TESIS – IF 185401

PEMBOBOTAN *TERM* EKSPANSI *QUERY* BERBASIS *WORD EMBEDDINGS* DAN *INVERSE BOOK FREQUENCY* UNTUK PENCARIAN DOKUMEN

DWI ARI SURYANINGRUM
05111850010047

Dosen Pembimbing
Prof. Dr. Agus Zainal Arifin, S.Kom, M.Kom
19720809 199512 1 001

Departemen Teknik Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
2020

[*Halaman ini sengaja dikosongkan*]

LEMBAR PENGESAHAN TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M. Kom)
di
Institut Teknologi Sepuluh Nopember
Oleh:
DWI ARI SURYANINGRUM
NRP: 05111850010047

Tanggal Ujian: 14 Juli 2020
Periode Wisuda: September 2020

Disetujui oleh:

Pembimbing:

1. Prof. Dr. Agus Zainal Arifin, S.Kom., M.Kom. .....
NIP: 19720809 199512 1 001

Penguji:

1. Prof. Ir. Handayani Tjandrasa, M.Sc., Ph.D. .....
NIP: 19490823 197603 2 001

2. Dr. Ahmad Saikhu, S.Si., M.T. .....
NIP: 19710718 200604 1 001

3. Hadziq Fabroyir, S.Kom., Ph.D. .....
NIP: 19860227 201903 1 006

Kepala Departemen Teknik Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas

Dr. Eng. Chastine Fatichah, S.Kom., M.Kom
NIP: 19751220 200112 2 002

[Halaman ini sengaja dikosongkan]

PEMBOBOTAN *TERM* EKSPANSI *QUERY* BERBASIS *WORD EMBEDDINGS* DAN *INVERSE BOOK FREQUENCY* UNTUK PENCARIAN DOKUMEN

Nama Mahasiswa	:	Dwi Ari Suryaningrum
NRP	:	05111850010047
Pembimbing	:	Prof. Dr. Agus Zainal Arifin, S.Kom., M.Kom

ABSTRAK

Penelitian pembobotan *term* TF-IDF dilakukan untuk mencari seberapa penting *term* dalam dokumen. Namun, TF-IDF tidak dapat mengatasi permasalahan dokumen yang memiliki beberapa topik. Pembobotan *term* IBF yang digabungkan dengan TF-IDF dapat mengatasi permasalahan tersebut. Pembobotan IBF menganggap *term* yang sering muncul pada banyak dokumen dan kategori akan bernilai kecil dan dianggap kurang penting.

Pencarian dokumen relevan dapat ditingkatkan dengan melakukan *query expansion* (QE) dengan *word embeddings* (WE). Penelitian QE yang telah dilakukan sebelumnya memilih *term* kandidat untuk dijadikan *term* hasil ekspansi *query* berdasarkan kedekatan semantiknya yang memiliki tingkat korelasi tinggi dengan *query* asli. Namun, tidak semua *term* dari hasil ekspansi *query* tersebut dianggap penting untuk dilakukan *retrieve* dokumen. Beberapa *term* hasil QE mungkin tidak berguna atau tidak relevan meskipun memiliki tingkat korelasi yang tinggi dengan *query* aslinya. Hal ini juga memungkinkan dapat mengurangi kualitas kemiripan dari hasil dokumen yang di*retrieve,* terutama ketika ada *term* QE yang lebih tidak relevan daripada yang relevan. Permasalahan yang muncul adalah cara memilih *term* hasil QE yang terbaik untuk meningkatkan kualitas hasil pencarian dokumen.

Penelitian ini mengusulkan sebuah metode baru pembobotan term pada hasil ekspansi *query* berdasarkan tingkat korelasi *term* terhadap *query* dan frekuensi *term* menggunakan metode *word embeddings* dan *Inverse Book Frequency* (IBF) untuk pencarian dokumen relevan. *Term* hasil QE dari WE dihitung pembobotannya dengan penggabungan metode TF-IDF dan IBF (TF-IDF-IBF). Bobot *term* hasil QE didapatkan dengan mengalikan nilai similaritas *term* dari WE dengan nilai bobot TF-IDF-IBF *term* tersebut. Tahap selanjutnya adalah memilih *term* QE yang memiliki bobot terbesar sebanyak 5 untuk digunakan pada proses pencarian dokumen. Metode baru pembobotan *term* digunakan untuk mendapatkan *term* hasil QE yang memiliki korelasi tinggi dengan *query* asli sekaligus merupakan *term* yang representatif dalam dokumen dan digunakan untuk proses pencarian dokumen.

Metode baru yang diusulkan dapat menghasilkan nilai *precision, recall*, dan *F-Score* yang lebih tinggi dengan menambahkan pemilihan QE terlebih dahulu sebelum proses pencarian dokumen, yaitu dengan *f-score* sebesar 0,743. Kinerja sistem yang didapatkan dapat lebih optimal jika menggunakan parameter jumlah

pemilihan *term* hasil QE yang kecil. Semakin besar pemilihan parameter, dapat menghasilkan dokumen yang kurang relevan terhadap *query* asli.

Kata Kunci: *Query Expansion*, *Word Embeddings*, pembobotan *Term*, TF, IDF, IBF

TERM WEIGHTING QUERY EXPANSION BASED ON WORD EMBEDDINGS AND INVERSE BOOK FREQUENCY FOR DOCUMENT SEARCH

Student Name	:	Dwi Ari Suryaningrum
Student Identity Number	:	05111850010047
Supervisor	:	Prof. Dr. Agus Zainal Arifin, S.Kom., M.Kom

ABSTRACT

The TF-IDF term weighting study was conducted to find out how important the terms are in the document. However, TF-IDF cannot overcome document problems that have multiple topics. IBF term weighting combined with TF-IDF can overcome this problem. IBF weighting considers terms that often appear in many documents and categories to be of little value and are considered less important.

The search for relevant documents can be increased by querying expansion (QE) with word embeddings (WE). QE research that has been done previously chose the candidate term to be used as the result of query expansion based on its semantic proximity which has a high degree of correlation with the original query. However, not all terms of the query expansion result are considered important to retrieve documents. Some QE terms may not be useful or relevant even though they have a high degree of correlation with the original query. This also makes it possible to reduce the quality of the similarity of the results of the documents that are returned, especially when there are QE terms that are more irrelevant than relevant. The problem that arises is how to choose the best QE term results to improve the quality of document search results.

This study proposes a new method of weighting the term on the results of query expansion based on the level of term correlation to the query and term frequency using the word embeddings and Inverse Book Frequency (IBF) method for searching relevant documents. The QE term results from WE are calculated by weighting them by combining the TF-IDF and IBF methods (TF-IDF-IBF). The QE term weights are obtained by multiplying the term similarity value from WE with the TF-IDF-IBF term weight value. The next step is to choose the term QE which has the largest weight of 5 to be used in the document search process. The new term weighting method is used to get QE term results that have a high correlation with the original query as well as being a representative term in the document and used for the document search process.

The proposed new method can produce a higher precision, recall, and F-Score by adding the QE selection first before the document search process, with f-score of 0.743. The system performance obtained can be optimized if it uses a small number of QE term selection parameters. The greater the selection of parameters, can produce documents that are less relevant to the original query.

Keywords : Query Expansion, Word Embeddings, Term Weighting, TF, IDF, IBF

KATA PENGANTAR

Puji syukur kehadirat Allah Subahanallahu Wa Ta'ala atas segala Rahmat dan Ridho-Nya penulis dapat menyelesaikan tesis dengan judul "Pembobotan *Term* Ekspansi *Query* Berbasis *Word Embeddings* dan *Inverse Book Frequency* untuk Pencarian Dokumen". Penulis mengucapkan terima kasih atas doa dan dukungan kepada:

1. Bapak Prof. Dr. Agus Zainal Arifin, S.Kom., M.Kom. sebagai Dosen Pembimbing yang telah membimbing penulis dalam menyusun tesis dengan dukungan semangat dan ilmu melalui *progress* mingguan di dalam grup Zemi.
2. Ibu Prof. Ir. Handayani Tjandrasa, M.Sc., PhD.; Bapak Dr. Ahmad Saikhu, S.Si., M.T.; serta Bapak Hadziq Fabroyir, S.Kom., Ph.D. sebagai Dosen Penguji yang telah memberikan masukan yang berharga bagi tesis ini.
3. Kedua Orang Tua dan seluruh keluarga besar atas segala nasehat, kasih sayang, perhatian dan kesabarannya dalam membesarkan dan mendidik penulis, serta senantiasa tiada henti memberikan doa dan semangat demi terselesaikannya tesis ini.
4. Teman-teman S2 TC 2018 : Mbak Ana, Arif, Fahmi, Adam, Shaza, *partner* dosen pembimbing sama yang selalu bersama-sama berproses dengan penulis, dan Maryamah, Mbak Rizka, Mbak Raras yang sudah membantu dalam memberikan ilmu di dalam grup bimbingan Zemi serta seluruh jajaran teman-teman S2 TC 2018 yang tidak dapat saya sebut secara satu persatu.
5. Sahabat-sahabatku, Dea, Ayu, Mbak Yuli, Annisa, dan Nanda yang selalu memberi semangat, saling berbagi, memberi bantuan dan saran menyelesaikan tesis ini.
6. Seluruh pihak yang terlibat dalam memberikan semangat kepada penulis untuk menyusun tesis ini dengan berani dan tepat.

Semoga Rahmad dan Ridho Allah menyertai setiap pihak yang telah terlibat dalam memberi motivasi kepada penulis untuk menyelesaikan tesis. Penulis menyadari bahwa penelitian ini masih sangat jauh dari sempurna. Oleh karena itu,

kritik dan saran sangat dibutuhkan oleh penulis agar bisa lebih baik lagi. Akhir kata, semoga penelitian yang tertuang dalam buku tesis ini bermanfaat bagi bidang informasi hadis sesuai cita-cita penulis dan dikembangkan lebih lanjut.

Surabaya, 14 Juli 2020

Dwi Ari Suryaningrum

DAFTAR ISI

DAFTAR GAMBAR

[*Halaman ini sengaja dikosongkan*]

DAFTAR TABEL

[*Halaman ini sengaja dikosongkan*]

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Perkembangan data saat ini sangat pesat terutama dalam bentuk teks. Data berupa teks banyak tersebar dari berbagai media antara lain media sosial, blog, berita. Perkembangan ini memberikan suatu istilah yang kini mulai populer yaitu *big data* dan cabang-cabang ilmu yang mengolahnya biasa dikategorikan ke dalam *data science*. Salah satu cabang ilmu yang berkembang adalah dalam sistem temu kembali informasi (*information retrieval system*), yaitu suatu sistem yang dapat mengakomodasi dalam pencarian teks secara tepat dan akurat berdasarkan suatu *query* dari pengguna.

Sistem temu kembali informasi adalah proses menemukan informasi yang paling relevan yang dapat digunakan untuk kebutuhan informasi pengguna. Sistem ini menggunakan metode tertentu untuk mendapatkan dokumen-dokumen yang relevan dengan *query* yang dimasukkan pengguna. Sistem temu kembali informasi yang baik memungkinkan pengguna untuk menentukan apakah isi dari dokumen yang diterima memenuhi kebutuhan secara cepat dan akurat.

Sebagian besar sistem *retrieval* menunjukkan kelemahan relatif dalam mengambil dokumen yang relevan, terutama ketika beberapa kata kunci digunakan untuk memodelkan kebutuhan informasi pengguna (Colace et al., 2015). Model *information retrieval* yang telah diusulkan sebelumnya, sering mengandalkan model *bag of words* untuk merepresentasikan dokumen dan *query* (Colace et al., 2015).

Pendekatan *query expansion* dilakukan untuk meningkatkan pencarian informasi dalam pencarian dokumen. *Query expansion* terbukti bermanfaat dalam meningkatkan efisiensi dan ketepatan pengambilan informasi dalam berbagai penelitian (Choi et al., 2016). Colace (Colace et al., 2015) mengusulkan metode *query expansion* yang secara otomatis dapat mengekstraksi sekumpulan pasangan kata yang berbobot dari kumpulan dokumen dengan topik yang terkait yang disediakan oleh *relevance feedback*.

Query expansion dapat dilakukan dengan menggunakan *word embeddings* untuk mendapatkan *query* baru. *Word embeddings* merupakan cara merepresentasikan kata-kata dalam Bahasa alami dengan mempertahankan kemiripan semantik dan sitaksis di antara kata-kata tersebut (Bintana et al., 2018). Ada beberapa teknik *word embeddings* yang dapat dilakukan antara lain dengan *Word2Vec*, *GloVe*, dan *fasttext*.

Mikolov (Mikolov et al., 2013) memperkenalkan *Word2Vec* dengan menggunakan model *Skip-gram*. Model ini tidak memprediksikan kata berdasarkan pada konteks, tapi mencoba untuk memaksimalkan klasifikasi sebuah kata berdasarkan kata lain dalam kalimat yang sama. Penelitian yang dilakukan Mikolov (Mikolov et al., 2013) menemukan bahwa kualitas vektor kata yang dihasilkan dapat diperbaiki dengan meningkatkan jarak dan kompleksitas komputasi.

Pennington (Pennington et al., 2014) mengusulkan model *word embeddings* yaitu *Global Vector* (*GloVe*). *GloVe* dibuat dengan menggunakan dua metode, yaitu *global matrix factorization* dan *local context window*. Wang (Wang et al., 2018) membandingkan *word embeddings* teknik *fasttext* dengan *GloVe* dan *Google News* untuk proses bahasa alami biomedis. Penelitian tersebut menggunakan *fasttext* untuk menghitung vektor kata untuk istilah-istilah medis yang tidak ada pada kosakata *word embeddings*. Penelitian Wang dkk dapat menghasilkan kesamaan *term* medis yang lebih relevan dibandingkan dari *GloVe* dan *Google News*.

Penelitian yang dilakukan Hidayatin dan Rahutomo (Hidayatin dan Rahutomo, 2018) mengevaluasi *query expansion* untuk aplikasi *chatbot*. Penelitian ini mengusulkan sistem *chatbot* dengan *knowledge base* dari pertanyaan-pertanyaan yang sering diajukan. Mekanisme *query expansion* diimplementasikan dengan Tesaurus. Antara *query* yang diinputkan dengan pertanyaan yang sering diajukan akan dihitung kemiripannya menggunakan *cosine similarity*. Nilai *cosine similarity* yang didapat dari penelitian ini memiliki akurasi yang rendah. Hal ini dikarenakan pada penelitian tersebut *query* asal dan *query* hasil ekspansi diperlakukan sama, sehingga kurang efektif untuk bisa mengembalikan dokumen yang lebih relevan. Penelitian tersebut tidak memperhatikan persebaran *term* pada kelas yang berbeda-beda.

Penelitian mengenai pencarian dokumen relevan dengan melakukan perangkingan dokumen telah banyak dilakukan. Perangkingan dokumen dilakukan dengan pembobotan frekuensi kemunculan *term* pada dokumen yang ada antara lain dilakukan oleh Hakim (Hakim et al., 2014). Penelitian tersebut mengusulkan klasifikasi artikel berita *online* Bahasa Indonesia dengan menggunakan pembobotan TF-IDF.

Ren dan Sohrab (Ren dan Sohrab, 2013) mengembangkan pembobotan TF-IDF dengan menambahkan pembobotan *term* berbasis indeks kelas yang disebut *Inverse Class Frequency* (ICF) dan variasinya, antara lain yaitu *Inverse Class Space density Frequency* (ICSdF) untuk pembobotan *term* pada dokumen Bahasa Inggris. ICF dan ICSdF digunakan untuk membobotkan *term* yang sering muncul pada banyak kelas akan bernilai kecil karena dianggap kurang penting.

Fauzi (Fauzi et al., 2014) mengembangkan metode perhitungan bobot *term* pada dokumen yang disebut TF-IDF-ICF-IBF. Metode yang diusulkan merupakan penggabungan antara ICF dan *Inverse Book Frequency* (IBF). ICF berbasis pada kelas yang merupakan pengelompokan secara otomatis melalui pola data statistik. Sementara *Inverse Book Frequency* (IBF) berbasis pada buku yang merupakan pengelompokan manual yang dilakukan sendiri oleh pengguna secara *semantic* (Fauzi et al., 2014). Namun, hasil dari metode-metode yang dilakukan sebelumnya belum dapat menampilkan dokumen yang relevan secara maksimal. Selain itu, penelitian tersebut belum dapat melakukan perluasan *query* dengan metode ekspansi *query*.

