TESIS

PENGUKURAN KEMIRIPAN TERM BERBASIS CO-OCCURRENCE DAN INVERSE CLASS FREQUENCY PADA PENGEMBANGAN THESAURUS BAHASA ARAB

DIKA RIZKY YUNIANTO
NRP. 5115201007

DOSEN PEMBIMBING:
Dr. Agus Zainal Arifin, S.Kom., M.Kom.

PROGRAM MAGISTER
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M.Kom.)
di
Institut Teknologi Sepuluh Nopember Surabaya

oleh:
Dika Rizky Yunianto
Nrp. 5115201007

Dengan judul :
PENGUKURAN KEMIRIPAN TERM BERBASIS CO-OCCURRENCE DAN INVERSE CLASS FREQUENCY PADA PENGEMBANGAN THESAURUS BAHASA ARAB

Tanggal Ujian : 4-1-2017
Periode Wisuda : 2016 Gasal

Disetujui oleh:

Dr.Agus Zainal Arifin, S.Kom, M.Kom
NIP. 197208091995121001
(Pembimbing 1)

Dr. Eng. Chastine Fatichah, S.Kom, M.Kom
NIP. 197512202001122002
(Penguji 1)

Diana Purwitasari, S.Kom, M.Sc
NIP. 197804102003122001
(Penguji 2)

Dini Adni Navastara, S.Kom, M.Sc
NIP. 198510172015042001
(Penguji 3)

an. Direktur Program Pascasarjana
Asisten Direktur

Prof. Dr. Ir. Tri Widjaja, M.Eng.
NIP. 19611021 198603 1 001

Direktur Program Pasca Sarjana,

Prof.Ir.Djauhar Manfaat, M.Sc., Ph.D.
NIP. 196012021987011001

PENGUKURAN KEMIRIPAN TERM BERBASIS CO-OCCURRENCE DAN INVERSE CLASS FREQUENCY PADA PENGEMBANGAN THESAURUS BAHASA ARAB

Nama mahasiswa : Dika Rizky Yunianto
NRP : 5115201007
Pembimbing : Dr. Agus Zainal Arifin, S.Kom., M.Kom.

ABSTRAK

Thesaurus merupakan *tools* yang bermanfaat untuk melakukan *query expansion* dalam pencarian dokumen. Thesaurus adalah kamus yang dibentuk dengan melihat kemiripan *term*. Kemiripan *term* dalam pembentukan thesaurus secara otomatis salah satunya dilakukan dengan pendekatan statistikal dari *term* pada dokumen-dokumen *corpus*. Beberapa thesaurus pada bahasa arab dibentuk dengan menggunakan pendekatan statistikal. Salah satu pendekatan statistikal adalah teknik *co-occurrence* yang memperhatikan frekuensi kemunculan *term* secara bersama-sama. Melihat kemiripan *term* dalam pembentukan thesaurus tidak hanya bergantung pada nilai informatif suatu *term* terhadap dokumen. Namun juga nilai informatif suatu *term* terhadap cluster. Dokumen-dokumen corpus dikumpulkan kemudian dilakukan proses preprocessing untuk medaptakan daftar term. Daftar *term* tersebut akan dihitung nilai TF-IDF nya sebagi fitur untuk melakukan clustering pada dokumen. Dokumen yang telah ter-cluster akan dijadikan patokan untuk menghitung nilai *Inverse Class frequency* (ICF). Nilai TF – ICF digunakan untuk perhitungan *cluster weight* pada teknik co-occurence dimana perhitungan tersebut memperhatikan kemunculan bersama kedua *term*. Hasil dari *cluster weight* yang melibatkan TF-ICF tersebut menjadi patokan nilai kemiripan term dalam pembentukan thesaurus. Pengujian terhadap thesaurus hasil bentukan metode usulan menghasilkan nilai *precision* tertinggi sebesar 76,7% sedangkan *recall* memiliki nilai terbesar 81,8% dan *f-measure* sebesar 54,1%.

Kata kunci: Teknik *Co-occurence, Inverse Class Frequency*, Kemiripan Term, Thesaurus Bahasa Arab

[Halaman ini sengaja dikosongkan]

TERM SIMILARITY MEASUREMENT BASED CO-OCCURRENCE TECHNIQUE AND INVERSE CLASS FREQUENCY ON ARABIC THESAURUS

Nama mahasiswa : Dika Rizky Yunianto
NRP : 5115201007
Pembimbing : Dr. Agus Zainal Arifin, S.Kom., M.Kom.

ABSTRACT

Thesaurus is a useful tool to perform query expansion in the document search. Dictionary Thesaurus is formed by looking at the similarities term. Similarities in the formation of a thesaurus term is automatically one of them carried out by statistical approach of the term in the document corpus. Some thesaurus in Arabic is formed by using a statistical approach. One approach is a statistical technique that takes into account the co-occurrence frequency of occurrence of terms together. See the resemblance in the formation of a thesaurus term depends not only on the informative value of a term of the document. But also informative value of a term to the cluster. The documents collected corpus preprocessing process is then performed to medaptakan term list. The term list will be calculated the value of its TF-IDF as a feature to perform clustering on the document. Documents that have already been cluster will be used as a benchmark to calculate the value of Inverse Class frequency (ICF). TF value - ICF is used for the calculation of weight in the engineering cluster co-occurence where the calculation of the notice of appearance with the two terms. Results of cluster weight involving TF-ICF has become a benchmark value of term similarity in the formation of a thesaurus. Tests on the thesaurus result form the proposed method produces the highest precision value amounted to 76.7%, while the recall has the greatest value 81.8% and f-measure of 54.1%.

Keywords: *Co-occuence Technique, Inverse Class Frequency*, *Term Similarity, Arabic Thesaurus*

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Segala puji syukur kepada Allah SWT yang telah melimpahkan rahmat dan hidayah Nya sehingga penulis dapat menyelesaikan Tesis yang berjudul " PENGUKURAN KEMIRIPAN TERM BERBASIS CO-OCCURRENCE DAN INVERSE CLASS FREQUENCY PADA PENGEMBANGAN THESAURUS BAHASA ARAB" sesuai dengan target dan waktu yang diharapkan.

Proses pembuatan dan pengerjaan Tesis ini merupakan pengalaman yang sangat berharga bagi penulis untuk memperdalam ilmu pengetahuannya khususnya di bidang komputasi cerdas dan *information retrieval* khususnya pengolahan teks. Oleh karena itu, penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada :

1. Allah SWT atas limpahan rahmat, karunia serta ilmu Nya sehingga penulis dapat menyelesaikan Tesis ini dengan baik.
2. Ibu Endah Wijayati dan Bapak Susapto selaku orang tua penulis yang selalu memberi bantuan dan dukungan baik secara moril maupun materil kepada penulis agar senantiasa diberi kelancaran dalam menyelesaikan Tesis ini.
3. Ibu Dr. Agus Zainal Arifin, S.Kom, M.Kom selaku Dosen Pembimbing penulis yang telah memberikan kepercayaan, perhatian, bimbingan, bantuan dan motivasi kepada penulis dalam proses menyelesaikan tesis ini.
4. Bapak Dr. Eng. Chastine Fatichah, S.Kom., M.Kom. , Ibu Diana Purwitasari, S.Kom., M.Sc., Ibu Dini Adni Navastara, S.Kom., M.Sc. selaku Dosen Penguji yang telah memberikan bimbingan, arahan, nasehat dan koreksi dalam pengerjaan Tesis ini.
5. Bapak Waskitho Wibisono, S.Kom., M.Eng., PhD selaku ketua program Pascasarjana Teknik Informatika ITS serta Dosen Pascasarjana Teknik Informatika ITS lainnya yang telah memberikan ilmunya.
6. Mbak Lina, Mas Kunto dan segenap staf Tata Usaha yang telah memberikan segala bantuan dan kemudahan kepada penulis selama menjalani kuliah di Teknik Informatika ITS.

7. Kakak penulis tersayang Diah Ivana Sari dan suami, ponakan tercinta Naufal Fayzan Almadina serta seluruh keluarga besar penulis yang selalui memberikan dukungan dan semangat kepada penulis.
8. Sahabat-sahabat penulis Dimas Widhiastara Putra, Devic Oktora, Razqyan Mas Bimatyugrajati yang selalu membantu demi terselesaikannya dan lancrnya Tesis ini.
9. Teman-teman seperjuangan Rizka Wakhidatus, Rarasmaya, Rosetya Septiyawan, Fawwaz Ali, M. Sonhaji, Wawan Gunawan, Andreyan, Nur Fajri Azhar, Didih serta teman-teman angkatan 2015 lain yang selalu ada di saat penulis mengalami suka dan duka.
10. Tidak lupa kepada semua pihak yang belum sempat disebutkan satu per satu disini yang telah membantu terselesaikannya Tesis ini

Penulis menyadari bahwa Tesis ini masih jauh dari kesempurnaan dan banyak kekurangan. Untuk itu dengan segala kerendahan hati penulis mengharapkan kritik dan saran yang membangun dari para pembaca.

Surabaya, Januari 2017

Dika Rizky Yunianto

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

[Halaman ini sengaja dikosongkan]

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Pada hakekatnya, temu kembali informasi harus dapat menampilkan dokumen-dokumen yang relevan sesuai dengan keiinginan pengguna pada proses pencarian dokumen. Terdapat permasalahan dimana kata kunci atau *query* yang digunakan untuk melakukan pencarian dokumen memiliki makna yang berbeda-beda melihat batas keamampuan pengguna dalam pemilihan kata-kata yang digunakan dalam pencarian (Otair, Ph, Amman, Kanaan, & Ph, 2013). Sebagai contoh, ketika pengguna menggunakan *query* "putaran partai final", yang dimaksud oleh kata atau *term* "partai" pada *query* tersebut apakah "partai" pada konteks politik ataukah "partai" pada konteks olahraga. Perbedaan makna sebuah *term* merupakan permasalahan yang terus dikaji dalam bidang temu kembali informasi agar dokumen-dokumen yang dihasilkan dalam pencarian dokumen relevan dengan keiinginan pengguna. Sistem harus dapat mengatasi permasalahan ambiguitas suatu kata serta harus dapat mengatasi ketidak cocokan antara dokumen dengan *query* dari pengguna. Untuk mengatasi hal tersebut terdapat *tools* atau kamus yang dinamakan dengan thesaurus (Y. Tseng, 2002).

Thesaurus merupakan kamus yang minimal berisi daftar kemiripan *term* dan dapat digunakan sebagai alat dalam melakukan *query expansion* sehingga meningkatkan relevansi hasil pencarian (Khafajeh, Refai, & Yousef, 2013). Yang dimaksud dengan kemiripan *term* bukan hanya *term* yang memiliki artian sama saja, namun *term-term* yang memiliki hubungan semantik atau kemiripan berdasarkan konsep konteks atau representasi objek yang sama (Li, Wang, Zhu, Wang, & Wu, 2013). Thesaurus merupakan *tools* yang efektif dan telah terbukti dalam penelitian bidang temu kembali informasi maupun bidang pemrosesan bahasa. Sebagai contoh thesaurus sangat membantu dalam melakukan *query expansion* untuk menemukan dokumen yang relevan (Ito, Nakayama, Hara, & Nishio, 2008). Selain itu thesaurus juga dapat digunakan untuk melakukan

pemrosesan tanya jawab secara otomatis atau yang sering dikenal dengan *question answering* (Z. Chen, Liu, Wenyin, Pu, & Ma, 2003).

Terdapat dua cara dalam melakukan pembangunan kamus thesaurus yaitu, pembangunan secara manual dan secara otomatis. Pembangunan thesaurus secara manual memiliki permasalahan dalam lama waktu proses pembangunan serta sumber daya yang dibutuhkan. Oleh sebab itu dibutuhkan pembangunan thesaurus dengan cara otomatis untuk menekan biaya dan efektifitas waktu (Otair et al., 2013).

Pembangunan thesaurus secara otomatis memiliki banyak cara, salah satunya dengan mencari kesamaan kata dilihat dari hubungan halaman *website* pada Wikipedia (Ito et al., 2008). Selain itu pembentukan thesaurus secara otomatis dapat dilakukan dengan menghitung kemiripan *term* secara statistikal. *Pointwise mutual information* dan *dice* dapat digunakan untuk menghitung kemiripan *term* secara simetris (Zohar, Liebeskind, Schler, & Dagan, 2013).

Khafajeh dkk, melakukan penelitian untuk membangun thesaurus Bahasa Arab secara otomatis dengan pendekatan statistikal *co-occurrence*. Teknik *co-occurrence* digunakan untuk menemukan kemiripan antar *term* dalam melakukan pembangunan kamus thesaurus (Khafajeh et al., 2013). Bahasa Arab digunakan sebagai studi kasus dalam pembangunan kamus thesaurus tersebut dikarenakan Bahasa Arab memiliki morfolologi yang kompleks sehingga masih sedikit pengembangannya. (Otair et al., 2013).

Teknik *co-occurrence* untuk membangun thesaurus Bahasa Arab merupakan perhitungan kemiripan antar *term* secara asimetris dengan melihat kemunculan bersama kedua term (Y. H. Tseng, 2002). Beberapa peneliti mengatakan bahwa perhitungan kemiripan secara asimetris lebih baik dibandingkan perhitungan secara simetris. Hal tersebut dikarenakan perhitungan secara simetris akan menyebabkan keberulangan atau kemunculan *term-term* yang bersama akan terhitung lebih sering, sehingga tidak begitu membantu dalam melakukan eksplorasi kata kunci atau *query expansion* pada saat pencarian dokumen (Khafajeh et al., 2013).

Teknik *co-occurence* dilakukan dengan memperhatikan frekuensi kemunculan bersama kedua *term*. Jika terdapat dua buah *term* yaitu *term J* dan *term*

K, maka perhitungan kemiripan kedua *term* tersebut dengan teknik *co-occurence* dilakukan dengan melihat probabilitas jumlah nilai keinformatifan kedua term terhadap *term J* pada dokumen yang kemudian diperkuat dengan probabilitas kemunculan bersama *term K* terhadap *term J* (H. Chen, Yim, Fye, & Schatz, 1995). Namun, melihat nilai keinformatifan suatu *term* tidak hanya dapat dilihat dari sisi dokumen saja, melainkan juga dapat dilihat dari kluster dokumen itu berada (Fauzi et al., 2015).

Fauzi dkk, pada penelitiannya melihat nilai keinformatifan suatu *term* dengan menghitung nilai *Inverse Class Frequency* atau ICF dari *term* tersebut. Hal tersebut dilakukan untuk melihat nilai keinformatifan atau penting tidaknya suatu *term* pada kelas atau kluster. Pembobotan yang tidak hanya memperhatikan frekuensi *term* pada dokumen namun juga memperhatikan frekuensi *term* pada kelas, akan menghasilkan *term-term* yang memiliki nilai informasi yang tinggi pada suatu kelas (Fauzi et al., 2015).

Untuk mendapatkan nilai keinformatifan suatu *term* tersebut, maka diperlukan klusterisasi pada dokumen. Clusterisasi dokumen-dokumen *corpus* dalam melakukan pembangunan thesaurus secara otomatis dapat membantu meningkatkan relevansi antar *term*, dikarenekan dokumen-dokumen tersebut berkelompok-kelompok berdasarkan karakternya yang sama sehingga *term-term* pada dokumen di cluster yang sama juga memiliki nilai kemiripan yang tinggi. Sedangkan *term-term* pada cluster yang berbeda merupakan *term-term* yang memiliki topik bahasan yang berbeda sehingga nilai relevansinya jauh.

Oleh sebab itu pada penelitian ini mengusulkan sebuah metode perhitungan kemiripan term yang memperhatikan pembobotan terhadap kelas atau cluster dengan menggabungkan teknik *co-occurrence* dan *inverse class frequency* pada pembentukan thesaurus Bahasa Arab untuk meningkatkan relevansi *term-term* pada thesaurus.

Inverse class frequency atau ICF akan membantu memberikan nilai pada *term-term* yang memiliki nilai informasi tinggi pada suatu cluster sehingga akan mempengaruhi nilai kemiripan dengan *term-term* yang berbeda cluster. Meningkatnya nilai relevansi *term-term* pada thesaurus akan membantu

meningkatkan nilai relevansi pada pencarian dokumen. Sehingga hasil pencarian dokumen akan sesuai dengan *query* yang dimasukkan oleh pengguna.

1.2 Perumusan Masalah

Dari latar belakang dapat dirumuskan sebagai berikut:

1. Bagaimana meningkatkan relevansi kemiripan *term* dengan berbasis pada *co-occurrence* dan *inverse class frequency* pada pengembangan thesaurus Bahas Arab?
2. Bagaimana pengaruh nilai treshold untuk hasil dari *co-occurence* - ICF pada pengembangan thesaurus Bahasa Arab?
3. Bagaimana pemanfaatan *Inverse Class Frequency* pada pengembangan thesaurus Bahasa Arab?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk meningkatkan relevansi kemiripan *term* dengan berbasis pada *co-occurrence* dan *inverse class frequency* pada pengembangan thesaurus Bahasa Arab.

