# PERBANDINGAN METODE *DICE SIMILARITY* DENGAN *COSINE SIMILARITY* MENGGUNAKAN *QUERY EXPANSION* PADA PENCARIAN *AYATUL AHKAM* DALAM TERJEMAH ALQURAN BERBAHASA INDONESIA

**SKRIPSI**

**Oleh:**
**AHMAD DZUL FIKRI**
**NIM. 13650031**

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

# PERBANDINGAN METODE *DICE SIMILARITY* DENGAN *COSINE SIMILARITY* MENGGUNAKAN *QUERY EXPANSION* PADA PENCARIAN *AYATUL AHKAM* DALAM TERJEMAH ALQURAN BERBAHASA INDONESIA

**SKRIPSI**

**Diajukan kepada:**
**Universitas Islam Negeri Maulana Malik Ibrahim Malang**
**Untuk Memenuhi Salah Satu Persyaratan dalam**
**Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh :**
**AHMAD DZUL FIKRI**
**NIM. 13650031**

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

**LEMBAR PERSETUJUAN**

**PERBANDINGAN METODE *DICE SIMILARITY* DENGAN *COSINE SIMILARITY* MENGGUNAKAN *QUERY EXPANSION* PADA PENCARIAN *AYATUL AHKAM* DALAM TERJEMAH ALQURAN BERBAHASA INDONESIA**

**SKRIPSI**

**Oleh :**
**AHMAD DZUL FIKRI**
**NIM. 13650031**

**Telah Diperiksa dan Disetujui untuk diuji**
**Tanggal November 2018**

| Dosen Pembimbing I, | Dosen Pembimbing II, |
|---|---|
| **Dr. Suhartono, M.Kom** | **Syahiduz Zaman, M.Kom** |
| **NIP. 19680519 200312 1 001** | **NIP. 19700502 200501 1 005** |

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

**Dr. Cahyo Crysdian**
**NIP. 19740424 200901 1 008**

## LEMBAR PENGESAHAN

**PERBANDINGAN METODE *DICE SIMILARITY* DENGAN *COSINE SIMILARITY* MENGGUNAKAN *QUERY EXPANSION* PADA PENCARIAN *AYATUL AHKAM* DALAM TERJEMAH ALQURAN BERBAHASA INDONESIA**

**SKRIPSI**

**Oleh :**
**AHMAD DZUL FIKRI**
**NIM. 13650031**

**Telah Dipertahankan di Depan Dewan Penguji Skripsi dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan untuk Memperoleh Gelar Sarjana Komputer (S.Kom)**
**Tanggal : Desember 2018**

| Susunan Dewan Penguji | | Tanda Tangan |
|---|---|---|
| Penguji Utama | : Fatchurrochman, M.Kom<br>NIP. 19700731 200501 1 002 | ( ) |
| Ketua Penguji | : A'la Syauqi, M.Kom<br>NIP. 19771201 200801 1 007 | ( ) |
| Sekretaris Penguji | : Dr. Suhartono, M.Kom<br>NIP. 19680519 200312 1 001 | ( ) |
| Anggota Penguji | : Syahiduz Zaman, M.Kom<br>NIP. 19700502 200501 1 005 | ( ) |

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

**Dr. Cahyo Crysdian**
**NIP. 19740424 200901 1 008**

## PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini :

Nama : Ahmad Dzul Fikri

Nim : 13650031

Fakultas / Jurusan : Sains dan Teknologi / Teknik Informatika

Judul Skripsi : **PERBANDINGAN METODE *DICE SIMILARITY* DENGAN *COSINE SIMILARITY* MENGGUNAKAN *QUERY EXPANSION* PADA PENCARIAN *AYATUL AHKAM* DALAM TERJEMAH ALQURAN BERBAHASA INDONESIA.**

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar – benar merupakan hasil karya saya sendiri, bukan merupakan pengambilalihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan Skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 26 November 2018

Yang membuat pernyataan


Ahmad Dzul Fikri
NIM. 13650031

## MOTTO

مَنْ عَرَفَ نَفْسَهُ فَقَدْ عَرَفَ رَبَّهُ

*Barang siapa yang telah mengenal siapa sebenarnya dirinya sendiri, maka sesungguhnya ia telah mengenal Tuhannya.*

- *Ulama Sufi*

*Much love will kill you. Except, much love to God.*

*- Ahmad D. Fikri.*

## HALAMAN PERSEMBAHAN

Alhamdulillah puji syukur ke Hadirat Allah SWT yang telah memberikan nikmat *dhohiriyah* dan *bathinyah* sehingga penulis mampu untuk menyelesaikan studi S1 di kampus UIN Malang ini. Salawat serta salam selalu tercurahkan kepada Baginda Nabi Muhammad SAW, yang telah membimbing umatnya menuju jalan yang benar.

Terima kasih kepada kedua orang tua, sang Ayah tercinta, Bapak Muhammad Yazid yang selalu mendidik dan memberikan contoh kehidupan terutama hal agama. Ibu Zubaidah yang tak lelah untuk menyayangi, rela berkorban, sehingga dapat merasakan kehidupan sampai saat ini. Tak lupa adik-adik saya, semoga seluruh tujuan tercapai dan diberikan yang terbaik.

Teruntuk seluruh guru, ustad, kiai dan dosen mulai Sekolah Dasar, Pondok Pesantren hingga Perguruan Tinggi. Pembimbing skripsiku Dr. Suhartono, M. Kom. dan Syahiduz Zaman, M.Kom yang dengan tulus, sabar, dan ikhlas membimbing serta menyalurkan pengetahuannya. Nasehat-nasehat bapak akan selalu diingat dan kita akan terus terhubung melalui sambung doa sampai akhir hayatku.

Teman seperjuangan *Fortinity* TI'13 UIN Maliki Malang, adik-adik angkatan, Memofoution, PMII rayon Pencerahan Galileo serta keluarga Kontrakan70an yang telah meluangkan waktunya. Rekan-rekan dan semua pihak yang tak bisa disebutkan satu persatu, terima kasih. Semoga terus terhubung meskipun dalam untaian doa yang mengiringi kesuksesan kita.

# KATA PENGANTAR

*Assalamualaikum Warahmatullahi Wabarokatuhu.*

*Alhamdulillahi Robbil 'Alamiin*, segala puji bagi Allah yang selalu memberikan nikmat *dhohiriyah* dan nikmat *bathiniyah* dalam proses penyelesaian skripsi ini. Sholawat serta salam selalu tercurahkan kepada junjungan, baginda dan pusaka umat islam, Nabi Muhammad SAW yang telah memberikan teladan, bimbingan dan petunjuk, sehingga umat manusia menjadi lebih beradab.

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

1. Dr. Suhartono, M. Kom., selaku dosen pembimbing I yang telah meluangkan waktu untuk membimbing, mengarahkan dan memberi masukan kepada penulis dalam pengerjaan skripsi ini hingga akhir.
2. Syahiduz Zaman, M.Kom, selaku dosen pembimbing II yang telah membimbing serta memberikan masukan kepada penulis dalam pengerjaan skripsi ini.
3. Bapak Dr. Cahyo Crysdian, selaku Ketua Jurusan Teknik Informatika yang telah memberikan motivasi untuk terus berjuang.
4. Segenap dosen teknik informatika yang telah memberikan bimbingan keilmuan kepada penulis selama masa studi.
5. Teman-teman seperjuangan teknik informatika Fortinity 2013.

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

*Wassalamualaikum Warahmatullahi.Wabarokatuhu.*

Malang, 26 November 2018

Penulis

## DAFTAR ISI

## DAFTAR GAMBAR

## DAFTAR TABEL

# ABSTRAK

Fikri, Ahmad Dzul. **Perbandingan Metode Dice Similarity Dengan Cosine Similarity Menggunakan Query Expansion Pada Pencarian Ayatul Ahkam Dalam Terjemah Alquran Berbahasa Indonesia**. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing (I) Dr. Suhartono, M. Kom, (II) Syahiduz Zaman, M.Kom.

Kata Kunci : *Ayatul ahkam*, Perbandingan, *Dice similarity*, *Cosine similarity*, *Query Expansion*.

---

Alquran merupakan sebuah pedoman kehidupan bagi umat islam. Dalam persoalan hukum yang berkaitan ibadah, muamalah, *siyasah* dan lain-lain, Alquran merupakan sumber hukum pertama yang dapat digunakan sebagai dasar hukum. Ayat-ayat Alquran yang digunakan sebagai dasar hukum disebut dengan *Ayatul Ahkam*. Sebuah sistem pencarian dibangun untuk dapat memudahkan pencarian *ayatul ahkam* dalam kumpulan dokumen terjemah Alquran berbahasa Indonesia. Pendekatan kemiripan antara kata kunci dengan dokumen dihitung dengan menggunakan *dice similarity* dan *cosine similarity*. Tujuan penelitian ini adalah untuk mengetahui perbandingan akurasi dan relevansi dokumen yang dihasilkan diantara kedua metode tersebut dengan atau tanpa menggunakan *query expansion*. Nilai yang menjadi ukuran adalah *recall, precision* dan *f-measure*. Hasil penelitian menunjukkan bahwa metode *dice similarity* dengan menggunakan *query expansion* memiliki nilai *recall* paling tinggi yaitu sebesar 85,357% dibanding metode lainnya. Metode *cosine similarity* menggunakan *query expansion* memiliki nilai *precision* yang tinggi dibanding dengan metode lainnya, yaitu sebesar 10,041% dan nilai *f-measure* yang tinggi dibandingkan metode lainnya yaitu sebesar 17,061%. Jadi, metode *cosine similarity* dengan menggunakan *query expansion* merupakan metode terbaik karena unggul dalam dua faktor uji evaluasi sistem yaitu *precision* dan *f-measure* dibandingkan metode lainnya.