Pendekatan ekspansi *query* dapat digunakan pada kasus pencarian dokumen yang relevan pada dokumen. Bintana (Bintana et al., 2018) melakukan penelitian untuk pencarian tanya-jawab menggunakan *convolutional neural network* pada topik agama bahasa Indonesia. Penelitian tersebut menggunakan *word embeddings* untuk pembelajaran representasi kata terdistribusi. Namun, hasil *mean average precision* (MAP) dari penelitian tersebut masih rendah. Selain itu, penelitian tersebut masih dalam bentuk pemodelan dari metode yang diusulkan, sehingga belum dilakukan proses pengelompokkan dokumen.

Metode pencarian dokumen yang relevan dilakukan dengan menghitung beberapa aspek antara lain kemiripan dokumen dengan *query* dan seberapa penting

term tersebut dalam beberapa dokumen. Seberapa penting suatu *term* dapat ditentukan apabila tingkat kemunculan kata yang dicari dalam beberapa dokumen kecil sehingga hasil pencarian dokumen yang diperoleh akan lebih relevan. Namun, bagaimana cara untuk dapat menemukan dokumen yang relevan dengan masukan *query* dari banyak dokumen yang memiliki berbagai topik/kategori.

Pencarian dokumen relevan dapat ditingkatkan dengan melakukan ekspansi *query* (QE) dengan *word embeddings* (WE). Teknik QE dengan WE memungkinkan setiap *term query* untuk menentukan kandidat ekspansi yang berkorelasi dengan *term query* asli. *Term-term* baru dari ekspansi *query* dapat digunakan untuk perluasan dalam pencocokan *query* dengan dokumen yang ada.

Metode QE menggabungkan *terms* kandidat berdasarkan proses *voting*, dimana *terms* baru dihasilkan oleh kedekatan semantik dari *terms* kosakata dengan semua *terms query*. metode ini bertujuan untuk menangkap pengertian *query* secara global untuk tujuan ekspansi. Namun, terdapat permasalahan mengenai metode QE yaitu *terms query* asli mana yang lebih berguna untuk diekspansi. Francis dkk (Francis et.al., 2019) mengatasi masalah tersebut dengan mengusulkan metode QE global baru (*V2Q*) menggunakan *word embeddings* yang menganggap *query* sebagai keseluruhan. Penelitian tersebut juga melakukan pengelompokkan *terms* kandidat untuk mengatasi masalah disambiguasi dan meningkatkan matrix *recall* dan *precision* di IR tanpa *relevance feedback*.

Penelitian QE yang telah dilakukan sebelumnya memilih *terms* kandidat untuk dijadikan *terms* hasil ekspansi *query* berdasarkan kedekatan semantiknya yang memiliki tingkat korelasi tinggi dengan *query* asli. Namun, tidak semua *terms* dari hasil ekspansi *query* tersebut dianggap penting untuk dilakukan *retrieve* dokumen. Beberapa *terms* hasil ekspansi *query* mungkin tidak berguna atau kurang relevan, meskipun memiliki tingkat korelasi yang tinggi dengan *query* asli. Selain itu, penggunaan hasil ekspansi *query* tersebut dapat mengurangi kualitas dari hasil yang diperoleh, terutama ketika ada *terms* ekspansi *query* yang lebih tidak relevan daripada yang relevan dengan dokumen yang digunakan.

Pemilihan *term* hasil QE bertujuan untuk menghapus *term* hasil QE yang tidak relevan dari kumpulannya. Set *term* hasil QE yang dipilih harus berisi informasi yang cukup dan berhubungan dengan dokumen asli. Dengan melakukan

pemilihan QE terlebih dahulu, diharapkan dapat meningkatkan kualitas dokumen yang di-*retrieve* sesuai dengan *query* asli yang diberikan pengguna. Permasalahan yang muncul adalah bagaimana cara memilih *term* hasil QE yang terbaik untuk meningkatkan kualitas hasil pencarian dokumen.

Penelitian ini mengusulkan sebuah metode baru pembobotan term pada hasil ekspansi *query* berdasarkan tingkat korelasi *term* terhadap *query* dan frekuensi *term* menggunakan metode *word embeddings* dan *Inverse Book Frequency* (IBF) untuk pencarian dokumen relevan. *Term-term* yang didapat dari hasil *preprocessing* dokumen tidak hanya memperhatikan frekuensi persebaran *term* dalam dokumen, namun juga memperhatikan persebaran *term* dalam dokumen yang telah dikelompokkan ke dalam berbagai kategori secara manual.

Word embeddings digunakan untuk mendapatkan ekspansi *query* secara otomatis. Penelitian ini menggunakan beberapa model *word embeddings* dan menyimpan semua hasil ekspansi *query* menjadi satu array. *Term-term* hasil ekspansi *query* akan dihitung pembobotan *term*-nya dengan TF-IDF-IBF.

Proses selanjutnya adalah pemilihan ekspansi *query* dengan menggunakan pembobotan baru. Pembobotan baru digunakan untuk mendapatkan *term* hasil ekspansi *query* yang memiliki korelasi tinggi dengan *query* asli sekaligus merupakan *term* yang representatif dalam dokumen dan digunakan untuk proses pencarian dokumen. *Query* awal dan ekspansinya akan dihitung kemiripannya terhadap dokumen yang ada untuk mendapatkan pencarian dokumen paling relevan.

Data yang digunakan adalah dokumen yang berupa kumpulan artikel *online* Bahasa Indonesia yang telah dikelompokkan sesuai dengan topiknya. Satu dokumen diasumsikan merupakan satu artikel. Metode baru yang diusulkan diharapkan dapat menghasilkan nilai *precision*, *recall*, dan *F-Score* yang lebih tinggi dalam menampilkan dokumen relevan dari *query* yang diberikan dengan menambahkan pemilihan ekspansi *query* terlebih dahulu sebelum proses pencarian dokumen.

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian ini adalah:

1. Bagaimana cara *preprocessing* terhadap dokumen yang digunakan?

2. Bagaimana perhitungan pembobotan kemunculan *term* pada berbagai dokumen dan memilki kategori beragam yang telah diklasifikasi secara manual (IBF)?
3. Bagaimana cara memilih *term* hasil ekspansi *query* dengan *word* embeddings dan IBF?
4. Bagaimana perhitungan pembobotan IBF berdasarkan *term* hasil ekspansi *query* dari *word embeddings*?
5. Bagaimana cara mengukur kemiripan dokumen dengan *query*?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah mengusulkan metode baru pembobotan *term* pada hasil ekspansi *query* berdasarkan metode *word embeddings* dan *Inverse Book Frequency* (IBF) untuk pemilihan *term* hasil ekspansi *query* yang memiliki tingkat korelasi *term* terhadap *query* dan frekuensi *term* yang tinggi.

1.4 Manfaat Penelitian

Penelitian ini diharapkan mampu menampilkan hasil pencarian dokumen relevan dengan memilih *term* hasil ekspansi *query* yang memiliki tingkat korelasi *term* terhadap *query* dan frekuensi *term* yang tinggi dengan menggunakan metode baru pembobotan *term* pada hasil ekspansi *query* berdasarkan *word embeddings* dan *Inverse Book Frequency* (IBF).

1.5 Kontribusi Penelitian

Kontribusi dari penelitian ini adalah mengusulkan metode baru pembobotan *term* pada hasil ekspansi *query* berdasarkan tingkat korelasi *term* terhadap *query* dan frekuensi *term* menggunakan metode *word embeddings* dan *Inverse Book Frequency* (IBF) untuk pencarian dokumen.

1.6 Batasan Masalah

Penelitian ini memiliki beberapa batasan antara lain:

1. Data yang digunakan merupakan kumpulan artikel *online* Bahasa Indonesia.
2. Dokumen yang digunakan telah dikelompokkan sesuai dengan topik/kategori yang dimiliki terlebih dahulu secara manual. Kategori yang digunakan adalah Ekonomi, Olahraga, dan Politik.

3. Proses pembentukan model *word embeddings* menggunakan *corpus* Wikipedia Bahasa Indonesia.
4. Model *word embbeddings* yang digunakan adalah model *Word2vec* dan *GloVe* dengan hasil *pretrained* yang sudah ada dan dibuat matriksnya menggunakan library *Gensim* yang ada di *Python*.

[*Halaman ini sengaja dikosongkan*]

BAB 2
KAJIAN PUSTAKA

2.1 *Information Retrieval*

Information Retrieval adalah proses pencarian dokumen yang dilakukan untuk menemukan suatu atau kumpulan dokumen yang relevan dengan suatu *query*. Proses pencarian dokumen biasanya dilakukan dengan mencari kemiripan dari beberapa dokumen. Mooers menciptakan istilah *Information Retrieval* pada tahun 1951. Hersh (Hersh, 2008) menyatakan *Information Retrieval* merupakan bidang persimpangan antara ilmu informasi dan ilmu komputer. *Information Retrieval* (IR) didefinisikan sebagai tindakan, metode, dan prosedur yang bertujuan untuk menemukan informasi kembali yang tersimpan, kemudian menyediakan informasi yang dibutuhkan, ISO 2382/1.

Sistem IR bertujuan untuk memenuhi kebutuhan informasi pengguna dengan me-*retrieve* semua dokumen yang mungkin relevan, pada waktu yang sama me-*retrieve* sedikit mungkin dokumen yang tidak relevan. Sistem IR yang baik adalah memungkinkan penguna menentukan secara cepat dan akurat apakah isi dari dokumen yang diterima sesuai dengan yang dibutuhkan. Kumpulan dokumen dengan topik atau isi yang mirip dikelompokkan agar dapat merepresentasikan dokumen lebih jauh.

Ada dua pekerjaan secara umum yang ditangani oleh sistem IR yaitu melakukan *preprocessing* terhadap data dalam bentuk teks dan kemudian diterapkan metode tertentu untuk menghitung kedekatan antara dokumen di dalam data yang telah di *preprocess* sebelumnya dengan *query* pengguna (Lukmana et al., 2014). *Tag* tertentu pada *term-term* dari dokumen biasanya diberikan oleh sistem yang berurusan dengan dokumen semi-*structured* pada tahapan dasar *preprocessing*. Pada dokumen yang tidak terstruktur, proses ini dilewati dan membiarkan *term* tanpa imbuhan *tag* (Lukmana, 2014).

Term-term penting akan diekstrak dengan mengkonversi *query* yang dimasukkan oleh pengguna sesuai aturan tertentu (Fauzi et al., 2014). Kemudian akan dilakukan penghitungan relevansi antara *query* dan dokumen berdasarkan

pada *term-term* yang sebelumnya telah diekstrak dari dokumen. Sebagai hasilnya, sistem akan mengembalikan suatu daftar dokumen terurut *descending* (*ranking*) sesuai nilai kemiripannya dengan *query* pengguna (Fauzi et al., 2014).

2.2 *Preprocessing* Dokumen

Preprocessing dokumen merupakan salah satu proses dalam melakukan *information retrieval*. Dokumen-dokumen yang akan digunakan akan dilakukan *preprocessing* terlebih dahulu hingga menghasilkan *term-term* yang siap untuk diproses ke tahap selanjutnya (Yunianto dan Arifin, 2017). *Preprocessing* terbagi menjadi beberapa tahapan sebagai berikut (Manning et al., 2009):

1. *Tokenization*

   Tokenization merupakan proses pemecahan terhadap isi dokumen berdasarkan *delimiter* antara lain spasi, digit, angka, tanda hubung dan tanda baca sehingga menjadi *term-term* tersendiri. Tahap ini memisahkan deretan kata yang ada di dalam kalimat, paragraf, maupun halaman menjadi *token* atau potongan kata tunggal.

2. *Case folding* dan *filtration*

   Tahap ini akan ditentukan *term* mana yang akan digunakan untuk mempresentasikan dokumen sehingga dapat mendeskripsikan isi dokumen dan membedakan dokumen tersebut dari dokumen lain di dalam koleksi. *Casefolding* mereupakan proses perubahan *term* yang pada awalnya terdapat huruf kapital menjadi huruf kecil semua. Sedangkan *filtration* merupakan proses menghilangkan simbol-simbol atau karakter-karakter yang tidak penting.

3. *Stopword Removal*

   Pada tahap ini *term-term* yang sering muncul di banyak dokumen atau dianggap tidak memiliki nilai informasi akan dihilangkan. Kamus yang berisi daftar *term-term* yang dianggap tidak memiliki nilai keinformatifan dapat digunakan untuk menghilangkan *term-term* tersebut.

4. *Stemming*

   Stemming merupakan proses konversi *term* ke bentuk dasarnya. Pada tahap ini *term-term* yang ada akan dirubah dengan menghilangkan imbuhan awalan serta akhiran.

2.3 Pembobotan *Term*

Vector space Model (VSM) direpresentasi dari proses pencarian dokumen dari kumpulan dataset. Dokumen dalam VSM direpresentasikan dalam bentuk matriks yang berisi bobot kata pada dokumen. Bobot tersebut menyatakan kepentingan kata terhadap suatu dokumen dan kumpulan dokumen. Frekuensi kemunculan suatu kata dalam dokumen dapat diartikan sebagai kepentingan kata tersebut terhadap dokumen yang digunakan. Terdapat beberapa metode pembobotan, yaitu:

a. *Term Frequency* (TF)

   Metode yang paling sederhana dalam membobotkan kata adalah TF. Setiap kata diasumsikan memiliki kepentingan yang proporsional terhadap jumlah kemunculan kata pada dokumen. Bobot dari kata *t* pada dokumen *d* dapat dihitung menggunakan Persamaan 2.1, seperti berikut:

   $$TF(d,t) = f(d,t), \tag{2.1}$$

   dimana *f(d,t)* merupakan frekuensi kemunculan *term t* pada dokumen *d*.

b. *Inverse Document Frequency* (IDF)

   Inverse document frequency berbeda dengan *term frequency*. TF memperhatikan kemunculan *term* di dalam dokumen, tetapi IDF memperhatikan kemunculan *term* pada kumpulan dokumen. Pada proses pembobotan ini, *term* yang dianggap sangat bernilai adalah *term* yang jarang muncul pada kumpulan dokumen. Persamaan 2.2 merupakan perhitungan faktor IDF dari *term t*, yaitu:

   $$IDF(t) = 1 + \log\left(N_d/df(t)\right), \tag{2.2}$$

   dimana N_d merupakan jumlah seluruh dokumen dan *df(t)* jumlah dokumen yang mengandung *term t*.

c. *Inverse Book Frequency* (IBF)

   Berbeda dengan IDF yang memperhatikan kemunculan *term* pada kumpulan dokumen, IBF memperhatikan kemunculan *term* pada kumpulan dokumen yang memiliki beberapa topik. *Term* yang bernilai untuk klasifikasi adalah *term* yang jarang muncul pada banyak dokumen dengan

beberapa topik. IBF dihitung dengan turunan langsung dari persamaan IDF, seperti pada Persamaan 2.3.

$$IBF(t) = 1 + \log(N_b/bf(t)), \quad (2.3)$$

dimana N_b adalah jumlah seluruh topik *book* dan *bf(t)* jumlah topik yang mengandung *term t*.

Beberapa perhitungan bobot tersebut dapat dikombinasikan untuk mendapatkan *term* yang tidak hanya dalam dokumen saja, namun juga kemunculan *term* tersebut pada kumpulan dokumen dengan banyak topik. Kombinasi bobot *term t* pada dokumen *d* tersebut disebut TF-IDF-IBF, seperti pada Persamaan 2.4 berikut:

$$TF - IDF - IBF(d,t) = TF(d,t) \times IDF(t) \times IBF(t). \quad (2.4)$$

Setelah didapatkan nilai kombinasi bobot *term* per dokumen pada sebaran topik *book*, dilakukan metode *mean* TF-IDF-IBF. Nilai *mean* TF-IDF-IBF *term* pada tiap dokumen dapat dihitung dengan Persamaan 2.5.

$$TF - IDF - IBF(t) = \frac{\sum_{i=1}^{n} TF-IDF-IBF(d_i,t)}{n}, \quad (2.5)$$

dimana d_i adalah dokumen ke-*i* dan *n* adalah jumlah keseluruhan dokumen. Dengan perumusan tersebut maka bobot akan semakin tinggi saat lebih banyak ditemukan di dalam satu dokumen (indikasi frekuensi *term*). Salton (Salton, 1989) menyatakan *term* yang sering muncul pada satu dokumen, tapi jarang muncul pada seluruh *dataset* akan diberikan nilai bobot yang lebih tinggi (indikasi IDF).

