presisi pada model LDA. Model LDA hanya mendapatkan nilai rata-rata P@1, P@3 dan P@5 secara berturut-turut 0.100, 0.140 dan 0.126, dengan nilai MAP=0.126.



	Average P@1	Average P@3	Average P@5	MAP
Model VSM	0.380	0.333	0.316	0.247
Model VSM+PP	0.440	0.347	0.348	0.258
Model LSI	0.500	0.407	0.408	0.237
Model LDA	0.100	0.140	0.136	0.126
Kombinasi (LSI+VSM)	0.700	0.573	0.476	0.579

Gambar 4.4 Perbandingan Hasil Eksperimen Antara Model Kombinasi (LSI & VSM) dengan *Baselines*

4.5 Analisa Hasil

Dalam mencari dokumen tanya-jawab yang relevan, temu kembali informasi menggunakan kombinasi LSI+VSM ini bekerja dengan cara mencari dokumen-dokumen yang memiliki kesamaan topik (menggunakan model LSI) yang tinggi terhadap *query* terlebih dahulu, kemudian setelah itu di-*ranking* kembali berdasarkan kesamaan *term* menggunakan model VSM. Pada umumnya setiap dokumen tersusun dari beberapa topik kemudian dalam model LSI kontribusi setiap topik tersebut dinyatakan dalam bobot dalam suatu vektor topik. Sehingga kesamaan topik suatu dokumen dengan dokumen lainnya dapat diukur dengan persamaan *cosine similarity*.

Dari hasil eksperimen yang dilakukan diketahui bahwa hasil presisi terbaik diperoleh saat jumlah topik k=33, maka di sini setiap dokumen akan direpresentasikan ke dalam vektor topik berdimensi 33. Bobot *term* (10 *term* dengan bobot tertinggi) untuk beberapa topik pada Model LSI tersebut disajikan pada Tabel 4.9 untuk hasil lengkap keseluruhan topik dapat dilihat pada lampiran 4. Dari tabel tersebut dapat diketahui topik-topik umum apa saja yang sering ditanyakan oleh pengguna SPSE. Sehingga dari model LSI yang dibuat dapat juga

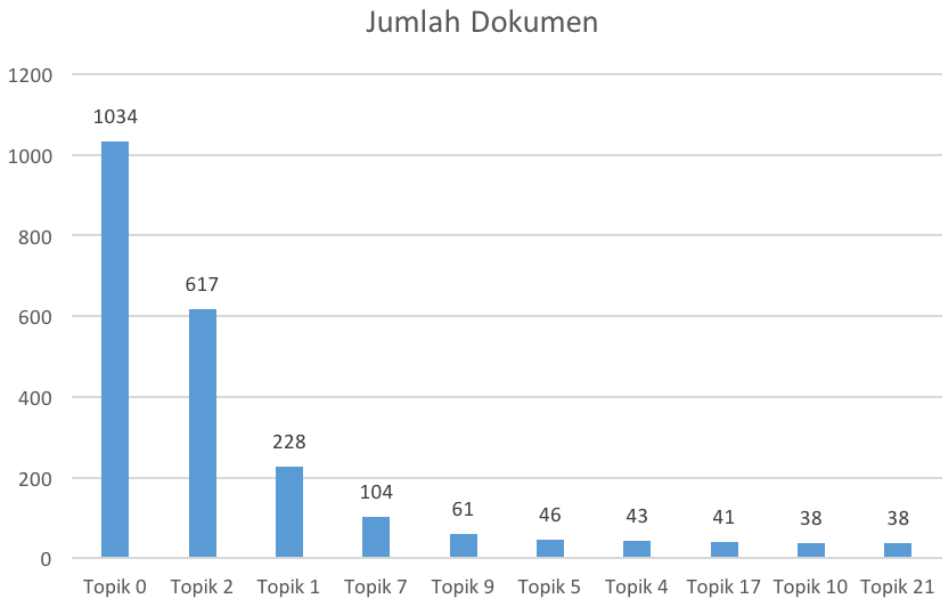
digunakan untuk mengklaster permasalahan yang ditanyakan oleh pengguna berdasarkan topik. Tentunya hal tersebut akan cukup membantu jika ingin membuat suatu FAQ untuk SPSE.

Tabel 4.9 Sepuluh *Term* Dengan Bobot Tertinggi Untuk Setiap Topik Pada Model LSI Dengan Jumlah Topik *k*=33

#	(*Keyword/Term*, Bobot)	Interpretasi Topik
Topik 0	0.392*"password" + 0.290*"email" + 0.290*"id" + 0.287*"lupa" + 0.274*"user" + 0.163*"login" + 0.142*"lpse" + 0.141*"bisa" + 0.137*"jabar" + 0.135*"verifikasi"	informasi tentang lupa user id, password dan email serta tentang proses verifikasi penyedia
Topik 1	-0.283*"password" + 0.273*"special" + 0.273*"content" + 0.267*"cara" + 0.260*"home" + 0.251*"menu" + -0.246*"lupa" + 0.237*"website" + 0.233*"mendaftar" + 0.222*"penyedia"	- Tata cara mendaftar - Lupa password
Topik 2	0.265*"lupa" + -0.243*"lelang" + 0.225*"password" + -0.212*"bisa" + -0.187*"tidak" + -0.173*"helpdesk" + 0.170*"special" + 0.170*"content" + -0.163*"jabar" + 0.154*"home"	- prosedur lupa password - informasi tentang pekerjaan lelang
Topik 3	-0.324*"id" + 0.299*"password" + -0.275*"email" + 0.272*"fasilitas" + 0.263*"lelang" + 0.257*"lupa" + -0.237*"user" + -0.203*"helpdesk" + 0.193*"halaman" + 0.147*"paket"	- informasi prosedur lelang - lupa password
Topik 4	-0.339*"email" + 0.286*"verifikasi" + 0.275*"login" + 0.235*"agregasi" + -0.227*"penggantian" + -0.203*"lelang" + 0.203*"mendaftar" + 0.183*"belum" + -0.173*"alamat" + 0.155*"inaproc"	- agregasi inaproc - penggantian email - cara mendaftar
Topik 5	0.369*"belum" + 0.323*"verifikasi" + -0.230*"helpdesk" + 0.196*"cek" + 0.195*"pendaftaran" + -0.186*"login" + 0.173*"online" + -0.162*"lupa" + -0.161*"kirim" + 0.160*"email"	Informasi pendaftaran dan verifikasi
. . . .	. . . .	. . . .
Topik 32	0.318*"npwp" + -0.289*"kualifikasi" + 0.240*"nama" + 0.201*"masuk" + 0.177*"akun" + 0.166*"jam" + 0.147*"penyedia" + -0.145*"login" + -0.138*"kirim" + -0.128*"surat"	Informasi data penyedia

Dari Tabel 4.9 tersebut dapat diketahui bahwa setiap nomor topik bisa terdiri lebih dari satu topik permasalahan dan di sisi lain setiap topik permasalahan bisa terdapat pada beberapa nomor topik. Sebagai contoh jika dokumen-dokumen pada korpus membicarakan tentang topik "lupa *user* dan *password*", maka dokumen-dokumen tersebut akan memiliki bobot yang tinggi pada topik 0, topik 1, topik 2, topik 6 dan akan memiliki bobot yang rendah pada nomor topik lainnya. Kemudian jika inputan *query* menanyakan permasalahan tentang lupa *user* dan *password*, maka *query* tersebut akan memiliki bobot yang tinggi pula pada topik 0, topik 1, topik 2, topik 6 dan akan memiliki bobot yang rendah pada nomor topik lainnya. Sehingga hal tersebut mengakibatkan representasi vektor topik dari dokumen-dokumen tersebut berdekatan dan akan mempunyai nilai *cosine similarity* yang tinggi.

Untuk kontribusi topik pada semua dokumen korpus dapat dilihat pada Gambar 4.5. Dari gambar tersebut dapat diketahui bahwa pertanyaan-pertanyaan yang diajukan oleh pengguna SPSE didominasi oleh topik tentang *user id*, *password*, email, proses pendaftaran dan verifikasi (topik 0, 1 dan 2), kemudian diikuti tentang topik informasi jadwal pelatihan SPSE dan topik pekerjaan lelang (topik 7 dan 9).



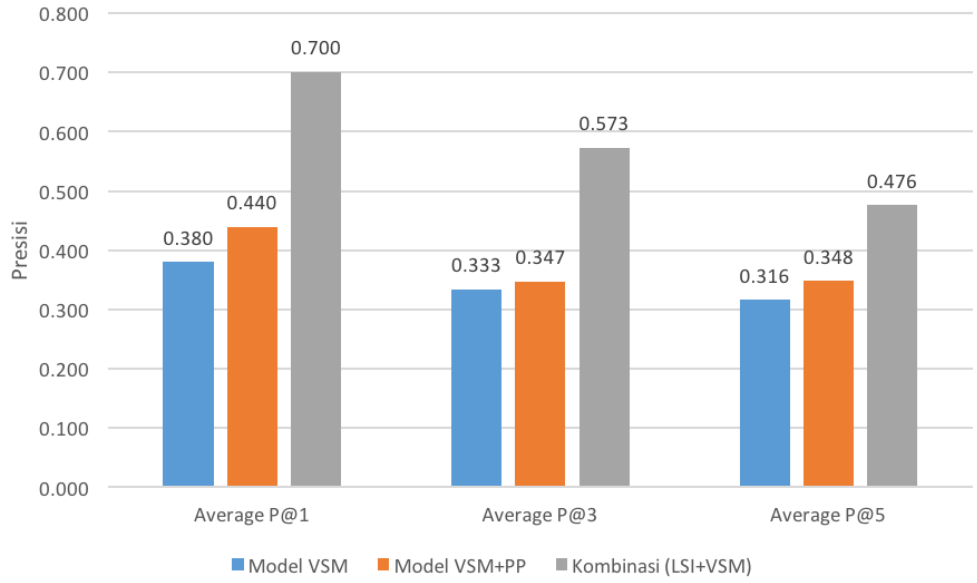
Gambar 4.5 Kontribusi 10 Nomor Topik Pada Semua Dokumen Korpus dengan Jumlah Dokumen Terbanyak.

Dengan memfilter dokumen-dokumen berdasarkan kesamaan topik tersebut, maka hal tersebut mampu meningkatkan kinerja temu kembali informasi. Hal ini dapat dilihat dari hasil yang diperoleh oleh model kombinasi, dimana model tersebut mampu memperoleh hasil yang lebih baik daripada model VSM baik yang VSM biasa maupun VSM+PP (seperti yang ditunjukkan oleh Gambar 4.4). Hal ini dikarenakan pemodelan topik LSI pada model kombinasi tersebut pada beberapa kasus mampu mengatasi permasalahan semantik dengan cukup baik. Seperti contoh hasil *query* yang ditunjukkan pada Tabel 4.10. Pada *query* nomor 2 menunjukkan model kombinasi mampu menangani masalah semantik seperti sinonim, dimana pertanyaan tentang "bagaimana cara ikuti tender" memiliki nilai *cosine similarity* tertinggi dengan pertanyaan "bagaimana caranya ikut lelang" pada korpus (id=863). Tetapi pada model VSM menghasilkan keluaran yang berbeda. Hal tersebut disebabkan karena model VSM bekerja dengan pencocokan *term,* sehingga memperoleh dokumen korpus dengan id=1094 sebagai dokumen paling relevan dengan nilai *cosine similarity* tertinggi. Pada kedua dokumen tersebut mengandung *term* "tender" dan "jadwalnya" yang diberi bobot yang tinggi oleh pembobotan TF-IDF.

Begitu juga halnya dengan *query* nomor 16, dimana makna "tanya apa USERID" pada *query* dan "tanya user name" pada korpus (id=1260) mempunyai makna yang sama maka oleh model kombinasi akan dianggap relevan karena mempunyai nilai *cosine similarity* yang tinggi. Sedangkan pada model VSM akan mencari dokumen yang mengandung *term* "pangandaran", karena *term* tersebut mempunyai bobot yang tinggi (0,893). Sehingga pencarian dokumen akan menghasilkan dokumen dengan id=372 sebagai yang paling relevan karena disana terdapat *term* "pangandaran" yang mempunyai bobot yang tinggi pula (0,766).