# ABSTRACT

Fikri, Ahmad Dzul. **Comparison of Dice Similarity and Cosine Similarity Using Query Expansion on The Search for Ayatul Ahkam in Indonesian Translation of Alquran**. Undergradute Thesis. Informatics Engineering Department of Science and Technology Faculty Islamic State University Maulana Malik Ibrahim Malang. Supervisor: (I) Dr. Suhartono, M. Kom, (II) Syahiduz Zaman, M.Kom.

Keyword : Ayatul ahkam, Comparison, Dice similarity, Cosine similarity, Query expansion.

Alquran is a life guide for Muslims.In legal matters relating to worship, muamalah, civil law and others, Alquran is the first legal source that can be used as a legal basis. The verses of Alquran which are used as a legal basis are called Ayatul Ahkam.. A search system was built to facilitate the search for ayat al-ahkam in the Indonesian translation of Alquran. The approach of similarity between keywords and documents is calculated using dice similarity and cosine similarity. The purpose of this research is to compare the accuracy and relevance of documents produced between the two methods with or without using query expansion. The value used as a comparison of methods is recall, precision and f-measure. The results showed that the dice similarity method using query expansion had the highest recall value of 85.357% compared to other methods. The cosine similarity method using query expansion has a high precision value compared to other methods, which is equal to 10.041% and a high f-measure value compared to other methods, which is 17.061%. So, the cosine similarity method using query expansion is the best method because it excels in two factors of system evaluation tests, namely precision and f-measure compared to other methods.

# ملخص

فكر، أحمد ذوال. **مقارنه بين تشابه dice وتشا به cosine باستخدام توسيع الاستعلام على البحث عن الاية الاحكم فى الترجمة ألإندونيسية فى القرأن.** أطروحة الجامعية . قسم هندسة المعلوماتية لكلية العلوم والتكنولوجيا في جامعة الدولة الإسلامية مولانا مالك إبراهيم مالانج . المشرف : (الأحد) الدكتور سوهارتونو، ماجيستير فى الكمبوت ، (الإثنان) شاهد الزمان، ما جيستير فى الكمبوت

الكلمات الدالة : اية الاحكام ، مقارنة ، تشابه dice ، تسابه cosine ، توسيع الاستعلام

---

القرآن هو دليل الحياة للمسلمين . فى المسائل القانونية المتعلقة بالعبادة ، والمعاملة ، القانون المدني و غيرها ، القرآن هو المصدر القنوني الاول الذي يمكن إستخدامه كأساس قانوني . وتسمّى آيات القرآن التي تُستخدم كأساس قانوني بإسم آية الاحكام . تم إنشاء نظام بحث لتسهيل البحث عن آيات الأحكام في الترجمة الإندونيسية للقرآن . يتم حساب نهج التشابه بين الكلمات الرئيسية والمستندات باستخدام تشابه dice وتشابه cosine . الغرض من هذا البحث هو دقة وأهمية المستندات التي تم إنتاجها بين الطريقتين مع أو بدون استخدام توسيع الاستعلام . القيمة المستخدمة كمقارنة بين الأساليب هي الاستدعاء والدقة وf-measure . ظهرت النتائج أن نفس التشابه dice باستخدام التوسع الاستعلام كانت أعلى قيمة سحب 85,357% مقارنة بالطرق الأخرى . تشابه جيب التمام باستخدام توسيع الاستعلام هو قيمة عالية الدقة مقارنة بالطرق الأخرى ، التي تساوي 10,041% وقيمة f-measure عالية مقارنة بالطرق الأخرى ، والتي هي 17,061% . و لذا ، فإن التشابه cosine باستخدام توسيع الاستعلام هو أفضل طريقة لأنه يتفوق في تقييم الاختبارات ، أي الدقة والقياس مقارنة بالطرق الأخرى.

# BAB I

# PENDAHULUAN

Pada bab ini akan dijelaskan mengenai latar belakang penelitian, identifikasi masalah, tujuan penelitian, manfaat penelitian dan batasan penelitian.

## 1.1 Latar Belakang

Allah SWT mewahyukan Alquran kepada Nabi Muhammad SAW untuk disampaikan kepada seluruh umat manusia. Proses pewahyuan/penurunan Alquran dari Allah SWT kepada Nabi Muhammad SAW dengan perantara Malaikat Jibril melalui berbagai cara yaitu mimpi, melalui suara, dan lain-lain. Alquran diwahyukan/diturunkan secara berangsur-angsur selama 23 tahun. Penurunan yang secara berangsur-angsur ini tidak lain adalah bertujuan agar mudah diresapi dan diamalkan dalam perilaku keseharian. Pengamalan Alquran dalam kehidupan merupakan sebuah keyakinan bahwa Alquran adalah kitab suci yang harus diposisikan sebagai pedoman hidup dalam berbagai aspek kehidupan, meliputi hubungan antara manusia dengan Tuhannya, hubungan antar sesama manusia, dan juga hubungan antara manusia dengan alam.

Umat Islam meyakini bahwa kandungan makna Alquran selalu relevan dalam kehidupan pada setiap zaman. Alquran merupakan sebuah buku pedoman, pegangan bagi umat Islam. Kepadanya seluruh aktivitas kehidupan manusia dirujukkan. Bukan hanya aktivitas yang menyangkut kehidupan manusia terkait dengan soal ukhrawi melainkan juga yang terkait dengan persoalan duniawi.

Semua ayat Alquran, walaupun diturunkan di masa lalu, selalu bisa digunakan sebagai pedoman untuk menyelesaikan permasalahan manusia di masa sekarang maupun yang akan datang. Permasalahan-permasalahan yang terkait dengan moral seperti pembunuhan, kesaksian palsu, perzinaan dan lain-lain selalu menjadi permasalahan yang akan terus menerus melilit umat manusia. Maka dari itu, Alquran disebut sebagai kitab petunjuk bagi seluruh manusia. Hal ini sesuai dengan firman Allah SWT surat *Ali-Imron* ayat 7:

هُوَ ٱلَّذِىٓ أَنزَلَ عَلَيْكَ ٱلْكِتَٰبَ مِنْهُ ءَايَٰتٌ مُّحْكَمَٰتٌ هُنَّ أُمُّ ٱلْكِتَٰبِ وَأُخَرُ مُتَشَٰبِهَٰتٌ ۖ فَأَمَّا ٱلَّذِينَ فِى قُلُوبِهِمْ زَيْغٌ فَيَتَّبِعُونَ مَا تَشَٰبَهَ مِنْهُ ٱبْتِغَآءَ ٱلْفِتْنَةِ وَٱبْتِغَآءَ تَأْوِيلِهِۦ ۗ وَمَا يَعْلَمُ تَأْوِيلَهُۥٓ إِلَّا ٱللَّهُ ۘ وَٱلرَّٰسِخُونَ فِى ٱلْعِلْمِ يَقُولُونَ ءَامَنَّا بِهِۦ كُلٌّ مِّنْ عِندِ رَبِّنَا ۗ وَمَا يَذَّكَّرُ إِلَّآ أُو۟لُوا۟ ٱلْأَلْبَٰبِ ٧

*Artinya (dari tafsir jalalain): (Dialah yang menurunkan kepadamu Alquran, di antara isinya ada ayat-ayat yang muhkamat) jelas maksud dan tujuannya (itulah dia pokok-pokok Alquran) yakni yang menjadi pegangan dalam menetapkan (sedangkan yang lainnya mutasyabihat) tidak dimengerti secara jelas maksudnya, misalnya permulaan-permulaan surah. Semuanya disebut sebagai 'muhkam' seperti dalam firman-Nya 'uhkimat aayaatuh' dengan arti tak ada cacat atau celanya, dan 'mutasyaabiha' pada firman-Nya, 'Kitaaban mutasyaabiha,' dengan makna bahwa sebagian menyamai lainnya dalam keindahan dan kebenaran. (Adapun orang-orang yang dalam hatinya ada kecenderungan pada kesesatan) menyeleweng dari kebenaran, (maka mereka mengikuti ayat-ayat mutasyabihat untuk membangkitkan fitnah) di kalangan orang-orang bodoh dengan menjerumuskan mereka ke dalam hal-hal yang syubhat dan kabur pengertiannya (dan demi untuk mencari-cari takwilnya) tafsirnya (padahal tidak ada yang tahu takwil) tafsirnya (kecuali Allah) sendiri-Nya (dan orang-orang yang mendalam) luas lagi kokoh (ilmunya) menjadi mubtada, sedangkan khabarnya: (Berkata, "Kami beriman kepada ayat-ayat mutasyaabihat) bahwa ia dari Allah, sedangkan kami tidak tahu akan maksudnya, (semuanya itu) baik yang muhkam maupun yang mutasyabih (dari sisi Tuhan kami," dan tidak ada yang mengambil pelajaran) 'Ta' yang pada asalnya terdapat pada 'dzal' diidgamkan pada dzal itu hingga berbunyi 'yadzdzakkaru' (kecuali orang-orang yang berakal) yang mau berpikir. Mereka juga mengucapkan hal berikut bila melihat orang-orang yang mengikuti mereka.*