2.4 *Query Expansion* dengan *Word Embeddings*

Query Expansion merupakan metode yang efektif untuk menangani masalah ketidak-cocokan kata dalam proses pencarian informasi (Zhang et al., 2016). Metode *query expansion* yang diusulkan oleh Cui (Cui et al., 2003) berbasis korelasi untuk mengekstraksi istilah ekspansi dari data log pencarian. Istilah yang telah diektraksi kemudian diintegrasikan ke dalam *query* asli dalam model peringkat *uni-fied* untuk meningkatkan kinerja dari pencarian pada web. Xu dan Croft (Xu dan Croft, 1996) menganalisis dokumen yang diambil oleh *query* asli sebagai informasi lokal yang kemudian diekprolasi dengan hubungan kata di

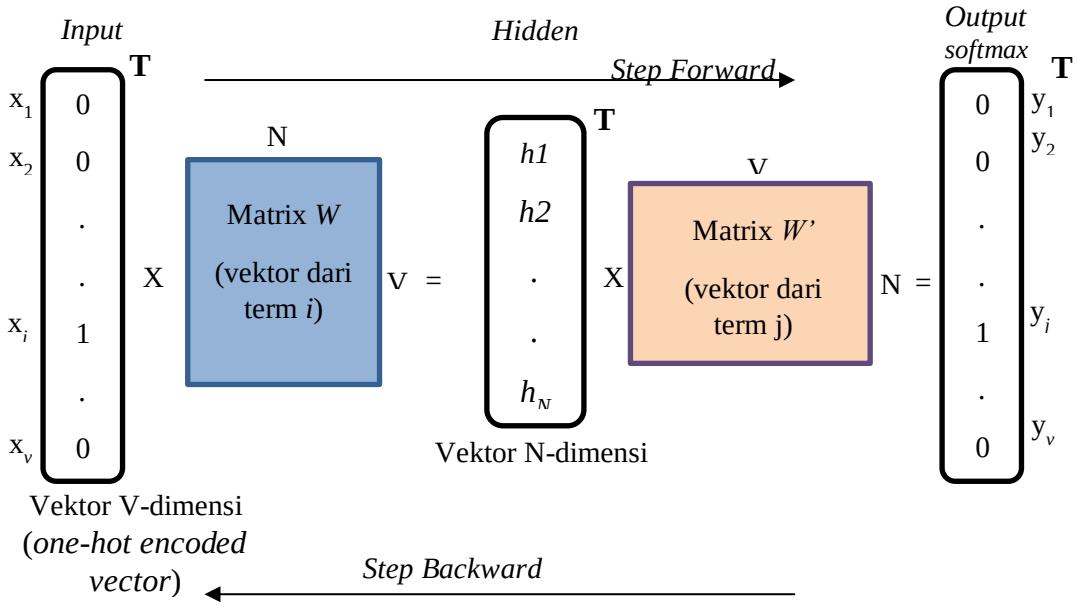
seluruh korpus sebagai informasi global. Informasi lokal dan global digabungkan sebagai perluasan *query* pada tugas pencarian informasi. Namun, pendekatan-pendekatan *query expansion* tersebut sepenuhnya didasarkan pada informasi statistik dan informasi istilah semantik diabaikan (Zhang et al., 2016). Salah satu metode yang dapat digunakan untuk *query expansion* yaitu dengan menggunakan metode *word embeddings*.

Word embeddings merupakan jenis representasi kata yang memungkinkan kata-kata tersebut bermakna yang sama dan memiliki representasi yang serupa. Model *word embeddings* didesain berdasarkan kata dengan arti yang mirip cenderung memiliki *word embeddings* yang sama (Suleiman dan Awajan, 2018). Kata yang memiliki semantik dan sintaksis kata dari korpus besar yang tidak berlabel dapat dikenali dengan menggunakan *word embeddings* (Mikolov et al., 2013). Kemiripan kata dengan lainnya dapat dihitung dengan persamaan *similarity* dengan nilai hasil berkisar antara -1 sampai 1, dimana 1 merupakan nilai *similarity* tertinggi (Elekes et al., 2017). *Word embeddings* dapat dihasilkan dengan beberapa teknik yang dapat digunakan, antara lain *word2vec*, GloVe, dan *fasttext*.

2.4.1 *Word2Vec*

Word2vec merupakan representasi vektor kata yang dibangun oleh Mikolov (Mikolov et al., 2013). Word2vec memiliki 2 model yaitu *Skip-Gram* dan *Continous Bag-of-Words* (CBOW) (Mikolov et al., 2013). Model *Skip-Gram* menggunakan proyeksi vektor kata-kata konteks untuk memprediksi vektor kata target, sedangkan CBOW memprediksi vektor kata-kata yang ada dikonteks dengan diberikan vektor kata tertentu (Mikolov et al., 2013).

Model word2vec yang digunakan adalah *Skip-Gram*. Model ini akan disesuaikan dengan metode yang diusulkan pada penelitian ini. Gambar 2.1 merupakan ilustrasi model *Neural Network* untuk word2vec *Skip-Gram*. *Word embeddings* dilakukan 2 tahap, yaitu tahap *forward* dan *backward*. Tahap *forward* digunakan untuk menghitung similaritas dari kata konteks dengan diberikan kata target sebagai inputan. Sedangkan tahap *backward* adalah memperbarui matrik *W* dan *W'* berdasarkan fungsi objektif.



Gambar 2.1 Ilustrasi Model *Neural Network* untuk word2vec *Skip-Gram* (Sumber: https://lilianweng.github.io/lil-log/2017/10/15/learning-word-embedding.html)

Berikut merupakan algoritma dari *Word2Vec* model *Skip-Gram* (Niasita et al., 2019):

1. Setiap kata dalam kalimat, ditentukan bobot target dari kata target dan kata konteks. Kata yang menjadi kata target dan kata konteks bernilai 1 sedangkan kata lainnnya bernilai 0.
2. Setiap kata yang menjadi kata target, dilakukan proses 3 hingga 7.
3. *Forward Pass*
   a. Perhitungan bobot dari *input layer* ke *hidden layer* dengan menggunakan Persamaan 2.6.

$$h = W^T \times x, \tag{2.6}$$

   dimana *h* adalah *hidden neuron*, *x* adalah *input vector*, dan W^T adalah nilai *transpose* pada bobot dari *input layer* ke *hidden layer*.

   b. Perhitungan bobor dari *hidden layer* ke *output layer* dengan menggunakan Persamaan 2.7.

$$o_j = W'^T \times h, \tag{2.7}$$

dimana o_j adalah *output* pada baris ke – j dan W'^T adalah nilai *transpose* bobot dari *hidden layer* ke *output layer*.

4. *Softmax Output*

   Perhitungan *softmax output* dengan menggunakan Persamaan 2.8.

$$y_j = S_{Q_i} = \frac{\exp\,(o_j)}{\sum_{j'=1}^{V} \exp\,(o_{j'})}, \tag{2.8}$$

   dimana y_j adalah *softmax output* pada baris ke – j, o_j' adalah *output* dari seluruh baris, dan V adalah jumlah kata unik (konteks). Nilai *output* ini merupakan tingkat kedekatan similaritas konteks dengan *query*.

5. Nilai *error*

   Perhitungan nilai *error* dengan menggunakan Persamaan 2.9.

$$e = \sum_1^c (y_j - x_c), \tag{2.9}$$

   dimana *e* adalah nilai *error* dan x_c adalah nilai vektor dari kata konteks.

6. *Backpropagation*

   a. Perhitungan bobot dari *output layer* ke *hidden layer* dengan menggunakan Persamaan 2.10.

$$dl_{dw'} = h \times e^T, \tag{2.10}$$

      dimana $dl_{dw'}$ adalah bobot dari *output layer* ke *hidden layer*.

   b. Perhitungan bobot dari *hidden layer* ke *input layer* dengan menggunakan Persamaan 2.11.

$$dl_{dw} = x \times (W' \times e)^T, \tag{2.11}$$

      dimana dl_{dw} adalah bobot dari *input layer* ke *hidden layer* dan W' adalah bobot dari *hidden layer* ke *output layer*.

7. *Update* bobot

   a. Perhitungan bobot baru dari *input layer* ke *hidden layer* dengan menggunakan Persamaan 2.12.

$$W = W(lama) - (learning\ rate \times dl_{dw}), \tag{2.12}$$

dimana W adalah bobot baru daru *input layer* ke *hidden layer*, $W(lama)$ adalah bobot lama dari *input layer* ke *hidden layer*, dan *learning rate* adalah parameter fungsi turunan dari waktu.

b. Perhitungan bobot baru dari *hidden layer* ke *output layer* dengan menggunakan Persamaan 2.13.

$$W' = W'(lama) - (learning\ rate \times dl_{dw'}), \tag{2.13}$$

dimana W' adalah bobot baru daru *hidden layer* ke *output layer* dan $W'(lama)$ adalah bobot lama dari *hidden layer* ke *output layer*.

8. Setelah bobot baru dari kata terakhir dalam kalimat diperoleh, selanjutnya dilakukan perhitungan *query* dengan masing-masing kata konteks untuk mendapatkan vektor kata dengan Persamaan 2.14.

$$\theta = \frac{V_{query} \times (V_{konteks})^T}{\sqrt{\sum {V_{query}}^2} \times \sqrt{\sum {V_{konteks}}^2}}, \tag{2.14}$$

dimana θ adalah vektor dari kata, V_{query} adalah bobot dari kata *query*, dan $V_{konteks}$ adalah bobot dari kata konteks.

9. Pengurutan nilai vektor pada tiap kata dari yang besar hingga kecil, lalu diambil *n* kata teratas untuk menentukan kata yang memiliki kedekatan similaritas dengan *query*.

Word embeddings akan menghasilkan *output term-term* yang memiliki kedekatan similaritas tinggi dengan *query* yang diberikan. *Term* tersebut tidak memperhatikan kemunculannya dalam berbagai dokumen dengan kategori yang beragam. Sehingga, peneliti mengusulkan hasil ekspansi *query* dari *word embeddings* dihitung pembobotan *term*-nya dengan IBF dan kemudian dilakukan pencarian dokumen untuk mendapatkan dokumen yang relevan. Pada penelitian ini, hasil ekspansi *query* dari *word embeddings* tidak dipilih semua sebagai *query expansion* akhir untuk proses pencarian dokumen. Pemilihan *term* hasil ekspansi *query* dilakukan dengan menghitung pembobotan *term* hasil ekspansi *query* (W_{Q_i}) menggunakan Persamaan 2.15.

$$W_{Q_i} = S_{Q_i} \times W_{Q_i \bullet TF-IDF-IBF}. \tag{2.15}$$

Term hasil ekspansi *query* yang dipilih adalah *term-term* yang memiliki similaritasa *query* (S_{Q_i}) tinggi dan nilai bobot TF-IDF-IBF ($W_{Q_i \bullet TF-IDF-IBF}$) juga tinggi. Sehingga hanya *term* dari hasil *query expansion* yang memiliki korelasi tinggi dengan *query* asli sekaligus merupakan *term* yang representatif dalam dokumen yang akan dipilih dan digunakan untuk proses pencarian dokumen.

2.4.2 *GloVe*

GloVe (*Global Vector*) merupakan representasi kata untuk menghasilkan *word embeddings*. *GloVe* merupakan metode *unsupervised learning* pada representasi kata. *GloVe* menggunakan kumpulan teks dari *corpus* yang akan dibangun *vocabulary* dan setiap kata pada *vocabulary* menghasilkan vektor yang berjumlah ratusan dimensi. Penelitian ini menggunakan *corpus* Wikipedia Bahasa Indonesia untuk dibangun vektor kata-kata yang ada. *GloVe* merupakan proses pembentukan *word co-occurence matrix* dari suatu kata. Penelitian yang dilakukan oleh Pennington (Pennington et al., 2014) membuktikan bahwa metode ini merupakan metode yang bisa menghasilkan *word embeddings* terbaik untuk tes analogi kata.

Teknik mendapatkan *word embeddings* dibagi dua metode. Metode pertama yaitu faktorisasi matrik, dengan *word embeddings* matrik dibuat berdasarkan jumlah kemunculan kata, kemudian dikonversikan ke dalam vektor berdimensi tertentu. Metode kedua yaitu *context window*. Metode ini merupakan proses untuk membandingkan antar kata yang sering muncul pada setiap kata dalam *corpus* yang akan dibandingkan.

2.5 Pembobotan *Term* Hasil Ekspansi *Query*

Ekspansi *query* dari hasil *word embeddings* dihitung pembobotannya menggunakan TF-IDF-IBF. Pembobotan IBF digunakan untuk melihat frekuensi kemunculan *term-term* hasil ekspansi *query* pada beberapa dokumen yang memiliki topik/kategori beragam. Penelitian ini sudah dilakukan pengelompokkan kategori secara manual, sehingga proses hanya perlu melihat *term* berada pada dokumen dan kategori mana saja, kemudian dihitung frekuensi kemunculannya. Pada penelitian ini, *term* dari *query* asli akan diberikan nilai faktor berbeda dari *term* hasil ekspansi *query*. Pemberian nilai faktor ini bertujuan agar *term* dari *query* asli tetap memiliki

bobot yang lebih tinggi dibandingkan dengan *term* hasil ekspansi *query*. Bobot akhir *term* dari *query* didapatkan dengan mengalikan bobot TF-IDF-IBF dengan nilai faktor yang dimiliki, seperti pada Persamaan 2.16.

$$W_{QA_i} = W_{Q_i \bullet TF-IDF-IBF} \times \alpha, \tag{2.16}$$

dimana W_{QA_i} adalah nilai akhir bobot TF-IDF-IBF dan α adalah nilai faktor pada *term* dari *query* dan ekspansi *query*. α bernilai 2 jika *term* berasal dari *query* asli dan bernilai 0,5 jika *term* berasal dari hasil ekspansi *query*.

2.6 Pengukuran Kemiripan *Query* dengan Dokumen

Representasi bobot yang telah dihitung sebelumnya menggunakan TF-IDF-IBF, dapat dihitung nilai kemiripan antara suatu dokumen dengan *query*. *Cosine similarity* biasa digunakan untuk tingkat kemiripan dengan berdasar pada besar sudut kosinus antara dua vektor (vektor dokumen). *Cosine similarity* menghitung nilai kosinus θ dari *query* dan dokumen-dokumen lain. Nilai *cosine similarity* menunjukkan derajat kemiripan dokumen dengan *query*. Persamaan 2.17 merupakan perhitungan derajat kemiripan antara *query* dan dokumen sebagai berikut (Fauzi et al., 2014):

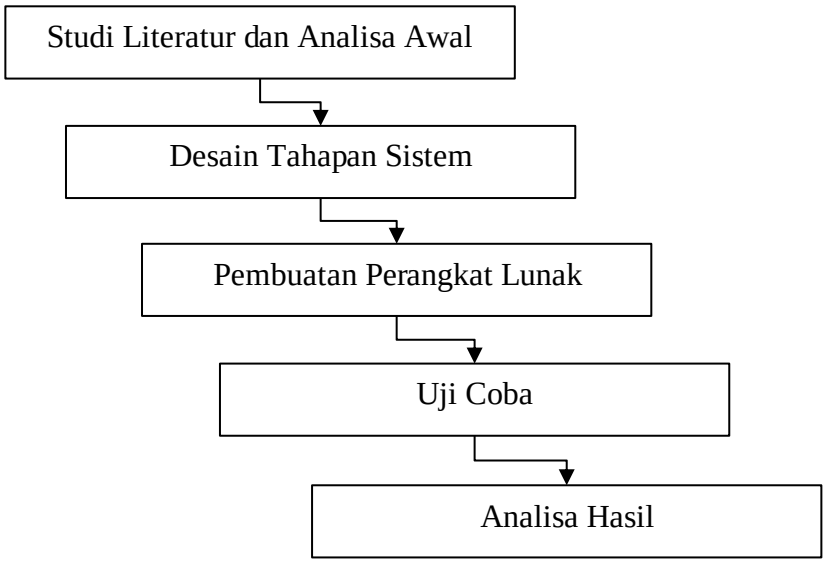
$$\cos(q, d_j) = \frac{\sum_{t_k}[TF-IDF-IBF(t_k,q)] \bullet [TF-IDF-IBF(t_k,d_j)]}{\sqrt{\sum |TF-IDF-IBF_q|^2} \bullet \sqrt{\sum |TF-IDF-IBF_{d_j}|^2}}, \tag{2.17}$$

dimana *TF-IDF-IBF*(t_k,d_j) dan *TF-IDF-IBF*(t_k,d_j) adalah pembobotan TF-IDF-IBF kata t_k pada *query* dan dokumen j. |*TF-IDF-IBF*$_q$| dan |*TF-IDF-IBF*$_{dj}$| adalah panjang dari vektor *query* q dan dokumen.