Tabel 4.10 Perbandingan Beberapa Hasil *Query* antara Model Kombinasi dan VSM

No. *Query*	***Query* (Q) dan Hasil *Query***
2	*Query* : Yth LPSE, bagaimana caranya untuk mengikuti tender pengadaan pencetakan dan informasi jadwalnya? Terimakasih Representasi dalam vektor *term*: Representasi dalam vektor *term*: [(informasi,0.345), (lpse,0.052), (tender,0.392), (cara,0.198), (mengikuti,0.299), (pengadaan,0.289), (jadwalnya,0.716)]
	Satu hasil *query* teratas (top-1) pada model kombinasi (id=863): P : bagaimana caranya ikut lelang di lpse??? J : Kepada Yth. Rohadi; Untuk informasi cara ikut lelang di LPSE, silahkan clik informasi yang ada di Menu Special Content
	Satu hasil *query* teratas (top-1) pada model VSM (id=1094): P : Mau tanya kalau tahap tidak ada jadwal itu maksudnya apa ya ? apakah tender belum dimulai atau bagaimana ? J : Lihat urutan jadwalnya pada paket tersebut. Representasi dalam vektor *term*: [(belum,0.168), (tidak,0.086), (paket,0.203), (tender,0.293), (tahap,0.375), (jadwal,0.311), (lihat,0.197), (maksudnya,0.524), (jadwalnya,0.536)]
16	*Query* : Pertanyaan/Query : pa mau tanya apa USERID CV. Awal dari pangandaran?????? Representasi dalam vektor *term*: [(id,0.186), (user,0.243), (cv,0.33), (pangandaran,0.893)]
	Satu hasil *query* teratas (top-1) pada model kombinasi (id=1260): P : mohon maaf panitia.boleh kami tanya user name utk CV Nopita.terima kasih.salam, J : Silahkan kirim e-mail ke helpdesk@lpse,jabarprov.go.id.
	Satu hasil *query* teratas (top-1) pada model VSM (id=372): P : bagaimana kalu mau buka pengumunan lelang yang kabupaten pangandaran? info nya masuk di lpse jabar J : Betul. Kabupaten Pangandaran masih menggunakan LPSE Provinsi Jawa Barat. Representasi dalam vektor *term*: [(informasi,0.185), (lpse,0.056), (jabar,0.151), (lelang,0.092), (pengumuman,0.212), (masuk,0.169), (buka,0.237), (pangandaran,0.766), (kaul,0.464)]



Gambar 4.6 Perbandingan Presisi Model VSM dan Model Kombinasi

Dengan demikian model kombinasi akan mampu meningkatkan kinerja dari model VSM (baik VSM biasa maupun VSM+PP) dalam proses temu temu kembali informasi. Pada penelitian ini, peningkatan presisi difokuskan pada peningkatan presisi pada P@1, P@3 dan P@5 karena kontribusi dari penelitian ini adalah sebagai studi awal untuk membangun sistem penjawab pertanyaan (QAS) atau sistem rekomendasi. Dimana sistem tersebut membutuhkan presisi P@1, P@3 dan P@5 yang cukup baik karena 5 jawaban teratas hasil *query* akan dijadikan sebagai kandidat jawaban yang akan diberikan kepada pengguna (penanya). Dari hasil eksperimen pada Gambar 4.6 menggambarkan peningkatan presisi yang signifikan oleh model kombinasi dari model dasarnya yakni VSM biasa ataupun VSM dengan pembobotan profesional (VSM+PP). Dari model tersebut didapatkan peningkatan presisi 32% untuk rata-rata P@1, 24% untuk rata-rata P@3 dan 16% untuk rata-rata P@5 untuk VSM biasa. Sedangkan untuk VSM+PP memperoleh peningkatan presisi 26% untuk rata-rata P@1, 22,7% untuk rata-rata P@3 dan 12,8% untuk rata-rata P@5. Dalam perbandingan antar model VSM yakni antara VSM biasa dan VSM+PP, hasil presisi yang diperoleh oleh VSM+PP lebih baik dari VSM biasa. Tetapi peningkatannya kecil yakni hanya 6% untuk rata-rata P@1, 1,3% untuk rata-rata P@3 dan 3,2% untuk rata-rata P@5.

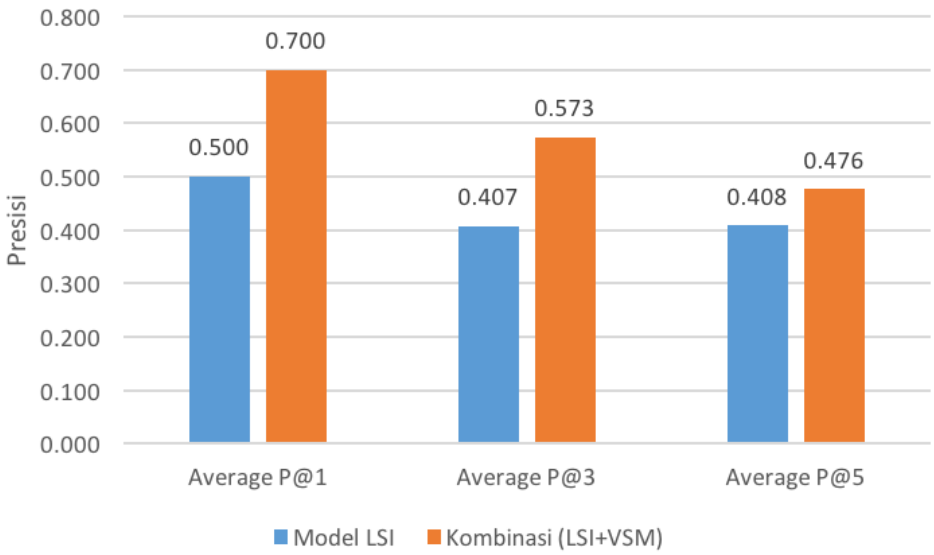
Di lain sisi, model LSI murni juga mempunyai kelemahan yakni jika pertanyaan atau *query* mengandung kata kunci dengan frekuensi kemunculan yang kecil pada korpus. Sebagai contoh yang disajikan pada Tabel 4.11, pada *query* nomor 11 kata kuncinya adalah "user id, password, diganti". Hasil dari LSI dan model kombinasi memang berisi topik tentang "user id", tetapi tidak memperhatikan kata kunci lain yang penting yakni *term* "diganti". Hal tersebut dapat dilihat pada lampiran 4, bahwa *term* "diganti" tidak mempunyai bobot yang berarti pada setiap nomor topik sehingga *term* tersebut diabaikan. Maka disini model VSM pada model kombinasi berperan penting dalam mengatasi masalah tersebut. Setelah mencari dokumen pada korpus dengan kesamaan topik oleh model LSI, maka selanjutnya dicari kesamaan *term* penting menggunakan model VSM. Sehingga model kombinasi akan menghasilkan dokumen dengan id=1555 sebagai dokumen yang paling relevan karena memiliki kesamaan topik dan kesamaan kata kunci dengan *query.*

Demikian halnya pada *query* nomor 20, dimana kata kuncinya adalah *term* "mendaftar" dan "SBU". Hasil dari LSI dan model kombinasi berisi topik tentang "mendaftar", tetapi mengabaikan kata kunci lain yang penting yakni *term* "SBU" karena *term* tersebut tidak memiliki bobot pada semua nomor topik model LSI (lihat Tabel 4.9). Dengan melakukan pencarian kata kunci "SBU" oleh model kombinasi pada dokumen dengan topik tentang "mendaftar" maka akan diperoleh hasil pencarian yakni dokumen korpus dengan id=565. Dimana pada dokumen tersebut memiliki kesamaan topik tentang "mendaftar" dan memiliki kesamaan kata kunci "SBU" dengan *query*.

Tabel 4.11 Perbandingan Beberapa Hasil *Query* antara Model Kombinasi dan LSI Murni

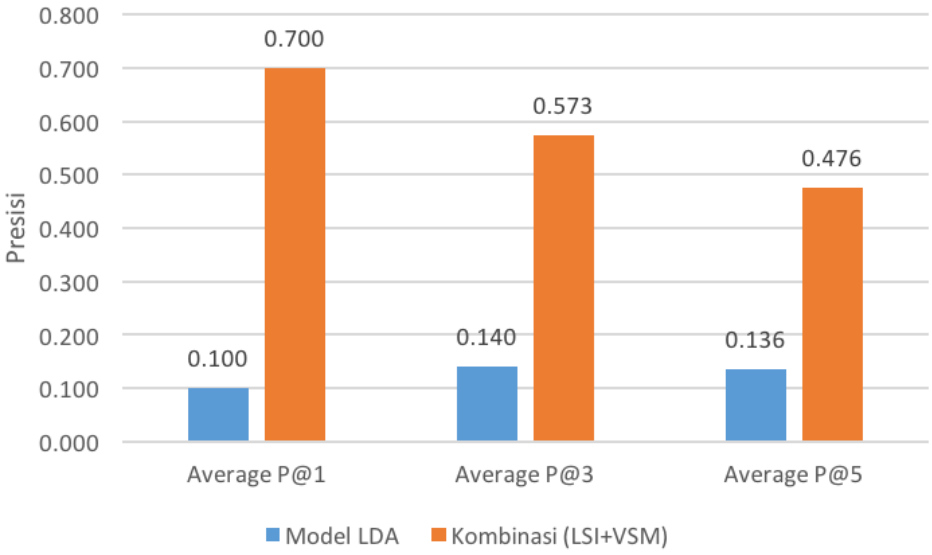
No. *Query*	***Query* (Q) dan Hasil *Query***
11	*Query* : Pertanyaan/Query : Kepada Yth LPSE PROP Jabar Kami dari CV Kresna Mulya mohon penjelasan apakah User ID dan passward LPS bisa diganti, kalau bisa gimana caranya. Sekian dan terima kasih Representasi dalam vektor *term*: [(password,0.165), (id,0.15), (lpse,0.105), (jabar,0.142), (user,0.196), (bisa,0.274), (diganti,0.551), (cara,0.2), (cv,0.266), (penjelasan,0.305), (sekian,0.547)]
	Satu hasil *query* teratas (top-1) pada model kombinasi (id=1555): P : user id perusahaan mendaftarke LPSE JABAR tahun sebelumnya itu bisa diganti atau tetap saja, karena lupa used dan passwod J : User ID tidak dapat diganti, silahkan tanya User ID Anda kepada Helpdesk di nomor 022-2536093. Representasi dalam vektor *term*: [(helpdesk,0.139), (password,0.098), (id,0.268), (lpse,0.031), (jabar,0.084), (user,0.349), (bisa,0.081), (lupa,0.139), (tidak,0.069), (perusahaan,0.116), (diganti,0.654), (nomor,0.193), (mendaftar,0.145), (used,0.489)]
	Satu hasil *query* teratas (top-1) pada model LSI (id=1906): P : yth lpse prov jabar, apakah user ID yg digunakan hanya satu pada LPSE? apakah ID CAU yang saya daftarkan di lpse.depkeu.go.id bisa juga di pakai di lpse.jabarprov.go.id? mohon penjelasannya J : Tidak bisa, Anda harus mendaptar ke LPSE Provinsi Jawa Barat.
20	*Query* : Pertanyaan/Query : yth. LPSE JABAR.CV kami ingin mendaftar. tetapi belum ada SBU dan belum ikut asosiasi apakah bisa mendaftar? Biaya pembuatan SBU dan asosiasi berkisar berapa. terimakasih. Representasi dalam vektor *term*: [(belum,0.179), (lpse,0.021), (jabar,0.056), (bisa,0.054), (mendaftar,0.193), (berapa,0.18), (cv,0.105), (sbu,0.423), (pembuatan,0.252), (biaya,0.252), (asosiasi,0.758)]
	Satu hasil *query* teratas (top-1) pada model kombinasi (id=565): P : untuk perusahaan yang belum memiliki sbu apa bisa mendaftar di lpse J : Bisa Representasi dalam vektor *term*: [(belum,0.27), (lpse,0.063), (bisa,0.327), (perusahaan,0.234), (mendaftar,0.291), (sbu,0.638), (memiliki,0.52)]
	Satu hasil *query* teratas (top-1) pada model LSI (id=2479): P : menindaklanjuti pemberitahuan di dalam harian umum Pikiran Rakyat pada tanggal 18 oktober 2010, yang diselenggarakan oleh Dispenda Prov Jabar untuk pekerjaan : 1.Pengadaan Proyektor; 2.Komputer Dekstop; 3.Notebook/Laptop. sapai hari ini kami belum bisa mendaftar karena belum muncul di halaman LPSE. kami mohon informasinya kenapa belum juga ditayangkan di LPSE sementara waktu terus berjalan. terima kasih J : Panitia akan mengumumkan pada waktunya. Cek kembali situs lpse.jabarprov.go.id.

Sehingga dengan keunggulan tersebut maka model kombinasi akan mampu meningkatkan tingkat presisi dari model LSI murni. Dari hasil eksperimen yang disajikan pada Gambar 4.7 didapatkan peningkatan 20% untuk rata-rata P@1, 16,6% untuk rata-rata P@3 dan 6,8% untuk rata-rata P@5.



Gambar 4.7 Perbandingan Presisi Model LSI dan Model Kombinasi

Jika dibandingkan dengan model *baselines* lainnya yakni model LDA, model kombinasi mampu jauh mengungguli model tersebut. Dari hasil eksperimen yang disajikan pada Gambar 4.8 didapatkan peningkatan 60% untuk rata-rata P@1, 43,3% untuk rata-rata P@3 dan 34% untuk rata-rata P@5. Hal ini disebabkan karena model LDA yang bekerja dengan sistem probabilistik mampu bekerja dengan baik pada dokumen dengan jumlah yang besar. Sedangkan jumlah data *training* pada penelitian ini kecil yakni hanya berjumlah 2.539 dokumen.



Gambar 4.8 Perbandingan Presisi Model LDA dan Model Kombinasi

Akan tetapi model kombinasi memiliki keterbatasan yakni mengabaikan urutan kata dalam kalimat, dimana pada beberapa kasus perubahan urutan kata pada kalimat akan mengubah makna kalimat itu sendiri. Hal yang sama juga terjadi pada model dasarnya yakni model VSM serta LSI. Contoh permasalahan urutan kata pada hasil *query* dapat dilihat pada Tabel 4.12. Pada *query* nomor 13 proses pencarian akan menghasilkan dokumen dengan id=647. Tetapi kedua dokumen tersebut tidaklah relevan, padahal keduanya memiliki topik yang sama tentang mencari alamat dan memiliki kesamaan *term* yang tinggi ("lpse", "jabar", "dimanakah"). Di sini urutan kata memiliki peran yang penting karena alamat yang ditanyakan oleh penanya adalah alamat LPSE Jawa Barat, bukan alamat LPSE kabupaten.