Sebagai sebuah pedoman bagi umat muslim dan umat manusia, Alquran memuat informasi dasar dalam berbagai masalah mengenai hukum, etika, *science*,

astronomi, kedokteran dan sebagainya. Dalam persoalan hukum yang berkaitan ibadah, muamalah, *siyasah* dan lain-lain, Alquran merupakan sumber hukum pertama yang dapat digunakan sebagai dasar hukum dan penetapan hukum islam untuk memecahkan suatu masalah dalam kehidupan sehari-hari. Ayat-ayat Alquran yang digunakan dalam pemecahan suatu masalah ataupun penetapan hukum disebut dengan *ayatul ahkam*. Karena Alquran merupakan referensi dalam penetapan hukum yang pertama kali dirujuk, maka diperlukan sebuah sistem untuk melakukan pencarian *ayatul ahkam* agar memudahkan bagi pengguna dalam mendapatkan sumber untuk penetapan sebuah hukum.

Untuk memecahkan masalah yang berkaitan dengan hukum Islam dalam Alquran tidak hanya mengacu pada satu atau dua ayat. Sedangkan Alquran memiliki 30 juz, 114 surat dan 6348 ayat. Sehingga bila pencarian *ayatul ahkam* dilakukan secara manual, maka akan membutuhkan waktu yang lama. Dan juga, kesusahan bagi pengguna dalam mencari *ayatul ahkam* adalah belum terdapat pengelompokan secara tematik. Pencarian data sederhana untuk mendapatkan informasi berdasarkan kata dan memasangkannya dengan suatu dokumen sudah umum dilakukan pada sistem komputer saat ini. Proses ini bisa memberikan hasil pencarian dokumen yang ditemukan pada sistem baik hasil yang relevan ataupun tidak. Namun pemrosesan ini memiliki banyak kelemahan seperti waktu proses yang lama (Manning, 2008).

Berdasarkan permasalahan di atas, dibutuhkan sebuah metode yang efektif dalam pencarian. Metode-metode yang efektif dalam sistem pencarian dipelajari dalam bidang *Information Retrieval System* (Manning, 2008). Teknik pencarian pada suatu sistem umumnya menggunakan pencocokan antara *query* masukan

pengguna dengan kata-kata yang terdapat pada sebuah dokumen terjemah Alquran. Ketika *query* pengguna tidak terdapat pada dokumen terjemah Alquran maka dokumen tersebut tidak termasuk dalam dokumen yang pengguna inginkan. Apabila dokumen tersebut, yang tidak cocok dengan *query* pengguna, memang tidak berkaitan dengan *query* pengguna, sistem pencarian akan menyajikan dokumen yang tepat pada pengguna. Akan tetapi jika dokumen tersebut, yang tidak cocok dengan *query* pengguna, ternyata berkaitan (relevan) dengan *query* pengguna, maka sistem pencarian dianggap kurang berhasil dengan tidak menyajikan dokumen yang terkait dengan *query*.

Untuk itu dibutuhkan sebuah cara dalam mengatasi hal tersebut. Cara yang akan dilakukan adalah pencarian tidak hanya berdasarkan kecocokan antara *query* pengguna dengan *term* yang ada di dokumen tetapi juga dengan memperhatikan keterkaitan *term* pada *query* pengguna dengan dokumen. Keterkaitan *term* pada pengguna diperoleh dengan menerapkan metode *query expansion* (perluasan kueri) dengan menggunakan Kamus Tesaurus Bahasa Indonesia. *Query Expansion* atau perluasan *query* adalah proses memformulasikan kembali *query* awal dengan melakukan penambahan beberapa *term* atau kata pada *query* untuk meningkatkan perfoma dalam proses *information retrieval* (Qiu, 1993).

Penelitian yang telah dilakukan terkait dengan sistem pencarian Alquran, yaitu penelitian yang dilakukan oleh Muhammad Muharrom Al-Haromainy (2017). Pada penelitian tersebut dilakukan pembobotan pada teks terjemahan Alquran, dan menjadikan terjemahan Alquran yang membahas hewan dan tumbuhan sebagai objeknya. Kemudian setelah hasil dari pembobotan yang termuat dalam *Vector Space Model* diperoleh, maka dilakukan sebuah proses penghitungan kemiripan

*query* pengguna dengan dokumen menggunakan metode *cosine similarity*. Pengujian sistem dilakukan dengan menggunakan 20 kata kunci yang menghasilkan *precision* sebesar 83,93%, *recall* 93,51% dan *accuracy* sebesar 98,27%.

Selain metode *cosine similarity*, terdapat beberapa metode untuk pengukuran kemiripan antara *query* dengan dokumen yaitu *dice similarity, jaccard similarity,* dan lain-lain. Setiap *similarity* memiliki kinerja tersendiri dalam pengukuran kemiripan antara *query* pengguna dengan dokumen. Kinerja dari metode tersebut dapat diukur keakuratannya. Ukuran keakuratan dokumen ditentukan berdasarkan relevansi dokumen yang dihasilkan dengan *query* pengguna. Untuk mengetahui tingkat keakuratan tersebut, dibutuhkan sebuah studi untuk membandingkan metode pengukuran kemiripan *query* pengguna dengan dokumen pada pencarian *ayatul ahkam* dalam terjemah Alquran berbahasa Indonesia. Peneliti menggunakan metode *dice similarity* dan *cosine similarity* dikarenakan menurut Chahal(2016) metode *dice similartiy* dan *cosine similarity* termasuk dalam ruang lingkup *vector space model* dalam *information retrieval system*.

Studi kinerja metode-metode *similarity* pernah dilakukan oleh Thada (2013) dengan membandingkan koefisien *Jaccard, Dice* dan *Cosine similarity* untuk menemukan dokumen web dengan algoritma genetika. Setelah dilakukan percobaan pada *query* yang sama pada setiap koefisien dan memilih 10 halaman web pertama yang di-retrieve dari Google menunjukkan bahwa koefisien yang sesuai yaitu koefisien *cosine* kemudian diikuti oleh koefisien *dice* dan *jaccard*. Penelitian ini berfokus pada perbandingan antara metode *cosine similarity* dengan *dice similarity* pada pencarian Ayatul Ahkam dalam terjemah Alquran berbahasa

Indonesia. Perbandingan tersebut diukur berdasarkan keakuratan setiap metode dalam menghasilkan dokumen yang relevan terhadap *query* pengguna. Perbandingan keakuratan didasarkan pada *recall*, *precision* dan *f-measure* yang diperoleh dari setiap pengujian metode tersebut. Selain itu juga ditambahkan *query expansion* pada setiap proses pengukuran kemiripan *query* dengan dokumen. *Query expansion* berguna untuk meningkatkan performa setiap hasil dari proses pencarian baik menggunakan metode *dice similarity* maupun *cosine similarity*. *Query expansion* pada penelitian ini didasarkan keterkaitan antara *term* dengan gugus kata yang terdaftar pada Kamus Tesaurus Bahasa Indonesia. Sehingga peneliti mengangkat judul "**Perbandingan metode *dice similarity* dengan *cosine similarity* menggunakan *query expansion* pada pencarian *ayatul ahkam* dalam terjemah** Alquran **berbahasa indonesia**" pada penelitian ini.

## 1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan sebelumnya maka terdapat permasalahan yang diangkat dalam penelitian ini yaitu:

1. Mengukur penggunaan metode *dice similarity* dan *cosine similarity* dengan atau tanpa menggunakan *query expansion* pada pencarian *ayatul ahkam* dalam terjemah Alquran berbahasa Indonesia.
2. Mengukur perbandingan penggunaan metode *dice similarity* dan *cosine similarity* dengan atau tanpa menggunakan *query expansion* pada pencarian *ayatul ahkam* dalam terjemah Alquran berbahasa Indonesia.

## 1.3 Tujuan Penelitian

Adapun maksud dan tujuan yang didapat dari penelitian ini adalah sebagai berikut ini.

1. Untuk mengukur penggunaan metode *dice similarity* dan *cosine similarity* dengan atau tanpa menggunakan *query expansion* pada pencarian *ayatul ahkam* dalam terjemah Alquran berbahasa Indonesia.
2. Untuk mengetahui perbandingan metode *dice similarity* dan *cosine similarity* dengan atau tanpa menggunakan *query expansion* pada pencarian *ayatul ahkam* dalam terjemah Alquran berbahasa Indonesia.