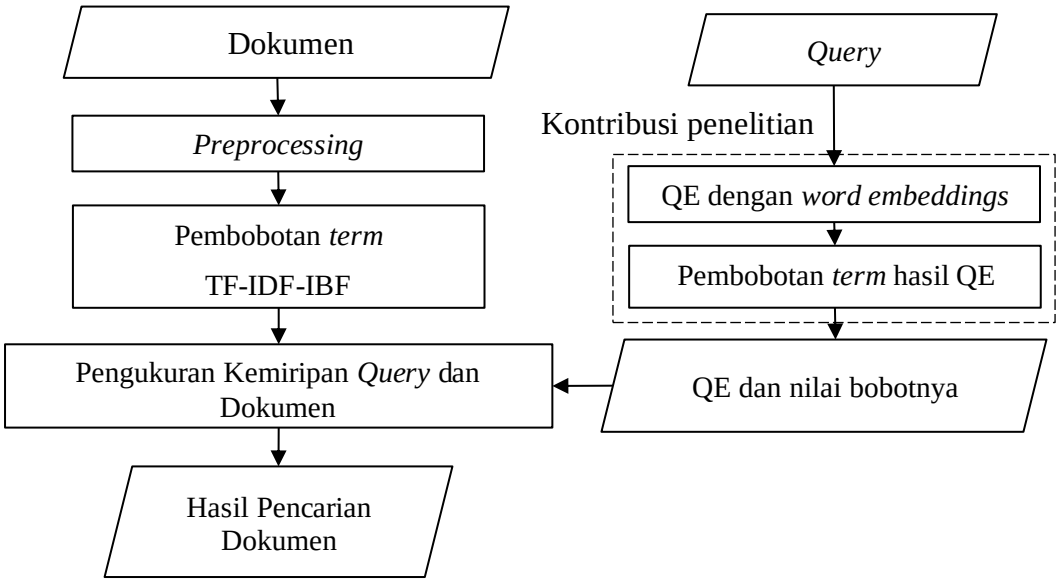
BAB 3
METODE PENELITIAN

Pada bab ini akan dijelaskan proses-proses atau tahapan yang dilalui untuk membentuk sistem yang sesuai dengan metode yang diusulkan. Gambar 3.1 merupakan gambaran dari tahapan penelitian yang dilakukan.

Tahapan penelitian yang dilakukan adalah mulai dari studi literatur dan analisa awal, desain tahapan sistem, pembuatan perangkat lunak, uji coba dan analisa hasil. Gambar 3.2 merupakan gambaran tahapan-tahapan secara garis besar sistem yang dibangun.



Gambar 3.1 Tahapan Penelitian



Gambar 3.2 Tahapan Proses Sistem

Tahapan utama yang dilakukan, yaitu proses *preprocessing*, *query expansion* (QE) dengan beberapa model *word embedding*, pembobotan *term*, dan *similarity matching* antara *query* dan dokumen untuk mendapatkan dokumen yang paling relevan.

3.1 Desain Model Sistem

Pada bagian ini akan dipaparkan mengenai format data input/output dan penggambaran alur proses yang terjadi dalam metode/sistem untuk menghasilkan output.

3.1.1 Data

Data yang digunakan dalam uji coba ini merupakan *corpus* atau kumpulan dokumen teks Bahasa Indonesia. Dokumen-dokumen tersebut merupakan kumpulan artikel berita *online* Bahasa Indonesia, antara lain dari website *www.kompas.com*, *www.tempo.co*, *www.liputan6.com*, *www.cnnindonesia.com*, dll. Dokumen tersebut berupa beberapa dokumen yang memiliki beragam topik atau kategori, antara lain Olahraga, Ekonomi, dan Politik. Dokumen yang digunakan dikelompokkan sesuai topik yang berbeda dan telah dilakukan secara manual terlebih dahulu. Jumlah keseluruhan artikel berita yang diambil sebanyak 11.245

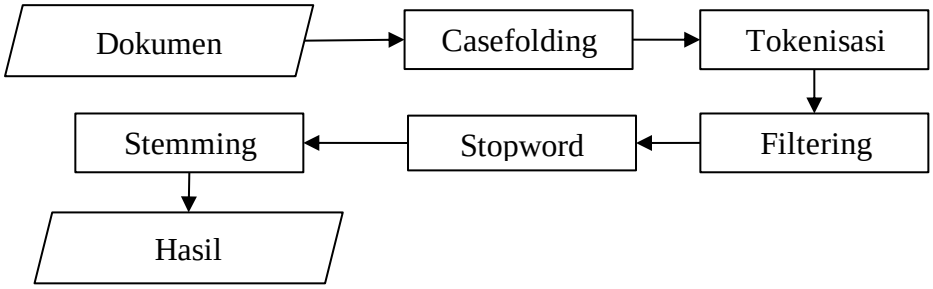
```
<artikel>
    <id>OLA_GP_01_003</id>
    <judul>Rossi: Saya Ogah Finis di Posisi Dua</judul>
    <tanggal>17/10/16</tanggal>
    <kata_kunci>terjatuh, debu-debu jalanan, MotoGP, GP Jepang, tergelincir</kata_kunci>
    <isi>Pebalap Movistar Yamaha, Valentino Rossi, mengaku telah mengeluarkan seluruh
        kekuatan terbaiknya di MotoGP Jepang dan berusaha terus bertarung dalam perebutan
        gelar juara dunia MotoGP. Pada akhirnya, upayanya itu justru membuat Rossi
        tergelincir dan terjerembab gagal menyelesaikan balapan. Marquez yang memenangi
        GP Jepang pun kemudian dinobatkan sebagai juara dunia 2016. Pada balapan yang
        digelar di Sirkuit Motegi, Minggu (16/10), Rossi mulai dari posisi pole tapi kemudian
        tercecer ke posisi tiga setelah disalip Jorge Lorenzo dan Marc Marquez. Sejak putaran
        keempat, Marquez memimpin balapan dan kemudian Rossi sukses menyalip Lorenzo
        untuk menduduki tempat kedua. Upaya Rossi berakhir sia-sia setelah ia tergelincir di
        tikungan ke-24 putaran ketujuh karena kehilangan kendali ban depannya. "Saya
        memberikan 100 persen upaya saya hari ini, karena saya tak tertarik finis di tempat
        kedua (di klasemen akhir)," kata pebalap yang dijuluki The Doctor itu, seperti dikutip
        dari GP One. "Saya mendorong keras. Saya tidak ingin Marquez kabur. Kecepatan saya
        tak terlalu berbeda dengannya, dan meski hal itu tak mudah dilakukan, saya bisa
        mencoba menyalipnya di akhir-akhir." Sebelum niat itu terlaksana, motor Rossi telah
        lebih dahulu mencium debu-debu jalanan. Rossi mengatakan dirinya tak merasakan
        adanya tanda-tanda kerusakan motor atau ban yang menyebabkan insiden tersebut.
        "Saya sadar terjatuh ketika saya telah berada di tanah. Pada akhirnya, memang seperti
        itu biasanya. Jika tidak, maka kami semua akan tetap bisa melaju di atas motor.". Kaki
        kanan Rossi sempat tertimpa motor ketika insiden tersebut. Ia tak mendapatkan
        cedera, tapi kegagalannya menghambat laju Marquez menjadi juara dunia menohok
        mentalnya. Rossi masih ingin bekerja sekeras mungkin hingga akhir balapan dan
        setidaknya mengamankan posisi kedua di klasemen akhir, meski tidak dengan
        antusiasme yang sama seperti sebelumnya. "Posisi runner-up masih ada untuk
        diperebutkan dengan Lorenzo. Ini memang tak sepenting posisi pertama, tapi
        setidaknya memberikan saya beberapa motivasi."</isi>
    <link>http://www.cnnindonesia.com/olahraga/20161017084254-156-165943/rossi-saya-
        ogah-finis-di-posisi-dua/</link>
</artikel>
```

Gambar 3.3 Contoh Dokumen Artikel Berita

dengan rincian per-kategori yaitu: kategori Ekonomi sebanyak 3.629 dokumen, Olahraga sebanyak 4.180 dokumen, dan Politik sebanyak 3.436 dokumen. Kumpulan artikel berita *online* tersebut dikumpulkan, kemudian diolah menjadi file berekstensi xml untuk kemudian dimasukkan ke dalam sistem yang dibentuk dalam penelitian ini. Setiap artikel berita yang diambil memiliki spesifikasi tag seperti contoh pada Gambar 3.3, antara lain: id artikel berita <id>, url link artikel berita <url>, kategori artikel berita <kategori>, judul artikel berita <judul>, isi artikel berita <isi>, dan kata kunci artikel berita <kata_kunci>.

Dokumen-dokumen tersebut akan diproses dari tahap *query expansion*, *preprocessing*, perhitungan pembobotan *term* hingga proses perhitungan *cosine similarity* untuk mendapatkan dokumen yang relevan. Selain dokumen yang berisi artikel *online*, sistem juga memproses *query* masukan oleh pengguna yang dilakukan ekspansi *query* untuk kemudian digunakan dalam mencari dokumen yang relevan.

3.1.2 Tahapan *Preprocessing Dokumen*

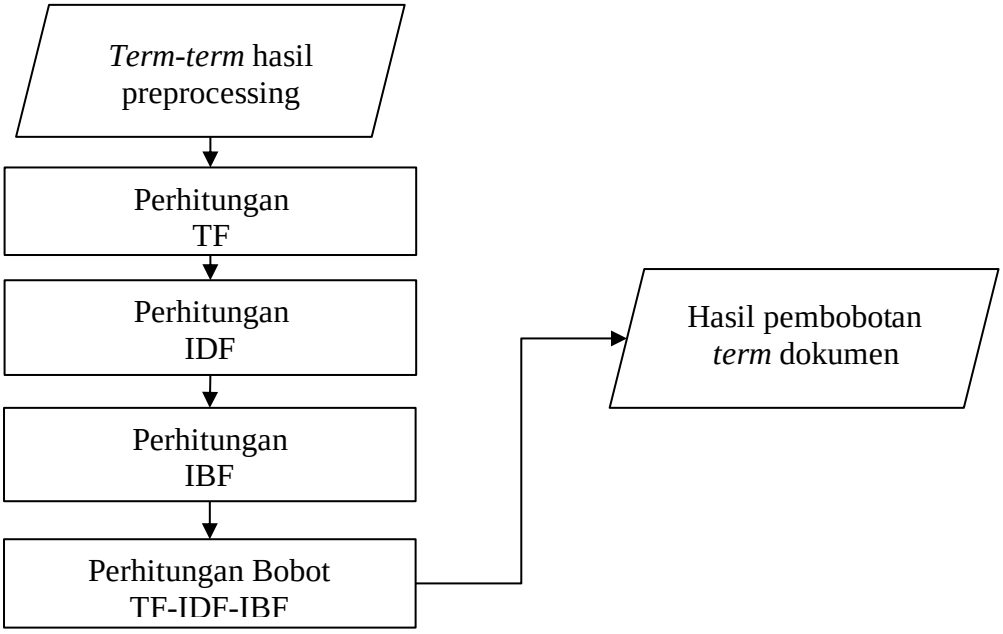


Gambar 3.4 Tahapan *Preprocessing* Dokumen

Dokumen yang digunakan akan dilakukan *preprocessing* terlebih dahulu untuk mendapatkan *term-term* yang digunakan untuk pembobotan. Tahapan *preprocessing* pada dokumen yang dilakukan dapat dilihat pada Gambar 3.4.

3.1.3 Tahapan Pembobotan *Term*

Tahapan pembobotan *term* TF-IDF-IBF dilakukan sebagai proses seleksi fitur. Bobot yang didapat menyatakan kepentingan *term* pada dokumen. Proses seleksi fitur untuk pemilihan subset fitur terbaik dilakukan dengan mengambil beberapa variasi pemilihan jumlah term. Proses perhitungan pembobotan *term* dapat dilihat pada Gambar 3.5.



Gambar 3.5 Tahapan Pembobotan *Term* dan Seleksi Fiturnya

Tabel 3.1 Contoh Representasi Dokumen

Topik	Dokumen	Isi Dokumen
To1	D1	Dolar naik harga naik penghasilan turun
	D2	Harga naik harusnya gaji juga naik
To2	D3	Premium tidak terpengaruh dolar
	D4	Harga laptop naik

Dokumen yang digunakan akan direpresentasikan sesuai dengan topik pertanyaan seperti pada Tabel 3.1. Contoh dari hasil perhitungan dokumen dapat dilihat pada Tabel 3.2 yang menampilkan hasil perhitungan TF pada dokumen dengan Persamaan 2.1. Contoh dari hasil perhitungan dokumen dapat dilihat pada Tabel 3.3 yang menampilkan hasil perhitungan IDF dan IBF pada dokumen dengan Persamaan 2.2 dan Persamaan 2.3.

Tabel 3.2 Contoh Hasil Perhitungan TF

Term-ke	*Term*	TF			
		D1	D2	D3	D4
T1	Dolar	1	0	1	0
T2	Naik	2	2	0	1
T3	Harga	1	1	0	1
T4	Hasil	0	0	0	0
T5	Turun	1	0	0	0
T6	Gaji	0	1	0	0
T7	Premium	0	0	1	0
T8	Pengaruh	0	0	1	0
T9	Laptop	0	0	0	1

Tabel 3.3 Contoh Hasil Perhitungan IDF dan IBF

Term	df(t)	bf(t)	IDF	IBF
T1	2	1	1	1
T2	3	2	0,41	0
T3	3	2	0,41	0
T4	1	1	2	1

Term	df(t)	bf(t)	IDF	IBF
T5	1	1	2	1
T6	1	1	2	1
T7	1	1	2	1
T8	1	1	2	1
T9	1	1	2	1

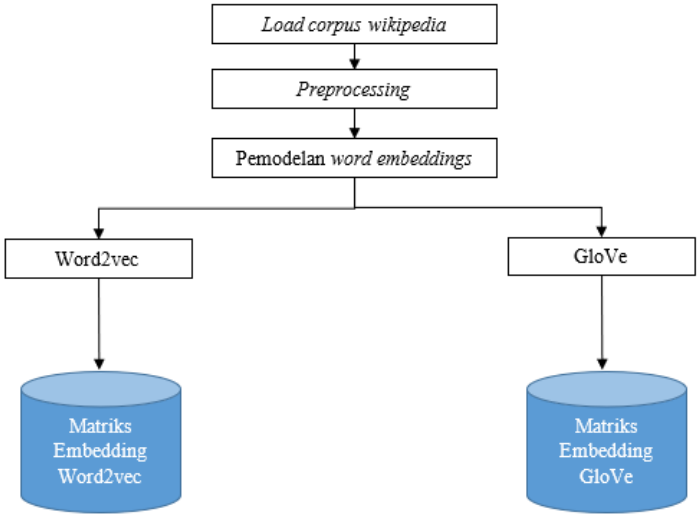
Contoh dari hasil perhitungan dokumen dapat dilihat pada Tabel 3.4 yang menampilkan hasil perhitungan kombinasi bobot TF-IDF-IBF pada dokumen dengan Persamaan 2.4.

Tabel 3.4 Contoh Hasil Perhitungan TF-IDF-IBF

	TF-IDF-IBF			
Term	D1	D2	D3	D4
Dolar	1	0	1	0
Naik	0	0	0	0
Harga	0	0	0	0
Hasil	0	0	0	0
Turun	2	0	0	0
Gaji	0	2	0	0
Premium	0	0	2	0
Pengaruh	0	0	2	0
Laptop	0	0	0	2

3.1.4 Tahapan *Query Expansion* dengan *Word Embeddings*

Query yang diinputkan akan diekspansi menggunakan beberapa pemodelan *word embeddings*. Sistem akan menyimpan urutan *query expansion* yang memiliki nilai similaritas terbesar sebanyak 5 dari masing-masing model. *Term-term* dari hasil ekspansi *query* akan dihitung pembobotan menggunakan TF-IDF-IBF untuk melihat frekuensi kemunculan *term* tersebut di dalam dokumen yang memiliki beragam topik/kategori. Tahap pemodelan *word embeddings* dengan beberapa model dapat dilihat pada Gambar 3.6.



Gambar 3.6 Tahapan Pemodelan *Word Embeddings*

Tahap pemilihan *query expansion* dengan *word embeddings* dapat dilihat pada Gambar 3.7. Pada tahap ini, hasil *query expansion* dari *word embeddings* tidak dipilih semua sebagai *query expansion* akhir untuk proses pencarian dokumen. Pemilihan *term* hasil ekspansi *query* dilakukan dengan menghitung pembobotan menggunakan Persamaan 2.15.