Contoh lainnya dapat dilihat pada *query* nomor 25 proses pencarian akan menghasilkan dokumen dengan id=1168. Hasil pencarian dokumen juga sama dengan di atas yakni tidak relevan dengan *query*, padahal keduanya memiliki kesamaan *term* yang tinggi ("lupa", "email", "id", "password") dengan permasalahan yang sama yakni tentang "lupa". Di sini urutan kata juga memiliki peran yang penting karena pengguna menanyakan tentang lupa email, bukan tentang lupa id dan password. Sehingga makna dari *query* dan dokumen hasil pencarian tidaklah sama walaupun memiliki nilai *cosine similarity* yang tinggi.

Tabel 4.12 Contoh Permasalahan Urutan Kata pada Model Kombinasi

No. *Query*	***Query* (Q) dan Hasil *Query***
13	*Query* : Pertanyaan/Query : Yth. LPSE Jabar. Saya selaku perwakilan dari PT. Pharma Kasih Sentosa , ingin menanyakan dimanakah alamat LPSE Jawa Barat ?Terima Kasih. Representasi dalam vektor *term*: [(lpse,0.115), (jabar,0.311), (alamat,0.323), (pt,0.403), (manakah,0.789)]
	Satu hasil *query* teratas (top-1) pada model kombinasi (id=647): P : dimanakah LPSE kab BDG Barat ? bagaimana melihat LPSE per/kabupaten, bisa ngak ? J : Bandung Barat masih menggunakan LPSE Provinsi Jawa Barat. Melihat LPSE Kabupaten dan Kota bisa dilihat di inaproc.go.id. Representasi dalam vektor *term*: [(id,0.079), (lpse,0.11), (jabar,0.074), (inaproc,0.209), (bisa,0.143), (tidak,0.061), (barang,0.194), (bandung,0.224), (barat,0.824), (manakah,0.376)]
25	*Query* : Saya punya Perusahaan PT Rajawali Citra Sarana , kemudian kami lupa Email : ID dan Pasword .. kira2 kami harus bagaimana agar kami bisa menggunakan kembali Email trsb ? Terima kasih Representasi dalam vektor *term*: [(email,0.509), (password,0.26), (id,0.237), (bisa,0.216), (lupa,0.37), (perusahaan,0.309), (pt,0.581)]
	Satu hasil *query* teratas (top-1) pada model kombinasi (id=1168): P : Yth. LPSE Jabar, Kami dari PT. Tashida Sejahtera Perkasa lupa User ID dan Password, mohon konfirmasi ke E-Mail Kami. Terima Kasih J : Sudah dijawab melalui e-mail Representasi dalam vektor *term*: [(email,0.433), (password,0.222), (id,0.202), (lpse,0.071), (jabar,0.191), (user,0.264), (lupa,0.315), (pt,0.495), (konfirmasi,0.517)]

Halaman ini sengaja dikosongkan

BAB 5
PENUTUP

5.1 Kesimpulan

Dari hasil-hasil eksperimen dan analisa yang telah dilakukan pada penelitian Temu Kembali Informasi Berbasis Pemodelan Topik Menggunakan Kombinasi LSI dan VSM Pada Sistem Tanya-Jawab Sistem, maka dapat disimpulkan beberapa poin diantaranya adalah:

1) Model kombinasi (LSI+VSM) mampu meningkatkan kinerja dari model menggunakan metode LSI dan VSM (VSM biasa maupun VSM+PP) yang berdiri sendiri dalam mencari pertanyaan pada arsip layanan tanya-jawab SPSE yang relevan atau mirip dengan *query*. Hal ini disebabkan karena model kombinasi mampu mengatasi beberapa kekurangan serta mengambil beberapa kelebihan yang ada pada kedua model tersebut. Dengan kombinasi tersebut menghasilkan model IR yang mampu mengatasi kekurangan model VSM dalam mengatasi permasalahan semantik atau makna kata menggunakan model LSI. Disisi lain kekurangan pada model LSI yakni permasalahan pencocokan kata kunci dapat diselesaikan menggunakan model VSM.
2) Dengan menggabungkan kelebihan pada kedua model dasar tersebut, model kombinasi mampu meningkatkan presisi keduanya dengan signifikan yakni memperoleh rata-rata P@1=0,700, rata-rata P@3=0,573 rata-rata P@5=0,476 dengan MAP=0,579. Dibandingkan dengan model LSI rata-rata P@1, P@3 dan P@5 secara berturut-turut memperoleh 0.500, 0.407 dan 0.408 dengan MAP=0,237; untuk model VSM biasa rata-rata P@1, P@3 dan P@5 secara berturut-turut hanya memperoleh 0.380, 0.333 dan 0.316 dengan MAP=0,247; sedangkan model VSM dengan pembobotan profesional (VSM+PP) rata-rata P@1, P@3 dan P@5 secara berturut-turut memperoleh 0.44, 0.347 dan 0.348 dengan MAP=0,258.
3) Dengan kelebihan yang ada, model kombinasi juga mempunyai keterbatasan yakni dalam permasalahan urutan kata dalam kalimat yang mempengaruhi

makna dari kalimat itu sendiri. Hal ini menyebabkan hasil temu kembali oleh model mendapatkan dokumen yang tidak relevan walaupun mempunyai nilai *cosine similarity* yang tinggi. Penyebabnya adalah karena model dasarnya yakni LSI dan VSM mengabaikan urutan kata dalam menentukan kemiripan suatu dokumen, sehingga berimbas pada model kombinasi yang tidak mampu menyelesaikan permasalahan tersebut.

5.2 Saran

Dengan kekurangan pada permasalahan urutan kata pada model kombinasi tersebut, dapat dilakukan pengembangan pada penelitian selanjutnya. Sehingga diharapkan dapat meningkatkan kinerja model kombinasi pada penelitian ini. Metode yang digunakan pada beberapa penelitian dalam mengatasi masalah urutan kata ini adalah dengan menggunakan n-grams atau *convolutional-pooling structure*.

DAFTAR PUSTAKA

[1] T. Kurniawan, "Hambatan dan Tantangan dalam Mewujudkan Good Governance melalui Penerapan E-Government di Indonesia," *Prosiding Konferensi Nasional Sistem Informasi,* 2006.

[2] Organisation for Economic Co-Operation and Development (OECD), The e-Government Imperative, Paris: OECD, 2003.

[3] International Telecommunication Union, Electronic Government for Developing Countries, Geneva: ITU, 2008.

[4] H. Isozaki, R. Higashinaka, M. Nagata dan T. Kato, Question Answering Systems, Japan: Corona Publishing Co. Ltd., 2009.

[5] D. Jurafsky dan J. H. Martin, "Question Answering," dalam *Speech and Language Processing*, stanford.edu, 2015.

[6] R. Baeza-Yates dan B. Ribeiro-Neto, Modern Information Retrieval, New York: ACM Press, 1999.

[7] A. Chandurkar dan A. Bansal, "Information Retrieval from a Structured KnowledgeBase," dalam *IEEE - 11th International Conference on Semantic Computing*, San Diego, 2017.

[8] A. N. Jamgade dan S. J. Karale, "Ontology based information retrieval system for Academic Library," dalam *IEEE - 2nd International Conference on Innovations in Information, Embedded and Communication systems*, Coimbatore, 2015.

[9] M. Sarrouti dan S. O. E. Alaoui, "A passage retrieval method based on probabilistic information retrieval model and UMLS concepts in biomedical question answering," *Journal of Biomedical Informatics,* vol. 68, pp. 96-103, 2017.

[10] T. B. Adji, Z. Abidin dan H. A. Nugroho, "System of Negative Indonesian Website Detection Using TF-IDF and Vector Space Model," dalam *IEEE International Conference on Electrical Engineering and Computer Science*, Bali, 2014.

[11] X. Liang, D. i. Wang dan M. Huang, "Improved Sentence Similarity Algorithm Based on VSM and Its Application in Question Answering System," dalam *IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS)*, Xiamen, 2010.

[12] H. Samuel, M.-Y. Kim, S. Prabhakar, M. S. M. Jabbar dan O. Z. ̈iane, "Community Question Retrieval in Health Forums," dalam *IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, Orlando, 2017.

[13] L. H. Xu, S. Sun dan Q. Wang, "Text Similarity Algorithm Based on Semantic Vector Space Model," dalam *IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*, Okayama, 2016.

[14] I. Chawla dan S. K. Singh, "Performance evaluation of VSM and LSI models to determine Bug reports similarity," dalam *IEEE Sixth International Conference on Contemporary Computing (IC3)*, Noida, 2013.

[15] B. V. Barde dan A. M. Bainwad, "An Overview of Topic Modeling Methods and Tools," dalam *IEEE International Conference on Intelligent Computing and Control Systems*, Madurai, 2017.

[16] M. Squire, Mastering Data Mining with Python - Find Patterns Hidden in Your Data, Birmingham: Packt Publishing, 2016.

[17] X. Hu, Z. Cai, P. Wiemer-Hastings, A. C. Graesser dan D. S. M. Namara, "Strengths, Limitations and Extentions of LSA," dalam *Handbook of Latent Semantic Analysis*, T. K. Landauer, D. S. M. Namara, S. Dennis dan W. Kintsch, Penyunt., New York, Routledge, 2007.

[18] D. Moldovan dan M. Surdeanu, "On the Role of Information Retrieval and Information Extraction in Question Answering Systems," dalam *Information Extraction in the Web Era* , Springer Berlin Heidelberg, 2003.

[19] V. Lopez, M. Pasin dan E. Motta, "AquaLog A Ontology-portable Question Answering interface for the Semantic Web," dalam *Proc. of the 2nd European Semantic Web Conference* , 2005.

[20] A. Mishra dan S. K. Jain, "A survey on question answering systems with classification," *Journal of King Saud University – Computer and Information Sciences,* vol. 28, pp. 345-361, 2016.

[21] K. Latha, Experiment and Evaluation in Information Retrieval Models, Boca Raton: CRC Press, 2016.

[22] C. X. Zhai dan S. Massung, Text Data Management and Analysis: A Practical Introduction to Information Retrieval and Text Mining, 2016.

[23] D. A. Grossman dan O. Frieder, Information Retrieval Algorithms and Heuristics, Springer, 2004.

[24] D. Sarkar, "Understanding Feature Engineering (Part 3)—Traditional Methods for Text Data," [Online]. Available: https://towardsdatascience.com/understanding-feature-engineering-part-3-traditional-methods-for-text-data-f6f7d70acd41. [Diakses 6 4 2018].

[25] M. A. Hearst, Search User Interfaces, New York: Cambridge University Press, 2009.

[26] G. B. Ivanov, "Complete Guide to Topic Modeling," 3 1 2018. [Online]. Available: https://nlpforhackers.io/topic-modeling/. [Diakses 6 4 2018].

[27] P.K.Srijith, M. Hepple, K. Bontcheva dan D. Preotiuc-Pietro, "Sub-story detection in Twitter with hierarchical Dirichlet processes," *Information Processing & Management,* vol. 53, no. 4, pp. 989-1003, 2017.

[28] R. N. Kenmogne, Understanding LSI via the Truncated Term-term Matrix, Saarbrücken: Max-Planck Institut, 2005.

[29] A. Thomo, "Latent Semantic Analysis (Tutorial)," [Online]. Available: http://webhome.cs.uvic.ca/~thomo/svd.pdf. [Diakses 12 6 2017].

[30] A. Kontostathis dan W. M. Pottenger, "A framework for understanding Latent Semantic Indexing (LSI) performance," *Information Processing and Management,* vol. 42, pp. 56-73, 2006.

[31] C. D. Manning, P. Raghavan dan H. Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008.

[32] D. M. Blei, A. Y. Ng dan M. I. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning Research,* vol. 3, pp. 993-1022, 2003.

[33] X.-H. Phan, C.-T. Nguyen, D.-T. Le dan L.-M. Nguyen, "A Hidden Topic-Based Framework toward Building Applications with Short Web Documents," dalam *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, 2011.

[34] S. N. Bahagia, "Sistem Pengadaan Publik Dan Cakupannya," *Senarai Pengadaan Barang/Jasa Pemerintah,* vol. 1, no. 1, pp. 8-25, 2011.

[35] Kementerian Hukum & HAM Republik Indonesia, "PERATURAN PRESIDEN NOMOR 54 TAHUN 2010 TENTANG PENGADAAN BARANG JASA PEMERINTAH," Kementerian Hukum & HAM Republik Indonesia, Jakarta, 2010.

[36] Lembaga Kebijakan Pengadaan Barang/Jasa Pemerintah, "Peraturan Kepala LKPP Nomor 2 Tahun 2010 Tentang Layanan Pengadaan Secara

Elektronik," Lembaga Kebijakan Pengadaan Barang/Jasa Pemerintah, Jakarta, 2010.

[37] LKPP, "Beranda | INAPROC," 2018. [Online]. Available: http://inaproc.id/. [Diakses 17 03 2018].

[38] LKPP, "Rekap Agregasi Data Penyedia pertahun berdasarkan unique NPWP," 2018. [Online]. Available: https://ppid.lkpp.go.id/information/public/view/237. [Diakses 2 4 2018].

[39] A. Kurniawan, "SPSE Versi 4.2 Resmi Diluncurkan," 25 1 2018. [Online]. Available: https://eproc.lkpp.go.id/news/read/406/spse-versi-42-resmi-diluncurkan. [Diakses 8 4 2018].