## 1.4 Manfaat Penelitian

Manfaat yang didapat dari penelitian ini dapat dipandang dari dua aspek yaitu akademik dan pembaca. Manfaat dari segi akademik adalah sebagai berikut:

1. Memudahkan pemilihan metode yang digunakan untuk mengukur kemiripan antara *query* dengan dokumen pada pencarian dalam terjemah Alquran berbahasa Indonesia.
2. Sebagai referensi pada perbandingan metode *cosine* dan *dice* dalam sistem temu balik informasi.

Manfaat dari sisi pembaca adalah memudahkan pembaca untuk melakukan pencarian *ayatul ahkam* dalam Alquran.

## 1.5 Batasan Penelitian

Agar pembahasan penelitian ini tidak menyimpang dari apa yang telah dirumuskan, maka diperlukan batasan-batasan. Batasan-batasan dalam penelitian ini adalah:

1. Menggunakan Alquran terjemahan terbitan Kementerian Agama Republik Indonesia edisi Tahun 2018 .
2. Data *ayatul ahkam* diambil dari situs http://alquranalhadi.com/.

3. Metode yang dibandingkan adalah *cosine similarity* dan *dice similarity*.
4. *Query expansion* menggunakan tesaurus bahasa Indonesia terbitan Pusat Bahasa Departemen Pendidikan Nasional.
5. *Query* berupa teks bahasa Indonesia.

# BAB II

# TINJAUAN PUSTAKA

Pada bagian ini membahas tentang penelitian yang terkait dan konsep tentang teori yang digunakan dalam melakukan penelitian ini.

## 2.1 Alquran

Alquran adalah kitab suci yang diwahyukan oleh Allah SWT kepada nabi Muhammad SAW untuk disampaikan kepada seluruh umat manusia. Alquran diturunkan dengan cara yang berangsur-angsur selama kurang lebih 23 tahun. Proses penurunan Alquran yang berangsur-angsur ini, bukan turun sekaligus dalam satu kitab, mempunyai sebuah hikmah tersendiri yaitu agar makna dalam Alquran mudah diresapi dan diaplikasikan dalam kehidupan sehari-hari. Dan juga, tujuan diturunkannya Alquran kepada seluruh umat manusia adalah sebagai petunjuk/pedoman dalam setiap aspek kehidupan, serta untuk membimbing dan mengeluarkan umat manusia dari jalan kegelapan menuju jalan yang terang benderang. Hal ini sesuai firman Allah SWT dalam surat *Al-A'rof* ayat 52:

وَلَقَدْ جِئْنَٰهُم بِكِتَٰبٍ فَصَّلْنَٰهُ عَلَىٰ عِلْمٍ هُدًى وَرَحْمَةً لِّقَوْمٍ يُؤْمِنُونَ ٥٢

*52. Dan sesungguhnya Kami telah mendatangkan sebuah Kitab (Al Quran) kepada mereka yang Kami telah menjelaskannya atas dasar pengetahuan Kami; menjadi petunjuk dan rahmat bagi orang-orang yang beriman.*

Dan juga firman Allah SWT pada surat *Yunus* ayat 57:

يَٰٓأَيُّهَا ٱلنَّاسُ قَدْ جَآءَتْكُم مَّوْعِظَةٌ مِّن رَّبِّكُمْ وَشِفَآءٌ لِّمَا فِى ٱلصُّدُورِ وَهُدًى وَرَحْمَةٌ لِّلْمُؤْمِنِينَ ٥٧

*57. Hai manusia, sesungguhnya telah datang kepadamu pelajaran dari Tuhanmu dan penyembuh bagi penyakit-penyakit (yang berada) dalam dada dan petunjuk serta rahmat bagi orang-orang yang beriman.*

Kedua ayat di atas menjelaskan bahwa Alquran adalah sebuah kitab yang diperuntukkan sebagai petunjuk bagi umat manusia yang beriman kepada-Nya. Sebagai sebuah kitab petunjuk, umat islam harus meyakini bahwa makna dan kandungan dalam Alquran selalu relevan pada setiap zaman. Permasalahan-permasalahan moral seperti pembunuhan, perzinaan dan lain-lain merupkan permasalahan yang akan selalu timbul dalam kehidupan manusia. Dan Alquran adalah pedoman dasar dalam memecahkan permasalahan yang timbul tersebut.

Sebagai sebuah pedoman bagi umat muslim dan umat manusia, Alquran memuat informasi dasar dalam berbagai masalah mengenai hukum, etika, *science*, astronomi, kedokteran dan sebagainya. Dalam persoalan hukum yang berkaitan ibadah, muamalah, *siyasah* dan lain-lain, Alquran merupakan sumber hukum pertama yang dapat digunakan sebagai dasar hukum dan penetapan hukum Islam untuk memecahkan suatu masalah dalam kehidupan sehari-hari. Ayat-ayat Alquran yang digunakan dalam pemecahan suatu masalah ataupun penetapan hukum disebut dengan *Ayatul Ahkam*.

### 2.1.1 *Ayatul Ahkam*

*Ayatul Ahkam* adalah ayat-ayat Alquran yang mengandung hukum terkait dengan perbuatan manusia. Tak seperti hukum *taklifi* yang dikategorisasikan oleh para ulama' fikih yakni wajib, sunah, haram, makruh, dan mubah, maka di dalam Alquran hanya menggunakan kata perintah dan kata larangan. Paling jauh, Alquran menggunakan diksi "halal" dan "haram" untuk menjelaskan sesuatu yang boleh dan

tidak boleh dilakukan. Misalnya dalam soal kata "halal" dengan segala derivasinya, Allah SWT berfirman dalam surat *Al-Baqoroh* ayat 187 :

أُحِلَّ لَكُمْ لَيْلَةَ ٱلصِّيَامِ ٱلرَّفَثُ إِلَىٰ نِسَآئِكُمْ ۚ هُنَّ لِبَاسٌ لَّكُمْ وَأَنتُمْ لِبَاسٌ لَّهُنَّ ۗ عَلِمَ ٱللَّهُ أَنَّكُمْ كُنتُمْ تَخْتَانُونَ أَنفُسَكُمْ فَتَابَ عَلَيْكُمْ وَعَفَا عَنكُمْ ۖ فَٱلْـَٰٔنَ بَـٰشِرُوهُنَّ وَٱبْتَغُوا۟ مَا كَتَبَ ٱللَّهُ لَكُمْ ۚ وَكُلُوا۟ وَٱشْرَبُوا۟ حَتَّىٰ يَتَبَيَّنَ لَكُمُ ٱلْخَيْطُ ٱلْأَبْيَضُ مِنَ ٱلْخَيْطِ ٱلْأَسْوَدِ مِنَ ٱلْفَجْرِ ۖ ثُمَّ أَتِمُّوا۟ ٱلصِّيَامَ إِلَى ٱلَّيْلِ ۚ وَلَا تُبَـٰشِرُوهُنَّ وَأَنتُمْ عَـٰكِفُونَ فِى ٱلْمَسَـٰجِدِ ۗ تِلْكَ حُدُودُ ٱللَّهِ فَلَا تَقْرَبُوهَا ۗ كَذَٰلِكَ يُبَيِّنُ ٱللَّهُ ءَايَـٰتِهِۦ لِلنَّاسِ لَعَلَّهُمْ يَتَّقُونَ ١٨٧

*187. Dihalalkan bagi kamu pada malam hari bulan puasa bercampur dengan isteri-isteri kamu; mereka adalah pakaian bagimu, dan kamupun adalah pakaian bagi mereka. Allah mengetahui bahwasanya kamu tidak dapat menahan nafsumu, karena itu Allah mengampuni kamu dan memberi maaf kepadamu. Maka sekarang campurilah mereka dan ikutilah apa yang telah ditetapkan Allah untukmu, dan makan minumlah hingga terang bagimu benang putih dari benang hitam, yaitu fajar. Kemudian sempurnakanlah puasa itu sampai (datang) malam, (tetapi) janganlah kamu campuri mereka itu, sedang kamu beri´tikaf dalam mesjid. Itulah larangan Allah, maka janganlah kamu mendekatinya. Demikianlah Allah menerangkan ayat-ayat-Nya kepada manusia, supaya mereka bertakwa.*

Ayat-ayat hukum dalam Alquran mencakup empat tema pokok (Kaltsum & Moqsith, 2015). Pertama, ayat-ayat ibadah. Yang termasuk di dalamnya adalah ayat yang membahas mengenai wudhu', salat, puasa, haji. Pada bagian ini, Alquran tak menjelaskan secara rinci. Tentang salat misalnya, Alquran tidak menjelaskan secara spesifik mengenai bagaimana mekanisme salat dan tata cara pelaksanaan ibadah salat, hanya menjelaskan tentang wajibnya menunaikan ibadah salat.