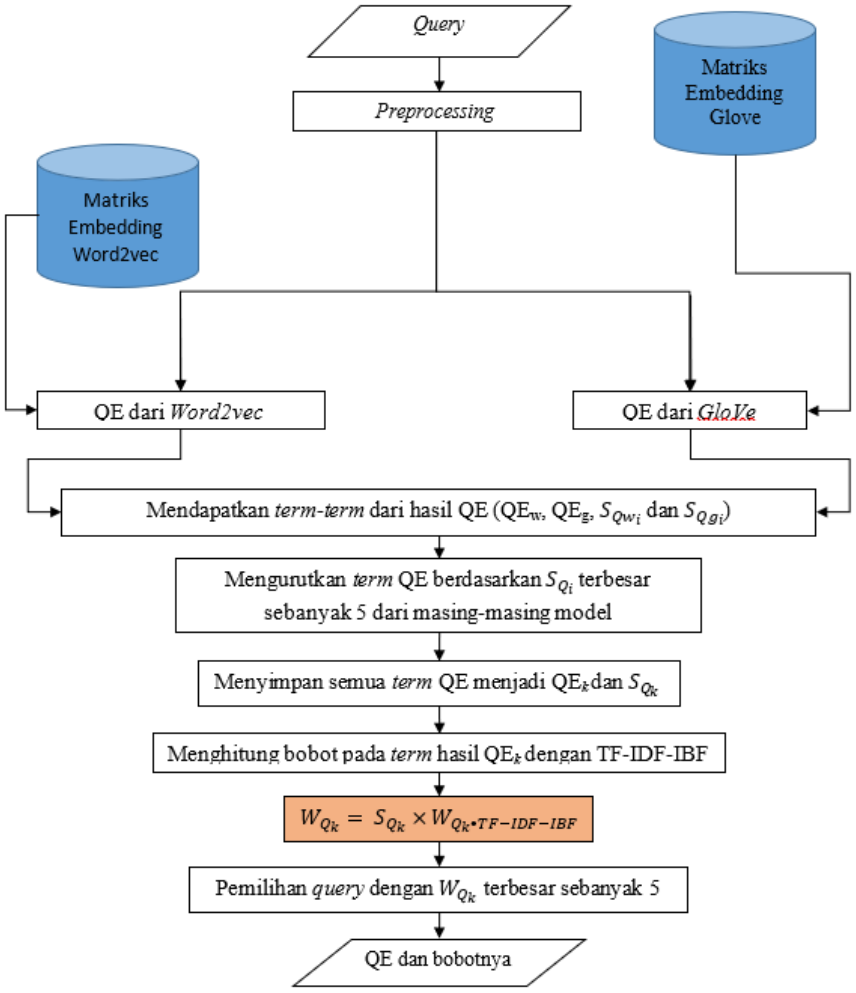
Term hasil ekspansi *query* yang dipilih adalah *term-term* yang memiliki similaritas *query* (S_{Q_k}) tinggi dan nilai bobot TF-IDF-IBF ($W_{Q_k \bullet TF-IDF-IBF}$) juga tinggi. Sehingga hanya *term* dari hasil *query expansion* yang memiliki korelasi tinggi dengan *query* asli sekaligus merupakan *term* yang representatif dalam dokumen yang akan dipilih dan digunakan untuk proses pencarian dokumen.

Query yang dimasukkan oleh pengguna akan dilakukan *preprocessing* untuk merubahnya menjadi *term-term* yang dapat diolah selanjutnya. *Term-term* tersebut akan diekspansi dengan menggunakan pemodelan dari beberapa model *word embeddings* yaitu Word2vec dan GloVe. *Term-term* tersebut akan dicocokkan dengan matriks embedding yang telah disimpan ke dalam *database*.

Hasil dari ekspansi *query* dengan beberapa model akan disebut sebagai QE_w (*query expansion* dari Word2vec) dan QE_g (*query expansion* dari GloVe). Sedangkan nilai similaritasnya disebut sebagai S_{Qw_i} (similaritas *query expansion* dari Word2vec) dan S_{Qg_i} (similaritas *query expansion* dari GloVe).

Term hasil ekspansi *query* akan diurutkan berdasarkan nilai similaritas *term- term* lain yang sering muncul (S_{Q_i}) terbesar sebanyak 5 *term* dari masing-

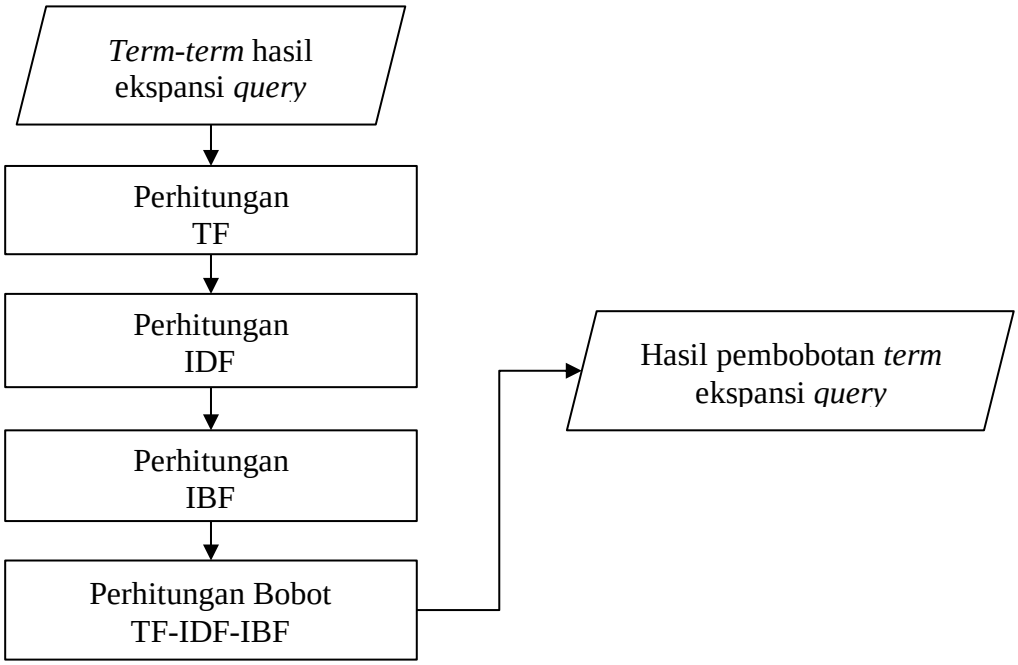
masing model. Semua *term* hasil ekspansi *query* disimpan menjadi satu beserta nilai similaritasnya (QE_k dan S_{Q_k}) yang selanjutnya akan dihitung bobotnya menggunakan TF-IDF-IBF. Kemudian, bobot baru akan dihitung dengan mengalikan nilai similaritas dengan bobot TF-IDF-IBF dari masing-masing *term* menggunakan Persamaan 2.15. Selanjutnya, *term* hasil ekspansi *query* akan dipilih sebanyak 5 *term* yang memiliki bobot terbesar. Perhitungan bobot ini dilakukan untuk mendapatkan *term* hasil ekspansi *query* yang memiliki tingkat korelasi yang tinggi dan merupakan *term* yang penting dalam dokumen.



Gambar 3.7 Tahapan Pemilihan *Query Expansion* dengan *Word Embeddings* dan IBF

3.1.5 Tahapan Pembobotan *Term* Hasil Ekspansi *Query*

Ekspansi *query* dari hasil *word embeddings* dihitung pembobotannya menggunakan TF-IDF-IBF. Pembobotan IBF digunakan untuk melihat frekuensi kemunculan *term-term* hasil ekspansi *query* pada beberapa dokumen yang memiliki



Gambar 3.8 Tahapan Pembobotan *Term* Hasil Ekspansi *Query*

topik/kategori beragam. Penelitian ini sudah dilakukan pengelompokkan kategori secara manual, sehingga proses hanya perlu melihat *term* berada pada dokumen dan kategori mana saja, kemudian dihitung frekuensi kemunculannya.

Pada penelitian ini, *term* dari *query* asli akan diberikan nilai faktor berbeda dari *term* hasil ekspansi *query*. Pemberian nilai faktor ini bertujuan agar *term* dari *query* asli tetap memiliki bobot yang lebih tinggi dibandingkan dengan *term* hasil ekspansi *query*. Bobot akhir *term* dari *query* didapatkan dengan mengalikan bobot TF-IDF-IBF dengan nilai faktor yang dimiliki, seperti pada Persamaan 2.16.

W_{QA_i} adalah nilai akhir bobot TF-IDF-IBF dan α adalah nilai faktor pada *term* dari *query* dan ekspansi *query*. α bernilai 2 jika *term* berasal dari *query* asli dan bernilai 0,5 jika *term* berasal dari hasil ekspansi *query*. Tahap perhitungan bobot terhadap ekspansi *query* dengan menggunakan TF-IDF-IBF dapat dilihat pada Gambar 3.8.

Tabel 3.5 Contoh Hasil Ekspansi *Query*

Query awal	Ekspansi *Query*		Semua *Query*	
Dolar	Q_1	Harga	Q_0	Dolar
	Q_2	Naik	Q_1	Harga

Query awal	Ekspansi *Query*		Semua *Query*	
	Q_3	Turun	Q_2	Naik
			Q_3	Turun

Tabel 3.6 Contoh Hasil Perhitungan TF pada *Term* Ekspansi *Query*

Term-ke	*Term*	TF			
		D1	D2	D3	D4
Q_0	Dolar	1	0	1	0
Q_1	Naik	1	1	0	1
Q_2	Harga	1	1	0	1
Q_3	Turun	1	0	0	0

Contoh dari hasil ekspansi *query* dari *word embeddings* dapat dilihat pada Tabel 3.5. Contoh dari hasil pembobotan ekspansi *query* dapat dilihat pada Tabel 3.6 yang menampilkan hasil perhitungan TF dengan Persamaan 2.1.

Contoh dari hasil pembobotan *term* ekspansi *query* dapat dilihat pada Tabel 3.7 yang menampilkan hasil perhitungan IDF dan IBF *term* ekspansi *query* pada dokumen yang memiliki beragam kategori dengan Persamaan 2.2 dan Persamaan 2.3.

Contoh dari hasil pembobotan *term* ekspansi *query* dapat dilihat pada Tabel 3.8 yang menampilkan hasil perhitungan kombinasi bobot TF-IDF-IBF *term* ekspansi *query* pada dokumen yang memiliki beragam kategori dengan Persamaan 2.4. Bobot tersebut akan dikalikan dengan nilai faktornya untuk mendapatkan bobot *term* ekspansi *query* terakhir dengan Persamaan 2.16 dan juga panjang vektor *query*.

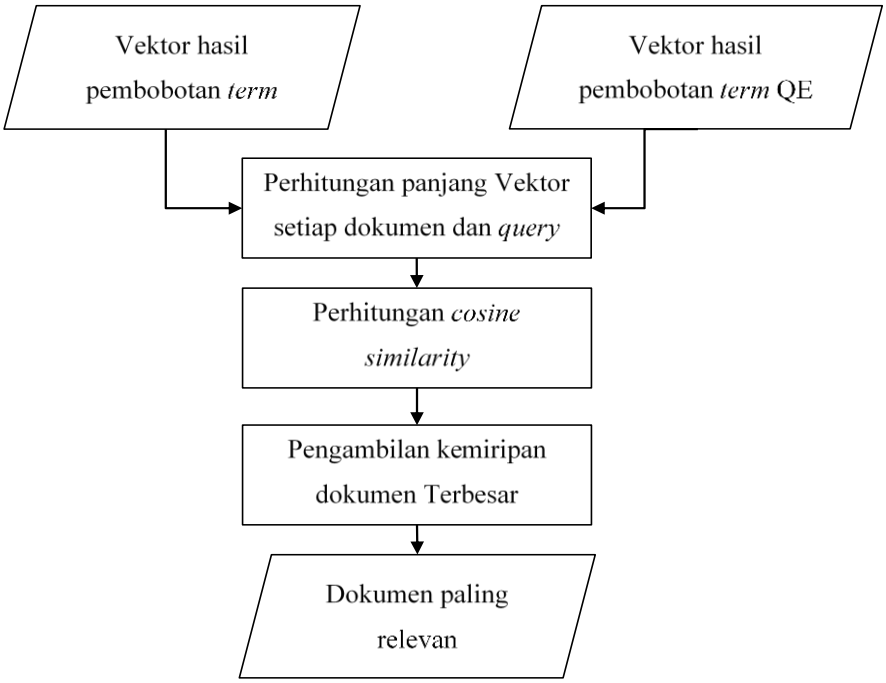
Tabel 3.7 Contoh Hasil Perhitungan IDF dan IBF pada *Term* Ekspansi *Query*

Term	df(t)	qf(t)	IDF	IBF
Q_0	2	1	1	1
Q_1	3	2	0,41	0
Q_2	3	2	0,41	0
Q_3	1	1	2	1

Tabel 3.8 Contoh Hasil Perhitungan TF-IDF-IBF pada *Term* Ekspansi *Query*

Term	TF(Q)	TF				IDF	IBF	TF-IDF-IBF	W_{QA}	|Q|
		D1	D2	D3	D4					
Dolar	1	1	0	1	0	1	1	1	2	2,24
Naik	1	1	1	0	1	0,41	0	0	0	
Harga	1	1	1	0	1	0,41	0	0	0	
Turun	1	1	0	0	0	2	1	2	1	

3.1.6 Tahapan Pengukuran Kemiripan *Query* dengan Dokumen



Gambar 3.9 Tahapan Perhitungan Kemiripan *Query* dengan Dokumen

Perhitungan *cosine similarity* digunakan untuk mengukur kemiripan *query* yang dimasukkan oleh pengguna dengan dokumen yang ada. *Query* yang digunakan diproses terlebih dahulu dengan melakukan *query expansion* dengan *word embeddings*. Perhitungan ini dapat menggunakan Persamaan 2.16. Tahapan pengukuran kemiripan *query* dengan dokumen dapat dilihat pada Gambar 3.9.

Tabel 3.9 Contoh Hasil Perhitungan *Cosine Similarity*

Cosim(Q_i, D_i)	W_{q1} dalam Q	W_{q1} dalam D*i*	W_{q2} dalam Q	W_{q2} dalam D*i*	W_{q3} dalam Q	W_{q3} dalam D*i*	W_{q4} dalam Q	W_{q4} dalam D*i*	$\|Q\|\times\|D_i\|$	Cosim
D1	1	1	0	0	0	0	2	2	5,02	0,99
D2	1	0	0	0	0	0	2	0	4,48	0
D3	1	1	0	0	0	0	2	0	6,72	0,15
D4	1	0	0	0	0	0	2	0	4,48	0

Contoh dari hasil perhitungan *cosine similarity* antara *query* awal dan ekspansi *query* dengan dokumen untuk pencarian dokumen yang relevan dengan Persamaan 2.17 dapat dilihat pada Tabel 3.9.

Cosime(Q_i, D_i) adalah *cosine similarity query* ke-*i* dengan dokumen ke-*i*. $W_{qi \text{ dalam } Q}$ merupakan bobot *term query expansion* ke-*i* dalam *query*. $W_{qi \text{ dalam } Di}$ adalah bobot *query* ke -*i* dalam dokumen ke-*i*. $|Q|\times|D_i|$ adalah panjang vektor matriks semua *query* dikalikan dengan panjang vektor matriks dokumen ke-*i*. Dokumen 1 merupakan dokumen paling relevan karena memiliki nilai *cosine similarity* paling besar. Sehingga, dokumen 1 akan ditampilkan sebagai dokumen yang di*retrieve* dari *query* yang dimasukkan.

3.2 Pembuatan Perangkat Lunak

Proses implementasi dari Tahapan sistem yang telah dirancang ke dalam bahasa pemrograman dimasukkan ke dalam tahapan pembuatan perangkat lunak. Tahapan ini menghasilkan suatu program sabagai representatif terhadap hasil dari metode yang diusulkan. Bahasa pemrograman yang digunakan untuk penelitian ini adalah bahasa pemrograman *Python*.

3.3 Uji Coba dan Evaluasi

Setelah tahapan pembuatan perangkat lunak selesai, maka tahapan penelitian ini akan dilanjutkan dengan melakukan suatu uji coba terhadap sistem yang telah dibuat. Dalam penelitian ini, metode evaluasi kesamaan dokumen yang digunakan adalah *precision*, *recall*, dan *f-score*. *F-Score* digunakan pada temu kembali informasi dengan mengkombinasikan konsep *recall* dan *precision*.