LAMPIRAN

Lampiran 1

Tabel 6.1 dataset untuk testing (sebagai *query*)

No	Id Query	Query
1	2456	Utk memenuhi syarat verifikasi, apakah copy data perusahaan yang diperlukan bisa dikirimkan melalui pos/ kurir? Terima kasih.
2	1884	Yth LPSE, bagaimana caranya untuk mengikuti tender pengadaan pencetakan dan informasi jadwalnya? Terimakasih
3	466	Bagaimana caranya untuk mengetahui nama pengguna dan password ketika pertama kali mendaftarkan (emailnya sudah tidak bisa dibuka lagi). Kemudian ganti email tapi belum mendapat kiriman balasan.
4	1605	Kepada yth Pokja ULP Dinas Kesehatan Provinsi Jawa Barat. kami selaku pengadaan ,mau nanya masalah d K3 mengenai sertifakt yg d maksud apa kami slaku penyedia punya sertifikat K3??????
5	32	Ass. Mohon pencerahan...1. Apakah boleh perusahaan kecil ikut lelang konstruksi di atas 2,5 M ?? (Yang seharusnya untuk perusahaan non kecil)2. Apakah bisa perusahaan kecil menang lelang tsb?
6	2081	saya sudah verifikasi dan sudah registrasi via internet tapi seperti yg sudah dijanjikan dapat password lewat email saya, tapi blum ada juga
7	653	saya mau tannya nilai berapa batasan antara siup kelas menengah dan besar yang bisa diikuti untuk leleang pegadaan
8	2054	As. Pa kami sudah beres verifikasi, namun belum mendapatkan password, gimana?
9	535	saya kok udah daftar tp setlah mau di cek kembli tdk dapat di buka..sedangkan udah ada laporan suksesnya lewat email amindojaya.pstnbr@yahoo.co.id
10	552	mau tanya apa userid CV Siaga Putra dari kabupaten ciamis
11	1925	Kepada Yth LPSE PROP Jabar Kami dari CV Kresna Mulya mohon penjelasan apakah User ID dan passward LPS bisa diganti, kalau bisa gimana caranya. Sekian dan terima kasih
12	111	kode lelang : 31482014Nama Lelang : Belanja renovasi gedung dan bangunaSBU BG 008 Bangunan Gedung Kesehatankenapa tdk di syaratkan ....lingkup pekerjaan area Rumah Sakit mhon penjelasan nya,thanks
13	1704	Yth. LPSE Jabar. Saya selaku perwakilan dari PT. Pharma Kasih Sentosa , ingin menanyakan dimanakah alamat LPSE Jawa Barat ?Terima Kasih.
14	742	Bpk Yth. Saya mau Tanya Bagaimana Cara mengikuti lelang dan bagaimana Supaya punya Pasword Login Terima kasih

15	1216	kepada YTH.LPSE Jabar kami lupa user id perusahaan kami,sehingga tidak bisa login, soalnya kami baru membuka lagi jd lupa user id nya tolong dibantu pak?trims
16	540	pa mau tanya apa USERID CV. Awal dari pangandaran??????
17	248	bagaimana cara mengganti email secara online
18	2363	Ass. Pak kami dari CV. Makmur Lestari ingin mendaftar ke LPSE Jabar. maaf tlong jelaskan cara mendaftarnya dan kalau bisa kirim ke email kami cara pendaftaran trsebut. wsslm
19	1798	Salam. Pak saya lupa password email saya bagaimana prosedur penggantian email dan format permohonanya seperti apa Ditunjukkan kemana?/ ke siapa? trm kasih
20	1277	yth. LPSE JABAR.CV kami ingin mendaftar. tetapi belum ada SBU dan belum ikut asosiasi apakah bisa mendaftar? Biaya pembuatan SBU dan asosiasi berkisar berapa. terimakasih.
21	1422	Pengelolah LPSE Jabar Yg terhormat, mohon kiranya dapat di jelaskan bagaimana caranya kami mendaftar sebagai rekanan di LPSE dan prosedur serta syaratnya apa saja, kami sangat membutuhkan bantuan dari tim pengelolah LPSE jabar, sebelumnya kami ucapkan terima kasih. Hormat sayaJON EDY
22	461	saya sudah mendaftar, tapi tidak mendapat balasan ke email
23	1384	Selamat pagi....saya ingin merubah data SIUP yang lama dengan SIUP yang baru, jika setelah saya merubah di web ini apakah data aslinya harus saya tunjukkan lagi ke kantor LPSE Provinsi Jawabarat?
24	2019	pak, kami dari CV. Pustaka saya mau mendaftar ke lpse jabar untuk mengikuti lelang. dokumen yg harus saya bawa apa hanya KTP, NPWP, SIUP, TDP, Formulir pendaftaran dan formulir keikutsertaan?
25	159	Saya punya Perusahaan PT Rajawali Citra Sarana , kemudian kami lupa Email : ID dan Pasword .. kira2 kami harus bagaimana agar kami bisa menggunakan kembali Email trsb ? Terima kasih
26	1474	selamat sore bapak/ibu Yth.LPSE,PT kami sudah lulus verifikasi dan kami tidak dapat longin kami lupa user ID dan Passwordnya apaka bisa diganti dengan yang baru mohon bantuanya Trimakasih
27	1221	Yth. Lpse Jabarkami mau daftar salah satu paket pekerjaan yaitu Pekerjaan Rehab Gd Makarti tetapi tidak ada pada portal padahal pemasukan masih cukup lama, apa yang harus saya lakukan
28	2283	perusahaan kami sudah selesai verikasi tapi password belum kami terima, proses tersebut sudah berjalan 1 minggu, mohon perhatiannya terima kasih
29	2211	kami sdh mendaftar kemarin tp blm bisa login kenepa

30	1000	cara mendapatkan user id
31	1795	di RKS melampirkan bukti kepemilikan/sewa 50 unit bubut kayu, bagaimana mengingat pekerjaan tsb tidak ada bahan/material kayu yg bulat
32	1858	Yth, LPSEApabila dalam pengumuman dicantumkan 2 sub bidang, apakah bisa 2 perusahan JO tetapi masing2 perusahaan hanya memiliki 1 sub bidang yang diminta utk saling melengkapi. terima kasih
33	1529	Yth. LPSE, pada saat pembuktian kualifikasi bagi calon pemenang apakah SBU harus sudah diperpanjang/dileges ? Apabila penyedia Jasa yang ditunjuk menjadi pemenang SBU nya belum diperpanjang/dileges, apakah syah ?
34	1362	kami dr PT Mega Dwitek bermaksud mempertanyakan untuk pekerjaan paket 502 Indramayu untuk jaminan Penawarannya di terbitkan melalui PT Asuransi saja atau dr Bank?
35	161	kenapa setiap buka email lpse slalu gagal sedangkan tahun kemarin email tersebut digunakan untuk lelang. mohon penjelasan dan bantuannya
36	76	bagaimana cara mendaftar menjadi penyedia. trims.
37	2392	Mau nanya Pak.....Password untuk User ID "GM" atas nama CV. Gunung Melati tolong di beritahu? soalnya sampai sekarang belum ada konfirmasi ke alamat e-mail kami yaitu "gunungmelati@hotmail.com" mohon secepatnya di beritahu lewat rubrik tanya jawab ini....terima kasih
38	1586	saya merubah password login panitia dan kelupaan password sehingga tidak bisa login bagaimana cara mengatasinya ? makasih
39	2201	saya ingin upload dok dalam 1 folder, diperbolehkan tidak menggunakan aplikasi ZIP/ RAR atau menggunakan aplikasi izarc
40	531	kenapa ya tiba-tiba tidak bisa login,komentarnya User ID dan Pasword salah padahal berum pernah di rubah dan tetep itu selama ini juga.mohon bantuanya,Trimakasih
41	1678	Ass. Pak sy mo tanya, perusahaan sy blm punya SBU dan IUJK Sedangkan di TDP tertulis Pengadaan dan Jasa Kontruksi, Apakah sy bs daftar ke LPSE sebagai penyedia. jwbn tolong di kirim ke e-mail sy.....terimakasih
42	1211	ga bisa login lpse jabar, sudah aktivasi inaproc tgl 27 Jan 2012 dan kami klik "Aktivasi Inaproc telah dilakukan tgl 27 Jan 2012" dan respon Maaf, operasi yang Anda lakukan terakhir salah .....
43	2480	Cara daftar dan mendapatkan ID
44	316	Kami peringkat 1 dalam penawaran.Tetapi kami tidak mendapatkan undangan,sedangkan waktu pembuktian terakhir sampai tgl 15 juli 2015, mohon penjelasannya!Bisa kirim sanggah kepada Pokja ULP.

45	522	Selamat Siang PAkMohon Diklarifikasi pak untuk Paket 503 Pekerjaan Pemasangan IR di UPTD Wil.V Cirebon perusahaan kmi digugurkan krna tidak melampirkan kalibrasi,yg sbnrnya ad SKDP kalibrasi
46	2465	mengapa user id dan password saya selalu dibilang salah
47	749	saya lupa password,dan user ID, Gimana cara buka dan mengunakan nya kembali?
48	198	cara untuk aggregasi
49	619	pada saat saya buka akun di LPSE ada tulisan "Data akun Anda saat ini tidak tersinkronisasi dengan sistem Aggregasi Inaproc" itu kenapa yah? dan apa yang harus dilakukan.terimakasih
50	196	Kenapa untuk login ke lpse selalu saja ngga bisa dan ada tulisan kesalahan inaproc, maksudnya apa ya pa

Lampiran 2

Tabel 6.2 Kata-kata yang diadopsi ke dalam *library* Aspell (file '.aspeel.id.pws')

aanwidzing	helpdesk	ntb	spam
account	home	outbox	sppbj
admin	inaproc	pangandaran	subang
adp	inbox	password	sukabumi
aktivasi	indramayu	pdam	sumedang
apendo	iujk	pelelangan	tasikmalaya
banjar	jabar	penggantian	tdp
bekasi	jateng	perda	teraktivasi
bermasalah	jatim	perpres	terkirim
bimtek	java	pln	tersinkronisasi
bpk	karawang	pokja	tkdn
ciamis	kemana	portal	training
cianjur	kementan	postgresql	ulp
cimahi	kemitraan	pt	unpad
cirebon	kso	purwakarta	upload
code	ktp	rar	user
cv	kuningan	rekanan	username
database	laptop	reset	wb
depok	link	rhs	website
dientry	linux	rks	wr
djk	lkpp	sbu	zip
download	login	sbujptl	
email	logout	script	
enkripsi	lpse	sent	
eproc	majalengka	server	
eprocurement	mendownload	sirup	
epurchasing	mengunduh	siujk	
error	net	siup	
form	nihil	ska	
garut	npwp	sockettimeoutexception	

Lampiran 3

Tabel 6.3 penyesuaian hasil pengkoreksian kata oleh *library* Aspell (file '.aspeel.id. prepl')

term	**hasil koreksi *term***
agresi	agregasi
aja	saja
anya	tanya
apaka	apakah
bari	baru
bbrp	beberapa
bgmn	bagaimana
blm	belum
brg	barang
bs	bisa
bsa	bisa
dah	sudah
daptar	daftar
dlm	dalam
dpt	dapat
dr	dari
ga	tidak
gak	tidak
gimana	bagaimana
hrs	harus
info	informasi
kab	kabupaten
knp	kenapa
mail	email
mengupload	upload
merubah	mengubah
ngak	agak
ngak	tidak
ngga	tidak

term	**hasil koreksi *term***
paspord	password
pasweed	password
paswordnya	password
paterima	terima
pd	pada
pendaftarannya	pendaftaran
peraratan	persyaratan
prifikasi	verifikasi
prop	provinsi
prov	provinsi
sdh	sudah
smpe	sampai
spek	spesifikasi
sprti	seperti
sy	saya
tdak	tidak
tgl	tanggal
tlg	tolong
tsb	tersebut
utk	untuk
yg	yang
yth	yang

Lampiran 4

Tabel 6.4 Sepuluh Term Dengan Bobot Tertinggi Untuk Setiap Topik Pada Model LSI Dengan Jumlah Topik k=33