Kedua, ayat-ayat *ahwal syahsiyah* yakni hukum keluarga. Yang dibahas pada bagian ini adalah soal nikah, talak, *iddah*, rujuk, dan nafkah. Pada bagian ini, ada ayat Alquran yang membahas secara rinci dan ada juga yang tidak. Misalnya soal *iddah* bagi perempuan, Alquran bukan hanya membahas mengenai wajibnya menjalankan *iddah*, melainkan juga menjelaskan tentang waktu-waktu *iddah*

berdasarkan kondisi perempuan itu. Namun, dalam soal nikah, Alquran menjelaskannya secara global dan umum tanpa menjelaskan tentang syarat dan rukun nikah.

Ketiga, ayat-ayat yang terkait dengan akad perdataan secara umum. Termasuk dalam kategori ini adalah ayat tentang jual beli, sewa-menyewa, gadai, *syuf'ah*, *mudharabah*, hutang piutang. Hukum waris yang mengatur mekanisme perpindahan dan pembagian harta dari satu keluarga pada anggota keluarga loain yang diakibatkan kematian juga menjadi bahasan rinci dalam Alquran.

Keempat, ayat-ayat yang terkait dengan *jinayat* atau pidana. Alquran menjelaskan tentang jenis-jenis pidana, baik pidana umum maupun pidana khusus, sampai dengan sanksi-sanksi hukum yang bisa dikenakan pada pelaku kriminal. Ayat-ayat yang terkait dengan pidana ini seperti pembunuhan, pencurian, perzinaan, disebutkan dalam Alquran bahkan hingga pada jenis sanksi hukumnya. Kemudian, juga terdapat ayat-ayat yang terkait dengan Fikih kontemporer.

## 2.2 *Information Retrieval System*

Definisi *information retrieval* menurut *oxford dictionary* adalah penelurusan dan pemulihan informasi dari data yang tersimpan. Istilah untuk *information retrieval* sendiri dikenalkan oleh Mooers pada tahun 1951. Mooers mendefinisikannya sebagai sebuah proses penemuan atau penemuan kembali sehubungan dengan informasi tersimpan yang mencakup aspek intelektual dari deskripsi informasi dan spesifikasi itu sendiri untuk pencarian, serta mencakup sistem, teknik, atau mesin apa pun yang digunakan untuk melakukan operasi tersebut. Seiring berjalannya waktu, definisi mengenai *information retrieval* berkembang secara luas, seperti yang ditetapkan dalam ISO 2382/1 yang

mendefinisikan *information retrieval* sebagai tindakan, metode dan prosedur untuk menemukan kembali data yang tersimpan, kemudian menyediakan kembali informasi mengenai subjek yang dibutuhkan. Tindakan tersebut mencakup *text indexing*, *inquiry analysis*. Informasi mencakup teks, tabel, gambar, ucapan dan video.

*Information retrieval system* atau sistem temu balik informasi adalah sistem yang digunakan untuk menemukan kembali (*retrieve)* informasi-informasi yang relevan terhadap kebutuhan pengguna dari suatu kumpulan informasi secara otomatis (Bunyamin, 2008). Tujuan dari *information retrieval system* adalah memenuhi kebutuhan informasi pengguna dengan me-*retrieve* semua dokukmen yang relevan, dan pada waktu yang sama me-*retrieve* dokumen-dokumen yang tidak relevan sesedikit mungkin. Kesuksesan sistem temu balik informasi dapat memungkinkan user untuk mendapatkan dokumen yang user inginkan secara akurat dan cepat (Murad, 2007). Contoh dari sistem temu balik informasi yang paling terkenal yang banyak digunakan oleh pengguna internet adalah *search engine* pada *world wide web*. Para pengguna web bisa mendapatkan informasi yang relevan berdasarkan kata yang dimasukkan ke dalam sebuah *seacrh engine*.

Bagian-bagian dari sistem *Information Retrieval* meliputi (Bunyamin, 2008):

1. *Text Operations* (operasi terhadap teks) yang meliputi pemilihan kata-kata dalam *query* maupun dokumen (*term selection*) dalam pentransformasian dokumen atau *query* menjadi *term index* (indeks dari kata-kata).

2. *Query Formulation* (formulasi terhadap *query*) yaitu memberi bobot pada indeks kata-kata *query*.
3. *Ranking* (perangkingan), mencari dokumen-dokumen yang relevan terhadap *query* dan mengurutkan dokumen tersebut berdasarkan kesesuaiannya dengan *query*.
4. *Indexing* (pengindeksan), membangun basis data indeks dari koleksi dokumen. Dilakukan terlebih dahulu sebelum pencarian dokumen dilakukan.

Secara umum, proses yang dilakukan oleh sistem *information retrieval* terdapat dua macam, yaitu melakukan tahap *preprocessing* terhadap data yang disimpan pada *database*, kemudian menerapkan suatu atau beberapa metode untuk menghitung kedekatan antar dokumen yang ada di *database* yang sudah diproses dengan *query* dari pengguna. Pada *preprocessing*, dilakukan tahapan-tahapan mulai dari menghilangkan tanda baca, menghilangkan kata tidak penting, kemudian menjadikan kata kerja menjadi kata dasar, dan yang terakhir adalah melakukan pembobotan pada setiap kata dari *term* yang ada di *database*. Lalu, pada *query* dilakukan proses yang sama, yaitu penghilangan tanda baca, menghilangkan kata tidak penting, lalu menjadikan kata dasar. Hanya saja, nilainya dikembalikan lalu dilakukan pendekatan untuk menghitung nilai kemiripan dengan dokumen, lalu dihasilkan urutan dokumen yang mirip dengan *query*. Hasil perangkingan yang diberikan kepada pengguna merupakan dokumen yang menurut sistem relevan dengan *query*. Namun relevansi dokumen terhadap suatu *query* merupakan penilaian pengguna yang subjektif dan dipengaruhi banyak faktor seperti topik, pewaktuan, sumber informasi maupun tujuan pengguna.

Wisnu (Wisnu & Anindita, 2015) membuat penelitian tentang perancangan *information retrieval* untuk pencarian ide pokok teks artikel berbahasa inggris dengan pembobotan vector space model. Tujuannya untuk mengambil sumber infomasi dengan mengutip sebagian besar isi yang penting dan menampilkan kepada pembaca dalam bentuk ringkas sesuai dengan kebutuhan pembaca. Dengan nilai rata-rata *recall* sebesar 66,86%, *precision* 72,29%, *f-measure* 70,38% hasil peringkasan belum bisa dianggap optimal, salah satu faktor yang mempengaruhi adalah *query* yang digunakan, diperlukan tambahan *query* yang relevan dengan artikel untuk hasil peringkasan yang lebih optimal.

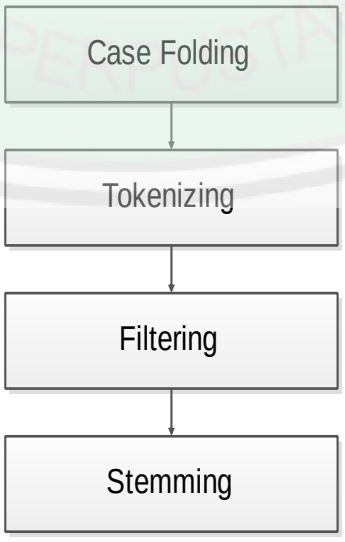
Bunyamin (2008) membuat aplikasi *information retrieval* (IR) CATA dengan metode *generalized vector space model***.** Tujuan aplikasi ini adalah mempermudah *user* untuk mencari informasi dalam koleksi dokumen. Hal yang dilakukan, menggunakan bobot index *term*, vektor dalam *query*, dan menghitung *cross product* untuk menentukan kesamaan *query* dan dokumen.

## 2.3 Penambangan Teks (*Text Mining*)

*Text Mining* proses ektraksi pola (informasi dan pengetahuan yang berguna) dari sejumlah bsar sumber data tak terstruktur. Penambangan teks memiliki tujuan dan menggunakan proses yang sama dengan penambangan data, namun memiliki input yang berbeda. Masukan untuk penambangan adalah data yang tidak atau kurang terstruktur, seperti kutipan teks, dokumen Word, PDF dan lain-lain, sedangkan masukan untuk penambangan data adalah data yang terstruktur (Feldman, 2007). Penambangan teks dapat dianggap sebagai proses dua tahap yang diawali dengan penerapan stuktur terhadapa sumber data teks dan dilanjutkan

dengan ektraksi informasi dan pengetahuan yang relevan dari data teks terstruktur ini, dengan menggunakan teknik dan alat yang sama dengan penambangan data.