Metode sistem temu kembali informasi diaplikasikan untuk mendapatkan informasi yang relevan dengan keinginan pengguna. Dari kumpulan dokumen yang ada, penelitian ini melakukan pencarian berdasarkan *query* yang diberikan pengguna. Q*uery* tersebut akan dicari dokumen yang relevan. Dalam satu *dataset* kemungkinan metode memberikan dokumen relevan.

Evaluasi dilakukan untuk menganalisa performa dari metode pencarian yang digunakan. Analisa performa ini menggambarkan efektifitas metode yang direlasikan dengan kemampuan metode untuk mengembalikan dokumen yang relevan dengan *query*. Metode temu kembali informasi berusaha untuk mengembalikan dokumen yang relevan dan tidak mengembalikan dokumen yang tidak relevan.

Nilai *recall* dan *precision* sistem temu kembali informasi tersebut dapat dinyatakan seperti pada Persamaan 3.2 dan Persamaan 3.3.

$$Precision = \frac{TP}{TP+FP}, \tag{3.2}$$

$$Recall = \frac{TP}{TP+FN}, \tag{3.3}$$

dimana *TP* merupakan dokumen yang dikembalikan dan *FP* dokumen yang dikembalikan yang tidak relevan. *FN* merupakan dokumen yang tidak dikembalikan tetapi relevan dan *TN* merupakan dokumen yang tidak dikembalikan dan tidak relevan.

Proses uji coba akan dilakukan dengan membandingkan data *testing* dengan *Ground Truth* yang dimiliki pada penelitian ini. Pada dasarnya, nilai *recall* dan *precision* berada pada rentang antara 0 sampai dengan 1. Oleh karena itu, suatu sistem temu kembali yang baik adalah yang dapat memberikan nilai *recall* dan *precision* mendekati 1.

Nilai *recall* atau *precision* saja belum cukup mewakili kinerja sistem. Oleh karena itu diperlukan metode evaluasi yang mengkombinasikan metode evaluasi *recall* dan *precision*, metode evaluasi ini adalah *f-score*. Formulasi *f-score* dinyatakan dalam Persamaan 3.4.

$$F = \frac{2rp}{r+p}, \tag{3.4}$$

dengan *r* adalah *recall*, *p* adalah *precision*.

[*Halaman ini sengaja dikosongkan*]

BAB 4
IMPLEMENTASI DAN PEMBAHASAN

Pada bab ini menjelaskan hasil implementasi dari setiap langkah yang tertera pada Bab 3. Selanjutnya hasil uji coba yang dilakukan akan dijelaskan sesuai dengan skenario pengujian yaitu dengan menggunakan perhitungan *precision*, *recall*, dan *F-Score* pada metode yang diusulkan dan dibandingkan dengan metode sebelumnya.

Peneliti mengimplementasikan metode yang diusulkan dengan menggunakan bahasa pemrograman Python 3 dan beberapa library yang digunakan antara lain: Sastrawi, scikit-multilearn, pandas, dan beberapa library lain sebagai pendukung pembuatan program. Platform yang digunakan adalah Microsoft Windows 10 Pro 64-bit dengan spesifikasi processor Core i5 8th Gen.

4.1 Persiapan Dataset

Data yang digunakan dalam penelitian ini merupakan *corpus* atau kumpulan dokumen teks bahasa Indonesia. Dokumen-dokumen tersebut merupakan kumpulan artikel berita *online* bahasa Indonesia, antara lain dari website *www.kompas.com*, *www.tempo.co*, *www.liputan6.com*, *www.cnnindonesia.com*, dll.

Dokumen tersebut berupa beberapa dokumen yang memiliki beragam topik atau kategori, antara lain Olahraga, Ekonomi, dan Politik. Dokumen yang digunakan dikelompokkan sesuai topik yang berbeda dan telah dilakukan secara manual terlebih dahulu. Jumlah keseluruhan artikel berita yang diambil sebanyak 11.245 dengan rincian per-kategori yaitu: kategori Ekonomi sebanyak 3.629, Olahraga sebanyak 4.180, dan Politik sebanyak 3.436. Kumpulan artikel berita *online* tersebut dikumpulkan, kemudian diolah menjadi file berekstensi xml untuk kemudian dimasukkan ke dalam sistem yang dibentuk dalam penelitian ini.

Setiap artikel berita yang diambil memiliki spesifikasi tag seperti contoh pada Gambar 3.3, antara lain: id artikel berita <id>, url link artikel berita <url>, kategori artikel berita <kategori>, judul artikel berita <judul>, isi artikel berita <isi>, dan kata kunci artikel berita <kata_kunci>. Penelitian ini hanya akan

menggunakan bagian isi dari artikel berita dan kategori dari berita tersebut, yang kemudian dilakukan proses *preprocessing* dokumen untuk diolah selanjutnya.

Gambar 3.3 merupakan contoh dari dataset berita yang digunakan pada penelitian ini. Dokumen-dokumen inilah yang akan diproses dari tahap *preprocessing*, perhitungan bobot *term*, hingga proses perhitungan *cosine similarity* untuk mendapatkan dokumen yang relevan. Selain dokumen yang berisi artikel berita *online*, sistem juga memproses *query* masukan oleh pengguna yang dilakukan ekspansi *query* untuk kemudian digunakan dalam mencari dokumen yang relevan.

4.2 Tahapan Preprocessing Dokumen

Dokumen yang digunakan akan dilakukan *preprocessing* terlebih dahulu untuk mendapatkan *term-term* yang digunakan untuk proses pembobotan. *Preprocessing* terhadap data dilakukan dalam 4 tahap yaitu: *case folding*, *tokenization*, *filtering*, dan *stemming*. Proses ini dapat menghasilkan fitur-fitur atau *term* yang nantinya akan digunakan di dalam perangkingan dokumen.

Penelitian ini menggunakan library Sastrawi untuk proses *stemming* dan mendapatkan daftar *stopword* untuk proses *filtering*. Saat tahap *preprocessing*, masih banyak ditemukan kata-kata yang tidak dapat di*stemming* dan tidak dapat dihapus dengan *filtering*. Hal ini dikarenakan masih banyak kesalahan penulisan kata-kata dalam dokumen. Tabel 4.1 merupakan contoh *term-term* yang didapatkan hasil dari tahap *preprocessing* dokumen.

Tabel 4.1 Contoh *Term-term* Hasil *Preprocessing*

Id_doc	Id_topik	term
1	1	international
1	1	monetary
1	1	fund
1	1	imf
1	1	proyeksi
1	1	tumbuh
1	1	ekonomi

4.3 Tahapan Pembobotan Term

Tahapan pembobotan *term* TF-IDF-IBF dilakukan sebagai proses seleksi fitur. Bobot dapat menyatakan kepentingan *term* pada dokumen. *Term-term* yang didapat dari tahap *preprocessing* sebelumnya akan dihitung nilai *TF* sebagai frekuensi kemunculan *term t* pada dokumen *d*. Tabel 4.2 merupakan contoh nilai *TF* untuk masing-masing *term* terhadap dokumen. Tahap selanjutnya yaitu menghitung nilai *IDF* dan *IBF term* pada dokumen.

Tahap selanjutnya adalah menghitung bobot *TF-IDF-IBF* untuk masing-masing *term* dengan mengalikan nilai *TF*, *IDF*, dan IBF. Selain itu dihitung juga panjang vektor pada setiap dokumen untuk digunakan pada tahap pengukuran kemiripan *query* dengan dokumen. Hasil perhitungan bobot *TF-IDF-IBF* dapat ditunjukkan pada Tabel 4.3.

Tabel 4.2 Contoh Hasil Perhitungan *TF*

Id_doc	term	tf
1	acu	1
1	akhir	2
1	aktivitas	1
1	anggar	1
1	angka	1
21	umum	1
21	verifikasi	1

Tabel 4.3 Contoh Hasil Perhitungan Bobot TF-IDF-IBF

Term	**TF-IDF**	**TF-IDF-IBF**
acu	1,243	1,836
akhir	1,532	1,532
aktivitas	1,845	2,725
anggar	1,544	2,281
angka	1,845	2,725

4.4 Tahapan Query Expansion dengan Word Embeddings

Subbab ini akan menampilkan beberapa contoh hasil dari proses *query expansion* dengan *word embeddings*. Tahapan pertama dalam *query expansion*

Tabel 4.4 Contoh Hasil *Query Expansion*

Term	**Similarity**
sepertiga	0,671
jumlah	0,833
persentase	0,639
sebesar	0,823
triliun	0,633
meningkat	0,822
juta	0,628
melebihi	0,815
pdb	0,624
juta	0,804

adalah *query* asli yang diinputkan oleh pengguna akan di-*preprocessing* dan diekspansi menggunakan model *word embeddings* yang sudah ada seperti *Word2Vec* dan *GloVe*. Contoh *query* asli yang dimasukkan adalah "Puji Ekonomi RI, IMF Ramal Pertumbuhan Capai 5,1 Persen". Peneliti menggunakan *library word2vec* dan *glove* untuk digunakan pada tahap *query expansion*. Hasil *preprocessing* pada *query* yang dimasukkan antara lain puji, ekonomi, ri, imf, ramal, tumbuh, capai, dan persen.

Peneliti menyimpan 5 *terms* hasil *query expansion* yang memiliki nilai similaritas *word embeddings* yang terbesar untuk masing-masing model. *Term-term* hasil *query expansion* tersebut akan digunakan pada tahap selanjutnya yaitu tahap pembobotan *term* hasil ekspansi *query*. Tabel 4.4 merupakan contoh *term-term* hasil ekspansi *query* dan nilai similaritas antara *term* dengan *query* asli menggunakan model *word2vec* dan *glove* yang disimpan dan digunakan pada tahap selanjutnya.

4.5 Tahapan Pembobotan Term Hasil Ekspansi Query

Subbab ini akan menampilkan contoh hasil pembobotan *term* hasil ekspansi *query* dan memilih *term* mana yang akan digunakan pada tahap pengukuran kemiripan *query* dengan dokumen, sehingga didapatkan dokumen yang paling relevan. *Term-term* hasil ekspansi *query* menggunakan *word embeddings* tahap

sebelumnya akan diolah dihitung bobot *TF-IDF-IBF* dengan melihat kepentingan *term* tersebut di dalam dokumen. *Term* hasil ekspansi *query* yang memiliki bobot paling tinggi akan terpilih untuk tahap pengukuran kemiripan *query* dengan dokumen. Dengan melakukan pemilihan *term* hasil ekspansi *query* terlebih dahulu, maka proses pencarian dokumen yang dilakukan dapat lebih relevan. Hal tersebut dikarenakan *term* hasil ekspansi *query* terpilih mempunyai kedekatan semantik yang tinggi dengan *query* asli ditandai dengan nilai similaritas *word embeddings*nya dan merupakan *term* yang representatif dalam dokumen ditandai dengan nilai bobot *TF-IDF-IBF*nya yang tinggi.

Tahap pertama dalam pembobotan *term* hasil ekspansi *query* adalah menggabungkan *term* hasil ekspansi *query* dari model *word2vec* dan *glove*. Tahap selanjutnya adalah menghitung menghitung bobot *TF-IDF-IBF* untuk *term* ekspansi *query* terhadap dokumen yang digunakan dalam dataset seperti pada Tabel 4.5. Tahap selanjutnya adalah pemilihan *term* hasil ekspansi *query* yang memiliki kedekatan semantik tinggi terhadap *query* asli dan merupakan *term* yang representatif dalam dokumen ditandai dengan nilai bobot *TF-IDF-IBF* tinggi yang ditandai dari nilai bobot *term* hasil ekspansi *query* akhir (W_q) sebanyak 4 terbesar. Nilai W_q didapatkan dengan mengalikan bobot *term* hasil QE (*TF-IDF-IBF*$_q$) dengan nilai similaritas *word embeddings* (S_q) atau pada Gambar 4.7 adalah kolom "similarity" dari proses *word embeddings* sebelumnya yang sudah disimpan. Tabel 4.5 merupakan contoh hasil *term query* asli dan *term* hasil ekspansi *query* yang terpilih berdasarkan 4 bobot *term* ekspansi *query* terbesar.

Tabel 4.5 menjelaskan bahwa *term* "juta", "pdb", dan "jumlah" merupakan *term* hasil ekspansi *query* yang terpilih karena memiliki nilai bobot W_q terbesar. *Term* hasil ekspansi *query* yang terpilih hanya 3 *term*, hal ini dikarenakan ada *term* yang memiliki nilai kedekatan semantik tinggi, namun dianggap tidak penting di dalam dokumen. Seluruh *term-term* tersebut yang akan digunakan pada tahap pengukuran kemiripan dokumen *query* dengan dokumen selanjutnya. Sedangkan *term-term* yang tidak terpilih akan dibuang.

Term-term hasil ekspansi *query* tersebut dihitung bobot *TF-IDF-IBF*$_q$ terhadap dokumen. Bobot akhir *term* dari ekspansi *query* didapatkan dengan

mengalikan bobot *TF-IDF-IBF*$_q$ dengan nilai faktor yang dimiliki, seperti pada Persamaan 2.16.

W_{QA_i} adalah nilai akhir bobot *TF-IDF-IBF*$_q$ pada *term* ke-*i* dan α adalah nilai faktor pada *term* dari *query* dan ekspansi *query*. α bernilai 2 jika *term* berasal dari *query* asli dan bernilai 0,5 jika *term* berasal dari hasil ekspansi *query*. Tujuan dari mengalikan bobot dengan nilai α adalah agar *query* asli akan tetap memiliki kedudukan lebih tinggi dari pada *term* hasil ekspansi *query*, sehingga dokumen yang di*retrieve* tetap sesuai dengan *query* asli bukan berdasarkan *term* hasil ekspansi *query*. Tabel 4.5 merupakan hasil perhitungan bobot akhir seluruh *query*.

Tabel 4.5. Contoh Hasil Perhitungan Bobot pada *Term Query* Asli dan QE

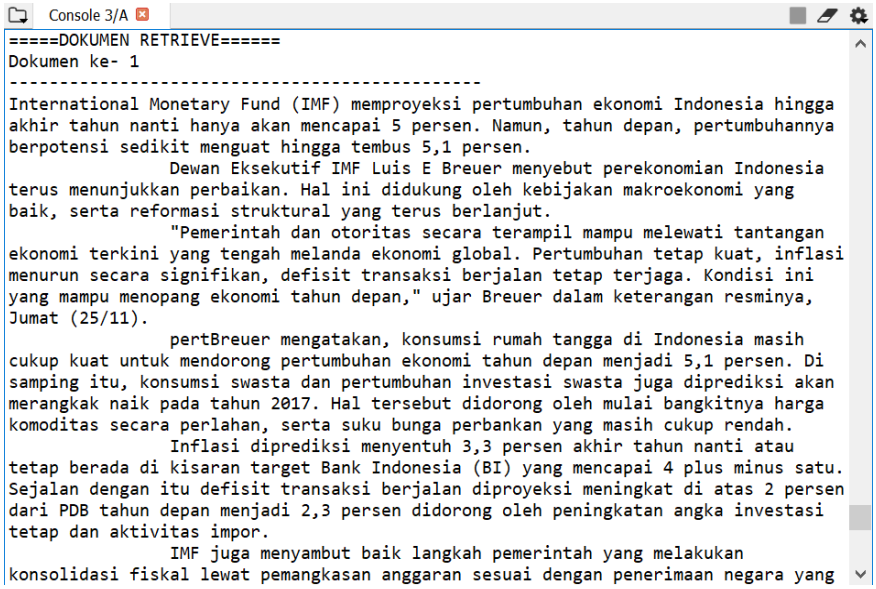
Asal query	**Term**	**TF-IDF-IBF**	**W_{qa}**
Query asli	ekonomi	14,689	14,689
	imf	13,627	13,627
	tumbuh	12,124	12,124
	persen	6,236	6,236
	capai	3,386	3,386
	puji	2,725	2,725
	ramal	2,281	2,281
Term QE	juta	5,451	1,712
	pdb	4,561	1,423
	jumlah	1,067	0,444

4.6 Tahapan Pengukuran Kemiripan Query dan Dokumen

Subbab ini akan menampilkan contoh hasil pencarian dokumen yang relevan dengan *query* yang diinputkan oleh pengguna dan dilakukan ekspansi *query*. Tahap pencarian dokumen yang relevan dengan menghitung nilai *cosine similarity* seluruh *query* dengan dokumen yang digunakan. Tabel 4.9 menjelaskan bahwa dokumen 1 (D1) mempunyai nilai cosim terbesar, sehingga dokumen 1 adalah dokumen yang paling relevan dengan *query* yang diinputkan oleh pengguna dan diekspansi. Gambar 4.1 merupakan contoh dokumen relevan yang berhasil di-*retrieve*.