#	(*Keyword/Term*, Bobot)	Interpretasi Topik
Topik 0	0.392*"password" + 0.290*"email" + 0.290*"id" + 0.287*"lupa" + 0.274*"user" + 0.163*"login" + 0.142*"lpse" + 0.141*"bisa" + 0.137*"jabar" + 0.135*"verifikasi"	informasi tentang lupa user id, password dan email serta tentang proses verifikasi penyedia
Topik 1	-0.283*"password" + 0.273*"special" + 0.273*"content" + 0.267*"cara" + 0.260*"home" + 0.251*"menu" + -0.246*"lupa" + 0.237*"website" + 0.233*"mendaftar" + 0.222*"penyedia"	- Tata cara mendaftar - Lupa password
Topik 2	0.265*"lupa" + -0.243*"lelang" + 0.225*"password" + -0.212*"bisa" + -0.187*"tidak" + -0.173*"helpdesk" + 0.170*"special" + 0.170*"content" + -0.163*"jabar" + 0.154*"home"	- prosedur lupa password - informasi tentang pekerjaan lelang
Topik 3	-0.324*"id" + 0.299*"password" + -0.275*"email" + 0.272*"fasilitas" + 0.263*"lelang" + 0.257*"lupa" + -0.237*"user" + -0.203*"helpdesk" + 0.193*"halaman" + 0.147*"paket"	- informasi prosedur lelang - lupa password
Topik 4	-0.339*"email" + 0.286*"verifikasi" + 0.275*"login" + 0.235*"agregasi" + -0.227*"penggantian" + -0.203*"lelang" + 0.203*"mendaftar" + 0.183*"belum" + -0.173*"alamat" + 0.155*"inaproc"	- agregasi inaproc - penggantian email - cara mendaftar
Topik 5	0.369*"belum" + 0.323*"verifikasi" + -0.230*"helpdesk" + 0.196*"cek" + 0.195*"pendaftaran" + -0.186*"login" + 0.173*"online" + -0.162*"lupa" + -0.161*"kirim" + 0.160*"email"	Informasi pendaftaran dan verifikasi
Topik 6	-0.458*"user" + -0.396*"id" + 0.379*"email" + 0.279*"penggantian" + 0.152*"login" + -0.141*"cara" + 0.139*"alamat" + -0.135*"mendaftar" + 0.127*"cek" + -0.115*"lelang"	- Prosedur penggantian email - user id
Topik 7	-0.289*"lelang" + 0.275*"upload" + 0.264*"pelatihan" + 0.239*"rabu" + -0.231*"paket" + 0.178*"kantor" + 0.169*"balai" + 0.165*"dokumen" + 0.163*"penawaran" + -0.153*"cari"	- informasi jadwal pelatihan - informasi pekerjaan lelang
Topik 8	0.337*"helpdesk" + -0.317*"agregasi" + -0.260*"terdaftar" + -0.216*"inaproc" + 0.208*"kontak" + 0.185*"kirim" + -0.165*"jabar" + -0.155*"alamat" + 0.153*"langsung" + -0.153*"user"	Agregasi data penyedia

Topik 9	0.310*"pelatihan" + -0.246*"penawaran" + 0.244*"rabu" + -0.219*"dokumen" + 0.166*"kamis" + 0.164*"mengikuti" + -0.164*"upload" + -0.161*"pendaftaran" + 0.158*"balai" + -0.146*"form"	- informasi jadwal pelatihan - informasi pekerjaan lelang
Topik 10	-0.382*"perusahaan" + -0.265*"data" + -0.222*"verifikasi" + 0.195*"konfirmasi" + -0.186*"npwp" + 0.181*"pendaftaran" + 0.157*"cek" + -0.152*"terdaftar" + 0.150*"pelatihan" + 0.148*"email"	verifikasi data perusahaan/penyedia
Topik 11	0.254*"verifikasi" + -0.244*"data" + -0.217*"pendaftaran" + -0.213*"mendaftar" + -0.208*"cara" + -0.195*"formulir" + -0.173*"penyedia" + 0.171*"penawaran" + -0.170*"klik" + -0.158*"npwp"	Tata cara penggunaan aplikasi SPSE
Topik 12	-0.280*"mendaftar" + -0.265*"cek" + 0.236*"penggantian" + 0.234*"pendaftaran" + -0.209*"belum" + -0.186*"cara" + 0.186*"login" + -0.177*"spam" + 0.176*"online" + 0.165*"bisa"	- Pendaftaran online - Email konfirmasi masuk spam
Topik 13	-0.334*"alamat" + 0.285*"lupa" + -0.249*"baru" + -0.244*"apendo" + 0.240*"kirim" + -0.192*"ulang" + -0.156*"bisa" + -0.150*"kontak" + 0.134*"data" + 0.134*"dokumen"	Aplikasi apendo
Topik 14	0.300*"login" + -0.255*"jasa" + -0.222*"barang" + -0.201*"mendaftar" + -0.195*"pengadaan" + 0.194*"perusahaan" + 0.186*"lelang" + -0.175*"langsung" + -0.160*"lkpp" + 0.146*"petunjuk"	Informasi pekerjaan lelang
Topik 15	-0.280*"data" + 0.238*"jabar" + -0.232*"baru" + 0.212*"login" + -0.208*"sistem" + 0.190*"halaman" + 0.182*"mendaftar" + -0.167*"password" + 0.157*"terdaftar" + -0.155*"mendapatkan"	Mendaftar lelang
Topik 16	0.378*"cv" + 0.222*"kontak" + 0.209*"paket" + -0.198*"ulang" + -0.190*"peserta" + 0.182*"daftar" + -0.160*"mengikuti" + 0.157*"nomor" + 0.151*"alamat" + -0.151*"email"	Informasi alamat kontak
Topik 17	0.303*"jasa" + 0.243*"barang" + 0.231*"penyedia" + 0.202*"login" + -0.199*"lelang" + -0.194*"inaproc" + -0.191*"dokumen" + 0.184*"masuk" + -0.184*"agregasi" + 0.175*"pengadaan"	Regulasi pengadaan barang dan jasa
Topik 18	-0.373*"daftar" + -0.278*"ulang" + -0.242*"cv" + 0.216*"apendo" + -0.175*"penawaran" + 0.169*"paket" + -0.167*"peserta" + -0.157*"nama" + 0.143*"baru" + 0.135*"verifikasi"	- Pendaftaran penyedia - Aplikasi apendo

Topik 19	-0.364*"apendo" + 0.356*"alamat" + -0.208*"penggantian" + -0.187*"data" + -0.186*"ulang" + -0.174*"penyedia" + 0.147*"dokumen" + -0.144*"download" + -0.144*"lupa" + 0.139*"sistem"	Cara mendownload aplikasi apendo
Topik 20	-0.351*"cv" + 0.299*"apendo" + -0.246*"mendaftar" + 0.198*"perusahaan" + -0.190*"data" + 0.182*"npwp" + 0.177*"lkpp" + 0.174*"daftar" + 0.157*"belum" + -0.139*"konfirmasi"	Kelengkapan data penyedia
Topik 21	-0.314*"dokumen" + -0.239*"terdaftar" + 0.239*"pemenang" + 0.214*"sanggah" + 0.200*"cv" + 0.162*"data" + -0.158*"kontak" + 0.155*"verifikasi" + -0.154*"lelang" + 0.150*"perubahan"	Informasi pekerjaan lelang
Topik 22	-0.363*"daftar" + -0.305*"masuk" + 0.237*"nomor" + 0.178*"kontak" + -0.155*"klik" + -0.153*"formulir" + 0.143*"petunjuk" + -0.140*"verifikasi" + -0.139*"alamat" + -0.133*"dokumen"	- Regulasi pengadaan barang & jasa - Verifikasi dokumen penyedia
Topik 23	-0.458*"daftar" + 0.290*"cv" + -0.247*"pt" + 0.179*"apendo" + 0.167*"verifikasi" + -0.166*"konfirmasi" + -0.165*"mengubah" + -0.162*"jabar" + 0.146*"mengikuti" + 0.142*"perusahaan"	Perubahan data penyedia
Topik 24	0.225*"pelatihan" + 0.225*"perusahaan" + 0.195*"jam" + 0.181*"rekanan" + -0.178*"jabar" + -0.170*"petunjuk" + 0.157*"cek" + -0.154*"lakukan" + -0.152*"penyedia" + -0.148*"terdaftar"	- Persyaratan pelatihan - Petunjuk penggunaan SPSE
Topik 25	0.251*"formulir" + -0.246*"akun" + -0.224*"cv" + 0.199*"pelatihan" + -0.194*"daftar" + -0.183*"cara" + -0.180*"memiliki" + -0.166*"registrasi" + 0.139*"klik" + 0.139*"terdaftar"	Petunjuk penggunaan SPSE
Topik 26	0.243*"login" + 0.225*"alamat" + 0.208*"verifikasi" + 0.180*"cek" + -0.179*"baru" + -0.176*"jabar" + 0.172*"akun" + 0.172*"halaman" + -0.160*"masuk" + -0.159*"kualifikasi"	Verifikasi data penyedia
Topik 27	0.262*"belum" + 0.260*"klik" + -0.177*"informasi" + 0.166*"sanggah" + 0.158*"langsung" + -0.149*"ulang" + 0.146*"log" + -0.145*"jabar" + 0.141*"panitia" + 0.141*"lakukan"	- Melakukan sanggah lelang - Penjelasan dokumen
Topik 28	-0.269*"lakukan" + 0.260*"apendo" + -0.185*"upload" + 0.175*"alamat" + -0.166*"kenapa" + 0.158*"login" + 0.155*"kirim" +	Penggunaan aplikasi apendo

	0.151*"daftar" + -0.151*"penggantian" + -0.147*"masuk"	
Topik 29	-0.315*"masuk" + 0.293*"log" + -0.235*"kualifikasi" + -0.180*"kirimkan" + 0.178*"kirim" + 0.173*"cv" + 0.151*"bisa" + 0.145*"gangguan" + -0.145*"akun" + 0.133*"petunjuk"	- Gangguan aplikasi - Pengiriman dokumen kualifikasi
Topik 30	-0.235*"ulang" + 0.226*"akun" + 0.212*"penawaran" + -0.209*"penjelasan" + 0.198*"langsung" + -0.189*"masuk" + 0.177*"file" + -0.167*"perubahan" + 0.156*"mendapatkan" + -0.145*"lkpp"	Dokumen penawaran
Topik 31	0.368*"jam" + -0.207*"konfirmasi" + -0.179*"log" + 0.172*"upload" + -0.159*"peserta" + 0.158*"mendaftar" + 0.155*"berapa" + -0.154*"bisa" + 0.153*"ulang" + -0.149*"mengikuti"	Informasi waktu dan fasilitas pelayanan
Topik 32	0.318*"npwp" + -0.289*"kualifikasi" + 0.240*"nama" + 0.201*"masuk" + 0.177*"akun" + 0.166*"jam" + 0.147*"penyedia" + -0.145*"login" + -0.138*"kirim" + -0.128*"surat"	Informasi data penyedia