1.4 Manfaat Penelitian

Manfaat yang diperoleh dalam penelitian ini antara lain :

1. Dari sisi ilmu pengetahuan adalah untuk menambah sumber kajian keilmuan khususnya dalam bidang pengukuran kemiripan *term*.
2. Perhitungan kemiripan *term* dengan menggabungkan teknik *co-occurrence* dan *inverse class frequency* melihat kemiripan informasi suatu *term* lebih spesifik karena melihat informasi berdasarkan dokumen dan *cluster* dari dokumen.

1.5 Kontribusi Penelitian

Kontribusi pada penelitian ini adalah mengusulkan metode baru dalam pengukuran kemiripan *term* dengan berbasis pada *co-occurrence* dan *inverse class frequency* pada pengembangan thesaurus Bahasa Arab.

1.6 Batasan Masalah

Batasan-batasan dari penelitian ini adalah sebagai berikut:

1. Thesaurus yang dibentuk merupakan daftar kata yang memiliki kemiripan berdasarkan konteks yang sama.
2. Dokumen yang digunakan sebagai studi kasus dari peneltian ini adalah dokumen fiqih berbahasa arab.

[Halaman ini sengaja dikosongkan]

BAB 2
KAJIAN PUSTAKA DAN DASAR TEORI

2.1 Kajian Pustaka

2.1.1 *Preprocessing* Dokumen

Preprocessing merupakan salah satu proses yang ada pada pemrosesan temu kembali informasi. Pada proses ini beberapa peneliti menggunakan *library* yang telah tersedia untuk melakukan *preprocessing* terhadap dokumen-dokumen *corpus*. Salah satu *library* yang dapat digunakan adalah *library lucene*.

Lucene sendiri memiliki fasilitas yang cukup lengkap mulai dari tahapan *tokenizing, filttration, stopword removal, stemming* hingga *indexing*. Rujia Gao dkk dalam penelitiannya menggunakan *library lucene* untuk membentuk sebuah sistem temu kembali informasi dalam pencarian dokumen berbahasa inggris. Pada sistem tersebut *library lucene* digunakan disetiap prosesnya mulai dari *preprocessing* hingga *indexing* (Gao, Li, Li, & Dong, 2012).

Dalam Bahasa Arab terdapat beberapa *stemmer* atau pengubah kata menjadi bentuk kata dasar. Majdi dan Eric dalam penelitiannya membandingkan tiga *stemmer* Bahasa Arab yaitu *Shereen Khoja Stemmer*, *Tim Buckwalter Morphological Analyzer*, dan *Tri-literal Root Extraction Algorithm*. Dari ketiga *stemmer* tersebut didapatkan bahwa *Shereen Khoja Stemmer* memiliki nilai akurasi yang lebih tinggi dibandingkan dengan kedua *stemmer* lainnya (Sawalha & Atwell, 2008).

2.1.2 *Clustering* Dokumen K-Means

Clustering dokumen memiliki tujuan untuk mengumpulkan dokumen-dokumen yang memiliki kemiripan dalam satu kelompok. Sudah cukup banyak metode clustering yang digunakan untuk melakukan pengelompokan atau clusterisasi dokumen.

Michael dkk, dalam penelitiannya membandingkan beberapa metode *clustering,* yaitu metode *hierarichal aglomerative* dengan K-Means. Dari percobaan yang dilakukan akurasi dari metode K-Means lebih baik dibandingkan

dengan hasil akurasi dari metode *hierarichal aglomerative* (Steinbach, Karypis, & Kumar, 2000).

Manjot dkk pada penelitiannya juga menggunakan metode K-Means untuk melakukan *clustering* pada dokumen *website.* Dalam penelitiannya dia melakukan optimasi dalam penentuan nilai *centroid* awal. Kelemahan dari metode K-Means adalah penentuan inisialisai centroid pada awl proses. Optimasi tersebut dilakukan agar inisialisai *centroid* di awal tidak acak, sehingga dapat mengurangi waktu pemrosesan. Dari hasil percobaan dengan adanya inisialisai *centroid* di awal yang tidak acak maka, waktu pemrosesan berkurang dibandingkan dengan inisialisai yang acak (Kaur & Kaur, 2013).

2.1.3 Inverse Class Frequency

Inverse Class Frequency atau ICF merupakan metode pembobotan *term* untuk melihat nilai keinformatifan suatu *term* terhadap kelas. Beberapa teknik pembobotan ICF dilakukan pada penelitian sebelumnya, dimana ICF digunakan untuk menghitung bobot *term* untuk dilakukan pengindeksan yang digunakan sebagai perankingan dokumen. Khadijah dkk, menggunakan pembobotan *term* ICF dan dikombinasikan dengan pembobotan berbasis preferensi pengguna untuk mendapatkan hasil perangkingan dokumen yang relevan dengan *query* dari pengguna (Holle, Arifin, & Purwitasari, 2015).

Wahib dkk, menggabungkan ICF dengan perhitungan *latent semantic indexing* untuk mendapatkan dokumen yang relevan pada pencarian dokumen (Wahib, Santika, & Arifin, 2015). Sedangkan Septyawan dkk, menggabungkan kedua metode *latent semantic indexing* dan pembobotan berbasis preferensi pengguna dikombinasikan dengan ICF untuk melakukan perankingan dokumen (Wardhana, Yunianto, Arifin, & Purwitasari, 2015).

2.1.4 Co-occurence

Teknik *co-occurence* merupakan teknik dalam pembentukan thesaurus yang melihat kemunculan term secara bersama-sama. Teknik tersebut telah digunakan oleh beberapa peneliti seperti Yuen dkk, menggunakan teknik *co-occurence* untuk membentuk thesaurus dalam bahasa cina. Teknik tersebut digabungkan dengan

metode segmentasi huruf cina mengingat morfologi bahasa cina yang kompleks dalam pembentukan kata dan frase (Y. H. Tseng, 2002).

Khafejah dkk, membentuk thesaurus Bahasa Arab dengan teknik *co-occurence.* Dimana peneliti membandingkan hasil pencarian dokumen menggunakan thesaurus dengan teknik *co-occurence* dengan pencarian dokumen tanpa menggunakan thesaurus. Hasil yang didapat dari percobaan tersebut nilai *recall* pada pencarian dokumen menggunakan thesaurus dengan teknik *co-occurence* lebih besar diabandingkan dengan pencarian dokumen tanpa menggunakan thesaurus (Khafajeh et al., 2013).

2.1.5 Thesaurus

Thesaurus merupakan *tools* yang membantu dalam melakukan pencarian dokumen. Beberapa penelitian sebelumnya telah melakukan pembentukan thesaurus secara otomatis dengan berbagai metode. Carolyn dkk, melakukan pembentukan thesaurus secara otomatis dengan pendekatan statistikal berbasis pada *discrimination value model* dan *complete link clustering.* Pendekatan ini dinyatakan pendekatan yang mampu mengurangi biaya dan memiliki waktu pemrosesan yang efektif (Crouch & Yang, 1992). Guntzer dkk, membangun thesaurus dengan melihat dari *session* pengguna. Dimana *machine learning* membangun sebuah model untuk menyamakan kemiripan kata dilihat dari *session* dari *query* pengguna (Gijntzer, Juttner, Seegmuller, & Sarre, 1989). Hoa Xu dkk, juga membangun sebuah thesaurus untuk *spam filtering* dengan pendekatan *revised back propagation neural network*. Pendekatan tersebut merupakan pengembangan dari *back propagation neural network* yang konvensional (Xu & Yu, 2010).

Thesaurus dibutuhkan di berbagai bahasa sebagai pengembangan di bidang temu kembali informasi. Bahasa Arab merupakan salah satu bahasa yang banyak digunakan di dunia. 23 negara menggunakan Bahasa Arab sebagai bahasa resminya, dan hamper 422 juta orang menggunakan Bahasa Arab sebagai Bahasa keseharian. Pengembangan thesaurus Bahasa Arab memiliki keunikan tersendiri dikarenakan Bahasa Arab memiliki morfologi yang berbeda jika dibandingkan dengan Bahasa Inggris atau Bahasa Indonesia (Otair et al., 2013). Pengembangan thesaurus dalam Bahasa Arab dapat bermanfaat jika dilihat dari topiknya. Fiqih

merupakan salah satu topik pada dokumen Bahasa Arab dimana *term-term* pada topik ini memiliki makna yang berbeda ketika digunakan pada bahasa sehari-hari. Sehingga pengembangan thesaurus Bahasa Arab pada topik Fiqih ini dapat menjadi suatu kontribusi tersendiri.

Dalam pembentukan thesaurus secara statistikal diperlukan pengukuran kemiripan antar *term* untuk melihat relasi atau hubungan kedua *term* tersebut. Beberapa penelitian terdahulu melakukan perhitungan kemiripan *term* dengan berbagai cara. Peter dalam penelitiannya melakukan perbandingan perhitungan kemiripan *term* yang dapat diterapkan dalam pembangunan thesaurus secara otomatis antara perhitungan probabilitas *term* dengan *pointwise mutual information* yang dengan perhitungan matriks *latent semantic indexing*. Dari penelitian tersebut didapatkan bahwa dalam perhitungan kemiripan *term* metode *pointwise mutual information* lebih baik dibandingkan dengan *latent semantic indexing* dalam hal jumlah cakupan dokumen yang besar (Turney, 2001).

Perhitungan Jaccard juga pernah dilakukan Suphakit dkk, untuk menghitung kemiripan term dengan studi kasus kemiripan kata kunci. Hasil dari penilitian tersebut dinyatakan bahwa Jaccard dapat digunakan untuk melakukan perhitungan kemiripan kata kunci namun kurang dapat mengatasi permasalahan *over-typed* kata kunci (Niwattanakul, Singthongchai, Naenudorn, & Wanapu, 2013) .

2.2 Dasar Teori

2.2.1 *Preprocessing* Dokumen

Preprocessing dokumen merupakan salah satu proses dalam melakukan temu kembali informasi. Dokumen-dokumen yang akan diproses akan melalui preprocessing terlebih dahulu hingga menghasilkan *term-term* yang siap untuk dihitung atau diproses ke tahapan berikutnya. Dalam proses ini terbagi menjadi beberapa tahapan yaitu (D. Manning, Ragavan, & Schutze, 2009):

1. *Tokenizing*

Pada tahapan ini dilakukan pemecahan terhadap isi dokumen berdasarkan *delimiter* seperti spasi sehingga menjadi *term-term* yang berdiri sendiri.

2. *Filtartion*

Filtration merupakan tahapan menghilangkan simbol-simbol atau karakter-karakter yang tidak penting.

3. *Normalization*

Pada tahapan ini akan dilakukan perubahan term yang pada awalnya terdapat huruf kapital menjadi huruf kecil semua.

4. *Stopwords Removal*

Pada tahapan ini term-term yang sering muncul di banyak dokumen atau dianggap tidak memiliki nilai informasi akan dihilangkan. Menghilangkan *term-term* tersebut akan dibantu dengan kamus daftar *term-term* yang dianggap tidak memiliki nilai keinformatifan.

5. *Stemming*

Pada tahapan ini dilakukan perubhan *term-term* yang ada menjadi bentuk kata dasar dengan menghilangkan imbuhan awalan serta akhiran.

2.2.2 Clustering Dokumen K-Means

Dokumen-dokumen *corpus* yang banyak dan tidak memiliki label perlu diclusterisasi untuk dikumpulkan dengan dokumen-dokumen yang sejenis. Metode untuk melakukan clusterisasi dapat dibagi dua yaitu secara hirarki dan secara partisi. Untuk melakukan clusterisasi pada dokumen, metode partisi sangat cocok karena kebutuhan komputasi yang rendah. Sedangkan metode hirarki memiliki kompleksitas waktu yang tinggi (Mahdavi & Abolhassani, 2009).

Metode clusterisasi K-Means merupakan salah satu metode partisi yang sangat mudah untuk diimplementasikan serta memiliki waktu kompleksitas yang rendah. Metode K-Means dilakukan dengan memilih K-dokumen sebagai *centroid*. Kemudian menghitung jarak setiap dokumen terhadap dokumen-dokumen *centroid*. Perubahan titik *centroid* dilakukan secara berulang hingga tercapainya *stop criteria* (Mahdavi & Abolhassani, 2009).

Perhitungan jarak dokumen dengan titik *centroid* dilakukan dengan konsep *vector space model* dimana perhitungan nya menggunakan *cosine similarity* seperti

pada **persamaan 2.1** dimana *d* merupakan titik *centroid* cluster ke *a*, sedangkan d' merupakan dokumen ke *i* (Mahdavi & Abolhassani, 2009).

$$\cos(d, d') = \frac{d \cdot d'}{|d||d'|} \quad (2.1)$$

Perubahan titik *centroid* dilakukan setelah semua dokumen yang ada telah terbagi kedalam cluster-cluster. Perubahan titik *centroid* akan berhenti hingga *stop criteria* terpenuhi. *Stop criteria* dapat terjadi bila perubahan titik *centroid* tidak signifikan atau telah didefinisikan di awal batas perulangan dari proses clusterisasi itu sendiri. Dalam perubahan titik *centroid* cluster dilakukan dengan mengikuti **persamaan 2.2** dimana d_i merupakan vector dokumen pada cluster s_j . c_j merupakan vector centroid sedangkan n_j merupakan jumlah dokumen yang terdapat pada kalstetr s_j (Gupta & Srivastava, 2014) .

$$c_j = \frac{1}{n_j} \sum_{\forall d_i \in s_j} d_i \quad (2.2)$$

2.2.3 *Inverse Class Frequency*

Inverse Class Frequency yang disingkat menjadi ICF merupakan salah satu metode pembobotan *term*. Pembobotan *term* dengan ICF memperhatikan kemunculan *term* pada kumpulan kategori atau kelas atau cluster. *Term* yang jarang muncul pada banyak cluster adalah *term* yang bernilai untuk klasifikasi. Kepentingan tiap *term* diasumsikan memiliki proporsi yang berkebalikan dengan jumlah kelas yang mengandung *term* (Fauzi et al., 2015). Pada **persamaan 2.3** Merupakan persamaan untuk mendapatkan bobot ICF. Dimana *Nc* adalah jumlah seluruh kelas, *cf(t)* jumlah kelas yang mengandung *term t*.

$$ICF(t) = 1 + \log\left(\frac{Nc}{cf(t)}\right) \quad (2.3)$$

2.2.4 Co-occurence

Teknik *co-occurrence* merupakan salah satu pendekatan secara statistik dalam menentukan kedekatan suatu term. Teknik *co-occurrence* termasuk dalam perhitungan asimetris. Yang dimaksud dengan perhitungan assimetris adalah dimana terdapat dua buah *term* A dan B yang memiliki nilai kemiripan yang berbeda antara A dan B serta B dan A (Y. Tseng, 2002). Teknik co-occurrence

memliki dua tahapan yaitu tahapan pembobotan dan tahapan perhitungan kemiripan term.

Tahapan awal teknik *co-occurrence* adalah menghitung frekuensi *term* pada setiap dokumen. Dilanjutkan dengan menghitung *inverse document frequency* atau IDF dari setiap *term*. Hasil kedua perhitungan tersebut akan menjadi bobot pada *term* tersebut seperti pada **persamaan 2.4** (Khafajeh et al., 2013).

$$d_{ij} = tf_{ij} \times \log \frac{N}{df_j} \quad (2.4)$$

Dimana d_{ij} merupakan bobot *term j* terhadap dokumen *i.* tf_{ij} merupakan frekuensi *term j* pada dokumen *i.* df_j merupakan jumlah dokumen yang mengandung *term j* dan notasi *N* merupakan jumlah dari keseluruhan dokumen yang ada (Khafajeh et al., 2013).

Setiap *term* akan dikombinasikan berpasangan untuk melihat kemiripannya. Setiap pasangan *term* akan dihitung frekuensi kemunculan bersamanya dilihat dari frekuensi terkecil dari kedua *term* tersebut. Kemudian bobot pasangan kedua *term* tersebut dihitung dengan mengalikan *term* frekuensi pasangan *term* dan IDF dari pasangan *term* tersebut seperti pada **persamaan 2.5** (Khafajeh et al., 2013).

$$d_{ijk} = tf_{ijk} \times \log \frac{N}{df_{jk}} \quad (2.5)$$

Dimana d_{ijk} merupakan bobot *term j* dan *term k* terhadap dokumen *i.* tf_{ijk} merupakan frekuensi terkecil dari *term j* dan *term k* pada dokumen *i.* df_j merupakan jumlah dokumen yang mengandung *term j* dan *term k* dan notasi *N* merupakan jumlah dari keseluruhan dokumen yang ada (Khafajeh et al., 2013).