*Text mining* mencoba untuk mengekstrak informasi yang berguna dari sumber data melalui identifikasi dan eksplorasi dari suatu pola menarik. Sumber data berupa sekumpulan dokumen dan pola menarik yang tidak ditemukan dalam bentuk *database record*, tetapi dalam data teks yang tidak terstruktur (Indranandita, 2008). Teks yang dilakukan proses *text mining*, pada umumnya memiliki beberapa karakteristik diantaranya adalah memiliki dimensi yang tinggi, terhadap noise pada data, dan terdapat struktur teks yang tidak baik. Cara yang digunakan dalam mempelajari struktur data teks adalah dengan terlebih dahulu menentukan fitur-fitur yang mewakili setiap kata untuk setiap fitur yang ada pada dokumen, sebelum menentukan fitur-fitur yang mewakili, diperlukan tahap *pre-processing* yang dilakukan secara umum dalam *text mining* pada dokumen, yaitu *case folding, tokenizing, filtering, dan stemming* (Raymond, 2006), seperti ditunjukkan pada Gambar 2.1




Gambar 2.1 Proses Penambangan Teks

Misalnya dalam terdapat ayat Alquran yang berbunyi "*Hai orang-orang yang beriman, taatilah Allah dan taatilah Rasul(Nya), dan ulil amri diantara kamu*".

### 2.3.1 *Case Folding*

Tidak semua dokumen teks konsisten dalam penggunaan huruf kapital. Oleh karena itu, case folding dibutuhkan untuk mengkonversi keseluruhan teks dalam dokumen menjadi suatu bentuk standar (biasanya huruf kecil atau lowercase). Dari contoh terjemahan ayat yang telah disebutkan, menjadi "*hai orang-orang yang beriman, taatilah allah dan taatilah rasul(nya), dan ulil amri diantara kamu*".

### 2.3.2 *Tokenizing*

*Tokenizing* yaitu proses penguraian deskripsi yang semula berupa kalimat–kalimat menjadi kata-kata berdasarkan pemisah kata yang ada dalam kalimat tersebut dan menghilangkan delimiter-delimiter seperti tanda titik(.), koma(,), spasi dan karakter angka yang ada pada kata tersebut (Weiss, 2005). Dari contoh ayat diatas, didapatkan kata-kata "*hai orang orang yang beriman taatilah allah dan taatilah rasul nya dan ulil amri diantara kamu*".

### 2.3.3 *Filtering*

Tahap *filtering* adalah tahap mengambil kata-kata penting dari hasil *term*. Tahap ini bisa menggunakan algoritma *stoplist* (membuang kata yang kurang penting) atau *wordlist* (menyimpan kata yang penting). Proses *Stoplist / Stopword* ini melakukan penghapusan kata-kata yang sering muncul dan tidak dipakai di dalam pemrosesan bahasa alami. Proses ini bertujuan untuk mengurangi volume

kata sehingga hanya kata-kata penting yang terdapat di dokumen (Wira, 2009). *Stopword* dapat berupa kata depan, kata penghubung, dan kata pengganti.

Kata yang terdapat dalam kalimat/teks, dibandingkan dengan *stoplist* yang ada dalam bahasa Indonesia. Jika terdapat *stoplist* di dalam kalimat tersebut, maka kata tersebut akan dihapus atau diganti dengan spasi. Kata-kata yang termasuk dalam *stoplist* adalah di, dengan, pada, dan, ini, dan sebagainya. Dari contoh diatas, maka hasil kalimatnya menjadi "*orang orang beriman taatilah allah taatilah rasul ulil amri kamu*".

### 2.3.4 *Stemming*

Setelah kalimat telah melewati proses *case folding, tokenization, filtering*, kalimat tersebut memasuki tahap *stemming*. Tahap *stemming* adalah tahap mencari kata dasar dari tiap kata dari kalimat hasil dari proses *filtering*. Proses ini bertujuan untuk mengurangi variasi kata yang sebenarnya memiliki kata dasar yang sama. Tahap ini kebanyakan dipakai pada kalimat atau teks berbahasa Inggris dan lebih sulit diterapkan pada teks berbahasa Indonesia. Hal ini dikarenakan bahasa Indonesia tidak memiliki rumus bentuk baku yang permanen. Proses ini dapat mengurangi representasi suatu kata dari dokumen tertentu, sehingga dapat mempercepat proses yang dijalankan, beban penyimpanan juga berkurang. Dari contoh terjemahan ayat yang telah disebutkan, menjadi "*orang orang iman taat allah taat rasul ulil amri kamu*".

Terdapat berbagai metode yang digunakan dalam proses *stemming*. Afuan Lasmedi (2013) menggunakan algoritma *porter* untuk melakukan proses *stemming* pada teks bahasa Indonesia. Penelitian ini dilakukan untuk membantu suatu penelitian yang dinamakan temu kembali, karena salah satu tahapnya adalah

*stemming*. *Stemming* mengubah kata-kata dalam dokumen menjadi *root word* atau kata dasar dari kata tersebut.

## 2.4 Algoritma TF-IDF(*Term Frequency – Inverse Document Frequency*)

Di dalam *information retrieval system*, pencarian informasi tidak terlepas dari *query* masukan pengguna dan kesesuaiannya dengan koleksi dokumen yang terdapat dalam *database*. Koleksi dokumen tersebut terdiri dari dokumen-dokumen yang beragam panjangnya dengan kandungan *term* yang berbeda pula. Hal yang perlu diperhatikan dalam pencarian informasi dari koleksi dokumen yang bermacam-macam isinya adalah pembobotan *term*. *Term* yang dimaksud dapat berupa kata, frase atau unit hasil pengindeksan lainnya dalam suatu dokumen yang dapat digunakan untuk mengetahui konteks dari dokumen tersebut. Karena di dalam setiap dokumen terdapat *term-term* yang memiliki tingkat kepentingan yang berbeda, maka pada setiap *term* tersebut diberikan sebuah indikator untuk mengenalinya, yaitu *term weight* (Zafikri, 2010).

Algoritma TF-IDF adalah suatu algoritma yang digunakan untuk memberi bobot hubungan suatu *term* terhadap dokumen. Terdapat dua konsep dalam algoritma ini dalam perhitungan bobot, yaitu *term frequency* atau frekuensi kemunculan sebuah *term* di dalam sebuah dokumen tertentu dan *inverse document frequency* atau *inverse* frekuensi dokumen yang mengandung *term* tersebut (Aziz, 2015). TF atau *term frequency* menentukan bobot *term* pada suatu dokumen berdasarkan jumlah kemunculannya dalam dokumen tersebut. Semakin besar jumlah kemunculan suatu *term* (tf tinggi) dalam nilai dokumen, semakin besar pula bobotnya dalam dokumen atau akan memberikan nilai kesesuaian yang semakin besar (Mandala, 2002). Nilai TF dapat diketahui melalui persamaan berikut ini:

$$TF(t_i, d_j) = f(d_j, t_i) \quad (2.1)$$

Dimana $tf_{ij}$ adalah frekuensi kemunculan *term* t ke-i pada dokumen d ke-j.

Kemudian, *Inverse Document Frequency* (IDF) yaitu pengurangan dominasi *term* yang sering muncul di berbagai dokumen. Banyak *term* yang sering muncul di dalam dokumen dianggap sebagai *term* umum (*common term*) sehingga *term* tersebut dianggap tidak penting nilainya. Latar belakang pembobotan ini adalah *term* yang jarang muncul pada kumpulan dokumen sangat bernilai. IDF merupakan frekuensi dokumen yang mengandung sebuah kata. Nilai IDF dapat diketahui melalui persamaan berikut ini :

$$IDF(t_i, d_j) = 1 + \left(log \frac{D}{dj}\right) \quad (2.2)$$

Dimana D merupakan jumlah total dokumen koleksi yang terdapat dalam *database* dan $d_j$ merupakan jumlah dokumen yang mengandung *term* i.

Berdasarkan TF dan IDF yang telah diketahui persamaanya di atas, bobot *term* i dalam *information retrieval system* ($W_{ij}$) dapat dihitung menggunakan penggabungan rumus TF dan IDF dengan cara mengalikan nilai TF dan IDF seperti ditunjukkan dalam persamaan berikut ini.

$$W_{ij} = TF_{ij} \text{ x } IDF_{fi} \quad (2.3)$$

Dimana $W_{ij}$ merupakan bobot *term* i terhadap dokumen d.

Setelah bobot masing-masing dokumen diketahui, maka dilakukan proses *sorting*/pengurutan di mana semakin besar nilai bobot, semakin besar tingkat similaritas dokumen tersebut terhadap suatu kata yang dicari, begitu pula

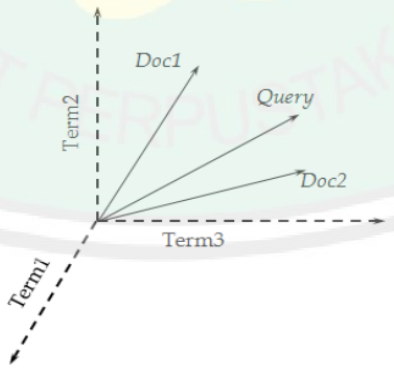
sebaliknya. Digunakan algoritma TF-IDF dikarenakan penelitian ini melakukan pembobotan berdasarkan dokumen ayat terjemahan Alquran.

## 2.5 *Similarity*

Setelah dilakukan pembobotan dalam setiap *term* dan direpresentasikan ke dalam sebuah model vektor, selanjutnya adalah proses untuk mencari kesamaan antar *term* atau proses perhitungan kesamaan antara *query* dengan dokumen.