Lampiran 5

a. Source code untuk menyimpan dokumen training ke dalam bentuk korpus

```
#
#Source code untuk preprocessing, simpan ke dalam korpus
#

#import library yang diperlukan
import logging
import os
import csv
import tempfile
import string
import glob
import re
import aspell
import pickle
from gensim import corpora
from pprint import pprint
from nltk.tokenize import RegexpTokenizer
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.tokenize import regexp_tokenize

#Inisialisasi file pendukung
folderInput = "file/data/input/"
folderOutput = "file/data/output/training/"
inputTraining = folderInput+"training.csv"
outputTraining = folderOutput+"outTraining.txt"
TEMP_FOLDER = ('file/temp')

#class preprocessing

indo = aspell.Speller('lang', 'id')
en = aspell.Speller('lang', 'en')
stop_words = set(stopwords.words('indonesian'))

#Membuat kustom kedua untuk mengganti kata tertentu (yang tidak bisa dilakukan oleh aspell)
import re
replacement_patterns = [
    (r'terima kasih', ''),
    (r'jawa barat', 'jabar'),
    (r' log in ', ' login '),
    (r' log out ', ' logout '),
    (r' log in', ' login '),
    (r' log out', ' logout '),
    (r' ma\'af ', ' maaf '),
    (r'selamat pagi', '')
]
class RegexpReplacer(object):
     def __init__(self, patterns=replacement_patterns):
       self.patterns = [(re.compile(regex), repl) for (regex, repl) in patterns]
     def replace(self, text):
        s = text
        for (pattern, repl) in self.patterns:
            s = re.sub(pattern, repl, s)
        return s
replacer = RegexpReplacer()

class WordReplacer(object):
    def __init__(self, word_map):
        self.word_map = word_map
    def replace(self, word):
        return self.word_map.get(word, word)

replacer1 =
WordReplacer({'bari':'baru','dah':'sudah','info':'informasi','paswad':'password','mail':'ema
il',\

'pass':'password','ngak':'tidak','no':'nomor','kab':'kabupaten','prop':'provinsi',\
                        'dok':'dokumen','caranya':'cara',\
                        'prov':'provinsi','sdgkan':'sedangkan','pendaftarannya ':'
pendaftaran','tgl':'tanggal'})


class teks:
    def __init__(self, teksInput):
        self.Teks = teksInput
```

```
    def preProcessing(self):
        word_tokens1 = regexp_tokenize(self, pattern='\w+')
        word_tokens1 = [z for z in word_tokens1 if not z in stop_words]
        #
        list1=[]
        a=len(word_tokens1)
        for b in range(0,a):
            huruf=word_tokens1[b]
            f=lambda a: int((abs(a)+a)/2)#untuk menghindari index negatif
            if word_tokens1[b].isupper():
                word_tokens1[b]=word_tokens1[b].lower()
            if huruf[0].isupper():
                #print(word_tokens1[b])
                if not word_tokens1[b].isupper():
                    if word_tokens1[f(b-1)]!="pt" and word_tokens1[f(b-1)]!="cv" and
word_tokens1[f(b-1)]!="ud":
                        if (word_tokens1[f(b-2)]!="pt" and word_tokens1[f(b-2)]!="cv" and
word_tokens1[f(b-2)]!="ud") or word_tokens1[f(b-1)].islower():
                            if (word_tokens1[f(b-3)]!="pt" and word_tokens1[f(b-3)]!="cv"
and word_tokens1[f(b-3)]!="ud") or word_tokens1[f(b-1)].islower():
                                #if word_tokens1[b]!="pt" and word_tokens1[b]!="cv" and
word_tokens1[b]!="ud":
                                word_tokens1[b]=word_tokens1[b].lower()
                                list1.append(word_tokens1[b])
                                #print(word_tokens1[b])
            else:
                list1.append(word_tokens1[b])
        word_tokens1=list1
        #
        kalimat2 = ''
        for k in word_tokens1:
            #Jika token mengandung angka/simbol maka lewati
            k = k.lower()
            k = replacer1.replace(k)
            if not (k.isalpha()):
                continue
            if len(k)<3:
                if (k!="id" and k!="pt" and k!="cv" and k!="ud"):
                     continue
            #jika angka jangan lakukan spell correction
            if not k.isdigit():
                #jika kata dalam bahasa inggris jngn lakukan spell correction
                if not (en.check(k)):
                    wSpell = indo.suggest(k)
                    if(len(wSpell) > 0):
                        kalimat2 = kalimat2+wSpell[0]+' '
                else:
                    kalimat2 = kalimat2+k+' '
        word_tokens1=word_tokenize(kalimat2)
        #Melakukan stopwords
        word_tokens1 = [r for r in word_tokens1 if not r in stop_words]
        filtered_sentence=' '.join(word_tokens1)
        filtered_sentence=replacer.replace(filtered_sentence)
        #
        return filtered_sentence
#

#menghitung statistik dataset sebelum preprocessing
with open(inputTraining, newline='') as f :
    reader = csv.reader(f, delimiter='|')
    q=0
    jumlahTotalWords=[]
    unikWords=[]
    terbesar=0
    for row in reader:
        row12=row[1]+" "+row[2]
        word_tokens1 = regexp_tokenize(row12, pattern='\w+')
        jumlahTokenDokumen = len(word_tokens1)
        if jumlahTokenDokumen > terbesar:
            terbesar = jumlahTokenDokumen
        for ww in word_tokens1:
            if ww not in unikWords:
                unikWords.append(ww)
        jumlahTotalWords=jumlahTotalWords+word_tokens1
print("Jumlah Total Term : ",len(jumlahTotalWords))
print("Jumlah term unik : ",len(unikWords))
print("Jumlah term terbanyak pada semua dokumen : ",terbesar)
#
```

```
#hapus file hasilPreprocessing.txt dulu
os.remove("file/data/output/hasilPreprocessing.txt")
#

#menghitung statistik dataset setelah preprocessing
with open(inputTraining, newline='') as f :
    reader = csv.reader(f, delimiter='|')
    for row in reader:
        row12=row[1]+" "+row[2]
        row12 = teks.preProcessing(row12)
        saveFile=open("file/data/output/hasilPreprocessing.txt",'a+')
        saveFile.write(row12)
        saveFile.write("\n")
        saveFile.close()
print("menghitung statistik dataset setelah preprocessing....done...")
#

#hapus isi folder output
files = glob.glob(folderOutput+"*")
for f in files:
    os.remove(f)

#ambil data training, simpan ke list/array
with open(inputTraining, newline='') as f :
    reader = csv.reader(f, delimiter='|')
    q=0
    for row in reader:
        row12=row[1]+" "+row[2]
        row12 = teks.preProcessing(row12)
        saveFile=open(outputTraining,'a+')
        saveFile.write(str(row[0])+". "+row12)
        saveFile.write("\n")
        saveFile.close()
        saveFile=open(folderOutput+str(q)+".txt",'w')
        saveFile.write(str(row[0])+". "+row12)
        saveFile.write("\n")
        saveFile.close()
        q = q+1
print("ambil data training... done...")
#

#Folder "file/temp" sebagai temporary
logging.basicConfig(format='%(asctime)s : %(levelname)s : %(message)s', level=logging.INFO)
NAMA_FILE_INPUT = outputTraining
print('Folder "{}" will be used to save temporary dictionary and
corpus.'.format(TEMP_FOLDER))
#

#hasil preprocessing sebagai dokumen per value untuk array
with open(NAMA_FILE_INPUT, 'r') as fin:
    dokumen=[]
    dokumen1=[]
    for line in fin:
        #for c in string.punctuation:
        #    line= line.replace(c,"")
        dokumen.append(line.strip())
for kalimat in dokumen:
    for c in string.punctuation:
        kalimat = kalimat.replace(c,"")
    for c in string.digits:
        kalimat = kalimat.replace(c,"")
    dokumen1.append(kalimat)
dokumen=dokumen1
#

#Setiap dokumen di tokenisasi
from collections import defaultdict

#tokenizer = RegexpTokenizer(r'\w+')
#texts = tokenizer.tokenize(dokumen)

texts = [[word for word in document.lower().split()]
         for document in dokumen]

frequency = defaultdict(int)
for text in texts:
    for token in text:
        frequency[token] += 1
```

```
texts = [[token for token in text if frequency[token] > 1] for text in texts]

from pprint import pprint  # pretty-printer
#

os.remove("file/data/output/hasilPreprocessing1.txt")

#hitung koleksi kata
koleksiKata = []
jumlahKata = 0
for aa in range(len(texts)):
    jumlahKata=jumlahKata+len(texts[aa])
    kalimatBaru=' '.join(texts[aa])
    saveFile=open("file/data/output/hasilPreprocessing1.txt",'a+')
    saveFile.write(kalimatBaru)
    saveFile.write("\n")
    saveFile.close()

    for bb in texts[aa]:
        if bb not in koleksiKata:
            koleksiKata.append(bb)

with open('file/data/koleksiKata.txt', 'wb') as fp:
    pickle.dump(koleksiKata, fp)
#

print("koleksi kata :",str(len(koleksiKata)))
print("jumlah kata :",str(jumlahKata))
#
os.remove("file/temp/lpse-jabar.dict")
os.remove("file/temp/lpse-jabar.mm")
os.remove("file/temp/lpse-jabar.mm.index")
#
#simpan hasil pembentukan korpus
dictionary = corpora.Dictionary(texts)
dictionary.save(os.path.join(TEMP_FOLDER, 'lpse-jabar.dict'))  # store the dictionary, for
future reference
corpus = [dictionary.doc2bow(text) for text in texts]
corpora.MmCorpus.serialize(os.path.join(TEMP_FOLDER, 'lpse-jabar.mm'), corpus)  # store to
disk, for later use
```

b. Source code untuk transformasi menggukan model

```
#
#source code untuk membuat model IR
#

#import library yang dibutuhkan
import logging
import tempfile
import os.path
import glob
import csv
from gensim import corpora, models, similarities

#inisialisasi file temporary
logging.basicConfig(format='%(asctime)s : %(levelname)s : %(message)s', level=logging.INFO)
TEMP_FOLDER = ('file/temp')

#mengambil korpus yang telah dibuat
from gensim import corpora, models, similarities
if os.path.isfile(os.path.join(TEMP_FOLDER, 'lpse-jabar.dict')):
    dictionary = corpora.Dictionary.load(os.path.join(TEMP_FOLDER, 'lpse-jabar.dict'))
    corpus = corpora.MmCorpus(os.path.join(TEMP_FOLDER, 'lpse-jabar.mm'))
    print("Mengambil korpus yang telah dibuat sebelumnya..")
else:
    print("Jalankan file pembuatan korpus terlebih dahulu..")
#

#Menghapus model yang telah ada sebelumnya
files = glob.glob("file/temp/model/*")
for f in files:
    os.remove(f)
#

#membuat model TF-IDF
```

```
tfidf = models.TfidfModel(corpus) # step 1 -- initialize a tfidf model
tfidf.save(os.path.join(TEMP_FOLDER, 'model/model.tfidf')) # save tfidf model
corpus_tfidf = tfidf[corpus]
#

#modifikasi TF-IDF (VSM+PP), simpan hasil indexing
termProf=[]
with open('file/data/input/profesionalWeight.csv', newline='') as f :
    reader = csv.reader(f, delimiter='|')
    for row in reader:
        #print(row[0])
        termProf.append(row[0])
kamusTerm = []
for i in range (len(dictionary)):
    kamusTerm.append(dictionary[i])
indexTermProf=[]
for b in range (len(termProf)):
    indexTermProf.append(kamusTerm.index(termProf[b]))
corpus_tfidf1=[]
corpus_tfidf_prof=[]
for c in range (len(corpus_tfidf)):
    lst = list(corpus_tfidf[c])
    for a in range (len(lst)):
        #periksa term apakah term profesional
        lst1 = list(lst[a])
        cekTerm=lst1[0]
        #print(cekTerm)
        if cekTerm not in indexTermProf:
            #print(cekTerm)
            lst1[1]=0.8*lst1[1]
        lst[a]=tuple(lst1)
    corpus_tfidf1=tuple(lst)
    corpus_tfidf_prof.append(corpus_tfidf1)
corpus_tfidf_prof=tuple(corpus_tfidf_prof)
indextfidfProf = similarities.MatrixSimilarity(corpus_tfidf_prof)
indextfidfProf.save(os.path.join(TEMP_FOLDER, 'index/tfidfProf.index'))#save tfidf index
#

#Pembuatan model LSI, dan menyimpannya
for jmlTopik in range(5,51):
    g="LSI"+str(jmlTopik)
    vars()[g] = models.LsiModel(corpus_tfidf, id2word=dictionary, num_topics=jmlTopik)
    vars()[g].save(os.path.join(TEMP_FOLDER, "model/model."+str(g)))#save LSI model
#

#Pembuatan model LDA, dan menyimpannya
for jmlTopik in range(5,51):
    g="LDA"+str(jmlTopik)
    vars()[g] = models.LdaModel(corpus, id2word=dictionary, num_topics=jmlTopik)
    vars()[g].save(os.path.join(TEMP_FOLDER, "model/model."+str(g)))#save LDA model
#

print("Selesai...")
```