Untuk menentukan nilai kemiripan dari pasangan antar *term*, dilakukan perhitungan seperti pada **persamaan 2.6** . Dimana perhitungan kemiripan term diistilahkan sebagai *cluster weight* yang merupakan nilai kemiripan antara *term j* dan *term k* . d_{ijk} yaitu merupakan bobot *term j* dan *term k* pada dokumen *i.* d_{ij} yaitu merupakan bobot *term j* pada dokumen *i.* Sedangkan *weighting factor* dari **persamaan 2.6** dapat dilihat pada **persamaan 2.8** dimana *N* yaitu jumlah dokumen keseluruhan serta df_k merupakan jumlah dokumen yang mengandung *term k.* **Persamaan 2.7** dan **persamaan 2.9** merupakan sisi dari asimetris *term k* dan t*erm j*. (Khafajeh et al., 2013).

$$ClusterWeight(t_j, t_k) = \frac{\sum_{i=1}^{n} d_{ijk}}{\sum_{i=1}^{n} d_{ij}} \times WeightingFactor(t_k) \quad (2.6)$$

$$ClusterWeight(t_k, t_j) = \frac{\sum_{i=1}^{n} d_{ijk}}{\sum_{i=1}^{n} d_{ik}} \times WeightingFactor(t_j) \quad (2.7)$$

$$WeightingFactor(t_k) = \frac{\log \frac{N}{df_k}}{\log N} \quad (2.8)$$

$$WeightingFactor(t_k) = \frac{\log \frac{N}{df_j}}{\log N} \quad (2.9)$$

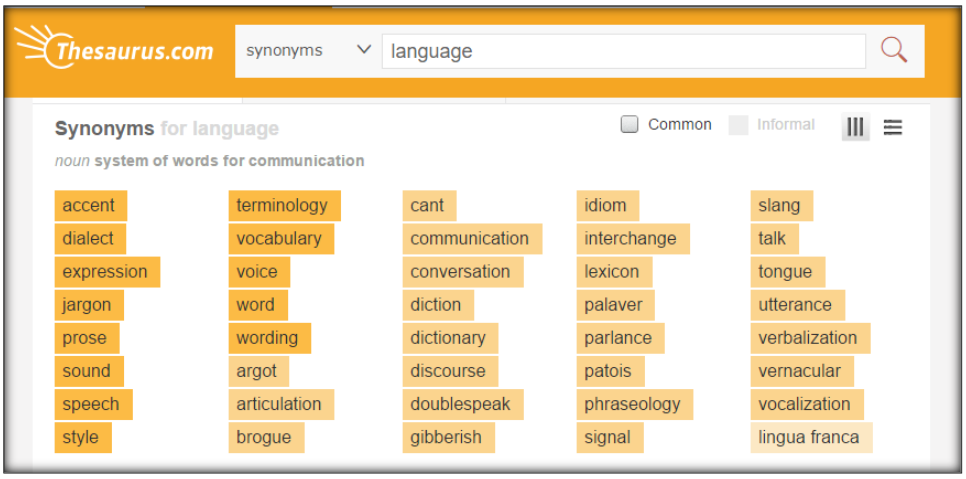
Dari hasil perhitungan *cluster weight* tersebut akan di *filter* untuk mendapatkan pasangan *term* yang dianggap mirip berdasarkan **persamaan 2.10** dimana *x* merpakan nilai *cluster weight* (Y. Tseng, 2002).

$$0.8 < x \leq 1 \quad (2.10)$$

2.2.5 Thesaurus

Thesaurus merupakan sebuah kamus yang sangat berguna dalam bidang temu kembali informasi. Thesaurus berisi mengenai daftar keterkaitan *term* satu dengan *term* lainnya. Keterkaitan yang dibentuk dapat diartikan sebagai kata yang relevan atau berkonteks sama. Kamus thesaurus tersebut dapat digunakan untuk mengembangkan *query* dari pengguna untuk mendapatkan dokumen yang relevan (Y. Tseng, 2002). Contoh dari thesaurus dapat dilihat pada **Gambar 2.1** dimana pada gambar tersebut merupakan contoh thesaurus Bahasa Inggris yang diambil dari www.thesaurus.com.

Thesaurus sendiri dibagi menjadi beberapa jenis, antara lain adalah global thesaurus. Global thesaurus dibangun dengan berbasis pada kemunculan bersama kata serta hubungannya pada *corpus*. Global thesaurus memiliki fokus pada sisi *corpus* atau dokumen tanpa memperhitungkan *query* permintaan pengguna sehingga menghasilkan solusi yang parsial dalam menghadapai permasalahan ketidakcocokan kata (Imran & Sharan, 2009).



Gambar 2.1 Contoh Thesaurus

Selain itu thesaurus dibagi menjadi dua berdasarkan pembentukannya yaitu thesaurus otomatis dan thesaurus manual. Yang dimaksud dengan thesaurus manual adalah, thesaurus yang dibangun secara general dimana tidak terdapat topik khusus. Thesaurus manual menggambarkan keterhubungan atau sinonim suatu kata. *Query expansion* pada thesaurus manual memanfaatkan *wordnet* yang menghubungkan antar katanya secara manual berdasarkan hubungan leksikal (Imran & Sharan, 2009). Thesaurus manual memiliki kelemahan dalam pembentukan yang membutuhkan biaya yang cukup tinggi serta waktu yang cukup lama (Otair et al., 2013).

Thusaurus otomatis dibangun dengan memperhatikan informasi suatu kata muncul bersama, informasi linguistik serta informasi keterhubungan. Thesaurus otomatis dapat menekan biaya serta kebutuhan sumber daya dalam pembangunan thesaurus. *Query expansion* dengan thesaurus otomatis adalah dengan melihat nilai similaritas antar *query* dengan kata-kata yang ada (Imran & Sharan, 2009).

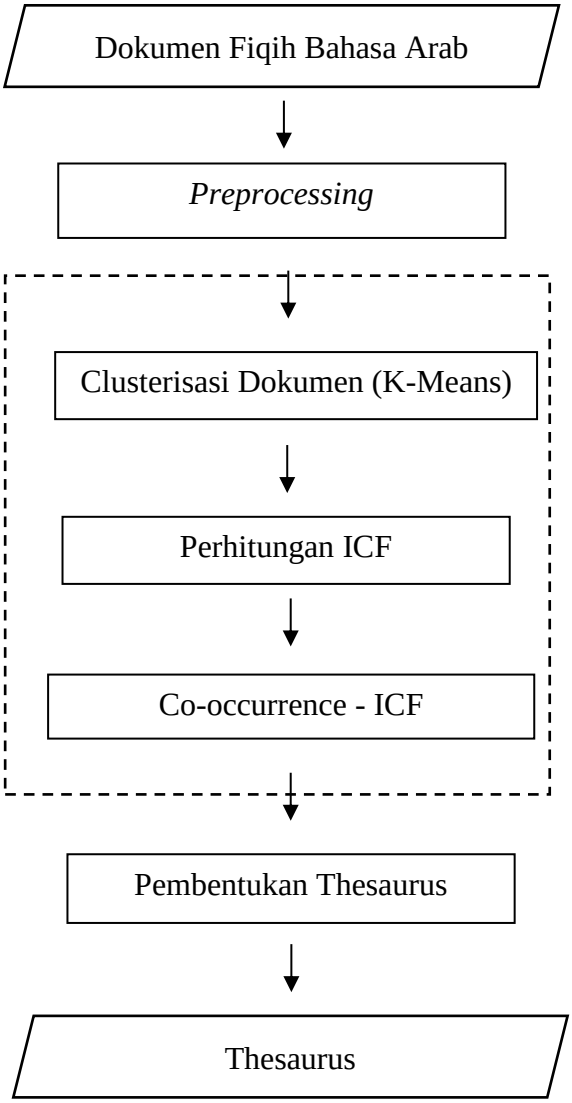
Dalam pembentukan thesaurus secara statistikal terdapat konsep t*erm similarity* atau kemiripan *term* yaitu merupakan sebuah konsep untuk menyatakan adanya hubungan semantik antar kedua *term* tersebut. Dikatakan kedua *term* memiliki hubungan semantik apabila kedua *term* tersebut memiliki arti yang hampir sama atau memiliki konsep atau objek yang merepresentasikan hal yang sama. Sebagai contoh “Google” dan “Microsoft” memiliki hubungan semantik karena merepresentasikan perusahaan software (Li et al., 2013).

Untuk melakukan pengukuran kemiripan term terdapat beberapa pendekatan. Pendekatan yang dilakukan dibagai menjadi dua yaitu pendekatan simetris dan pendekatan asimetris. Pendekatan secara simetris merupakan pendekatan dengan konsep bahwa jika *term* A memiliki kemiripan dengan *term* B, maka *term* B juga memiliki kemiripan dengan *term* A. Beberepa metode yang menerapkan pendekatan simetris antara lain Cosine Cimilarity, Jaccard's dan Dice (Khafajeh et al., 2013).

Pendekatan asimetris merupakan pendekatan dengan konsep bahwa jika *term* A memiliki kemiripan dengan *term* B, belum tentu *term* B juga memiliki kemiripan dengan *term* A. Konsep pendekatan asimetris ini digunakan oleh metode teknik *co-occurrence* untuk menemukan nilai kemiripan antar *term* (Y. Tseng, 2002).

BAB 3
METODA PENELITIAN

Pada bab ini akan dijelaskan proses-proses atau tahapan yang dilalui untuk membentuk sistem yang sesuai dengan metode yang telah diusulkan. Secara garis besar sistem yang dibangun memiliki gambaran tahapan-tahapan yang dapat dilihat pada **Gambar 3.1**



Gambar 3.1 Tahapan Proses Sistem.

3.1 Data

Dokumen yang digunakan merupakan dokumen-dokemen fiqih berbahasa arab yang diambil dari E-Book pada Maktabah Syamilah. Dokumen yang digunakan sebanyak 1000 dokumen. Satu halaman diasumsikan sebagai satu dokumen. Pada **Gambar 3.2** merupakan gambaran dokumen fiqih berbahasa arab.

[كتاب الصلاة]

ثم إنه بدأ بكتاب الصلاة؛ لأن الصلاة من أقوى الأركان بعد الإيمان بالله - تعالى - قال الله تعالى: {فإن تابوا وأقاموا الصلاة} [التوبة: 5] وقال - عليه الصلاة والسلام -: «الصلاة عماد الدين» فمن أراد نصب خيمة بدأ بنصب العماد، والصلاة من أعلى معالم الدين ما خلت عنها شريعة المرسلين - صلوات الله وسلامه عليهم أجمعين -

وقد سمعت شيخنا الإمام الأستاذ شمس الأئمة الحلواني - رحمه الله تعالى - يقول في تأويل قوله تعالى: {وأقم الصلاة لذكري} [طه: 14] أي لأني ذكرتها في كل كتاب منزل على لسان كل نبي مرسل وفي قوله - عز وجل -: {ما سلككم في سقر} [المدثر: 42] {قالوا لم نك من المصلين} [المدثر: 43] ما يدل

Gambar 3.2 Contoh Dokumen Fiqih Bahasa Arab.

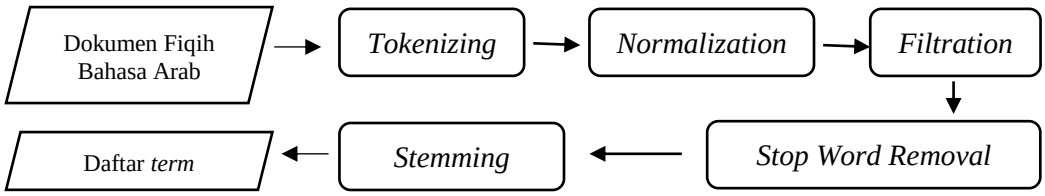
Dokumen yang digunakan memiliki label yang diambil dari topik-topik bahasan Fiqih dari dokumen tersebut. Daftar label topik dari dokumen-dokumen dapat dilihat pada **Tabel 3.1**.

Tabel 3.1 Label Topik Dokumen Fiqih Bahasa Arab

No	Label Topik	Jumlah Dokumen
1	Haji	200
2	Nikah	200
3	Sholat	200
4	Puasa	200
5	Zakat	200

3.2 *Preprocessing* Dokumen

Pada tahapan ini dilakukan *preprocessing* dari dokumen untuk mendapatkan *term-term* yang siap diolah pada proses berikutnya. Dokumen-dokumen fiqih berbahasa arab yang telah dikumpulkan akan di proses dengan menggunakan *library Lucene* dan bantuan *stemmer* Khoja pada Bahasa pemrograman Java. Pada proses ini akan dilakukan pemisahan rangkaian kata berdasarkan delimiter atau pemisah kata seperti karakter spasi. Proses pemisahan tersebut sering disebut dengan *tokenizing.* Kemudian proses dilanjutkan dengan *normalization* dan *filtration* yaitu menghilangkan harokat serta simbol-simbol yang tidak penting. Kemudian proses berlanjut dengan *stopword removal* atau menghapus kata-kata yang dianggap tidak penting. Untuk mendapatkan bentuk kata dasar maka kata-kata atau yang disebut dengan term dilakukan *stemming.* Untuk memperjelas bagan dari *preprocessing* data dapat dilihat pada **Gambar 3.3.**



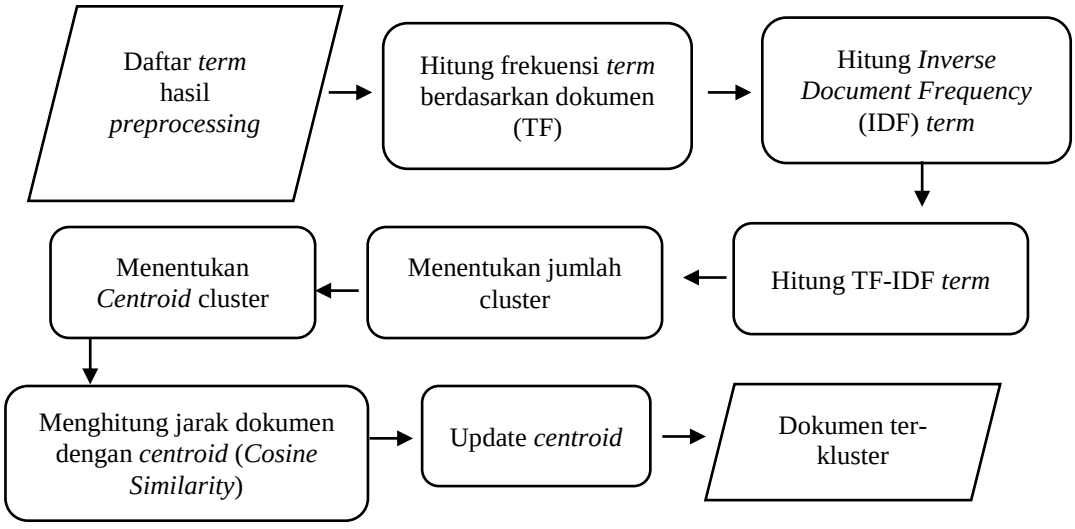
Gambar 3.3 Tahapan *Preprocessing* Data

Sebagai contoh sebuah kalimat teks bahsa Arab ثم إنه بدأ بكتاب الصلاة mengalami proses *tokenizing*, *normalization* dan *filtration* untuk memisahkan rangkaian kata dan membuang tanda baca serta harokat maka akan terbentuk *term-term* yaitu الصلاة - بكتاب - بدأ - إنه - ثم. . *Term-term* yang telah terbentuk akan mengalami proses *stopword removal* dan *stemming* yaitu menghilangkan kata-kata yang dianggap tidak penting serta membentuk ke kata dasar sehingga menjadi ان - بدأ - كتب – صلي . Pada **Gambar 3.4** merupakan potongan daftar stopword dari bahasa arab.

id	word	stemmedword
2	بيد	بيد
3	وبيد	و-بيد
4	فبيد	ف-بيد
5	سوى	سوى
6	وسوى	و-سوى
7	فسوى	ف-سوى
8	غير	غير
9	بغير	ب-غير
10	كغير	ك-غير

Gambar 3.4 Contoh daftar stopwords

3.3 *Clustering* Dokumen K-Means



Gambar 3.5 Tahapan *Clustering* K-Means

Tahapan clustering digunakan untuk mengelompokkan dokumen-dokumen berdasarkan kedekatannya. Tahapan ini digunakan untuk membantu mendapatkan nilai *inverse class frequency* pada tahapan berikutnya. Sebagai gambaran dari tahapan *clustering* dokumen dapat dilihat pada **Gambar 3.5** Dari dokumen-

dokumen fiqih yang telah melalui proses *preprocessing* akan menjadi daftar *term*. *Term-term* tersebut kemudian dihitung frekuensinya terhadap dokumen-dokumen. **Gambar 3.6** merupakan ilustrasi *term* frekuensi terhadap dokumen.

	Dok 1	Dok 2	Dok 3	…	Dok n
Term 1	tf_{11}	tf_{12}	tf_{13}	…	tf_{1n}
Term 2	tf_{21}	tf_{22}	tf_{23}	…	tf_{2n}
Term 3	tf_{31}	tf_{32}	tf_{33}	…	tf_{3n}
⋮	⋮	⋮	⋮	…	⋮
Term i	tf_{i1}	tf_{i2}	tf_{i3}	…	tf_{in}

Gambar 3.6 Ilustrasi *Term Frequency*

Kemudian setiap *term* dihitung nilai *Inverse Document frequency* (IDF) seperti pada **persamaan 3.1** dimana nilai IDF *term j* dipengaruhi oleh *N* jumlah dokumen yang ada dibagi dengan df_j jumlah dokumen yang mengandung *term j*.

$$IDF_j = 1 + \log \frac{N}{df_j} \tag{3.1}$$

Pada Gambar 3.7 merupakan ilustrasi term terhadap hasil perhitungan IDF.