Tabel 4.6 Contoh Hasil Perhitungan *Cosine Similarity*

Dokumen ke-	Cosine similarity	Dokumen ke-	Cosine similarity
1	0,525	7	0,0004
2	0,142	8	0,035
3	0,155	9	0,0004
4	0,305	14	0,038
5	0,315	15	0,027
6	0,045	20	0,007



Gambar 4.1 Contoh Tampilan Dokumen yang Terpilih

4.7 Dokumentasi Uji Coba

Uji coba penelitian ini menggunakan 100 *query* yang telah dikumpulkan dari beberapa artikel berita *online*. 100 *query* tersebut merupakan kata_kunci dari artikel berita *online* dengan hasil dokumen yang di-*retrieve* yang telah disimpan sebagai *ground truth*. Uji coba yang dilakukan pada 100 *query* ini dibandingkan dengan menggunakan 5 metode yang berbeda yaitu metode yang diusulkan (WE dengan TF-IDF-IBF), TF-IDF, TF-IDF-IBF, *word embeddings* model *Word2Vec*, dan model *GloVe*. Uji coba dilakukan dengan mengambil 4 *term* hasil ekspansi *query* (*top term QE*), 8 *term* hasil ekspansi *query*, dan 10 *term* hasil ekspansi *query*

teratas untuk proses pemilihan jumlah *term* hasil ekspansi *query*. *Term-term* tersebut kemudian digunakan untuk menentukan *retrieve document* terpilih.

Hasil *retrieve document* yang didapatkan, selanjutnya dilakukan pengecekan dengan *ground thruth* untuk menghitung keakuratan hasil kebenarannya yang kemudian dievaluasi menggunakan *recall*, *precision*, dan *F-Score*. Tabel 4.7 merupakan tabel perbandingan hasil *precision*, *recall*, dan *F-Score* pada metode yang diusulkan yaitu *word embeddings* dengan *TF-IDF-IBF* dengan menggunakan beberapa paramater jumlah *term* hasil ekspansi ekspansi *query* (*top term QE*).

Hasil uji coba dengan penggunaan 4 *top term QE*, 8 *top term QE*, dan 10 *top term QE*. Hasil *precision*, *recall*, dan *f-score* yang diperoleh dengan penggunaan 4 *top term QE* mendapatkan hasil yang tertinggi dibandingkan dengan penggunaan parameter lainnya. Nilai *precision* yang diperoleh sebesar 0,751, *recall* sebesar 0,742, dan *f-score* sebesar 0,743. Sedangkan, hasil *precision*, *recall*, dan *f-score* dari penggunaan parameter 8 *top term QE* lebih baik daripada penggunaan parameter 10 *top term QE*, dimana nilai *precision* sebesar 0,705, *recall* sebesar 0,691, dan *f-score* sebesar 0,696. Nilai hasil dengan penggunaan 10 *top term QE* yang didapat yaitu *precision* sebesar 0,639, *recall* sebesar 0,639, dan *f-score* sebesar 0,638.

Uji coba juga dilakukan dengan membandingkan metode yang menggunakan *query expansion* yang sudah ada dan dilakukan dengan 3 parameter

Tabel 4.7 Hasil *Precision*, *Recall*, dan *F-Score Top Term QE* pada Metode WE dengan TF-IDF-IBF

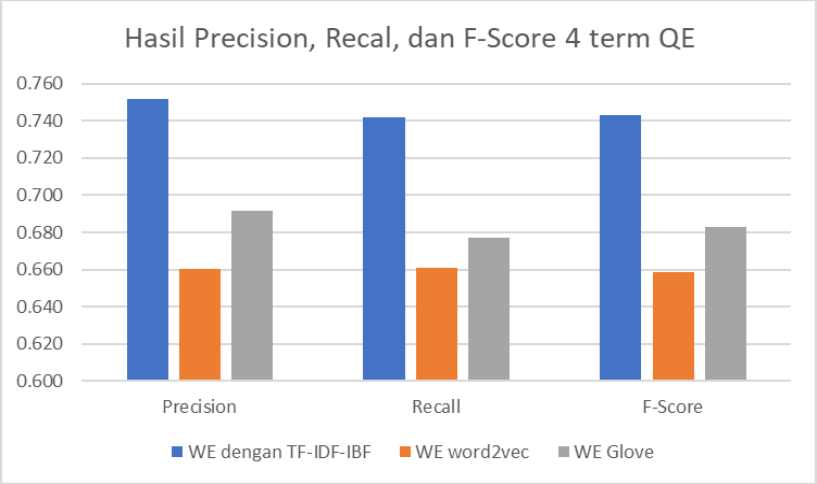
Top Query	4 top term QE	8 top term QE	10 top term QE
Precision	**0,751**	0,705	0,639
Recal	**0,742**	0,691	0,639
F-Score	**0,743**	0,696	0,638

jumlah *top term QE* yang sama yaitu 4 *top term QE*, 8 *top term QE*, dan 10 *top term QE*. Tabel 4.8 dan Gambar 4.2 merupakan hasil *precision*, *recall*, dan *f-score* yang didapat dengan menggunakan parameter 4 *top term QE*. Nilai *precision, recall*, dan *f-score* tertingi diperoleh pada penggunaan metode yang diusulkan yaitu nilai *precision* dari *word embeddings* dengan *TF-IDF-IBF* sebesar 0,751, *recall* sebesar

0,742, dan *f-score* sebesar 0,743. Sedangkan hasil *precision*, *recall*, dan *f-score* terendah didapat dari penggunaan metode *word embeddings* dengan *Word2Vec* yaitu dengan nilai *precision* sebesar 0,660, *recall* sebesar 0,661, dan *f-score* sebesar 0,659.

Tabel 4.8 Hasil *Precision*, *Recall*, dan *F-Score* pada 4 *Top Term QE*

Top Query	Precision	Recall	F-Score
WE dengan TF-IDF-IBF	**0,751**	**0,742**	**0,743**
WE dengan Word2Vec	0,660	0,661	0,659
We dengan Glove	0,692	0,677	0,683

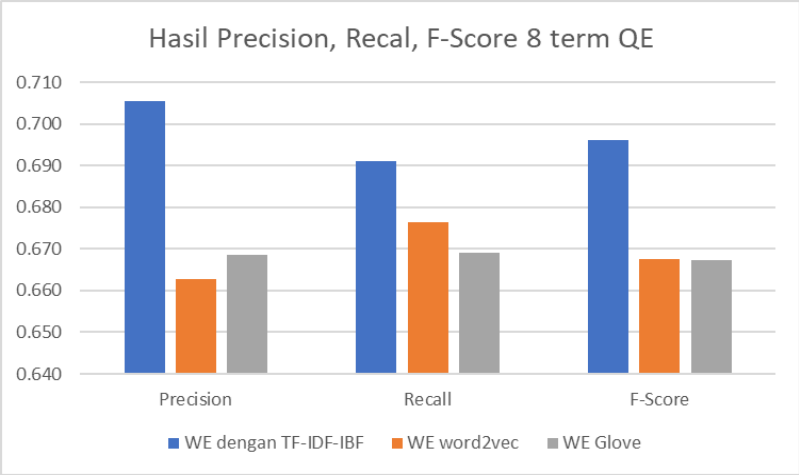


Gambar 4.2 Grafik Perbandingan Hasil *Precision*, *Recall*, dan *F-Score* pada 4 *Top Term QE*

Tabel 4.9 dan Gambar 4.3 merupakan hasil *precision*, *recall*, dan *f-score* yang diperoleh dengan menggunakan parameter 8 *top term QE*. Nilai *precision*, *recall*, dan *f-score* tertingi diperoleh pada penggunaan metode yang diusulkan yaitu nilai *precision* dari *word embeddings* dengan *TF-IDF-IBF* sebesar 0,705, *recall* sebesar 0,691, dan *f-score* sebesar 0,696. Sedangkan hasil *precision* terendah didapat dari penggunaan metode *word embeddings* dengan *Word2Vec* yaitu dengan nilai *precision* sebesar 0,663. Namun, hasil *recall*, dan *f-score* terendah didapat dari penggunaan metode *word embeddings* dengan *GloVe* yaitu dengan nilai *recall* sebesar 0,669, dan *f-score* sebesar 0,667.

Tabel 4.9 Hasil *Precision*, *Recall*, dan *F-Score* pada 8 *Top Term QE*

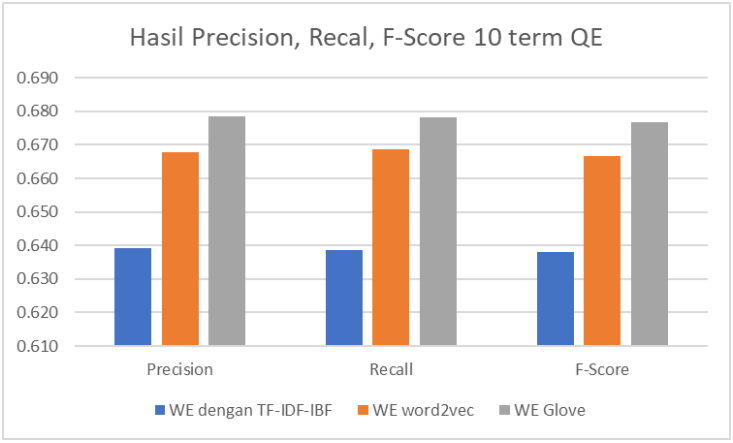
Top Query	Precision	Recall	F-Score
WE dengan TF-IDF-IBF	**0,705**	**0,691**	**0,696**
WE dengan Word2Vec	0,663	0,676	0,668
We dengan Glove	0,668	0,669	0,667



Gambar 4.3 Grafik Perbandingan Hasil *Precision*, *Recall*, dan *F-Score* pada 8 *Top Term QE*

Tabel 4.10 Hasil *Precision*, *Recall*, dan *F-Score* pada 10 *Top Term QE*

Top Query	Precision	Recall	F-Score
WE dengan TF-IDF-IBF	0,639	0,639	0,638
WE dengan Word2Vec	0,668	0,669	0,667
We dengan Glove	**0,678**	**0,678**	**0,677**



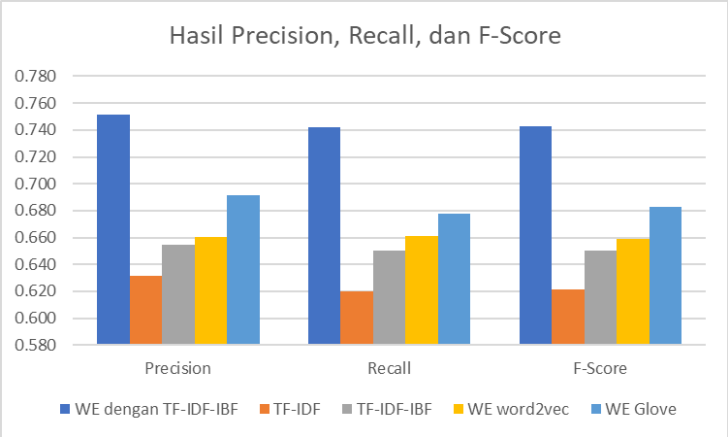
Gambar 4.4 Grafik Perbandingan Hasil *Precision*, *Recall*, dan *F-Score* pada 10 *Top Term QE*

Tabel 4.10 dan Gambar 4.4 merupakan hasil *precision*, *recall*, dan *f-score* yang didapat dengan menggunakan parameter 10 *top term QE*. Nilai *precision*, *recall*, dan *f-score* tertingi didapatkan pada penggunaan metode *word embeddings* dengan *GloVe* yaitu nilai *precision* sebesar 0,678, *recall* sebesar 0,678, dan *f-score* sebesar 0,677. Sedangkan hasil *precision*, *recall*, dan *f-score* terendah didapat dari penggunaan metode yang diusulkan yaitu nilai *precision* dari *word embeddings* dengan *TF-IDF-IBF* sebesar 0,639, *recall* sebesar 0,639, dan *f-score* sebesar 0,638.

Uji coba terakhir juga dilakukan dengan membandingkan metode yang diusulkan dengan metode pembobotan *term* yang sudah ada sebelumnya dan metode *word embeddings*. Metode pembobotan *term* sebelumnya yang dibandingkan adalah metode pembobotan *TF-IDF* dan *TF-IDF-IBF*. Tabel 4.11 dan Gambar 4.5 merupakan hasil perbandingan *precision*, *recall*, dan *f-score* dari metode yang diusulkan dengan metode yang sudah ada sebelumnya. Parameter yang digunakan pada metode yang diusulkan dan metode *word embeddings* dengan *Word2Vec* dan *Glove* yaitu parameter jumlah 4 *top term QE*. Nilai uji coba perbandingan pada Tabel 4.11 dan Gambar 4.5 didapatkan hasil *precision*, *recall*, dan *f-score* terbaik terdapat pada penggunaan metode yang diusulkan yaitu *word embeddings* dengan *TF-IDF-IBF*. Hasil *precision* yang didapat yaitu sebesar 0,751, *recall* sebesar 0,742, dan *f-score* sebesar 0,743. Sedangkan hasil *precision*, *recall*, dan *f-score* didapat dengan

Tabel 4.11 Hasil Perbandingan *Precision*, *Recall*, dan *F-Score*

Top Query	Precision	Recall	F-Score
WE dengan TF-IDF-IBF	**0,751**	**0,742**	**0,743**
TF-IDF	0,632	0,620	0,622
TF-IDF-IBF	0,655	0,650	0,650
WE dengan Word2Vec	0,660	0,661	0,659
We dengan Glove	0,692	0,677	0,683



Gambar 4.5 Grafik Perbandingan Hasil *Precision*, *Recall*, dan *F-Score*

menggunakan metode pembobotan *TF-IDF* yaitu nilai *precision* sebesar 0,632, *recall* sebesar 0,620, dan *f-score* sebesar 0,622. Hasil uji coba yang telah dilakukan menunjukkan bahwa metode yang diusulkan yaitu penggunaan metode *word embeddings* dengan pembobotan *TF-IDF-IBF* lebih optimal jika dibandingkan dengan metode pembobotan *term* dan *word embeddings* lainnya dengan pemilihan parameter 4 *top term QE* pada sistem.

4.8 Analisa Hasil

Hasil penelitian yang telah dilakukan menunjukkan bahwa metode yang diusulkan mampu untuk mencapai tujuan utama penelitian yaitu meningkatkan hasil pencarian dokumen yang relevan dengan melakukan pemilihan *term* hasil ekspansi *query* yang memiliki korelasi tinggi dengan *query* asli berdasarkan metode *word embeddings* sekaligus merupakan *term* yang representatif dalam dokumen dengan *Inverse Book Frequency* (IBF).

Uji coba dilakukan dengan 100 *query* dan memberikan hasil terbaik pada penggunaan parameter 4 *top term QE* untuk proses pemilihan jumlah *term* hasil ekspansi *query* yang terpilih dengan memperoleh hasil *f-score* sebesar 0,743. Hasil uji coba mengalami penurunan jika menggunakan parameter 8 *top term QE* yaitu hasil rata- *f-score* sebesar 0,696. Hasil uji coba semakin menurun jika menggunakan parameter 10 *top term QE* yaitu hasil *f-score* sebesar 0,638. Hal ini menunjukkan bahwa semakin banyak jumlah *term* hasil ekspansi *query* yang digunakan, maka nilai *f-score* akan semakin turun. Penurunan kinerja ini disebabkan oleh semakin

banyak *term* hasil ekspansi *query*, dokumen yang akan diseleksi semakin banyak dan menjadi kurang relevan terhadap *query* asli yang dimasukkan.

Kekurangan di dalam sistem yang dibuat ini adalah hanya dapat menemukan satu dokumen yang dianggap paling relevan dengan *query*. Hal tersebut dikarenakan sistem hanya melakukan pemilihan pada dokumen yang memiliki nilai *cosine similarity* paling tinggi terhadap *query* yang dimasukkan dan *query expansion*nya. Kesimpulan dari uji coba pada penelitian ini adalah kinerja sistem yang didapatkan dapat lebih optimal jika menggunakan parameter jumlah *top term* hasil ekspansi *query* kecil. Semakin besar pemilihan parameter jumlah *top term* hasil ekspansi *query*, dapat menghasilkan dokumen yang kurang relevan terhadap *query* asli yang dimasukkan dengan dokumen yang digunakan.