c. Source code untuk pengukuran *similarity*

```
#
#source code penghitungan similarity query terhadap korpus
#

#inisialisasi library yang diperlukan
import logging
import tempfile
import os.path
import csv
import glob
import re
import aspell
from gensim import corpora, models, similarities
from collections import defaultdict
from gensim import corpora
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.tokenize import regexp_tokenize

#inisialisasi folder temporary dan file pendukung
```

```
logging.basicConfig(format='%(asctime)s : %(levelname)s : %(message)s',
level=logging.INFO)
TEMP_FOLDER = ('file/temp')
inputPertanyaan = "file/data/input/questions.csv"
fileJawaban="file/data/input/IndexJawaban.csv"
folderHasil="file/data/output/similarity"

#class normalisasi
def normalisasi(data1):
    newData1 = (data1-(-1))*(1-0)/(1-(-1))+0
    return newData1

#class preprocessing
#preprocessing
#
indo = aspell.Speller('lang', 'id')
en = aspell.Speller('lang', 'en')
stop_words = set(stopwords.words('indonesian'))

#Membuat kustom kedua untuk mengganti kata tertentu (yang tidak bisa dilakukan oleh
aspell)
import re
replacement_patterns = [
    (r'terima kasih', ''),
    (r'jawa barat', 'jabar'),
    (r' log in ', ' login '),
    (r' log out ', ' logout '),
    (r' log in', ' login '),
    (r' log out', ' logout '),
    (r' ma\'af ', ' maaf '),
    (r'selamat pagi', '')
]
class RegexpReplacer(object):
     def __init__(self, patterns=replacement_patterns):
       self.patterns = [(re.compile(regex), repl) for (regex, repl) in patterns]
     def replace(self, text):
        s = text
        for (pattern, repl) in self.patterns:
            s = re.sub(pattern, repl, s)
        return s
replacer = RegexpReplacer()

class WordReplacer(object):
    def __init__(self, word_map):
        self.word_map = word_map
    def replace(self, word):
        return self.word_map.get(word, word)

replacer1 =
WordReplacer({'bari':'baru','dah':'sudah','info':'informasi','paswad':'password','mail
':'email',\

'pass':'password','ngak':'tidak','no':'nomor','kab':'kabupaten','prop':'provinsi',\
                        'dok':'dokumen','caranya':'cara',\
                        'prov':'provinsi','sdgkan':'sedangkan','pendaftarannya ':'
pendaftaran','tgl':'tanggal'})


class teks:
    def __init__(self, teksInput):
        self.Teks = teksInput
    def preProcessing(self):
        word_tokens1 = regexp_tokenize(self, pattern='\w+')
        word_tokens1 = [z for z in word_tokens1 if not z in stop_words]
        #
        list1=[]
        a=len(word_tokens1)
        for b in range(0,a):
            huruf=word_tokens1[b]
            f=lambda a: int((abs(a)+a)/2)#untuk menghindari index negatif
            if word_tokens1[b].isupper():
                word_tokens1[b]=word_tokens1[b].lower()
            if huruf[0].isupper():
                #print(word_tokens1[b])
```

```
                if not word_tokens1[b].isupper():
                    if word_tokens1[f(b-1)]!="pt" and word_tokens1[f(b-1)]!="cv" and
word_tokens1[f(b-1)]!="ud":
                        if (word_tokens1[f(b-2)]!="pt" and word_tokens1[f(b-2)]!="cv"
and word_tokens1[f(b-2)]!="ud") or word_tokens1[f(b-1)].islower():
                            if (word_tokens1[f(b-3)]!="pt" and word_tokens1[f(b-
3)]!="cv" and word_tokens1[f(b-3)]!="ud") or word_tokens1[f(b-1)].islower():
                                #if word_tokens1[b]!="pt" and word_tokens1[b]!="cv"
and word_tokens1[b]!="ud":
                                word_tokens1[b]=word_tokens1[b].lower()
                                list1.append(word_tokens1[b])
                                #print(word_tokens1[b])
            else:
                list1.append(word_tokens1[b])
        word_tokens1=list1
        #
        kalimat2 = ''
        for k in word_tokens1:
            #Jika token mengandung angka/simbol maka lewati
            k = k.lower()
            k = replacer1.replace(k)
            if not (k.isalpha()):
                continue
            if len(k)<3:
                if (k!="id" and k!="pt" and k!="cv" and k!="ud"):
                    continue
            #jika angka jangan lakukan spell correction
            if not k.isdigit():
                #jika kata dalam bahasa inggris jngn lakukan spell correction
                if not (en.check(k)):
                    wSpell = indo.suggest(k)
                    if(len(wSpell) > 0):
                        kalimat2 = kalimat2+wSpell[0]+' '
                else:
                    kalimat2 = kalimat2+k+' '
        word_tokens1=word_tokenize(kalimat2)
        #Melakukan stopwords
        word_tokens1 = [r for r in word_tokens1 if not r in stop_words]
        filtered_sentence=' '.join(word_tokens1)
        filtered_sentence=replacer.replace(filtered_sentence)
        #
        return filtered_sentence
#end of preprocessing

#
os.popen('rm -f file/data/output/result/preProcessingQuery.txt')

#mengambil query dan melakukan preprocessing
idQuery = []
inputQuery = []
nn=1
with open(inputPertanyaan, newline='') as f :
    reader = csv.reader(f, delimiter='|')
    for row in reader:
        idQuery.append(row[1])
        inputQuery.append(row[2])
        doc=row[2]
        print(nn,".",doc,"\n")
        doc = teks.preProcessing(doc)
        print(nn,".",doc,"\n")
        saveFile=open("file/data/output/result/preProcessingQuery.txt",'a+')
        saveFile.write(doc)
        saveFile.write("\n")
        saveFile.close()
        nn=nn+1

#mengambil data training untuk menampilkan hasil query dalam dokumen, bukan nomor id
corpusPertanyaan = []
corpusJawaban = []
corpusID = []

inputTraining = "file/data/input/training.csv"
with open(inputTraining, newline='') as f :
    reader = csv.reader(f, delimiter='|')
```

```
    for row in reader:
        row01 = row[0]+". "+row[1]
        corpusID.append(row[0])
        corpusPertanyaan.append(row[1])
        corpusJawaban.append(row[2])

#Mengambil term dengan pembobotan profesional
termProf=[]
with open('file/data/input/profesionalWeight.csv', newline='') as f :
    reader = csv.reader(f, delimiter='|')
    for row in reader:
        #print(row[0])
        termProf.append(row[0])

#mengambil korpus
dictionary = corpora.Dictionary.load(os.path.join(TEMP_FOLDER, 'lpse-jabar.dict'))
corpus = corpora.MmCorpus(os.path.join(TEMP_FOLDER, 'lpse-jabar.mm'))

#inisialisasi TFIDF (VSM)
tfidf = models.TfidfModel.load('file/temp/model/model.tfidf')
indextfidf = similarities.MatrixSimilarity(tfidf[corpus])
indextfidf.save(os.path.join(TEMP_FOLDER, 'index/tfidf.index'))#save tfidf index
#indextfidf = similarities.MatrixSimilarity.load(os.path.join(TEMP_FOLDER,
'index/tfidf.index'))

#inisialisasi TFIDF + Pembobotan profesional (VSM+PP)
indextfidfProf = similarities.MatrixSimilarity.load(os.path.join(TEMP_FOLDER,
'index/tfidfProf.index'))

#inisialisasi LSI
for jmlTopik in range(5,51):
    g="LSI"+str(jmlTopik)
    vars()[g] = models.LsiModel.load('file/temp/model/model.'+str(g))
for jmlTopik in range(5,51):
    g="LSI"+str(jmlTopik)
    h="index"+str(jmlTopik)
    vars()[h] = similarities.MatrixSimilarity(vars()[g][corpus])
for jmlTopik in range(5,51):
    g="LSI"+str(jmlTopik)
    h="index"+str(jmlTopik)
    vars()[h].save(os.path.join(TEMP_FOLDER, 'index/'+str(g)+'.index'))#save LSI index
#for jmlTopik in range(5,51):
#    g="LSI"+str(jmlTopik)
#    h="index"+str(jmlTopik)
#    vars()[h] = similarities.MatrixSimilarity.load(os.path.join(TEMP_FOLDER,
'index/'+str(g)+'.index'))#load LSI index

#inisialisasi LDA
for jmlTopik in range(5,51):
    g="LDA"+str(jmlTopik)
    vars()[g] = models.LdaModel.load('file/temp/model/model.'+str(g))
for jmlTopik in range(5,51):
    g="LDA"+str(jmlTopik)
    h="indexLDA"+str(jmlTopik)
    vars()[h] = similarities.MatrixSimilarity(vars()[g][corpus])
for jmlTopik in range(5,51):
    g="LDA"+str(jmlTopik)
    h="indexLDA"+str(jmlTopik)
    vars()[h].save(os.path.join(TEMP_FOLDER, 'index/'+str(g)+'.index'))#save LDA index
#for jmlTopik in range(5,51):
#    g="LDA"+str(jmlTopik)
#    h="indexLDA"+str(jmlTopik)
#    vars()[h] = similarities.MatrixSimilarity.load(os.path.join(TEMP_FOLDER,
'index/'+str(g)+'.index'))#load LDA index

#mengambil index dari term profesional
kamusTerm = []
for i in range (len(dictionary)):
    kamusTerm.append(dictionary[i])
indexTermProf=[]
for b in range (len(termProf)):
    indexTermProf.append(kamusTerm.index(termProf[b]))
print(indexTermProf)
```

```
#transformasi query menjadi vector
vec_tfidf = []
idQuery = []
inputQuery = []
for jmlTopik in range(5,51):
    h="vec_lsi"+str(jmlTopik)
    i="vec_lda"+str(jmlTopik)
    vars()[h]=[]
    vars()[i]=[]

with open(inputPertanyaan, newline='') as f :
    reader = csv.reader(f, delimiter='|')
    for row in reader:
        idQuery.append(row[1])
        inputQuery.append(row[2])
        doc=row[2]
        doc = teks.preProcessing(doc)
        vec_bow=dictionary.doc2bow(doc.lower().split())
        vec_tfidf.append(tfidf[vec_bow])
        #LSI
        for jmlTopik in range(5,51):
            g="LSI"+str(jmlTopik)
            h="vec_lsi"+str(jmlTopik)
            vars()[h].append(vars()[g][vec_bow])
        #LDA
        for jmlTopik in range(5,51):
            g="LDA"+str(jmlTopik)
            h="vec_lda"+str(jmlTopik)
            vars()[h].append(vars()[g][vec_bow])

#transformasi menggunakan pembobotan profesional
vec_tfidf1=[]
vec_tfidf_prof=[]
for c in range (len(vec_tfidf)):
    lst = list(vec_tfidf[c])
    for a in range (len(lst)):
        #periksa term apakah term profesional
        lst1 = list(lst[a])
        cekTerm=lst1[0]
        #print(cekTerm)
        if cekTerm not in indexTermProf:
            #print(cekTerm)
            lst1[1]=0.8*lst1[1]
        lst[a]=tuple(lst1)
    vec_tfidf1=tuple(lst)
    vec_tfidf_prof.append(vec_tfidf1)

#penghitungan similarity
#test bentar
simsTfidf = []
simsLSI = []

for jmlTopik in range(5,51):
    j="simsTfidfLsi"+str(jmlTopik)
    k="simsLSI"+str(jmlTopik)
    l="simsTfidfLda"+str(jmlTopik)
    o="simsLDA"+str(jmlTopik)
    vars()[j]=[]
    vars()[k]=[]
    vars()[l]=[]
    vars()[o]=[]

for y in range (0,50):
    simsTfidfA = indextfidf[vec_tfidf[y]] # perform a similarity query against the
corpus
    simsTfidf.append(sorted(enumerate(simsTfidfA), key=lambda item: -item[1]))
    #Similarity untuk LSI-proposed
    for jmlTopik in range(5,51):
        g="simsLSI"+str(jmlTopik)+"A"
        h="vec_lsi"+str(jmlTopik)
        i="index"+str(jmlTopik)
        j="simsTfidfLsi"+str(jmlTopik)
        k="simsLSI"+str(jmlTopik)
        #simsLSIXX
```

```
        simsNew=[]
        simsTemp=[]
        vars()[g] = vars()[i][vars()[h][y]]

        for bb in range(len(vars()[g])):
            #normalisasi
            simsTemp.append(normalisasi(vars()[g][bb]))
            #simsTemp.append (vars()[g][bb])
        vars()[g]=simsTemp
        simsNew = sorted(enumerate(simsTemp), key=lambda item: -item[1])
        vars()[k].append(simsNew)
        #
        ambang=0.9*vars()[k][y][0][1]
        #print("No Query:",y,", Nomor LSI Topik:",jmlTopik,", ambang:", ambang)
        #
        simsNew=[]
        indexOut=[]
        for bb in range(len(vars()[g])):
            if vars()[g][bb] < 0.7 or vars()[g][bb] < ambang :
                indexOut.append(bb)
            if vars()[g][bb]>= ambang :
                simsNew.append((1*simsTfidfA[bb])+(1*vars()[g][bb]))
            else:
                simsNew.append(vars()[g][bb])

                #simsNew.append((1*simsTfidfA[bb])+(1*vars()[g][bb]))
        simsNew = sorted(enumerate(simsNew), key=lambda item: -item[1])

        for value in simsNew[:]:
            if value[0] in indexOut:
                simsNew.remove(value)

        vars()[j].append(simsNew)
        #

        #simsLDAXX
        l="simsLDA"+str(jmlTopik)+"A"
        m="vec_lda"+str(jmlTopik)
        n="indexLDA"+str(jmlTopik)
        o="simsLDA"+str(jmlTopik)

        vars()[l] = vars()[n][vars()[m][y]] # perform a similarity query against the
corpus
        vars()[o].append(sorted(enumerate(vars()[l]), key=lambda item: -item[1]))

#profesional tf-idf
simsTfidfProf = []
for y in range (0,50):
    simsTfidfA = indextfidfProf[vec_tfidf_prof[y]] # perform a similarity query
against the corpus
    simsTfidfProf.append(sorted(enumerate(simsTfidfA), key=lambda item: -item[1]))

#buat variabel kosong untuk q0 - q49
for a in range(0,50):
    b="q"+str(a)
    vars()[b] = []

with open(fileJawaban, newline='') as f :
    reader = csv.reader(f, delimiter='|')
    for row in reader:
        for a in range(0,50):
            b="q"+str(a)
            if row[a]:
                vars()[b].append(row[a])

#TF-IDF hitung MAP
files = glob.glob(folderHasil+'/TFIDF/'+'/*')
for f in files:
    os.remove(f)
os.popen('rm -f file/data/output/result/hasilTFIDF.txt')

for a in range(0,50):
    b="a"+str(a)
    vars()[b] = []
```

```
MAP=0
prec1=0
prec3=0
prec5=0
averagePrecR=0
for a in range (0,50):
    m=0
    n=0
    p=0
    w=0
    p3=0
    p5=0
    pr=0

    b="q"+str(a)
    c="a"+str(a)
    panjangJawaban=len(vars()[b])
    e=[]
    u=1
    saveFile=open(folderHasil+'/TFIDF/'+str(a)+'.txt','a+')
    saveFile.write("Nomor ID Query : "+str(idQuery[a])+"\n")
    saveFile.write("Pertanyaan/Query : "+str(inputQuery[a])+"\n")
    saveFile.write("\n")
    saveFile.close()
    for y in range (len(simsTfidf[a])):
        indexPertanyaan=simsTfidf[a][y][0]
        pertanyaan = corpusPertanyaan[indexPertanyaan]
        idPertanyaan = corpusID[indexPertanyaan]
        #print(y+1,".")
        jawaban = corpusJawaban[indexPertanyaan]
        #print("Q: ",pertanyaan,"\nA: ",jawaban)
        saveFile=open(folderHasil+'/TFIDF/'+str(a)+'.txt','a+')
        saveFile.write("Nomor ID Pertanyaan : "+idPertanyaan+"\n")
        saveFile.write("Q : "+pertanyaan)
        saveFile.write("\n")
        saveFile.write("A : "+jawaban)
        saveFile.write("\n")
        saveFile.close()
        e.append(idPertanyaan)
        if idPertanyaan in vars()[b]:
            vars()[c].append(1)
            if u <4:
                p3=p3+1
            if u <6:
                p5=p5+1
        else:
            vars()[c].append(0)
        u=u+1
    prec1=prec1+vars()[c][0]
    prec3=prec3+(p3/3)
    prec5=prec5+(p5/5)
    #Hitung MAP
    patr = 0 #P@R
    benar = 0
    for dd in range (len(vars()[c])):
        if vars()[c][dd]==1:
            benar = benar+1
            patr = patr+(benar/(dd+1))
    patr=patr/panjangJawaban
    MAP=MAP+patr
    #
MAP=MAP/50
prec1=prec1/50
prec3=prec3/50
prec5=prec5/50

print("Nilai MAP : ",MAP)
print("prec@1 ,",prec1)
print("prec@3 ,",prec3)
print("prec@5 ,",prec5)

for f in range(0,50):
    g="a"+str(f)
    saveFile=open('file/data/output/result/hasilTFIDF.txt','a+')
```

```
    saveFile.write("Pertanyaan "+str(f+1)+",")
    for h in range(len(vars()[g])):
        saveFile.write(str(vars()[g][h])+",")
    saveFile.write("\n")
    saveFile.close()

saveFile=open('file/data/output/result/hasilTFIDF.txt','a+')
saveFile.write("TFIDF, Rata-rata
P@1,P@3,P@5,MAP:,"+str(prec1)+","+str(prec3)+","+str(prec5)+","+str(MAP)+"\n")
saveFile.close()
print("TFIDF model, done..")

#TF-IDF (VSM+PP) hitung MAP
files = glob.glob(folderHasil+'/TFIDF/'+'/*')
for f in files:
    os.remove(f)
os.popen('rm -f file/data/output/result/hasilTFIDF.txt')

for a in range(0,50):
    b="a"+str(a)
    vars()[b] = []
MAP=0
prec1=0
prec3=0
prec5=0
averagePrecR=0
for a in range (0,50):
    m=0
    n=0
    p=0
    w=0
    p3=0
    p5=0
    pr=0

    b="q"+str(a)
    c="a"+str(a)
    panjangJawaban=len(vars()[b])
    #print(vars()[b])
    e=[]
    u=1
    saveFile=open(folderHasil+'/TFIDF/'+str(a)+'.txt','a+')
    saveFile.write("Nomor ID Query : "+str(idQuery[a])+"\n")
    saveFile.write("Pertanyaan/Query : "+str(inputQuery[a])+"\n")
    saveFile.write("\n")
    saveFile.close()
    for y in range (len(simsTfidfProf[a])):
        indexPertanyaan=simsTfidfProf[a][y][0]
        pertanyaan = corpusPertanyaan[indexPertanyaan]
        idPertanyaan = corpusID[indexPertanyaan]
        #print(y+1,".")
        jawaban = corpusJawaban[indexPertanyaan]
        #print("Q: ",pertanyaan,"\nA: ",jawaban)
        saveFile=open(folderHasil+'/TFIDF/'+str(a)+'.txt','a+')
        saveFile.write("Nomor ID Pertanyaan : "+idPertanyaan+"\n")
        saveFile.write("Q : "+pertanyaan)
        saveFile.write("\n")
        saveFile.write("A : "+jawaban)
        saveFile.write("\n")
        saveFile.close()
        e.append(idPertanyaan)
        if idPertanyaan in vars()[b]:
            vars()[c].append(1)
            if u <4:
                p3=p3+1
            if u <6:
                p5=p5+1
        else:
            vars()[c].append(0)
        u=u+1
    prec1=prec1+vars()[c][0]
    prec3=prec3+(p3/3)
    prec5=prec5+(p5/5)
    #Hitung MAP
```

```
    patr = 0 #P@R
    benar = 0
    for dd in range (len(vars()[c])):
        if vars()[c][dd]==1:
            benar = benar+1
            patr = patr+(benar/(dd+1))
    patr=patr/panjangJawaban
    MAP=MAP+patr
    #
MAP=MAP/50
prec1=prec1/50
prec3=prec3/50
prec5=prec5/50

print("Nilai MAP : ",MAP)
print("prec@1 ,",prec1)
print("prec@3 ,",prec3)
print("prec@5 ,",prec5)

for f in range(0,50):
    g="a"+str(f)
    saveFile=open('file/data/output/result/hasilTFIDFPP.txt','a+')
    saveFile.write("Pertanyaan "+str(f+1)+",")
    for h in range(len(vars()[g])):
        saveFile.write(str(vars()[g][h])+",")
    saveFile.write("\n")
    saveFile.close()

saveFile=open('file/data/output/result/hasilTFIDFPP.txt','a+')
saveFile.write("TFIDF, Rata-rata
P@1,P@3,P@5,MAP:,"+str(prec1)+","+str(prec3)+","+str(prec5)+","+str(MAP)+"\n")
saveFile.close()
print("TFIDF (VSM+PP) model, done..")

#SEMUA LSI Baru hitung MAP
os.popen('rm -f file/data/output/result/hasilSemuaLSI.txt')

for jmlTopik in range(5,51):
    LSIx="LSI"+str(jmlTopik)
    h="simsLSI"+str(jmlTopik)

    for a in range(0,50):
        b="a"+str(a)
        vars()[b] = []
    MAP=0
    prec1=0
    prec3=0
    prec5=0
    averagePrecR=0

    for a in range (0,50):
        m=0
        n=0
        p=0
        w=0
        p3=0
        p5=0
        pr=0

        b="q"+str(a)
        c="a"+str(a)
        panjangJawaban=len(vars()[b])
        e=[]
        u=1
        for y in range (len(vars()[h][a])):
            indexPertanyaan=vars()[h][a][y][0]
            pertanyaan = corpusPertanyaan[indexPertanyaan]
            idPertanyaan = corpusID[indexPertanyaan]
            #print(y+1,".")
            jawaban = corpusJawaban[indexPertanyaan]
            #print("Q: ",pertanyaan,"\nA: ",jawaban)
            e.append(idPertanyaan)
            if idPertanyaan in vars()[b]:
                vars()[c].append(1)
```

```
                if u <4:
                    p3=p3+1
                if u <6:
                    p5=p5+1
            else:
                vars()[c].append(0)
            u=u+1
        prec1=prec1+vars()[c][0]
        prec3=prec3+(p3/3)
        prec5=prec5+(p5/5)
        #Hitung MAP
        patr = 0 #P@R
        benar = 0
        for dd in range (len(vars()[c])):
            if vars()[c][dd]==1:
                benar = benar+1
                patr = patr+(benar/(dd+1))
        patr=patr/panjangJawaban
        MAP=MAP+patr
        #
    MAP=MAP/50
    prec1=prec1/50
    prec3=prec3/50
    prec5=prec5/50

    print("Nilai MAP : ",MAP)
    print("prec@1 ,",prec1)
    print("prec@3 ,",prec3)
    print("prec@5 ,",prec5)

    saveFile=open('file/data/output/result/hasilSemuaLSI.txt','a+')
    saveFile.write("["+str(LSIx)+"] Rata-rata
P@1,P@3,P@5,MAP:,"+str(prec1)+","+str(prec3)+","+str(prec5)+","+str(MAP)+"\n")
    saveFile.close()
    print(str(LSIx),", done..")

#SEMUA LDA Baru hitung MAP
os.popen('rm -f file/data/output/result/hasilSemuaLDA.txt')

for jmlTopik in range(5,51):
    LDAx="LDA"+str(jmlTopik)
    h="simsLDA"+str(jmlTopik)

    for a in range(0,50):
        b="a"+str(a)
        vars()[b] = []
    MAP=0
    prec1=0
    prec3=0
    prec5=0
    averagePrecR=0

    for a in range (0,50):
        m=0
        n=0
        p=0
        w=0
        p3=0
        p5=0
        pr=0

        b="q"+str(a)
        c="a"+str(a)
        panjangJawaban=len(vars()[b])
        #print(vars()[b])
        e=[]
        u=1
        for y in range (len(vars()[h][a])):
            indexPertanyaan=vars()[h][a][y][0]
            pertanyaan = corpusPertanyaan[indexPertanyaan]
            idPertanyaan = corpusID[indexPertanyaan]
            #print(y+1,".")
            jawaban = corpusJawaban[indexPertanyaan]
            #print("Q: ",pertanyaan,"\nA: ",jawaban)
```

```
            e.append(idPertanyaan)
            if idPertanyaan in vars()[b]:
                vars()[c].append(1)
                if u <4:
                    p3=p3+1
                if u <6:
                    p5=p5+1
            else:
                vars()[c].append(0)
            u=u+1
        prec1=prec1+vars()[c][0]
        prec3=prec3+(p3/3)
        prec5=prec5+(p5/5)
        #Hitung MAP
        patr = 0 #P@R
        benar = 0
        for dd in range (len(vars()[c])):
            if vars()[c][dd]==1:
                benar = benar+1
                patr = patr+(benar/(dd+1))
        patr=patr/panjangJawaban
        MAP=MAP+patr
        #
    MAP=MAP/50
    prec1=prec1/50
    prec3=prec3/50
    prec5=prec5/50

    print("Nilai MAP : ",MAP)
    print("prec@1 ,",prec1)
    print("prec@3 ,",prec3)
    print("prec@5 ,",prec5)

    saveFile=open('file/data/output/result/hasilSemuaLDA.txt','a+')
    saveFile.write("["+str(LDAx)+"] Rata-rata
P@1,P@3,P@5,MAP:,"+str(prec1)+","+str(prec3)+","+str(prec5)+","+str(MAP)+"\n")
    saveFile.close()
    print(str(LDAx),", done..")

#SEMUA VSM+LSI Baru hitung MAP --pake  threshold
os.popen('rm -f file/data/output/result/hasilTFIDF+LSI.txt')

for jmlTopik in range(5,51):
    LSIx="LSI"+str(jmlTopik)
    h="simsTfidfLsi"+str(jmlTopik)

    #VSM+LSI
    files = glob.glob(folderHasil+'/'+str(LSIx)+'/*')
    for f in files:
        os.remove(f)

    for a in range(0,50):
        b="a"+str(a)
        vars()[b] = []
    MAP=0
    prec1=0
    prec3=0
    prec5=0
    averagePrecR=0

    for a in range (0,50):
        m=0
        n=0
        p=0
        w=0
        p3=0
        p5=0
        pr=0

        b="q"+str(a)
        c="a"+str(a)
        panjangJawaban=len(vars()[b])
        e=[]
        u=1
```

```
        saveFile=open(str(folderHasil)+'/'+str(LSIx)+'/'+str(a)+'.txt','a+')
        saveFile.write("Nomor ID Query : "+str(idQuery[a])+"\n")
        saveFile.write("Pertanyaan/Query : "+str(inputQuery[a])+"\n")
        saveFile.write("\n")
        saveFile.close()
        for y in range (len(vars()[h][a])):
            indexPertanyaan=vars()[h][a][y][0]
            pertanyaan = corpusPertanyaan[indexPertanyaan]
            idPertanyaan = corpusID[indexPertanyaan]
            #print(y+1,".")
            jawaban = corpusJawaban[indexPertanyaan]
            #print("Q: ",pertanyaan,"\nA: ",jawaban)
            saveFile=open(str(folderHasil)+'/'+str(LSIx)+'/'+str(a)+'.txt','a+')
            saveFile.write("Nomor ID Pertanyaan : "+idPertanyaan+"\n")
            saveFile.write("Q : "+pertanyaan)
            saveFile.write("\n")
            saveFile.write("A : "+jawaban)
            saveFile.write("\n")
            saveFile.close()
            e.append(idPertanyaan)
            if idPertanyaan in vars()[b]:
                vars()[c].append(1)
                if u <4:
                    p3=p3+1
                if u <6:
                    p5=p5+1
            else:
                vars()[c].append(0)
            u=u+1
        prec1=prec1+vars()[c][0]
        prec3=prec3+(p3/3)
        prec5=prec5+(p5/5)
        #Hitung MAP
        patr = 0 #P@R
        patrTemp = 0
        benar = 0
        for dd in range (len(vars()[c])):
            if vars()[c][dd]==1:
                benar = benar+1
                patrTemp = benar/(dd+1)
                patr = patr+patrTemp
        #patr=patr/benar
        if benar != 0:
            patr=patr/benar
        else:
            patr=0
        saveFile=open("file/data/output/result/patrProposed.txt",'a+')
        saveFile.write(str(patr))
        saveFile.write("\n")
        saveFile.close()
        MAP=MAP+patr
        #
    MAP=MAP/50
    prec1=prec1/50
    prec3=prec3/50
    prec5=prec5/50

    print("Nilai MAP : ",MAP)
    print("prec@1 ,",prec1)
    print("prec@3 ,",prec3)
    print("prec@5 ,",prec5)

    saveFile=open('file/data/output/result/hasilTFIDF+LSI.txt','a+')
    saveFile.write("["+str(LSIx)+"] Rata-rata
P@1,P@3,P@5,MAP:,"+str(prec1)+","+str(prec3)+","+str(prec5)+","+str(MAP)+"\n")
    saveFile.close()
    print(str(LSIx),"+VSM, done..")
```