	IDF
Term 1	IDF_1
Term 2	IDF_2
Term 3	IDF_3
⋮	⋮
Term i	IDFi

Gambar 3.7 Ilustrasi IDF *Term*

Kemudian setiap *term j* pada dokumen-dokumen *i* akan dihitung bobotnya w_{ji} dengan perhitungan pada **persamaan 3.2** dimana tf_{ji} frekuansi *term j* pada dokumen *i* dikali dengan IDF *term j*. Pada **Gambar 3.8** merupakan ilustrasi bobot TF-IDF *term* terhadap dokumen.

$$w_{ji} = tf_{ji} \times IDF_j \tag{3.2}$$

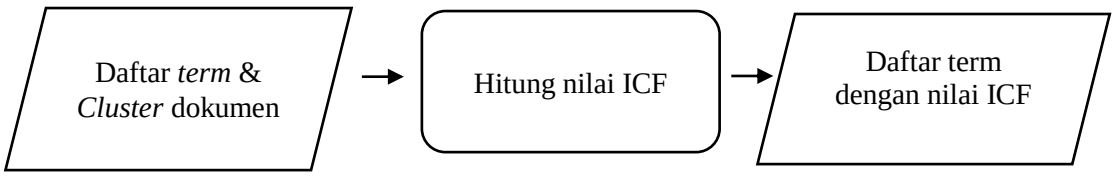
	Dok 1	Dok 2	Dok 3	…	Dok n
Term 1	$tf.idf_{11}$	$tf.idf_{12}$	$tf.idf_{13}$	…	$tf.idf_{1n}$
Term 2	$tf.idf_{21}$	$tf.idf_{22}$	$tf.idf_{23}$	…	$tf.idf_{2n}$
Term 3	$tf.idf_{31}$	$tf.idf_{32}$	$tf.idf_{33}$	…	$tf.idf_{3n}$
⋮	⋮	⋮	⋮	⋯	⋮
Term i	$tf.idf_{i1}$	$tf.idf_{i2}$	$tf.idf_{i3}$	…	$tf.idf_{in}$

Gambar 3.8 Ilustrasi Bobot TF-IDF *Term.*

Setiap dokumen memiliki daftar *term* dengan bobotnya masing-masing. Kemudian menentukan jumlah cluster yang akan dibentuk serta nilai titik *centroid* secara acak dari masing-masing kluster. Setiap dokumen dihitung jaraknya atau similaritasnya terhadap titik *centroid* setiap cluster menggunakan *cosine similarity* seperti pada **persamaan 2.1**.

Hasil dari perhitungan dokumen terhadap semua titik *centroid* cluster akan diurutkan, nilai yang mendekati 1 atau nilai yang terbesar memiliki arti bahwa dokumen tersebut memiliki kesamaan atau kedekatan terhadap cluster tersebut. Sehingga dokumen tersebut akan dijadikan sebagai anggota dari cluster yang memiliki nilai similaritas paling besar. Nilai *centroid* akan terus diupdate dengan **persamaan 2.2**

3.4 *Inverse Class Frequency*



Gambar 3.9 Tahapan Perhitungan ICF

Setelah dokumen telah di kelompokkan berdasar cluster-clusternya maka dilanjutkan dengan menghitung nilai ICF pada term-term yang ada. **Gambar 3.9** merupakan tahapan yang dilakukan dalam melakukan perhitungan ICF. Perhitungan ICF dilakukan sesuai dengan **persamaan 3.3** dimana nilai ICF pada *term j* dipengaruhi dengan jumlah cluster yang ada *C*, dan cf_j jumlah cluster yang mengandung *term j*. **Gambar 3.10** merupakan ilustrasi dari hasil perhitungan ICF.

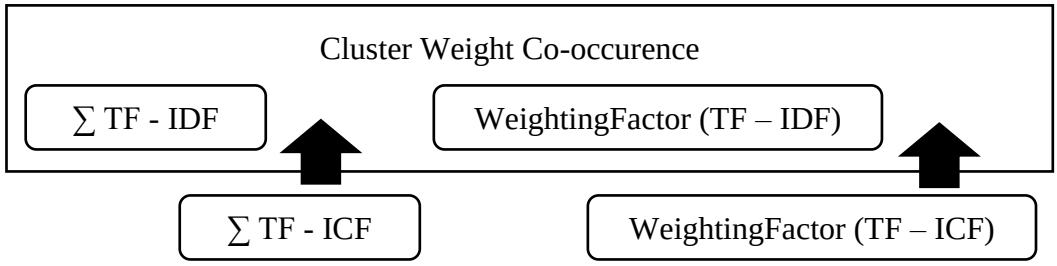
$$ICF_j = 1 + \log \frac{C}{cf_j} \tag{3.3}$$

	ICF
Term 1	ICF_1
Term 2	ICF_2
Term 3	ICF_3
⋮	⋮
Term i	ICFi

Gambar 3.10 Ilustrasi ICF *Term*

3.5 Co-occurence – ICF

Setelah dokumen-dokumen fiqih Bahasa Arab di clusterisasi menggunakan metode K-Means dan setiap term memiliki nilai ICF, tahapan yang dilakukan selanjutnya adalah melakukan tahapan teknik *co-occurrence* pada *term-term* yang ada. Pada penelitian ini diusulkan perhitungan yang juga memperhatikan bobot sebuah *term* tidak hanya pada dokumen saja melainkan juga dengan cluster dari dokumen. Bagan penggabungan teknik *co-occurrence* dan ICF dapat dilihat pada **Gambar 3.11.**



Gambar 3.11 Penggabungan ICF pada *Cluster Weight Co-occurrence*

Tujuan dari perhitungan cluster weight adalah menemukan nilai kemiripan antar kedua term. Perhitungan kemiripan tersebut dilakukan terhadap semua *term* yang ada dengan mengikuti **persamaan 3.4**. **Persamaan 3.4** merupakan perhitungan kemiripan term pada tahapan teknik *co-occurrence* yang digabungkan dengan ICF dimana **persamaan 2.6** hingga **persamaan 2.9** merupakan persamaan yang asli dari perhitungan kemiripan term *clusterweight* pada teknik *co-occurrence*. Diketahui pula bahwa penggabungan teknik *co-occurrence* dan ICF ini merupakan perhitungan statistikal asimetris dimana $ClusterWeight(t_j, t_k) \neq ClusterWeight(t_k, t_j)$.

$$ClusterWeight(t_j, t_k) = \frac{\sum_{i=1}^{n} d_{ijk}}{\sum_{i=1}^{n} d_{ij}} \text{ x } \frac{\sum_{\text{h=1}}^{m} C_{hjk}}{\sum_{\text{h=1}}^{m} c_{hj}} \text{ x } \frac{\log \frac{N}{df_k}}{\log N} \text{ x } \frac{\log \frac{M}{Cf_k}}{\log M} \quad (3.4)$$

Keterangan:

d_{ijk} = Bobot *term j* dan *term k* muncul bersama pada dokumen *i.*

d_{ij} = Bobot *term j* pada dokumen *i*.

C_{hjk} = Bobot *term j* dan *term k* muncul bersama pada kelas *h* (TF-ICF *term j* dan *term k.*

C_{hj} = Bobot *term j* pada kelas *h (*TF-ICF *term j).*

Cf_{jk} = Jumlah kelas yang terdapat *term j* dan *term k* muncul bersama.

Cf_j = Jumlah kelas yang terdapat *term j*

M = Jumah kelas keseluruhan

N = Jumlah dokumen keseluruhan

Persamaan 3.4 merupakan persamaan metode usulan. Dimana nilai kemiripan antar *term* juga melihat pada nilai bobot suatu *term* pada cluster dari dokumen. *Term frequency* pada TF - ICF merupakan frekuensi *term j* pada kelas *h.* Hasil dari pembobotan akan dijumlahkan, kemudian akan dikalikan dengan *weighting factor* seperti pada **persamaan 2.7** dan **persamaan 2.8**. *Weighting factor* ini sendiri merupakan probabilitas kemunculan *term k* terhadap *term j*. Sehingga

diperlukan pula *weighting factor* yang mempengaruhi kemunculan term k terhadap term j dilihat dari cluster dokumennya.

Pada **Gambar 3.12** merupakan ilustrasi hasil perhitungan *cluster weight* pasangan term. Hasil dari perhitungan ini memiliki rentang nilai antara 0 hingga 1 dimana semakin besar nilai *cluster weight* maka kedua term tersebut dianggap semakin mirip. Hasil perhitungan *cluster weight* ini digunakan untuk membentuk thesaurus.

	ClusterWeight
Term 1. Term 2	$\propto_{12}$
Term 1. Term 3	$\propto_{13}$
Term 1. Term 4	$\propto_{14}$
⋮	
Term i, Term (i-1)	$\propto_{i(i-1)}$

Gambar 3.12 Ilustrasi *Cluster Weight* Antar Term.

3.6 Thesaurus

Pembentukan thesaurus dilakukan dengan melakukan filterisasi dari hasil perhitungan kemiripan *term* pada tahapan teknik *co-occurrence* dan ICF. Filterisasi dimaksudkan untuk memberi nilai batasan atau nilai *threshold*. Pasangan term yang berada pada persamaan *threshold* merupakan pasangan term yang dianggap memiliki kemiripan secara konteks. Threshold tersebut sesuai dengan **persamaan 3.5** dimana $\propto$ merupakan nilai dari hasil perhitungan kemiripan *term* pada tahapan teknik *co-occurrence* dan ICF. Nilai threshold tersebut akan diubah untuk menemukan nilai yang terbaik untuk mendapatkan kemiripan term yang relevan.

$$0 < \propto \leq 1 \tag{3.5}$$

Pasangan *term* yang sudah di filter akan didaftar untuk mendapatkan sebuah daftar *term* yang saling berkaitan atau relevan. Pada **Gambar 3.13** merupakan contoh daftar term yang saling relevan dilihat dari hasil filterisasi.

Term 1: Term 2. Term 5. Term 7
Term 3: Term 2. Term 6. Term 7
Term 3: Term 1. Term 4. Term 5

Gambar 3.13 Contoh Daftar Term yang Relevan.

3.7 Rancangan Uji Coba

Untuk melihat kebenaran hasil thesaurus yang telah dibentuk dilakukan pengujian terhadap sistem. Pengujian dilakukan dengan membuat mesin pencari dokumen fiqih berbahasa arab dengan bantuan kamus thesaurus. Dokumen-dokumen fiqih yang telah dikumpulkan akan dibentuk menjadi vektor berdasarkan bobot dari *term-term* yang terdapat pada dokumen tersebut. Pembobotan yang digunakan merupakan pembobotan TF-IDF seperti pada **persamaan 2.4.** Kemudian vektor dari dokumen tersebut dihitung similaritasnya dengan vektor dari *query* yang dimasukkan oleh pengguna. Perhitungan similaritas vektor dokumen dan vektor *query* menggunakan metode *cosine similarity* seperti pada **persamaan 2.1.**

Dalam pembentukan vektor *query* , hasil dari thesaurus dimasukkan juga kedalam daftar *term* pada query. Sehingga hasil similaritas *query* dan dokumen merupakan hasil similaritas yang dibantu dengan thesaurus. Dari hasil similaritas tersebut akan ditemukan dokumen-dokumen mana saja yang relevan dan tidak relevan dengan *query*.

Pengujian dilakukan dengan uji coba sebanyak 10 *query*. Setiap *query* akan dibuatkan daftar dokumen-dokumen yang relevan dengan *query* tersebut oleh pakar. Kemudian pengujian *query* dilakukan sebanyak dua kali, percobaan pertama menggunakan sistem pencarian dokumen dengan bantuan thesaurus co-occurence konvensional sedangkan percobaan kedua menggunakan bantuan dari thesaurus yang sudah dibentuk dengan metode usulan.

Untuk menguji coba hasil dari perhitungan kemiripan *term* dengan menggabungkan tahapan teknik *co-occurrence* dan ICF tersebut digunakan

perhitungan *precission, recall* dan *f-measure* seperti pada **persamaan 3.6** , **persamaan 3.7** dan **persamaan 3.8** dengan keterangan seperti pada **Tabel 3.2**.

$$Precision = \frac{tp}{tp+fp} \quad (3.6)$$

$$Recall = \frac{tp}{tp+tn} \quad (3.7)$$

$$F - measure = \frac{2rp}{r+p} \quad (3.8)$$

Dimana *tp* merupakan *true positive, fp* adalah *false postive* dan *tn* adalah *true negatif*. Sedangkan notasi *r* merupakan *recall* dan *p* adalah *precisión*.

Tabel 3.2 Tabel *Recall Precision*

	Dokumen relevan	**Dokumen yang tidak relevan**
Dokumen yang Ditemukan	*True Positive (tp)*	*False Positive (fp)*
Dokumen yang Tidak Ditemukan	*True Negative (tn)*	*False Negative (fn)*

Percobaan diatas akan dilakukan kembali dengan beberapa keadaan sebagai berikut:

1. Menguji hasil pencarian dokumen dengan menggunkan nilai *threshold* pada **persamaan 3.5** untuk menemukan nilai *threshold* paling optimal.
2. Menguji metode dengan dokumen berita Bahasa Indonesia.

- Dokumen berasal dari harian online kompas yang diakses melalui www.kompas.com.
- Jumlah dokumen berita Bahasa Indonesia yang digunakan sebanyak 200 dokumen dengan pembagian 40 dokumen berkategori Ekonomi, 40 dokumen berkategori Teknologi, 40 dokumen berkategori Traveling , 40 dokumen berkategori Olahraga dan 40 dokumen berkategori Hiburan.
- Library yang digunakan untuk melakukan *preprocessing* data dokumen Bahasa Indonesia adalah library dari JSastrawi untuk Bahasa pemrograman Java.
- Hasil thesaurus Bahasa Indonesia akan diujicobakan menggunakan thesaurus yang dibentuk dengan metode usulan.

[Halaman ini sengaja dikosongkan]

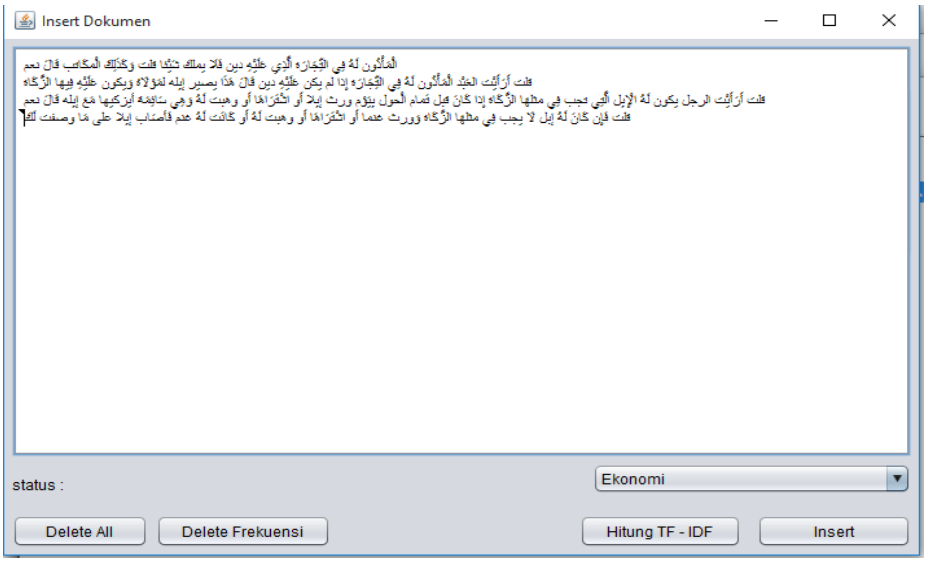
BAB 4
HASIL DAN PEMBAHASAN

Pada bagian ini akan membahas hasil implementasi dari langkah-langkah yang telah dijelaskan pada bab sebelumnya. Kemudian dilanjutkan dengan menampilkan hasil dari uji coba dan pembahasannya. Metode yang diusulkan diimplementasikan dengan menggunakan bahasa pemrograman *Java* pada *platform Java Development Kit (JDK) 1.8.0* dan *IDE Netbeans 8.1*. *Database server* yang digunakan adalah *MySQL*, dengan desain antarmuka *swing Java*, dan *library lucene* sebagai framework *preprocessing*. Aplikasi ini dibangun di atas *platform Microsoft Windows 10.* dengan spesifikasi *processor Core i3* dan *Memory DDR III 8 GB.*

4.1 Hasil Uji Coba

4.1.1 Preprocessing Dokumen

Setiap dokumen yang telah dikumpulkan disimpan kedalam *database* MySQL melalui sebuah program yang dibuat dengan tampilan user interface pada **Gambar 4.1**.