Pengukuran yang pertama adalah *dice similarity*. *Dice Similarity* atau *Dice Coefficient* merupakan salah satu ukuran kemiripan atau kesamaan dalam *information retrieval*. Dokumen yang di-*retrieve* atau dikembalikan merupakan hasil pengukuran antara *query* dan dokumen (Chahal, 2016). *Dice coefficient* merupakan suatu formula untuk menghitung nilai kesamaan antara dua buah objek pengamatan. Bentuk persamaannya adalah sebagai berikut.

$$Dice\ Coefficient = \frac{2*|X \cap Y|}{|X| + |Y|} \quad (2.4)$$




Gambar 2.2 Representasi *dice coefficeint* dalam model ruang vektor.

Pada Gambar 2.2 tersebut, dokumen terjemahan Alquran dan *query*, yang merupakan masukan dari pengguna, direpresentasikan sebagai vektor-vektor dalam ruang vektor. Komponen-komponen dari vektor tersebut adalah bobot *term* yang

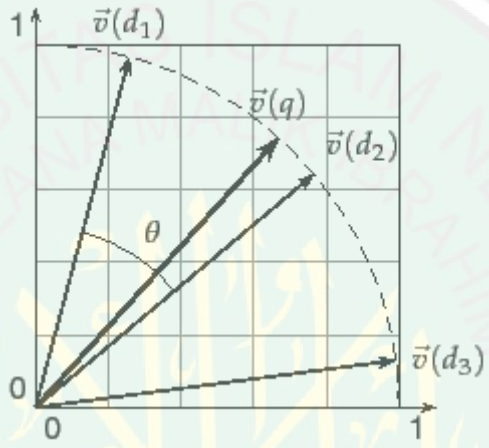
bersesuaian. *Dice similarity* merupakan panjang normalisasi dari *inner product* dari dua buah vektor. Perhitungan *Dice similarity* dapat dilakukan dengan persaman berikut ini :

$$similarity(\overrightarrow{d_j}, q) = \frac{2|\overrightarrow{d_j}.\overrightarrow{q}|}{|\overrightarrow{d_j}|^2+|\overrightarrow{q_j}|^2} = \frac{2 .\sum_{i=1}^{t}(w_{ij} .w_{iq})}{\sum_{i=1}^{t} w_{ij}^2 + \sum_{i=1}^{t} w_{iq}^2} \quad (2.5)$$

Pada persamaan diatas, $d_j$ adalah vektor dokumen yang merupakan representasi matrik dengan komponen $w_{ij}$. Sedangkan q adalah vektor *query* yang merupakan representasi matrik dengan komponen $w_{iq}$. Sehingga dari hasil perhitungan ini jika diurutkan secara *descending* akan menghasilkan dokumen terjemah Alquran yang berurutan sesuai dengan terjemah Alquran yang paling mirip dan relevan.

Evana Ainaaul Novita (2014) membuat sebuah rancang bangun *search engine* terjemahan tafsir ayat-ayat Alquran pada dokumen teks berbahasa Indonesia menggunakan metode *Dice Similirity*. *Dice Similarity* atau *Dice Coefficient* digunakan dalam mengukur kesamaan antara dokumen terjemahan tafsir Alquran dengan dokumen masukan pengguna yang akan dicocokkan dengan terjemahan tafsir tersebut. Pengukuran dilakukan dengan melakukan perhitung *precision*, *recall*, dan *accuracy* pada 30 dokumen sampel. Dari 30 sampel tersebut, rata-rata nilai *recall* yang diperoleh adalah 58.19%, *precision* sebesar 10.63% dan *accuracy* sebesar 99.76%. Nilai yang dihasilkan masih tinggi karena tingginya dokumen yang ditemukan relevan. Dari 30 dokumen yang telah diuji coba, ada 80% atau 24 dokumen yang ditemukan padanan ayatnya dan sebanyak 20% atau 6 dokumen tidak ditemukan padanan ayatnya.

Pengukuran kemiripan dokumen yang kedua adalah *cosine similarity*. Ukuran ini menghitung nilai kosinus sudut antara dua vektor. Dalam Gambar 2.3 terdapat tiga vektor dokumen d1, d2, d3 dan satu vektor *query* q. *Cosine similiarity* menghitung nilai kosinus θ dari *query* dan tiga dokumen lain. Nilai ini menunj ukkan derajat kemiripan dokumen dengan *query*.




Gambar 2.3 Representasi *cosine similarity* pada model ruang vektor.

Karena berdasarkan sudut antar dua vektor, maka nilainya berkisar pada 0 sampai dengan 1, dimana 0 menandakan bahwa kedua dokumen tidak mirip sama sekali, dan 1 menandakan bahwa antar *query* dan dokumen benar-benar identik. *Cosine similarity* dinyatakan pada persamaan 2.6 berikut.

$$\cos(d_j, q_k) = \frac{\sum_{i=1}^{n} (td_{ij} \times tq_{ik})}{\sqrt{\sum_{i=1}^{n} td_{ij}^2 \times \sum_{i=1}^{n} tq_{ik}^2}} \quad (2.6)$$

Keterangan :

| | |
|---|---|
| $\cos(d_j, q_k)$ | : tingkat kesamaan suatu dokumen dengan *query* tertentu |
| $td_{ij}$ | : *term* ke-*i* dalam vektor untuk dokumen ke-*j* |
| $tq_{ik}$ | : *term* ke-*i* dalam vektor untuk *query* ke-*k* |
| n | : jumlah *term* yang unik dalam *dataset*. |

## 2.6 *Query Expansion*

*Query expansion* (perluasan kata kunci) adalah teknik untuk memodifikasi kueri yang bertujuan memenuhi kebutuhan informasi. Modifikasi yang dilakukan pada umumnya berupa penambahan istilah ke dalam kata kunci, meskipun sebenarnya juga meliputi penyesuaian bobot dan penghapusan istilah kata kunci (Selberg, 1997). Dalam konteks web *search engine*, hal ini termasuk evaluasi *input user* dan memperluas *query* pencarian untuk mendapatkan dokumen yang cocok dengan *query*. Proses perluasan kata kunci dilakukan dengan menggunakan *wordnet* maupun tesaurus.

Kata tesaurus berasal dari kata *thesauros*, bahasa Yunani, yang bermakna 'khazanah'. Lambat laun, kata tersebut mengalami perkembangan makna, yakni 'buku yang dijadikan sumber informasi'. Tesaurus berisi seperangkat kata yang saling bertalian maknanya. Pada dasarnya, tesaurus merupakan sarana untuk mengalihkan gagasan ke dalam sebuah kata, atau sebaliknya. Oleh karena itu, lazimnya tesaurus disusun berdasarkan gagasan atau tema. Namun, untuk memudahkan pengguna dalam pencarian kata, penyusunan tesaurus pun berkembang, kini banyak tesaurus yang dikemas berdasarkan abjad. Tesaurus dibedakan dari kamus. Di dalam kamus dapat dicari informasi tentang makna kata, sedangkan di dalam tesaurus dapat dicari kata yang akan digunakan untuk mengungkapkan gagasan pengguna. Dengan demikian, tesaurus dapat membantu penggunanya dalam mengungkapkan atau mengekspresikan gagasan sesuai dengan apa yang dimaksud.

Misalnya, pencarian kata lain untuk kata hewan, pengguna tesaurus dapat mencarinya pada lema **hewan**.

**hewan** n binatang, dabat, fauna, sato, satwa.

Sederet kata yang terdapat pada lema hewan tersebut menunjukkan bahwa kata tersebut bersinonim sehingga dapat saling menggantikan sesuai dengan konteksnya. Tesaurus ini berguna dalam pengajaran bahasa sehingga dapat dimanfaatkan oleh pengajar dan pelajar.

Dalam perluasan kata kunci terdapat tiga jenis perluasan, yaitu (Rahayuni, 2011):

1. *Manual Query Expansion* (MQE)

   Dalam metode ini, pengguna memodifikasi kueri secar manual tanpa bantuan sistem.

2. *Automatic Query Expansion*(AQE)

   Metode ini akan memodifikasi kueri tanpa bantuan pengguna. Ekspansi kueri dilakukan secara otomatis melalui sistem. Menurut penelitian sebelumnya terdapat beberapa teknik yang digunakan dalam *Automatic Query Expansion*, yaitu: *Global Analysis* yang prinsip dasarnya adalah memanfaatkan konteks suatu kata untuk menentukan kesamaannya dengan kata yang lain, *Local Analysis* yang menggunakan dokumen yang di-retrieve pengguna untuk mendapatkan kueri baru, dan *Local Context Analysis* yang merupakan gabungan antara teknik analisis lokal dan analisis global.

3. *Interactive Query Expansion*(IQE)

   Metode ini membutuhkan interaksi antara pengguna dengan sistem untuk melakukan proses ekspansi kueri.