[*Halaman ini sengaja dikosongkan*]

BAB 5
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini mengusulkan metode baru pembobotan *term* pada hasil ekspansi *query* berdasarkan tingkat korelasi *term* terhadap *query* dan frekuensi *term* menggunakan metode *word embeddings* dan *Inverse Book Frequency* (IBF) untuk pencarian dokumen. Sistem yang dibuat dapat melakukan proses *preprocessing* dokumen, menghitung bobot TF-IDF-IBF, memilih *term* hasil ekspansi *query* dengan *word embeddings* dan IBF, menghitung bobot pada *term* hasil ekspansi *query* dengan metode baru yang diusulkan, dan mengukur kemiripan dokumen dengan *query* untuk mendapatkan dokumen yang relevan.

Pemilihan *terms* hasil ekspansi *query* bertujuan untuk menghapus *terms* hasil ekspansi *query* yang tidak relevan dari kumpulan dokumen. Hasil ekspansi *query* yang dipilih harus berisi informasi yang cukup dan berhubungan dengan dokumen yang digunakan. Dengan melakukan pemilihan hasil ekspansi *query* terlebih dahulu dapat meningkatkan kualitas dokumen yang di-*retrieve* sesuai dengan *query* asli yang diberikan.

Metode kombinasi pembobotan *term* TF-IDF dengan pembobotan IBF digunakan pada penelitian ini bertujuan agar *term-term* yang didapat dari hasil *preprocessing* dokumen tidak hanya memperhatikan frekuensi persebaran *term* dalam dokumen. Namun, juga memperhatikan persebaran *term* dalam dokumen yang telah dikelompokkan ke dalam berbagai kategori secara manual.

Word embeddings digunakan untuk mendapatkan ekspansi *query* secara otomatis. Penelitian ini menggunakan beberapa model *word embeddings* dan menyimpan semua hasil ekspansi *query* menjadi satu array. *Term-term* hasil ekspansi *query* akan dihitung pembobotan *term*-nya dengan TF-IDF-IBF. Beberapa kesimpulan dari hasil uji coba yang dilakukan adalah sebagai berikut:

1. Hasil uji coba dengan parameter 4 *top term QE* memperoleh nilai tertinggi yaitu dengan nilai *precision* sebesar 0,751, *recall* sebesar 0,742, dan *f-score* sebesar 0,742.

2. Hasil uji coba metode yang diusulkan dengan parameter 4 *top term QE* dan 8 *top term QE* memperoleh hasil yang lebih optimal dibandingkan dengan metode *word embeddings* yang sudah ada antara lain *Word2Vec* dan *GloVe* yaitu *f-score* sebesar 0,751 dan 0,696. Namun, hasil uji coba metode yang diusulkan dengan parameter 10 *top term QE* mendapatkan hasil yang rendah jika dibandingkan metode *word embeddings* lainnya yaitu *f-score* sebesar 0,638.
3. Hasil uji coba metode yang diusulkan dibandingkan dengan metode pembobotan dan metode *word embeddings* yang sudah ada menunjukkan hasil yang lebih optimal untuk pencarian dokumen yang relevan.

Kesimpulan dari uji coba pada penelitian ini adalah kinerja sistem yang didapatkan dapat lebih optimal jika menggunakan parameter jumlah *top term* hasil ekspansi *query* kecil. Semakin besar pemilihan parameter jumlah *top term* hasil ekspansi *query*, dapat menghasilkan dokumen yang kurang relevan terhadap *query* asli yang dimasukkan dengan dokumen yang digunakan.

5.2 Saran

Saran untuk penelitian selanjutnya adalah sebagai berikut:

1. Penerapan metode yang diusulkan pada dataset lainnya.
2. Penambahan metode untuk mengatasi *Word Sense Disambiguation* agar dapat mengidentifikasi makna kata yang ambigu dalam kalimat tertentu ketika kata memiliki sejumlah makna yang berbeda.

DAFTAR PUSTAKA

Bintana, R. R., Fatichah, C. and Purwitasari, D. (2018) "Pencarian Book-Answer Menggunakan Convolutional Neural Network Pada Topik Agama Berbahasa Indonesia," *Ultimatics*, vol. x, no. 1, pp. 57-64.

Choi, J., Park, Y. and Yi, M. (2016) "A Hybrid Method for Retrieving Medical Documents with Query Expansion," *IEEE International Conference on Big Data and Smart Computing*, pp. 411-414.

Colace, F., Santo, M. D., Greco, L. and Napoletano, P. (2015) "Weighted Word Pairs for query expansion," *Elsevier, Information Processing and Management*, vol. 51, pp. 179-193

Cui, H., Wen, J, Nie, J., and Ma, W. (2003) "Query expansion by mining user logs," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 4, pp. 829–839.

Elekes, A., Schaler, M. and Bohm, K. (2017) "On the Various Semantics of Similarity in Word Embedding Tahapans," *ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, 19-23.

Fauzi, A., Arifin, A. Z. and Yuniarti, A. (2014) "Term Weighting Berbasis Indeks Buku Dan Kelas Untuk Pencarian Dokumen Berbahasa Arab," *Lontar Komputer: Jurnal Ilmiah Teknologi Informasi*, vol. 5, no. 2.

Francis, C. F., Jorge, H. and Manuel, M. (2018) "A Prospect-Guided global query expansion strategy using word embeddings," *ELSEVIER Information Processing and Management*, vol. 54, pp. 1-13.

Hakim, A. A., Erwin, A., Eng, K. I., Galinium, M. and Muliady, W. (2014) "Automated Document Classification for News Article in Bahasa Indonesia based on Term Frequency Inverse Document Frequency (TF-IDF) Approach," *IEEE 6th International Conference on Information Technology and Electrical Engineering (ICITEE)*.

Hersh, W. (2008) “Information Retrieval: A Health and Biomedical Perspective,” *Springer*.

Hidayatin, L. and Rahutomo, F. (2018) “Query Expansion Evaluation for Chatbot Application,” *IEEE International Conference on Applied Information Technology and Innovation (ICAITI)*.

Lukmana, I., Arifin, A. Z. and Purwitasari, D. (2016) “Pencarian Dokumen Berbahasa Indonesia Berdasarkan Model Topik Menggunakan Latent Dirichlet Allocation dan Metadata,” *Master Theses of Informatics Engineering, Institut Teknologi Sepuluh Nopember*

Manning, D., Raghavan, P. and Schutze, H. (2009) “Introduction to Information Retrieval,” *Cambridge University* Press. https://doi.org/10.1109/LPT.2009.2020494

Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013) “Efficient estimation of word representations in vector space,” arXiv preprint arXiv: 1301.3781.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. and Dean, J. (2013) “Distributed Representations of Words and Phrases and their Compositionality,” arXiv:1301.3781

Mikolov, T., Yih, W.-t. and Zweig, G. (2013) “Linguistic regularities in continous space word representations,” *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 746-751.

Niasita, A. N., Adikara, P. P. and Adinugroho, S. (2019) “Analisis Sentimen Pembangunan Infrastruktur di Indonesia *Automated Lexicon Word2Vec* dan *Naive-Bayes*,” Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, Vol. 3, No. 3, pp. 2673-2679.

Pennington, J., Socher, R. and Manning, C. D. (2014) “GloVe: Global Vectors for Word Representation,” *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1531-1542.

Ren, F. and Sohrab, M. G. (2013) "Class-indexing-based term weighting for automatic text classification," *ELSEVIER Journal Information Science*, vol. 236, pp. 109-125.

Salton, G. (1989) "Automatic Text Processing: the Transformation, Analysis, and Retrieval of Information" *by Computer.: Addison-Wesly*

Suleiman, D and Awajan, A. (2018) "Comparative study of word embeddings models and their usage in Arabic language applications," *IEEE International Arab Conference on Information Technology (ACIT)*.

Wang, Y., Liu, S., Afzal, N., Rastegar-Mojarad, M., Wang, L., Shen, F., Kingsbury, P. and Liu, H. (2018) "A Comparison of Word Embeddings for the Biomedical Natural Language Processing," *ELSEVIER, Journal of Biomedical Informatics*, https://doi.org/10.1016/j.jbi.2018.09.008

Xu, J. and Croft, W. B. (1996) "Query expansion using local and global document analysis," in *Proc. 19th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, pp. 4–11.

Yunianto, D. R. and Arifin, A. Z. (2017) "Pengukuran Kemiripan Term Berbasis Co-occurrence dan Inverse Class Frequency pada Pengembangan Thesaurus Bahasa Arab," *Master Theses of Informatics Engineering, Institut Teknologi Sepuluh Nopember*

Zhang, W., Ming, Z., Zhang, Y., Liu, T. and Chua, T. (2016) "Capturing the Semantics of Key Phrases Using Multiple Languages for Question Retrieval," *IEEE Transactions on Knowledge and Data Engineering.*, vol. 28, no. 4, pp. 888–900

[*Halaman ini sengaja dikosongkan*]

LAMPIRAN

Lampiran. Daftar *Query* yang digunakan pada uji coba.

No	*Query*
1	Kenaikan Harga Barang Penyebab Inflasi Berlanjut ke Desember
2	Seluruh Harga Barang dan Jasa Naik, Inflasi Juni Tembus 0,54%
3	BI Minta Pemerintah Hati-hati Naikkan Harga Gas dan Listrik
4	PLN Dinilai Seharusnya Tak Perlu Naikkan Tarif Listrik
5	Tarif Listrik Rumah Mewah, Mal, dan Industri Naik
6	Pengusaha Ritel: Tarif Listrik Naik, Inflasi Bakal Menanjak
7	Pemerintah Klaim Harga Barang Kebutuhan Pokok Tidak Naik
8	Ditunda Sebulan, Tarif Listrik Dipastikan Naik Mei 2015
9	Kenaikan Harga BBM dan Transportasi Picu Inflasi Maret 0,17%
10	Tarif Naik, KAI Klaim Tak Ada Penurunan Jumlah Penumpang
11	Tarif Kereta Resmi Naik, Ongkos Mudik Tahun Ini Lebih Mahal
12	Harga Barang Jasa Naik Moderat, BI Ramal Inflasi Juli 0,6%
13	Harga Makanan dan Minuman akan Naik 10 Persen
14	Organda Naikkan Tarif Angkutan Umum 30 Persen
15	Menhub Izinkan Tarif Angkutan Naik 10 Persen
16	Mandiri Sekuritas: Kenaikan Harga Energi Picu Inflasi April
17	Harga-Harga Naik, BI Prediksi Inflasi April 0,44%
18	Prediksi Harga Naik, Indeks Keyakinan Konsumen Melemah
19	ICP Maret Turun, Pemerintah Malah Naikkan Harga BBM
20	Perhitungan Harga BBM Tak Jelas, Faisal Cs Minta Formula Baku
21	Juni, Harga Produk Pertanian Naik Tinggi di Tingkat Produsen
22	2015, Tarif Kiriman Barang Naik 15 Persen
23	Politikus Gerindra Tolak Kenaikan Harga BBM
24	Kenaikan BBM Risiko Presiden Jokowi
25	Kenaikan BBM Dongkrak Biaya Logistik Industri
26	Polisi Amankan SPBU Jelang Harga BBM Naik
27	Naikkan Harga BBM, Pemerintah Harus Lapor DPR
28	JK Isyaratkan Kenaikan Harga BBM Mendadak
29	BBM Naik Karena Kondisi Fiskal Sedang Rawan

30	Harga BBM Naik Setelah 16 November 2014
31	Harga Rokok Naik, November Inflasi 0,21 Persen
32	25 Juta PKL Serempak Naikkan Harga Jual
33	BBM Naik, Rabu Angkutan Umum Mogok Nasional
34	Rossi: Honda Melesat Tinggalkan Yamaha
35	Akhir Sempurna Honda dan Marquez di Motegi
36	Rossi: Saya Ogah Finis di Posisi Dua
37	Valentino Rossi: Marquez Pantas Juara Dunia
38	Marquez di Jepang: Tahun Lalu Membuang, Tahun Ini Mengunci
39	Valentino Rossi Tak Merasa Tanda-tanda Akan Tergelincir
40	Dilanda Kegugupan, Marquez Lupa Ganti Persneling
41	Marquez: Saya Lihat Rossi Gugup
42	Pendukung Rossi Langsung Tinggalkan Sirkuit
43	Marc Marquez Juara Dunia MotoGP 2016
44	Marquez Ingat Nenek Usai Juara Dunia MotoGP
45	Marquez Kaget Juara Dunia di Motegi
46	Konsistensi, Kunci Marquez Juara Dunia MotoGP 2016
47	Rossi: Pebalap Spanyol Sukar Dikalahkan di Aragon
48	Mengintip Museum Nostalgia Honda di Sirkuit Motegi
49	Berkemah di Sirkuit Motegi, Pilihan Pintar Nonton MotoGP
50	Rossi: Marquez dan Lorenzo Masih Lebih Cepat
51	Jorge Lorenzo Gembira Start di Baris Depan
52	Kecelakaan Pedrosa Buat Marquez Makin Berhati-hati
53	Jalan Berliku Marquez Kembali ke Puncak Dunia
54	Ketika Raja Sirkuit Motegi Terlempar ke Udara
55	Bebas dan Tertib Ala MotoGP Jepang
56	Marquez Takut Jatuh di Jepang
57	Dovizioso Ungguli Marquez dan Rossi di FP 1 GP Jepang
58	Lorenzo Tercepat di FP2 GP Jepang, Pedrosa Kecelakaan
59	Lorenzo Pertanyakan Sikap Yamaha
60	Lorenzo Ingin Lengkapi Karier dengan Juara Dunia di Ducati
61	Mengenal Motegi, Sirkuit yang Sukar Ditaklukkan Rossi

62	Lorenzo Tuding Ban Michelin Sebabkan Kecelakaan
63	Parade Moge Tarik Perhatian di Sirkuit Motegi
64	Vinales Terlihat `Bete` Usai Marquez Jadi Juara Dunia
65	Berbisnis Lewat Tanda Tangan Bintang MotoGP
66	Rossi vs Lorenzo, Api yang Tersisa di MotoGP 2016
67	Dana Awal Kampanye Ahok Paling Kecil, Anies Dominasi Medsos
68	Dana Awal Kampanye Agus-Sylvi Hanya Rp5 Juta
69	Djarot 'Gerilya' Tanpa Ahok di Hari Pertama Kampanye
70	Diskusi Isu Pemuda Jadi Awal Kampanye Agus Yudhoyono
71	Anies Baswedan Memulai Kampanye dari Tebet
72	Pilkada Jakarta Disebut Miniatur Demokrasi Indonesia
73	Aksi Iwan Bule dan Pilkada DKI Jakarta
74	Plafon Dana Kampanye DKI Jakarta Belum Ditetapkan
75	Total Dana Kampanye Anies-Sandi Bertambah Rp3 Miliar
76	Peserta Pilkada Tak Bisa Terima Sumbangan dalam Bentuk Tunai
77	KPU Minta Kandidat Serahkan Laporan Awal Dana Kampanye Besok
78	Anies Baswedan Klaim Tak Biasa Berkoar-koar
79	Tak Hadiri Penetapan Calon, Ahok-Djarot Berdalih Pilih Kerja
80	Bidik Pedagang, Anies Janjikan Pembenahan Pasar Tradisional
81	KPU Jakarta Taksir Batas Dana Kampanye Rp70 Miliar
82	Refleksi Sikap Politik Ahok dan Gemuruh Pendukung Agus
83	Ahok Artikan Nomor Dua sebagai Dua Periode Jabatan
84	Pilkada Jakarta dan 'Kutukan' Nomor Urut
85	Nomor 'Terkutuk' di Tangan Agus Yudhoyono
86	Ahok Wajib Hadiri Pengundian Nomor Urut Calon Gubernur
87	Pendukung Ahok-Djarot Gemakan Salam Dua Jari
88	Sophia Latjuba Diberi Waktu Seminggu Kuasai Program Ahok
89	Ahok Gunakan Mobil Bekas Selama Kampanye
90	Satu Dua dan Tiga untuk Agus, Ahok, dan Anies
91	Undian Nomor Urut, Agus Datang Awal dan Anies Paling Terakhir
92	Pengganti Ahok dan Rano Karno Diminta Fokus Keamanan Pilkada
93	Jelang Pilkada, Polisi Maksimalkan Patroli Siber di Medsos

94	Kapolda Metro Jaya Ajak Kampanye Santun dan Beretika
95	Anies Janji Legalisasi Kampung Ilegal
96	Timses Ahok: Anies Salah Soal Larangan Pemegang KJP
97	Agus Yudhoyono Tidak Percaya Kutukan Nomor Urut Satu
98	Bawaslu DKI-Polda Koordinasi soal Pelanggaran Pilkada
99	Ahok Tak Dapat Dukungan dari Koalisi Buruh Jakarta di Pilkada
100	PDIP: Keputusan Dukung Ahok-Djarot Final, yang Tak Suka Suarakan Lewat TPS