Gambar 4.1 *User Interface* Untuk Memasukkan Dokumen

Database MySQL yang dibangun memiliki tabel dokumen dengan kolom id dokumen, isi dokumen dan topik dokumen. Contoh dari database MySQL untuk menyimpan dokumen dapat dilihat pada **Gambar 4.2**.

id_dokumen ▲ 1	isi_dokumen	topik
6	...ولأنها نسك لا تختص بوقت معين، فلم تكن واجبة بالشرع	Haji
7	...وروي عن ابن عمر، وابن عباس: أنهما كانا يقرآن، (وأق	Haji
8	... وأما الخبر الأول: فغير صحيح؛ لأنه رواية الحجاج بن	Haji
9	...وقال بعض الناس: يجب في كل سنة. وهذا القائل محجوج ب	Haji
10	... فقالهم، أو يدخلها خائفا من ظالم، أو يخاف عريما له	Haji
11	...زاد فهو تطوع» ولأنه داخل إلى مكة لغير نسك، فلم يجب	Haji
12	...تقضى، ألا ترى أن النوافل الراتبة تقضى، وليست بواجب	Haji
13	...في حال كفره؟ فيه وجهان، بناء على أن الكفار مخاطبون	Haji
14	... دليلنا: ما روى ابن عباس - - رَضِيَ اللَّهُ عَنْهُ	Haji
15	...أحدهما: يصح، وهو قول أبي إسحاق؛ لأنه عبادة، فصح إح	Haji
16	...تثبت بمجرد المال، وزعم المحققون منهم أن هذه الرواي	Haji
17	...ويدل عليه: أنه لو كان مستطيعاً للحج بنفسه فطرأ على	Haji
18	وأما المعنى: ... قالوا: الاستطاعة بالزاد والراحلة شرط	Haji
19	...قال: هذا لا يدل على أنه يجب عليه ابتداء كالميت يبق	Haji
20	وأما قصل العبد الذي تعلقوا به: ...،قلنا: وإنما لم يجب	Haji
21	وأما قصل الكفّاره: ...قلا نسلم، ونقول: إذا بذل الابن	Haji
22	...قلنا: وبأن كان يوجد الماء مباحاً في الأصل بخلاف ال	Haji
23	وقد صححوا كونه على الفور. لنا: ...إن مطلق الأمر لا يق	Haji
24	...أما المشروع: فلأنه قد وجد نظير ذلك في الزكوات والك	Haji
25	...يبينه: أن الأمر بالفعل قد تناول أول ستي الإمكان فأ	Haji

Gambar 4.2 Contoh *database* penyimpanan dokumen.

Setiap dokumen yang tersimpan pada *database* terlebih dahulu dilakukan *preprocessing* untuk memudahkan proses selanjutnya. Implementasi *preprocessing* ini terdiri dari beberapa tahapan, diantanya adalah *tokenisasi*, *filtering*, *stemming*, dan *stopword removal*. Tokenisasi dilakukan untuk memecah keseluruhan isi dokumen menjadi suku kata tunggal. Sedangkan pada tahapan *filtering* dilakukan pembuangan harokat-harokat bahasa Arab. Penghapusan *stopword* dilakukan untuk menghilangkan kata-kata yang dianggap tidak penting karena sering muncul dalam

dokumen tetapi tidak mempunyai nilai yang berarti pada sebuah dokumen. Kemudian dilakukan *stemming* untuk memperoleh kata dasar dari masing-masing kata dengan cara mencari kata dasar. Kemudian hasil *preprocessing* tersebut disimpan kedalam database. **Gambar 4.3** merupakan contoh term hasil *preprocessing* yang tersimpan pada *database*. Dari 1000 dokumen yang dimasukkan kedalam *database* didapatkan 3.986 *term* yang telah melalui proses *preprocessing.* Pada proses preprocessing ini diperlukan waktu kurang lebih 3000 detik untuk seluruh dokumen.

id_term ▲ 1	term
1	لأن
2	نسك
3	تختص
4	بوق
5	عون
6	فلم
7	كني
8	أجب

Gambar 4.3 Potongan daftar *term.*

4.1.2 *Clustering* Dokumen

Untuk melakukan *clustering* pada dokumen dilakukan perhitungan *term frequency* terhadap dokumen. Hasil dari perhitungan tersebut disimpan kedalam database. Pada **Gambar 4.4** merupakan hasil perhitungan *term frequency* yang disimpan dalam *database*. *Term* ‘طلق’ yang berarti ‘cerai’ memiliki jumlah frekuensi terbesar. *Term* tersebut banyak ditemukan pada dokumen_id 83 dimana dokumen tersebut merupakan dokumen yang memiliki topik pernikahan.

term	id_dok	TF 1
طلق	83	85
ركع	204	63
طلق	85	62
طلق	86	53
كبر	192	52
صلي	167	47
صلي	191	46
طلق	84	45
قول	504	41
طلق	87	40

Gambar 4.4 Potongan Daftar *Term Frequency*.

Dari hasil perhitungan *term frequency* tersebut maka dilakukan perhitungan IDF setiap *term*. Pada **Gambar 4.5** merupakan hasil perhitungan IDF yang disimpan dalam *database*.

id_term 1	term	idf
1	لأن	1.32
2	نسك	2.34
3	تختص	3.70
4	بوق	2.72
5	عون	1.80
6	فلم	1.73
7	كني	1.39
8	أجب	2.77
9	كطواف	4.00
10	قدم	1.71

Gambar 4.5 Potongan Daftar IDF *Term*

term	id_dok	tf	tf_idf
طلق	83	85	153.77
ركع	204	63	118.33
طلق	85	62	112.16
طلق	86	53	95.88
كبر	192	52	90.47
صلي	167	47	58.43
صلي	191	46	57.19
طلق	84	45	81.41
قول	546	42	43.59
قول	504	41	42.55

Gambar 4.6 Potongan Daftar TF-IDF.

Setelah didapatkan nilai IDF setiap *term* maka dilakukan perkalian dari hasil *term frequency* dan IDF. Kemudian hasil tersebut disimpan dalam *database* untuk digunakan dalam perhitungan *clustering dokumen* dan juga *co-occurence* – ICF. **Gambar 4.6** merupakan hasil perhitungan TF-IDF yang disimpan dalam *database*.

Setiap dokumen memiliki fitur berupa *term* dan nilai fitur berupa TF-IDF. Fitur-fitur tersebut digunakan untuk melakukan *clustering* dokumen. Penentuan nilai centroid dilakukan secara acak dengan menyamakan nilai fitur pada suatu dokumen. Iterasi perhitungan dilakukan sebanyak 1000 iterasi.

Setiap dokumen akan diberi tanda atau *flag* untuk menandai termasuk *cluster* mana dokumen tersebut berada. **Gambar 4.7** merupakan hasil *clustering* dokumen yang tersimpan pada *database.*

id_dok ▼ 1	dokumen	label	cluster_flag
505	فَإِنْ فَقِدَا أَوْ أَحَدُهُمَا فَلَهُ عِبْرَةُ بِا...	Zakat	4
504	أَبْدَلَهُ بِمِصَابِ عَيْنٍ أَوْ مَاشِيَةٍ مِنْ نَ...	Zakat	4
503	وَأَوْلَى بِفَسَادِ بَيْعٍ عَلَى حَوْلِهَا الْأَصْ...	Zakat	4
502	فَيَخْرُجُ الثَّبِيعُ الثَّانِي مِنْهَا لِأَنَّهَا...	Zakat	4
501	يُخْتُ إِبِلٌ خُرَاسَانَ (لِجَرَابِ) بِكَسْرِ الْعِ...	Zakat	4
500	(الْبَقَرِ) يَجِبُ (فِي كُلِّ ثَلَاثِينَ) مِنْهَا ...	Zakat	1

Gambar 4.7 Potongan daftar dokumen dengan hasil *clustering*.

Untuk mendapatkan nilai akurasi, setiap hasil clustering akan disamakan dengan label asli dari dokumen itu sendiri. Kemudian setiap *cluster* akan dihitung prosentase akurasi dengan melihat jumlah dokumen yang cocok terbanyak pada *cluster* tersebut. **Gambar 4.8** merupakan contoh hasil perhitungan akurasi. Dimana indeks label 0 merupakan dokumen dengan label asli "haji" berikutnya secara berurutan "pernikahan", "puasa", "sholat" dan indeks label 4 adalah "zakat".

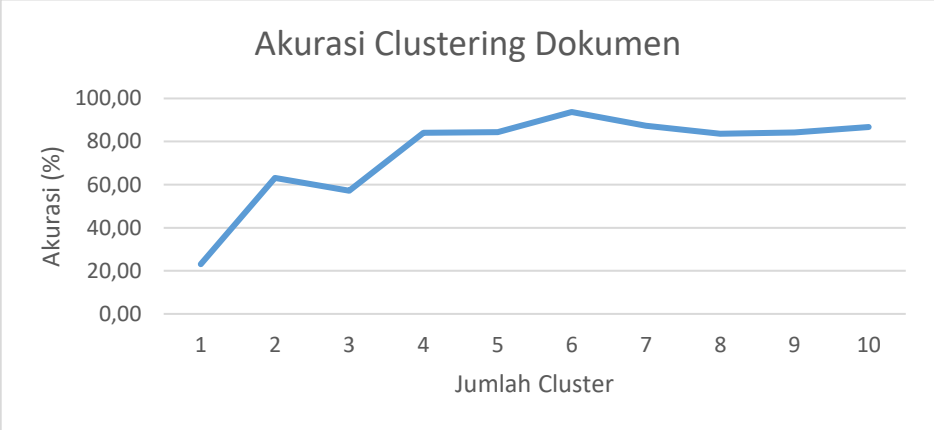
```
 indeks label : 0 -jumlah dok : 180.0
 indeks label : 1 -jumlah dok : 7.0
 indeks label : 2 -jumlah dok : 1.0
 indeks label : 3 -jumlah dok : 22.0
 indeks label : 4 -jumlah dok : 5.0
Cluster ke : 4 Akurasi : 180.0 / 215.0 = 0.8372093023255814

 indeks label : 0 -jumlah dok : 0.0
 indeks label : 1 -jumlah dok : 1.0
 indeks label : 2 -jumlah dok : 0.0
 indeks label : 3 -jumlah dok : 0.0
 indeks label : 4 -jumlah dok : 137.0
Cluster ke : 5 Akurasi : 137.0 / 138.0 = 0.9927536231884058

 indeks label : 0 -jumlah dok : 1.0
 indeks label : 1 -jumlah dok : 1.0
 indeks label : 2 -jumlah dok : 0.0
 indeks label : 3 -jumlah dok : 0.0
 indeks label : 4 -jumlah dok : 80.0
Cluster ke : 6 Akurasi : 80.0 / 82.0 = 0.975609756097561


-------> Akurasi total = 0.9371506436146378
BUILD SUCCESSFUL (total time: 27 minutes 49 seconds)
```

Gambar 4.8 Potongan hasil perhitungan akurasi *clustering*.

Clustering dokumen dilakukan berulang-ulang dengan mengubah jumlah *cluster* yang ditentukan. Hal ini dilakukan untuk mendapatkan jumlah *cluster* yang optimal yang dapat digunakan untuk proses berikutnya. Selain merubah jumlah *cluster* proses *clustering* dilakukan berulang untuk melihat perubahan nilai akurasi dikarenakan penentuan nilai *centroid* awal yang dilakukan secara acak. Penentuan nilai *centroid* awal berpengaruh pada hasil *clustering* dokumen. Dari 10 percobaan yang dilakukan dengan mengubah nilai *centroid* awal didapatkan nilai akurasi terbaik. Pada **Tabel 4.2** merupakan hasil akurasi uji coba *clustering* dokumen. Sedangkan **Gambar 4.9** merupakan bentuk diagram dari hasil *clustering*.

Tabel 4.1 Akurasi *clustering* dokumen.

Jumlah Cluster	Akurasi %
1	23.08
2	63.12
3	57.13
4	84.00
5	84.30
6	93.72
7	87.34
8	83.60
9	84.25
10	86.70



Gambar 4.9 Diagram akurasi *clustering* dokumen.

Nilai akurasi pada *clustering* 5 *cluster* sebesar 84% kemudian meningkat menjadi 93% pada *clustering* 6 *cluster* dan berubah tidak terlalu signifikan pada *clustering* berikutnya. Menggunakan *elbow method* maka *clustering* dengan jumlah 6 *cluster* digunakan untuk proses selanjutnya. Pada tahapan *clustering* ini diperlukan waktu total 57 menit 43 detik dalam pemrosesannya.

4.1.3 Perhitungan TF – ICF

Hasil clustering dokumen digunakan untuk melakukan perhitungan TF - ICF. Perhitungan *term frequency* pada kali ini merupakan jumlah *term j* pada *cluster*

h. **Gambar 4.10** merupakan hasil perhitungan *term frequency* terhadap *cluster* dokumen yang tersimpan pada *database.*

term	id_class	TF 1
صلي	2	2513
قول	2	2489
قول	4	2199
قول	3	1851
قول	1	1818
قول	5	1396
صوم	1	1311
زكي	5	1142
زوج	3	1118
نكح	3	917
حرم	4	904
ولي	3	864

Gambar 4.10 Potongan Daftar TF *term* terhadap *cluster*.

Dari hasil perhitungan *term frequency* tersebut maka dilakukan perhitungan ICF setiap *term*. Pada **Gambar 4.11** merupakan hasil perhitungan ICF yang disimpan dalam *database*.

Setelah didapatkan nilai ICF setiap *term* maka dilakukan perkalian dari hasil *term frequency* dan ICF. Kemudian hasil tersebut disimpan dalam database untuk digunakan dalam perhitungan *co-occurence* – ICF. **Gambar 4.12** merupakan hasil perhitungan TF-ICF yang disimpan dalam *database*. Total waktu untuk melakukan perhitungan TF-ICF adalah 28 menit 14 detik.

id_term ▲ 1	term	icf
1	لأن	1.00
2	نسك	1.18
3	تختص	1.48
4	بوق	1.08
5	عون	1.00
6	فلم	1.00
7	كني	1.00
8	أجب	1.00
9	كطواف	1.78
10	قدم	1.00

Gambar 4.11 Potongan Daftar ICF t*erm*

term	id_class ▲ 1	tf_icf
لأن	1	193.00
صحب	1	55.00
قدم	1	46.00
تبع	1	73.00
عون	1	77.00
عبس	1	38.00
جبر	1	11.00
سيب	1	3.00
قول	1	1818.00
كني	1	82.00

Gambar 4.12 Potongan daftar TF-ICF t*erm*

4.1.4 *Co-occurence* – ICF

Untuk mendapatkan nilai kemiripan antar term diperlukan pengkombinasian *term-term* yang ada. **Gambar 4.13** merupakan hasil kombinasi term-term yang ada yang disimpan dalam *database.* Pengkombinasian seluruh *term*

dilakukan dengan *term-term* yang terdapat pada *query* pengujian sehingga menghasilkan kombinasi *term* sebanyak 40.547 kombinasi *term*.

id_term ▲ 1	term1	term2
1	نوع	عقد
2	نوع	علن
3	نوع	وقل
4	نوع	حزر
5	نوع	حمل
6	نوع	قول
7	نوع	قلل
8	نوع	ثوب

Gambar 4.13 Potongan kombinasi *term*

Dari hasil kombinasi tersebut dihitung pula nilai TF – IDF dan TF - ICF kombinasi *term* tersebut seperti sebelumnya. **Gambar 4.14** merupakan hasil perhitungan TF-IDF kombinasi *term* yang disimpan dalam *database*. Sedangkan **Gambar 4.15** merupakan hasil perhitungan TF-ICF kombinasi *term* yang disimpan dalam *database*.

term1	term2	id_dok	tf_idf
زكي	بدل	503	94.74
زوج	ألف	568	69.29
زكي	بدل	502	65.14
زكي	قضي	677	62.09
صلي	مطر	165	57.68
صلي	عقل	158	56.95
صلي	كبر	192	53.83
صلي	نيأ	191	52.08
وصي	ولي	562	51.06

Gambar 4.14 Potongan TF-IDF kombinasi *term*

term1	term2	id_dok	tf_icf
صلي	قول	2	1674.00
صوم	قول	1	1015.00
زوج	قول	3	863.00
زكي	قول	5	866.58
صلي	ولي	2	750.00
صلي	سلم	2	648.00
زوج	نكح	3	688.52
زوج	ولي	3	608.00
حرم	قول	4	608.00

Gambar 4.15 Potongan TF-ICF kombinasi *term*

Dari semua perhitungan yang telah dilakukan sebelumnya maka dapat dihasilkan hasil kemiripan antar term dengan melakukan perhitungan modifikasi clusterweight dimana perhitungan tersebut merupakan perhitungan gabungan *co-occurence* dengan ICF. **Gambar 4.16** merupakan hasil *co-occurence*-ICF yang disimpan dalam *database*. Total waktu yang dibutuhkan untuk pemrosesan co-occurence-ICF adalah 141 jam 38 menit 17 detik.

id_term	term1	term2	clusterweight
31633	رسم	نوع	0.349
31773	رسم	سبخ	0.325
31716	رسم	قررناه	0.325
31758	رسم	ربغ	0.325
31608	رسم	توبيب	0.325
31583	رسم	تتفق	0.325
31596	رسم	عقف	0.301
31628	رسم	قحم	0.301
31607	رسم	كفأ	0.277
31578	رسم	قرأ	0.271

Gambar 4.16 Potongan hasil perhitungan *co-occurence* - ICF

4.1.5 Thesaurus

Perhitungan *clusterweight* menghasilkan nilai kemiripan antar *term* dimana nilai tersebut digunakan untuk membentuk thesaurus. Contoh hasil kemiripan *term* jika digambarkan dapat dilihat pada **Tabel 4.2.** Lebih lengkap hasil perhitungan dapat dilihat pada **Lampiran 2.**