BIOGRAFI PENULIS

Syamsul Bahri, Magister Teknik Elektro, Bidang Keahlian Telematika/Pengelola TIK Pemerintahan Angkatan Tahun 2016, Institut Teknologi Sepuluh Nopember, Surabaya. Lahir di Lombok Barat, pada tanggal 11 Juli 1982. Pada saat ini bekerja sebagai Pejabat Fungsional Pranata Komputer pada Dinas Komunikasi, Informatika & Statistik (Diskominfotik) Pemerintah Daerah Provinsi Nusa Tenggara Barat. Jika ada yang ingin berkorespondensi dapat melalui alamat email: syam785@yahoo.com.

Riwayat Pendidikan :

1988 - 1994	SDN 1 Dasan Tapen, Gerung, Lombok Barat
1994 - 1997	SMPN 1 Gerung, Lombok Barat
1997 - 2000	SMKN 3 Mataram
2000 - 2004	D3 Teknik Elektronika, UNY, Yogyakarta
2005 - 2011	S1 Teknik Informatika, STMIK Bumigora, Mataram

Riwayat Pekerjaan :

2005 - 2013	Pengumpul dan Pengolah Data Telematika, Biro Umum, Setda Provinsi Nusa Tenggara Barat
2013 - 2016	Fungsional Pranata Komputer, Biro Umum, Setda Provinsi Nusa Tenggara Barat
2017 - sekarang	Fungsional Pranata Komputer, Dinas Kominfotik Provinsi Nusa Tenggara Barat