Tabel 4.2 Contoh Hasil Thesaurus

صلي				
Doa				
سلم	ولي	ملك	سجد	قلل
Salam	Wakil	Raja	Sujud	Menurunkan

Pengujian terhadap hasil pembentukan thesaurus dilakukan dengan menerapkannya terhadap pencarian dokumen. Untuk pengujian ini digunakan *query* yang telah ditentukan seperti pada **Tabel 4.3**. Query yang ditentukan tersebut merupakan kalimat-kalimat yang dibuat sendiri dan disesuaikan dengan topik yang digunakan dalam pembentukan thesaurus sesuai dengan **Tabel 3.1.**

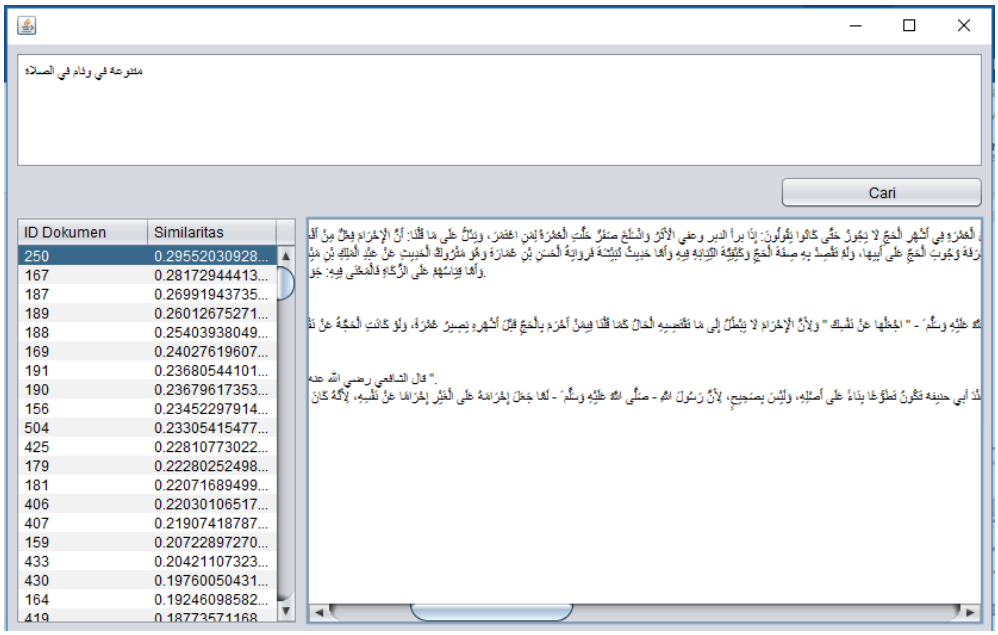
Jumlah dokumen yang digunakan merupakan 238 dokumen fiqih bahasa arab yang diambil dari *E-Book* Maktabah Syamilah. 238 dokumen tersebut telah diberi label *query* oleh pakar untuk mempermudah penghitungan *recall, precision* dan *f-measure*. Pakar memberikan label terhadap setiap dokumen, dimana dokumen tersebut relevan dengan *query* yang mana. Hasil pelabelan oleh pakar dapat dilihat pada **Lampiran 1.**

Query yang telah ditentukan tersebut dimasukkan kedalam sistem seperti pada **Gambar 4.17** untuk mendapatkan hasil perankingan dokumen fiqih bahasa arab. Hasil perankingan dokumen menampilkan dokumen sebanyak 30 dokumen dengan nilai similaritas paling tinggi. Pemilihan jumlah dokumen yang di-*retrive* dilakukan dengan mencoba beberapa jumlah dokumen yang di-*retrieve* terlebih dahulu dan didapatkan angka 30. Angka tersebut didapatkan karena pada angka 10

dan 20 dokumen yang di-*retrieve* beberapa query memiliki nilai *precision* 0 sehingga tidak dapat dilakukan analisa.

Tabel 4.3 Daftar *query* pengujian

ID	QUERY
Q1	متنوعة في وئام في الصلاة " Rukun-rukun dalam sholat"
Q2	كيفية تنفيذ القانون من الزواج " Bagaimana hukum melaksanakan nikah"
Q3	شروط الحج " Syarat-syarat melaksanakan ibadah haji"
Q4	لماذا ينبغي أن أداء الحج " Mengapa harus melaksanakan ibadah haji"
Q5	أمور محرمة في الحج " Hal-hal yang dilarang saat berhaji"
Q6	أي شخص ملزم يصدر الزكاة " Siapa saja yang wajib mengeluarkan zakat"
Q7	المراسيم عشر " Tata cara berzakat"
Q8	كيف يمكن للقانون الصوم " Bagaimana hukum dalam berpuasa"
Q9	ما هي الأشياء التي تفطر الصائم فقط " Hal-hal apa saja yang membatalkan puasa"
Q10	لماذا يجب الصيام " Mengapa harus berpuasa"



Gambar 4.17 Masukan *query* terhadap sistem

Untuk mendapatkan nilai *threshold* yang optimal dilakukan pengujian dengan mengubah nilai *threshold*. Nilai *threshold* dirubah dengan memenuhi syarat seperti pada **persamaan 3.5.** Hasil dari pengujian tersebut dapat dilihat pada **Tabel 4.4** dimana nilai *treshold* yang digunakan secara berturut turut adalah 0.25 ; 0.16 ; dan 0.09. Nilai treshold yang digunakan merupakan barisan geometri dimana melihat hasil perhitungan metode usulan yang memiliki nilai kemiripan antar term paling besar sebesar 0.35. Selain itu metode usulan merupakan perhitungan kemiripan term yang dilihat dari sisi dokumen dan dari sisi *cluster* dimana masing-masing memiliki nilai kemiripan yang berkisar antara 0 hingga 1. Sehingga perkalian keduanya akan menyebabkan nilai dibelakang koma semakin besar, oleh sebab itu dipilih barisan geometri untuk menentukan nilai treshold dengan pola $(0.3)^2$, $(0.4)^2$ dan $(0.5)^2$.

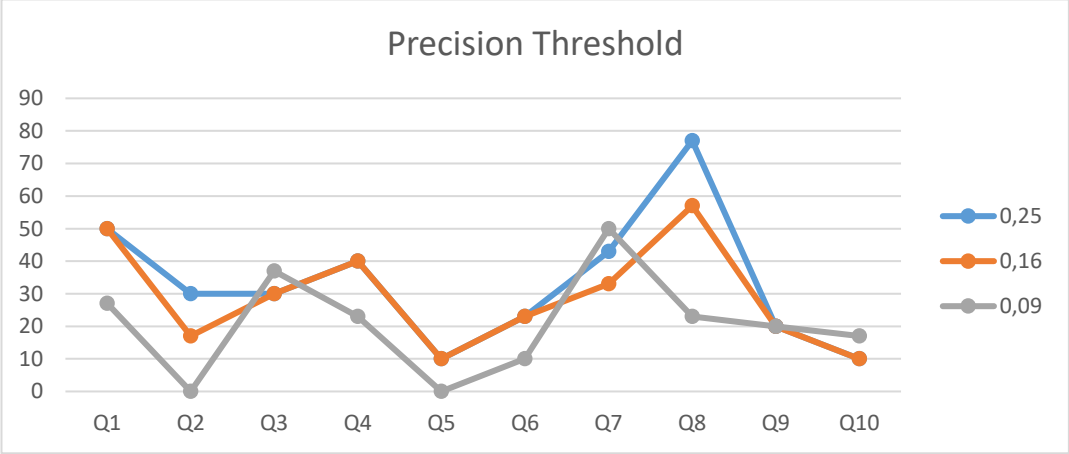
Tabel 4.4 Hasil Pengujian terhadap nilai *treshold*

Query	**α > 0.25**			**α > 0.16**			**α > 0.09**		
	P(%)	**R(%)**	**F(%)**	**P(%)**	**R(%)**	**F(%)**	**P(%)**	**R(%)**	**F(%)**
Q1	50	44	47	50	44	47	27	24	25
Q2	30	82	44	17	45	24	0	0	0
Q3	30	64	41	30	64	41	37	79	50
Q4	40	80	53	40	80	53	23	47	31
Q5	10	43	16	10	43	16	0	0	0
Q6	23	78	36	23	78	36	10	33	15
Q7	43	21	28	33	16	22	50	24	32
Q8	77	42	54	57	31	40	23	13	16
Q9	20	43	27	20	43	27	20	43	27
Q10	10	43	16	10	43	16	17	71	27
Rata-rata	**33.3**	**54**	**36.2**	**29**	**48.7**	**32.2**	**20.7**	**33.4**	**22.3**

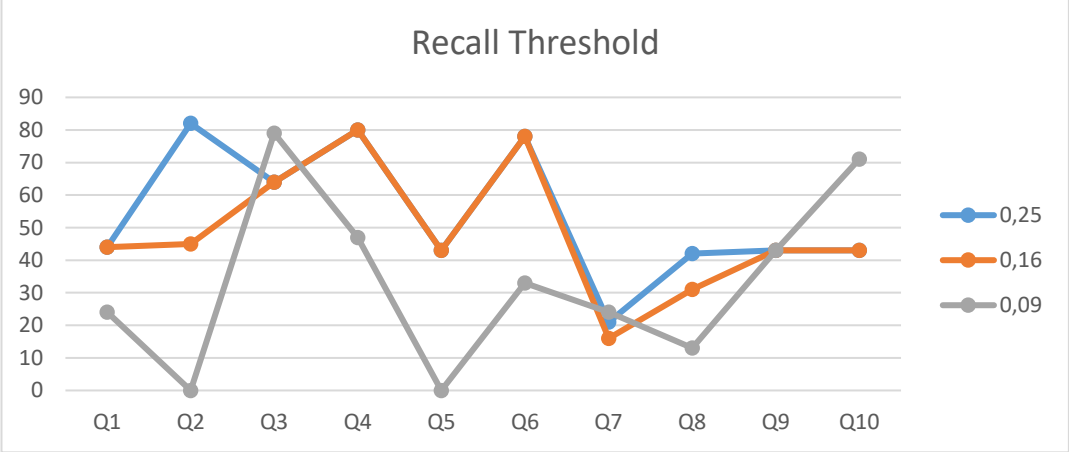
*) P = Precision, R = Recall, F = F-Measure

Dari **Tabel 4.4** dapat dilihat nilai *precision* tertinggi berada pada nilai *threshold* 0.25 dengan nilai sebesar 77% pada query ke-7. Sedangkan nilai *recall* tertinggi juga terdapat pada nilai *threshold* yang sama dengan nilai 82% dan nilai *f-measure* tertinggi sebesar 54%. Untuk memeperjelas dapat dilihat pada grafik

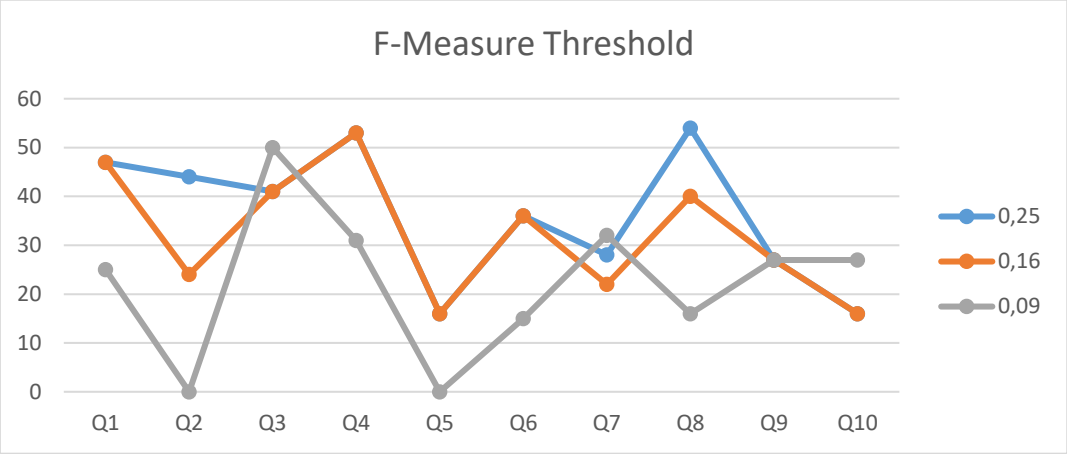
Gambar 4.18 sebagai visualisai dari *precision* **, Gambar 4.19** sebagai visualisasi dari hasil *recall* dan **Gambar 4.20** sebagai visualisasi dari *f-measure*.



Gambar 4.18 Hasil *Precision* Pengujian *Threshold*



Gambar 4.19 Hasil *Recall* Pengujian *Threshold*



Gambar 4.20 Hasil *F-Measure* Pengujian *Threshold*

Pengujian berikutnya dilakukan untuk membandingkan metode usulan dengan metode sebelum dikembangkan. Pada **Tabel 4.5** merupakan hasil dari perbandingan kedua metode. Dimana nilai *precision* tertinggi sebesar 76.7% sedangkan *recall* memiliki nilai terbesar 81.8% dan *f-measure* sebesar 54.1%.

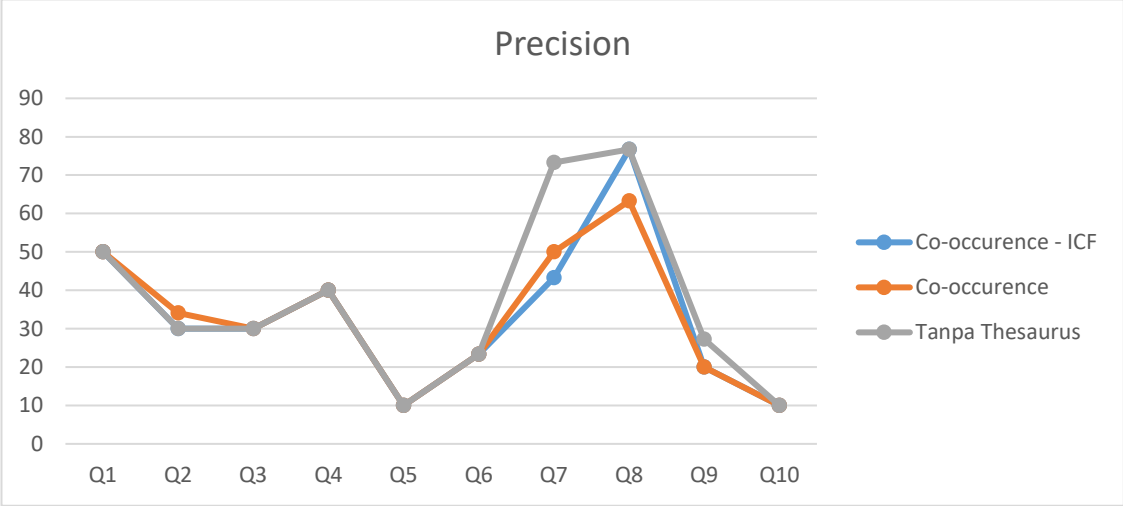
Tabel 4.5 Hasil perbandingan metode usulan

Query	Co-occurence - ICF			Co-occurence			Tanpa Thesaurus		
	P (%)	R (%)	F (%)	P (%)	R (%)	F (%)	P (%)	R (%)	F (%)
Q1	50.0	44.1	46.9	50.0	44.1	46.9	50.0	44.1	46.9
Q2	30.0	**81.8**	43.9	34.1	63.6	34.1	30.0	**81.8**	43.9
Q3	30.0	64.3	40.9	30.0	64.3	40.9	30.0	64.3	40.9
Q4	40.0	80.0	**53.3**	40.0	80.0	53.3	40.0	80.0	**53.3**
Q5	10.0	42.9	16.2	10.0	42.9	16.2	10.0	42.9	16.2
Q6	23.3	77.8	35.9	23.3	77.8	35.9	23.3	77.8	35.9
Q7	43.3	20.6	28.0	50.0	23.8	32.3	73.3	34.9	47.3
Q8	**76.7**	41.8	54.1	63.3	34.5	44.7	**76.7**	41.8	54.1
Q9	20.0	42.9	27.3	20.0	42.9	27.3	27.3	42.9	20.0
Q10	10.0	42.9	16.2	10.0	42.9	16.2	10.0	42.9	16.2
Rata-rata	**33.3**	**53.91**	**36.27**	**33.07**	**51.68**	**34.78**	**37.06**	**55.34**	**32.06**

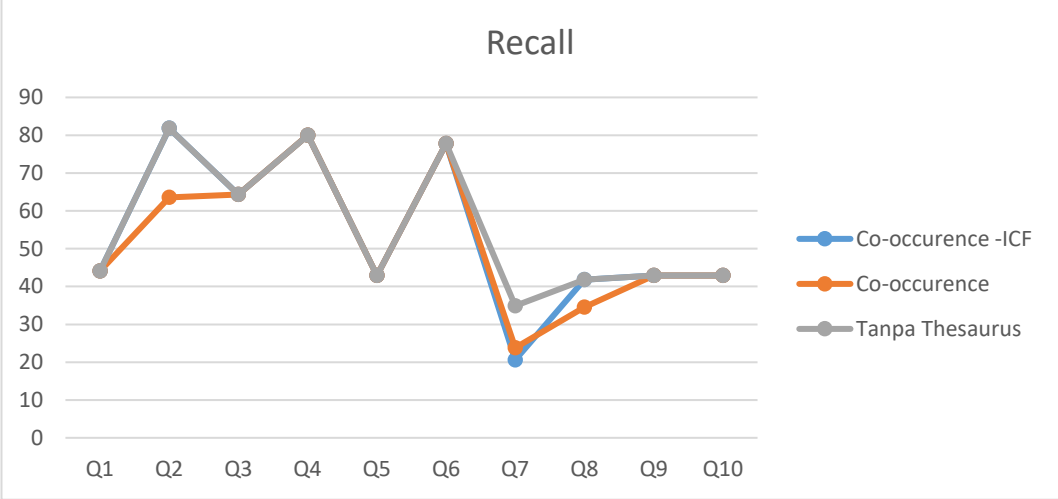
*) P = Precision, R = Recall, F = F-Measure

Sebagai bentuk visualisasi perbandingan **Gambar 4.21** merupakan grafik perbandingan nilai *precision*. Sedangkan **Gambar 4.22** merupakan grafik

perbandingan nilai *recall* dan **Gambar 4.23** merupakan perbandingan nilai *f-measure*.



Gambar 4.21 Hasil Perbandingan Precision



Gambar 4.22 Hasil Perbandingan Recall

Gambar 4.23 Hasil Perbandingan F-Measure

Dilihat dari nilai *precision* query Q8 memiliki nilai tertinggi. **Tabel 4.6** merupakan gambaran dari pemanfaatan thesaurus dalam pencarian dokumen yang dicontohkan pada *query* Q8 yang memiliki nilai *precision* tertinggi. Dari tabel 4.6 dapat dilihat bahwa kemiripan *term* dengan batas *treshold* 0.25 menghasilkan 8 term tambahan pada *query* Q8.

Tabel 4.6 Hasil *query expansion* Q8

***Query* awal**	كيف يمكن للقانون الصوم	
***Term* Hasil *preprocessing query* (A)**	صوم (puasa)	قون (senjata)
***Term* yang memiliki nilai kemiripan dengan term hasil *preprocessing* (B)**		- رحم (Pengasih) - ربع (Keempat) - شيأ (Diagungkan) - قرأ (Baca) - رسل (Rosul) - قلل (Mengurangi) - نهي (Dilarang) - سقط (Menjatuhkan)
***Term* yang digunakan untuk pencarian (A+B)**	قون- صوم - رحم – ربع - شيأ – قرأ - رسل – قلل - نهي - سقط -	

Tabel 4.7 merupakan potongan hasil pengujian pada Q8 dimana ditampilkan 10 dokumen teratas dari 30 dokumen yang di-*retrieve*. Dilihat dari tabel tersebut , dari 10 dokumen teratas hanya 1 dokumen saja yang tidak sesuai dengan kalimat *query* masukan. Selain itu topik dari kesepuluh dokumen memiliki topik yang sama yaitu topik 'puasa'. Hasil yang lebih lengkap dapat dilihat pada **Lampiran 3.**

Tabel 4.7 Potongan Contoh Hasil Pengujian Q8

NO	Potongan Isi Dokumen	Topik	Ground Truth
1	النية][620 [مسألة :لا يصح الصيام إلا بنية .خلافاً لزفر في] قوله :إن صوم رمضان يصح بغير نية، لقوله عليه السلام) :لا	Puasa	Benar
2	كتاب بيان أحكام الصيام وهو والصَوم مصدران، معناهما لغةً الإمساك، وشرعًا إمساك عن مفطر بنية مخصوصة، جميعَ	Puasa	Benar
3	فرْقَ بين ذلك .وهو قول ابن عباسٍ، وابنِ عمر في التَّبْييت في الصيامقال مالكٌ، وأصحابه :لا صيامَ إلاَّ لمَنْ بيَّتَهُ؛ لأنَّ الله	Puasa	Benar
4	فليقل مرتين أو ثلاثا :»إني صائم«، إما بلسانه - كما قال النووي في الأذكار - أو بقلبه - كما نقله الرافعي عن الأئمة.	Puasa	Salah
5	على الاتصال، فإذا أعاده صحت صلاته ولا يمكنه في الصوم إعادة ما فعله بنية النفل على الاتصال لأن زمان الليل لا يقبل	Puasa	Benar
6	بَابُ النِّيَّةِ فِي الصَّوْمِ)قَالَ الشَّافِعِيُّ : (وَلَا يَجُوزُ لِأَحَدٍ ِ صِيَامُ فَرْضٍ مِنْ شَهْرِ رَمَضَانَ وَلَا نَذْرٍ وَلَا كَفَّارَةٍ إِلَّا أَنْ يَنْوِي	Puasa	Benar
7	عبد الرحمن :أنه سمع معاوية رضي الله عنه يوم عاشوراء على المنبر يقول يا أهل المدينة أين عطاؤكم سمعت رسول	Puasa	Benar
8	صام، ومنا من أفطر، فلم يعب الصائم على المفطر، ولا المفطر على الصائم «.إذا ثبت هذا :فإن كان ممن لا يجهده ا	Puasa	Benar
9	سنة ثم صوم كل يوم منه كفارة سنة ثم بعد رجب شهر شعبان قال النبي صلى الله عليه وسلم" :من سره أن يذهب ،" (1)	Puasa	Benar
10	اضغط للبحث عن الكلمة داخل الكتاب تحميل الكتاب الأولى السابقة التالية الأخيرةكتاب الصيام	Puasa	Benar

Untuk melihat posisi hasil dokumen yang relevan ter-*retrieve* maka dihitung rata-rata hasil perankingan dokumen yang sesuai dengan *query* masukan. Tabel 4.8 merupakan hasil rata-rata posisi dokumen relevan yang ter-*retrieve*.

Tabel 4.8 Rata-rata posisi hasil perankingan dokumen

Query	Co-occurence - ICF	Co-occurence	Tanpa Thesaurus
Q1	47.79	47.79	47.79
Q2	34.27	61.18	49.36
Q3	25.28	25.28	25.28
Q4	16.7	16.73	16.73
Q5	87.5	58.14	85.29
Q6	25.11	25.11	25.11
Q7	111.85	95.77	70.73
Q8	55.97	56.4	46.76
Q9	42.21	42.21	42.21
Q10	38.14	38.14	38.14
Rata-rata	**44.33**	**43.12**	**43,12**

Pengujian pada bahasa Indonesia juga dilakukan. Dokumen yang diujikan pada Bahasa Indonesia sebanyak 200 dokumen dengan query seperti pada **Tabel 4.9.** Hasil perankingan dokumen menampilkan dokumen sebanyak 10 dokumen dengan nilai similaritas paling tinggi.

Tabel 4.9 Daftar query pengujian Bahasa Indonesia

ID	Query
QI1	Destinasi wisata yang cocok untuk bertualang di alam
QI2	Prestasi badminton Indonesia
QI3	Bank dengan profit tertinggi di akhir tahun
QI4	Pernikahan selebriti yang gempar dan megah
QI5	Aplikasi mobile yang banyak diunduh oleh pengguna

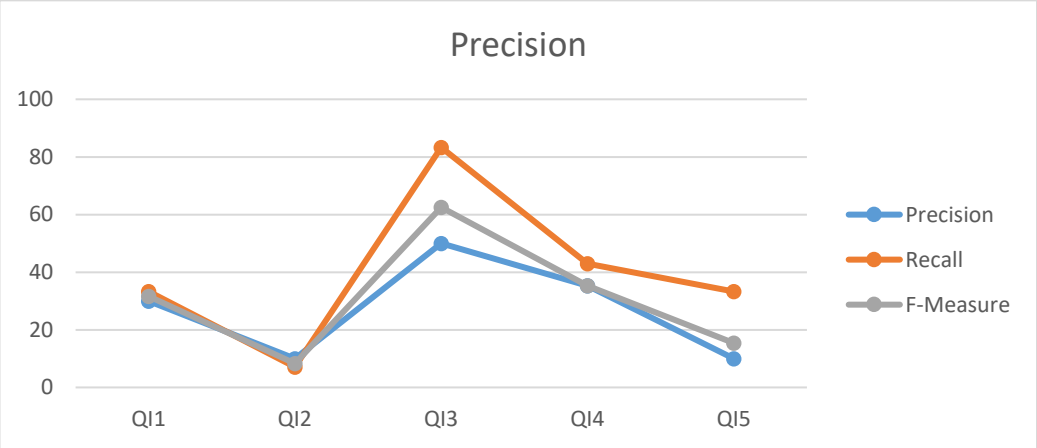
Pengujian dilakukan dengan membandingkan metode usulan dengan metode yang sudah ada ebelumnya. Pada **Tabel 4.10** merupakan hasil dari pengujian bahasa Indonesia. Dimana nilai *precision* tertinggi sebesar 50% sedangkan *recall* memiliki nilai terbesar 83.3% dan *f-measure* sebesar 62.5%.

Tabel 4.10 Hasil Pengujian Bahasa Indonesia

Query	P (%)	R (%)	F (%)
QI1	30.0	33.3	31.6
QI2	10.0	7.1	8.3
QI3	50.0	83.3	62.5
QI4	35.3	42.9	35.3
QI5	10.0	33.3	15.4
Rata-rata	**27.06**	**39.98**	**30.62**

*) P = Precision, R = Recall, F = F-Measure

Sebagai bentuk visualisasi **Gambar 4.24** merupakan grafik dari hasil pengujian bahas Indonesia.



Gambar 4.24 Hasil Pengujian Bahasa Indonesia

4.2 Pembahasan

4.2.1 Preprocessing Dokumen

Hasil dari tahapan preprocessing adalah daftar term-term yang terdapat pada dokumen-dokumen corpus. Dari 1000 dokumen fiqih bahasa arab yang melalui tahapan ini, dihasilkan term sebanyak 3.983 term.

Library lucene dengan bantuan *stemmer* Khoja mampu memecah dokumen menjadi penggalan-penggalan term berdasarkan delimiter spasi. *Library* tersebut

juga berhasil menghilangkan harokat serta simbol-simbol karakter yang tidak penting dan juga angka.

Namun pada tahapan stemming masih terdapat beberapa kata yang belum menjadi kata dasarnya. Seperti term "بسم" pada rangkaian "بسم الله" dimana seharusnya term tersebut berpisah menjadi "ب" dan "اسم" . Kesalahan tersebut akan berpengaruh terhadap arti dari kata tersebut.

4.2.2 *Clustering* Dokumen

Clustering dokumen menggunakan metode K-Means mampu menghasilkan nilai akurasi tertinggi sebesr 93%. Hasil dari *clustering* dokumen dengan jumlah *cluster* dibawah 5 *cluster* memiliki nilai akurasi yang rendah dengan rata-rata 56%. Hal tersebut dikarenakan dokumen-dokumen yang digunakan pada dasarnya memiliki label yang terdiri dari 5 kategori. Sehingga akurasi *clustering* dengan jumlah *cluster* dibawah 5 *cluster* akan memiliki nilai yang rendah.

Dilihat dari **Gambar 4.9** posisi "elbow" terletak pada akurasi dengan 6 *cluster*. Hal tersebut dilihat juga dengan perubahan nilai akurasi pada 7 *cluster*, 8 *cluster* dan seterusnya yang memiliki selisih yang tidak terlalu jauh. Oleh sebab itu *clustering* dengan jumlah 6 *cluster* digunakan untuk *clustering* dokumen pada pembentukan thesaurus ini. Proses perhitungan *clustering* ini tidak terlalu lama hanya membutuhkan waktu selama 27 menit untuk pengambilan data dari *database* dan juga pengujian hingga 10 *cluster.*

4.2.3 Perhitungan TF – ICF

Dalam perhitungan TF-ICF mampu dianalisa nilai keinformatifan suatu term. Persebaran *term* pada *cluster-cluster* yang ada dapat dilihat dari nilai ICF dan TF – ICF nya. Semakin besar nilai ICF suatu *term* maka *term* tersebut memiliki nilai informatif pada *cluster* tertentu. Namun nilai tersebut juga perlu diimbangi dengan jumlah TF yang tinggi pula.

Term dengan nilai TF-ICF tertinggi adalah pasanagan *term* " صلي" dan " قول" dimana yang memiliki arti "berdoa" dan "berkata", pasangan *term* tersebut memiliki nilai TF-ICF sebesar 1674 dimana nilai tersebut sama dengan nilai TF dari pasangan *term* itu. Oleh sebab itu dapat disimpulkan pasangan *term* tersebut

kurang memiliki nilai informatif karena muncul pada semua *cluster* dengan frekuensi yang besar.

Dari hasil perhitungan ini terdapat beberapa *term* yang muncul dibanyak *cluster* dengan nilai yang tinggi pula seperti *term* tunggal " صلي" yang selalu muncul di setiap *cluster* dengan frekuensi yang rata-rata mencapai lebih dari 500 kali kemunculan disetiap *cluster*. Keadaan tersebut tidak berimbang dengan adanya *term-term* yang hanya muncul satu kali di satu *cluster*. *Term* tersebut merupakan *term* yang khas dari *cluste*r tersebut dan memiliki nilai informatif namun nilai TF-ICF dari *term* tersebut masih lebih kecil dibandingkan dengan *term* yang memiliki kemunculan besar. Berbeda dengan *term* yang muncul juga satu kali disetiap *cluster*. *Term-term* tersebut dapat ditafsirkan sebagai *term* yang tidak memiliki nilai informatif dan *term* tersebut menjadi beban saat melakukan proses perhitungan berikutnya. Oleh sebab itu untuk kedepannya, *term* dengan frekuensi yang sedikit dengan batasan yang ditentukan dapat dihilangkan dari daftar *term*.

4.2.4 *Co-occurence* – ICF

Perhitungan metode usulan ini merupakan perhitungan dimana probabilitas kemunculan bersama suatu *term* pada dokumen akan diperkuat dengan probabilitas kemunculan bersama *term* tersebut pada *cluster*. Jika suatu pasangan *term* memiliki nilai *cluster weight* yang tinggi terhadap dokumen (dimisalkan memiliki nilai 1) dan juga memiliki nilai yang tinggi pula terhadap *cluster* (dimisalkan memiliki nilai 1), maka nilai *cluster weight* pasangan *term* tersebut dengan metode usulan tetap memiliki nilai yang tinggi yaitu 1. Hal tersebut dapat diartikan bahwa *term* tersebut memiliki kemiripan karena selalu muncul pada dokumen yang sama di cluster yang sama.

Berbeda dengan nilai pasangan *term* yang selalu muncul di dokumen yang sama namun pada *cluster* yang berbeda. Nilai *cluster weight* pasangan *term* tersebut terhadap dokumen memiliki nilai 1 namun nilai *cluster weight* terhadap *cluster* hanya memiliki nilai 0.5. maka nilai *cluster weight* pasangan tersebut dengan metode usulan menjadi 0.5. Hal ini dapat diartikan bahwa pasangan *term* tersebut memiliki konteks yang berbeda ketika berada pada *cluster* yang berbeda.

Namun perhitungan *cluster weight* dengan metode usulan menyebabkan perbedaan range nilai yang besar atau menambah nilai angka dibelakang koma. Hal tersebut menyebabkan penetuan nilai *threshold* dalam pembentukan thesaurus menjadi kecil dibandingkan metode sebelumnya.

Permasalahan yang timbul pada tahap ini adalah lamanya waktu perhitungan dikarenakan pengkombinasian *term-term* yang ada, Hal ini dapat disiasati dengan menghilangkan *term-term* yang memiliki nilai keinformatifan yang kecil dilihat dari nilai TF-ICF maupun TF-IDF nya.

4.2.5 Thesaurus

Dapat dilihat dari hasil pengujian bahwa metode usulan mampu memperbaiki metode sebelumnya dimana rata-rata *recall* dan *f-measure* meningkat sebanyak 2% meskipun nilai *precision* hanya meningkat 0.3%. Hal ini dapat disimpulkan bahwa melihat nilai keinformatifan sutau *term* terhadap *cluster* juga mempengaruhi nilai kemiripan antar *term*.

Dari hasil pengujian pembentukan thesaurus terhadap nilai *threshold* didapatkan hasil bahwa semakin kecil nilai *threshold* maka semakin kecil pula nilai *precision, recall* dan *f-measure*-nya. Hal ini dikarenakan *term* yang diikutkan dalam *query expansion* merupakan *term-term* yang memiliki nilai kemiripan yang kecil.

Dilihat dari tabel hasil pengujian, nilai-nilai *precision, recall* dan *f-measure* hasil pengujian banyak yang bernilai kecil atau dibawah 50%, hal ini dikarenakan oleh beberapa faktor. Yang pertama adalah metode *query expansion* dimana *term-term* pada thesaurus ditambahkan secara langsung terhadap *term-term query* hal tersebut menyebabkan bobot term *query* asli dengan *term* thesaurus menjadi sama dan mengubah nilai informasi dari *query* tersebut. Faktor kedua adalah perseberan *term* terhadap *cluster* yang ada dimana suatu term memiliki frekuensi yang besar pada setiap *cluster* namun memiliki makna atau konteks yang berbeda mengingat kategori asli pada dokumen yang digunakan merupakan kategori-kategori yang memiliki konteks yang berbeda. Permasalahan tersebut dapat disiasati dengan lebih mengkrucutkan kategori pada dokumen-dokumen *corpus* dan juga menggunakan metode *query expansion* yang tidak mengurangi nilai keinformatifan *query* awal.

Hasil dari pengujian dengan menggunakan thesaurus metode usulan dan tanpa thesaurus memiliki nilai yang tidak jauh berbeda bahkan ada yang sama. Namun dilihat dari urutan dokumen pencarian, kedua pengujian memiliki hasil urutan yang berbeda. Dilihat dari rata-rata posisi hasil pengujian dimana rata-rata posisi pada metode usulan lebih besar dibandingkan dengan rata-rata posisi hasil tanpa thesaurus. Hal tersebut dapat diartikan bahwa posisi dokumen yang relevan pada hasil perankingan dokumen metode usulan berada pada urutan yang rendah dibandingkan dengan uji coba tanpa menggunakan thesaurus.

Dilihat pula pada hasil dokumen yang di-retrieve pada *query* Q8. Dimana dokumen-dokumen tersebut memiliki topik yang sama yaitu topik “puasa”, hal tersebut dapat diartikan bahwa *term-term* yang dihasilkan pada thesaurus berada pada satu topik. Meskipun beberapa *term* juga terdapat pada topik lain, namun jika *term-term* hasil thesaurus digabungkan sesuai *term query* maka akan menunjuk pada topik yang sama. Dari sini dapat disimpulkan bahwa term-term hasil thesaurus berkelompok dan relevan dengan topik-topik yang ada.

[Halaman ini sengaja dikosongkan]

BAB 5
KESIMPULAN

5.1 Kesimpulan

Dari penelitian yang telah dilakukan dapat diambil kesimpulan sebagai berikut:

1. Metode pengukuran kemiripan *term* dengan menggabungkan teknik *co-occurrence* dan *inverse class frequency* pada pembentukan thesaurus Bahasa Arab berhasil meningkatkan relevansi antar term dibuktikan dengan nilai *precision* tertinggi sebesar 76.7% sedangkan *recall* memiliki nilai terbesar 81.8% dan *f-measure* sebesar 54.1%. Nilai tersebut lebih baik dibandingkan dengan metode teknik *co-occurence* sebelumnya.
2. Nilai *threshold* perhitungan kemiripan *term* yang terbaik untuk pembentukan thesaurus sebesar 0.25. Nilai tersebut mampu menyaring *term-term* yang memiliki nilai kemiripan yang sesuai dengan konteks.
3. *Inverse class frequency* dapat mengetahui nilai keinformatifan suatu *term* terhadap *cluster* sehingga dapat digunakan untuk melihat kemiripan antar *term* yang tidak hanya berdasar dari dokumen saja.
4. Metode yang diusulkan juga dapat digunakan pada Bahasa Indonesia dilihat dari nilai *precision* tertinggi sebesar 50.0% sedangkan *recall* memiliki nilai terbesar 83.3% dan *f-measure* sebesar 62.5%.

5.2 Saran

Dari hasil pengujian dan analisa dapat disimpulkan saran untuk pengembangan berikutnya sebagai berikut:

1. Dokumen corpus yang digunakan lebih terkonsentrasi terhadap satu topik saja dikarenakan perbedaan makna sebuah kata antar topik yang tidak sama.
2. Dalam uji coba menggunakan metode *query expansion* yang tidak mengurangi nilai keinformatifan *query* awal.
3. Menghilangkan *term-term* yang memiliki nilai keinformatifan yang kecil dilihat dari nilai TF-ICF maupun TF-IDF nya.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

Chen, H., Yim, T., Fye, D., & Schatz, B. (1995). Automatic thesaurus generation for an electronic community system. *Journal of the American Society for Information Science*, *46*(3), 175–193. https://doi.org/10.1002/(SICI)1097-4571(199504)46:3<175::AID-ASI3>3.0.CO;2-U

Chen, Z., Liu, S., Wenyin, L., Pu, G., & Ma, W. (2003). Building a Web Thesaurus from Web Link Structure. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval* (pp. 0–7).

Crouch, C. J., & Yang, B. (1992). Experiments in Automatic Statistical Thesaurus Construction. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval. ACM* (pp. 77–88).

D. Manning, C., Ragavan, P., & Schutze, H. (2009). *Introduction to Information Retrieval*. *Cambridge University Press*. https://doi.org/10.1109/LPT.2009.2020494

Fauzi, M. A., Arifin, A. Z., Kom, S., Kom, M., Yuniarti, A., Kom, S., & Sc, M. C. (2015). Term Weighting Berbasis Indeks Buku Dan Kelas Untuk Perangkingan Dokumen Berbahasa Arab. *Lontar Komputer*, *5*(2), 110–117.

Gao, R., Li, D., Li, W., & Dong, Y. (2012). Application of Full Text Search Engine Based on Lucene, *2*(October), 106–109. https://doi.org/10.4236/ait.2012.24013

Gijntzer, U., Juttner, G., Seegmuller, & Sarre, F. (1989). Automatic thesaurus construction by machine learning from retrieval sessions. *Information Processing & Management*, *25*(3), 265–273.

Gupta, H., & Srivastava, R. (2014). k-means Based Document Clustering with Automatic " k " Selection and Cluster Refinement. *International Journal of Computer Science and Mobile Applications*, *2*(5), 7–13.

Holle, K. F. H., Arifin, A. Z., & Purwitasari, D. (2015). Preference Based Term Weighting For Arabic Fiqh Document Ranking. *Jurnal Ilmu Komputer Dan Informasi*, *8*(1), 45–52.

Imran, H., & Sharan, A. (2009). Thesaurus and Query Expansion. *Journal of Computer Science and Information Technology*, *1*(2), 89–97.

Ito, M., Nakayama, K., Hara, T., & Nishio, S. (2008). Association thesaurus construction methods based on link co-occurrence analysis for wikipedia. *Proceeding of the 17th ACM Conference on Information and Knowledge Mining - CIKM '08*. 817. https://doi.org/10.1145/1458082.1458191

Kaur, M., & Kaur, N. (2013). Web Document Clustering Approaches Using K-Means Algorithm. *International Journal of Advanced Research in Computer Science and Software Engineering*, *3*(5), 861–864. Retrieved from http://www.ijarcsse.com/docs/papers/Volume_3/5_May2013/V3I5-0380.pdf

Khafajeh, H., Refai, M., & Yousef, N. (2013). Building Arabic Automatic Thesaurus Using Co-occurence Technique. In *Proceedings of International Conference on Communication, Media, Technology and Design* (pp. 28–32).

Li, P., Wang, H., Zhu, K. Q., Wang, Z., & Wu, X. (2013). Computing term similarity by large probabilistic isA knowledge. *Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management - CIKM '13*. 1401–1410. https://doi.org/10.1145/2505515.2505567

Mahdavi, M., & Abolhassani, H. (2009). Harmony K -means algorithm for document clustering, (November 2008), 370–391. https://doi.org/10.1007/s10618-008-0123-0

Niwattanakul, S., Singthongchai, J., Naenudorn, E., & Wanapu, S. (2013). Using of Jaccard Coefficient for Keywords Similarity. In *International MultiConference of Engineers and Computer Scientists* (Vol. I, pp. 380–384). https://doi.org/ISBN 978-988-19251-8-3

Otair, M., Ph, D., Amman, J., Kanaan, R., & Ph, D. (2013). Optimizing an Arabic Query using Comprehensive Query Expansion Techniques. *International Journal of Computer Applications*, *71*(17), 42–49.

Sawalha, M., & Atwell, E. (2008). Comparative Evaluation of Arabic Language

Morphological Analysers and Stemmers. In *International Conference on Comptational Linguistics (Poster* (pp. 107–110). https://doi.org/10.1016/j.aap.2006.08.007

Steinbach, M., Karypis, G., & Kumar, V. (2000). A Comparison of Document Clustering Techniques. *KDD Workshop on Text Mining*, *400*(X), 1–2. https://doi.org/10.1109/ICCCYB.2008.4721382

Tseng, Y. (2002). Automatic Thesaurus Generation for Chinese Documents. *Journal of the American Society for Information Science and Technology*, *53*(September), 1130–1138. https://doi.org/10.1002/asi.10146

Tseng, Y. H. (2002). Automatic thesaurus generation for Chinese documents. *Journal of the American Society for Information Science and Technology*, *53*(13), 1130–1138. https://doi.org/10.1002/asi.10146

Turney, P. D. (2001). Mining the Web for synonyms: PMI-IR versus LSA on TOEFL. *Proceedings of the 12th European Conference on Machine Learning (ECML-2001), Freiburg, Germany*, 491–502. https://doi.org/10.1007/3-540-44795-4_42

Wahib, A., Santika, P. P., & Arifin, A. Z. (2015). Perangkingan Dokumen Berbahasa Arab Menggunakan Latent Semantic Indexing. *Buana Informatika*, *6*(2), 83–92.

Wardhana, S. R., Yunianto, D. R., Arifin, A. Z., & Purwitasari, D. (2015). PEMBOBOTAN KATA BERBASIS PREFERENSI DAN HUBUNGAN SEMANTIK PADA DOKUMEN FIQIH BERBAHASA ARAB. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *2*(2), 132–137.

Xu, H., & Yu, B. (2010). Expert Systems with Applications Automatic thesaurus construction for spam filtering using revised back propagation neural network. *Expert Systems With Applications*, *37*(1), 18–23. https://doi.org/10.1016/j.eswa.2009.02.059

Zohar, H., Liebeskind, C., Schler, J., & Dagan, I. D. O. (2013). Automatic Thesaurus Construction for Cross Generation Corpus. *Journal on Computing and Cultural Heritage (JOCCH)*, *6*(1).

[Halaman ini sengaja dikosongkan]

Lampiran 1

Ground Truth Dokumen Pengujian Bahasa Arab

	Query									
IDE_Dok	1	2	3	4	5	6	7	8	9	10
6			1							
7			1							
8				1						
9				1						
10				1						
11				1						
12			1							
13			1							
14			1	1						
15				1						
16				1						
17				1						
18				1						
19				1						
24			1							
26				1						
28			1							
29			1							
33			1							
36			1							
39				1						
41					1					
46				1						
47			1							
48			1							
49			1							
54				1						
55				1						
56		1								
57		1								
58										
59										
60										

106	1							1		
107									1	
108										1
109										1
112										1
113										1
116								1		
117								1		
118										1
124								1		
125								1		
126								1		
127								1		
128								1		
129								1		
130								1		
131								1		
132								1		
133									1	
139								1		
140								1		
141								1		
142								1		
143								1		
144								1		
145								1		
146								1		
147								1		
148									1	
149								1		
150									1	
151								1		
152								1		
170	1									
171	1									
172	1									
173	1									
188	1									
189	1									
190	1									
191	1									
192	1									

193	1									
194	1									
195	1									
196	1									
197	1									
198	1									
199	1									
200	1									
201	1									
202	1									
203	1									
204	1									
205	1									
206						1	1			
207							1			
208							1			
209							1			
210							1			
211							1			
212							1			
213							1			
214							1			
215							1			
216							1			
217							1			
218							1			
219						1				
220						1				
221						1				
222							1			
223							1			
224							1			
225							1			
226							1			
227							1			
228							1			
229							1			
230						1				
231							1			
232							1			
233							1			
234							1			

235							1			
236							1			
237							1			
238							1			
239							1			
240							1			
241							1			
242							1			
243							1			
244							1			
245							1			
246							1			
247							1			
248							1			
249							1			
250							1			
251							1			
252							1			
253							1			
254							1			
255							1			
258			1							
260					1					
261					1					
262					1					
264					1					
282					1					
283					1					
335										
338		1								
340		1								
344		1								
345										
347		1								
348		1								
349		1								
350		1								
352		1								
355		1								
356								1		
357								1		
360									1	

361									1	
362									1	
363									1	
364									1	
365									1	
366								1		
367								1		
368								1		
369								1		
370								1		
371								1		
372								1		
373								1		
374									1	
375								1		
376								1		
377								1		
378								1		
380										1
381								1		
382								1		
383								1		
384								1		
385								1		
386								1		
387								1		
388								1		
393									1	
394									1	
395									1	
396								1		
397								1		
398								1		
399										1
400								1		
401								1		
402								1		
403								1		
404								1		
405								1		
407	1									
408										

416	1									
417	1									
418	1									
428										
429										
432										
433										
438	1									
439	1									
440	1									
441	1									
442	1									
445	1									
447	1									
468							1			
469							1			
471							1			
473							1			
476							1			
478							1			
479							1			
480							1			
481						1				
483							1			
484							1			
489							1			
490						1				
493							1			
495							1			
497										
498						1				
499							1			
500						1				
501							1			
502							1			
503							1			
504							1			

Lampiran 2

Hasil Thesaurus

- Term:
نوع
Expansion:
مشي بدل بيع نصب ولي قول
- Term:
وئام
Expansion:

- Term:
صلي
Expansion:

- Term:
فضل
Expansion:
ولي قول
- Term:
فود
Expansion:
زكي جبي مول
- Term:
اذا
Expansion:
عمر فلي رمض قلل قول صلي شيأ غور ولي ملك عبد
أبن فعل كره ومأ أمر
- Term:
ترك

Expansion:

ولي قول

- Term:

وصي

Expansion:

قول

- Term:

زوج

Expansion:

قول

- Term:

نفذ

Expansion:

جمع نمي لأن غور ولي قول

- Term:

قون

Expansion:

ربع بوب ولم نبه رحم قول نهر حلي سلم ولي صلي
عقل عمر وسن رسل بتل برك غور صبح قرأ شيأ
نسخ دبر حمد جزأ نزل علم نهي يا قلل بني
صلت سقط أبي قرن

- Term:

جبي

Expansion:

قول

- Term:

شرط

Expansion:

- Term:

لحج

Expansion:

حوج قول

- Term:

نبغ

Expansion:

رأى قلل ولي قول غور

- Term:

اداء

Expansion:

فرض وجب غور لأن قول صلي ولي

- Term:

مور

Expansion:

أبن ملك عمر ولي سلم صلي قول غور

- Term:

صدر

Expansion:

قول

- Term:

شخص

Expansion:

مكة حوج وعي وجد صفف شرط حرم غور بطل جمع قرب
ضيف نوع طرق خلي ذكر وأم

- Term:

لزم

Expansion:

قول

- Term:

رسم

Expansion:

نقص ملأ لفظ تتفق ثلث نسخ نوص قرأ حقق نهج ثني
رجح قول حشي ذكر عقف حجز كثر كلم ثبت فإني
قلل سول فإن عمر وضع توبيب كفأ فحينئذ الي كون

وأر	تيح	دخل	يلي	لظي	سوق	خلل	فلي	سور	ترك
صلي	لزم	كني	نوع	غور	بقي	كمل	قحم	سلأ	وثب
أجب	بخر	أكد	طمس	خوذ	ولم	ولي	متن	حرص	قرب
شرك	يقت	رمض	بوق	قود	مرد	تمم	فود	أتم	جوب
بذأ	فهم	عمم	مكة	حطب	درأ	صبأ	وكذا	غوي	فقأ
حلف	ذوي	مدين	سيب	مدي	دنا	نفي	قررناه	زوم	حجر
بسر	رفق	جور	عظم	ثلثمائة	صغر	لوم	همل	لحي	وضم
رحل	هوع	قوا	سكن	لجم	جحف	عرب	أرض	مصر	شوم
غين	همز	بالراء	ربغ	فلذ	خفي	بوأ	طرق	شمل	مغرب
كسر	شرق	نهر	وري	خرس	عرق	نصف	حوط	ربض	عجم
	مطل	بسأ	قرن	سلك	نجد	جبل	ستت	سبخ	الراء

- Term:

عشر

Expansion:

قول

- Term:

صوم

Expansion:

قول

- Term:

شيأ

Expansion:

قول ولي

- Term:

فطر

Expansion:

قول صوم

Lampiran 3

Hasil Dokumen Relevan Query Q8

Ide_Dokumen	Topik Dokumen	Ground Truth
386	puasa	Benar
147	puasa	Benar
377	puasa	Benar
150	puasa	Salah
125	puasa	Benar
116	puasa	Benar
129	puasa	Benar
143	puasa	Benar
130	puasa	Benar
396	puasa	Benar
124	puasa	Benar
127	puasa	Benar
113	puasa	Salah
128	puasa	Benar
140	puasa	Benar
369	puasa	Benar
118	puasa	Salah
131	puasa	Benar
393	puasa	Salah
148	puasa	Salah
145	puasa	Benar
399	puasa	Salah
387	puasa	Benar
117	puasa	Benar
370	puasa	Benar
388	puasa	Benar
106	puasa	Benar
142	puasa	Benar
362	puasa	Salah
364	puasa	Salah

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS

Penulis, **Dika Rizky Yunianto**, lahir di Malang, 06 Juni 1992. Biasa dipanggil dengan nama Dika. Anak kedua dari 2 bersaudara dan dibesarkan di kota Malang, Jawa Timur. Penulis menempuh pendidikan formal di MI Negeri 1 Malang (1998-2004), MTs Negeri I Malang (2004-2007), SMA Negeri 8 Malang (2007-2010) Pada tahun 2010-2014 penulis melanjutkan studinya di jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya Malang. Pada tahun 2014-2016, penulis melanjutkan pendidikan Magister S2 di jurusan yang sama, yaitu Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya, Jawa Timur. Di Jurusan Teknik Informatika, penulis mengambil bidang minat Komputasi Cerdas dengan konsentrasi ilmu Data Mining serta pengolahan teks.