Pada penelitian yang dilakukan oleh Baiti Nur Amalia (2017) menggunakan ekspansi kueri dalam *Query Answering System* hadis *Muttafaqun 'Alaih*. Perluasan kueri dilakukan dengan mencocokkan setiap kata atau *term* dari kueri yang sudah melalui proses *preprocessing* dengan daftar gugus kata yang berkaitan pada tesaurus yang ada dalam kamus tesaurus. Kamus tesaurus tersebut mengacu pada Tesaurus Bahasa Indonesia Pusat Bahasa yang diterbitkan oleh Departemen Pendidikan Nasional. Daftar kata yang digunakan sebagai kamus tesaurus adalah kata-kata yang berkaitan dengan topik penelitian. Kemudian kata-kata tersebut disusun dalam tabel beserta dengan daftar sinonim pada gugus tesaurusnya. Hasil yang didapatkan sebelum menggunakan ekspansi adalah nilai *precision* 68.05%, *recall* 84.7%, F-*measure* 72.46%. Dan setelah menggunakan ekspansi kueri nilai *precision* naik sebesar 6.63% menjadi 74.68% , nilai *recall* naik sebesar 3.54% menjadi 88.41% dan nilai F-*measure* naik sebesar 6.27% menjadi 78.73%. Hal tersebut terjadi karena dengan proses ekspansi, kata dalam *query* dapat mengambil gugus kata dalam tesaurus yang kemungkinan ada dalam indeks database hadis. Sehingga sistem mampu me*retrieve* dokumen hadis yang relevan dengan *query* asli namun *query* asli tidak mengandung kata dalam indeks dokumen hadis atau dapat juga menaikkan posisi hadis yang paling relevan dengan *query* menjadi posisi teratas pencarian.

Dari hasil yang diperoleh pada penelitian yang dilakukan oleh Baiti (2017), ditunjukkan bahwa *query expansion* dapat menaikkan nilai *precision, recall* dan f-*measure*. Maka dari itu, peneliti akan menggunakan metode *query expansion* untuk meningkatkan kinerja sistem dalam mendapatkan dokumen yang relevan.

Perbandingan antara *dice* dan *cosine similarity* dapat dilihat kinerjanya sebelum menggunakan *query expansion* dan juga setelah menggunakan *query expansion*.

## 2.7 Uji Evaluasi *Similarity*

Evaluasi *similarity* dilakukan untuk mengukur kemampuan sistem dalam melakukan kinerjanya. Evaluasi juga dilakukan untuk mengetahui perbandingan kinerja dalam pengukuruan kemiripan *query* dengan dokumen terjemah Alquran berbahasa Indonesia antara dua metode *similarity* yaitu *dice similarity* dan *cosine similarity*.

Sistem temu kembali informasi mengembalikan sekumpulan dokumen sebagai jawaban dari *query* pengguna. Terdapat dua kategori dokumen yang dihasilkan oleh sistem temu kembali informasi terkait pemrosesan *query*, yaitu relevant documents (dokumen yang relevan dengan *query*) dan retrieved documents (dokumen yang diterima pengguna). Ukuran umum yang digunakan untuk mengukur kualitas dari data retrieval adalah kombinasi *precision* dan *recall*. *Precision* merupakan proporsi dari suatu set yang diperoleh yang relevan. *Recall* merupakan proporsi dari semua hasil yang relevan di koleksi termasuk hasil yang diperoleh atau dikembalikan. *F-measure* biasa digunakan pada bidang sistem temu kembali informasi untuk mengukur klasifikasi pencarian dokumen dan performa *query* classification. F-*measure* merupakan bobot *harmonic mean* dari *precision* dan *recall* yang merupakan ukuran timbal balik di antara keduanya.

Penelitian Alkautsar (2012) mencoba membandingkan efisiensi model ruang vektor pada sistem temu kembali informasi berdasarkan tiga ukuran koefisien kesamaan tersebut. Hasil pemrosesan dokumen menggunakan tiga koefisien tersebut menunjukkan hasil pengurutan dokumen yang tidak jauh berbeda. Masing-

masing koefisien dalam ukuran kesamaan model ruang vektor memiliki nilai yang sama untuk *recall* dan AVP (*Average Precision*). Namun, koefisien *cosine* lebih baik dibanding dengan koefisien *jaccard* dan koefisien *dice* dalam hal kompleksitas algoritma dan waktu komputasi. Perbandingan tiga koefisien tersebut juga dilakukan oleh Thada(2013) dengan membandingkan koefisien *jaccard*, *dice* dan *cosine similarity* untuk menemukan dokumen web dengan algoritma genetika. Setelah dilakukan percobaan pada *query* yang sama pada setiap koefisien dan memilih 10 halaman web pertama yang di-*retrieve* dari Google menunjukkan bahwa koefisien yang sesuai yaitu koefisien *cosine* kemudian diikuti oleh koefisien *dice* dan *jaccard*.

# BAB III

# METODOLOGI PENELITIAN

Pada bab ini akan dibahas mengenai beberapa hal, diantaranya adalah tahapan penelitian yang akan dilakukan, perancangan sistem yang akan dibuat dan penyelesaian masalah pada implementasi metode *dice similarity* dan *query expansion* dalam pencarian *ayatul ahkam* dalam terjemahan Alquran.

## 3.1 Tahapan penelitian

Adapun tahapan-tahapan yang akan dilakukan dalam penelitian ini akan direpresentasikan pada Gambar 3.1


Gambar 3.1 Tahapan penelitian

Tahapan yang pertama adalah studi literatur yang digunakan untuk mendapatkan referensi-referensi yang terkait dengan penelitian ini. Kemudian, tahapan identifikasi masalah yang sudah dijelaskan pada bab sebelumnya. Setelah itu tahap pengumpulan data. Tahap pengumpulan data berisi mengenai penjelasan data-data yang digunakan dalam penelitian beserta cara perolehan data tersebut. Tahap berikutnya adalah perancangan sistem yang menentukan alur proses sistem yang selanjutnya akan diimpelementasikan ke dalam sistem. Tahap uji evaluasi dan pembahasan berisi pengujian sistem serta pembahasan mengenai perbandingan akurasi dan relevansi hasil perangkingan dokumen yang diperoleh dengan metode *dice similarity* dan *cosine similarity*. Kemudian, tahap yang terakhir adalah analisa hasil dari penelitian ini.

## 3.2 Studi Literatur

Tahapan ini dilakukan untuk mendapatkan informasi yang berkaitan dengan lingkup pembahasa dalam penelitian, perkembangan mengenai penelitian yang terkait, serta perkembangan metode yang digunakan dalam penelitian yang terkait. Studi literatur yang dilakukan diharapkan dapat memberikan data, informasi, dan fakta mengenai pencarian ayat-ayat Alquran dengan metode *Dice Similarity* dan *Query Expansion* yang akan dikembangkan. Studi literatur yang dilakukan mencakup pencarian dan pemahaman mengenai beberapa referensi serta penelitian yang terkait hal-hal dibawah ini, yaitu :

1. *Information Retrieval System.*
2. Penambangan teks yaitu, *case folding*, *tokenizing*, *filtering* dan *stemming* kata berbahasa Indonesia.
3. Algoritma pembobotan *term* dan dokumen, TF-IDF.

4. Metode pengukuran kesamaan dokumen dengan *query* yaitu *dice similarity* dan *cosine similairity*.
5. Metode ekpasnsi *query* yang digunakan untuk memodifikasi kueri agar dapat menghasilkan dokumen yang relevan.
6. Evaluasi hasil perangkingan dokumen dengan *dice similarity* dan *cosine similarity* menggunakan perhitungan *recall* dan *precision*.

### 3.3 Pengumpulan Data

Data yang digunakan dalam penelitian ini merupakan data sekunder berupa terjemahan ayat dan surat Alquran dari Kementrian Agama Republik Indonesia. Data tersebut yang akan diolah untuk sistem ini. Dan untuk data Ayatul Ahkam diambil dari situs http://alquranalhadi.com/ . Situs tersebut dikelola oleh Pusat Kajian Hadis Al-Mughni Center Jakarta. Situs tersebut menyediakan indeks Alquran dalam bentuk tematik.

Pada data *ayatul ahkam* tersebut, terdapat indeks Alquran yang sudah dijadikan atau dikumpulkan menjadi tematik bahasan dalam Alquran. Setiap tema terdapat bab dan subbab dari penjabaran terhadap tema terkait. Pada bab dan subbab terdapat ayat Alquran yang berisi tentang tema tersebut. Data yang disajikan dibuat oleh ahli dalam bidang terkait, di dalamnya terdapat tulisan ayat Alquran dalam bahasa Arab serta terjemahan setiap ayat. Termasuk pula keterangan ayat, surat, maupun juz dari bab terkait. Sesuai dengan penelitian ini, tema yang diambil adalah Tema Syariah. Tema Syariah terdiri atas beberapa sub tema, di antaranya adalah Konsep, Fikih ibadah, Fikih muamalah, Fikih *ahwal al-syahsiyah* atau *siyasah*, Fikih *jinayah* dan Fikih kontemporer.

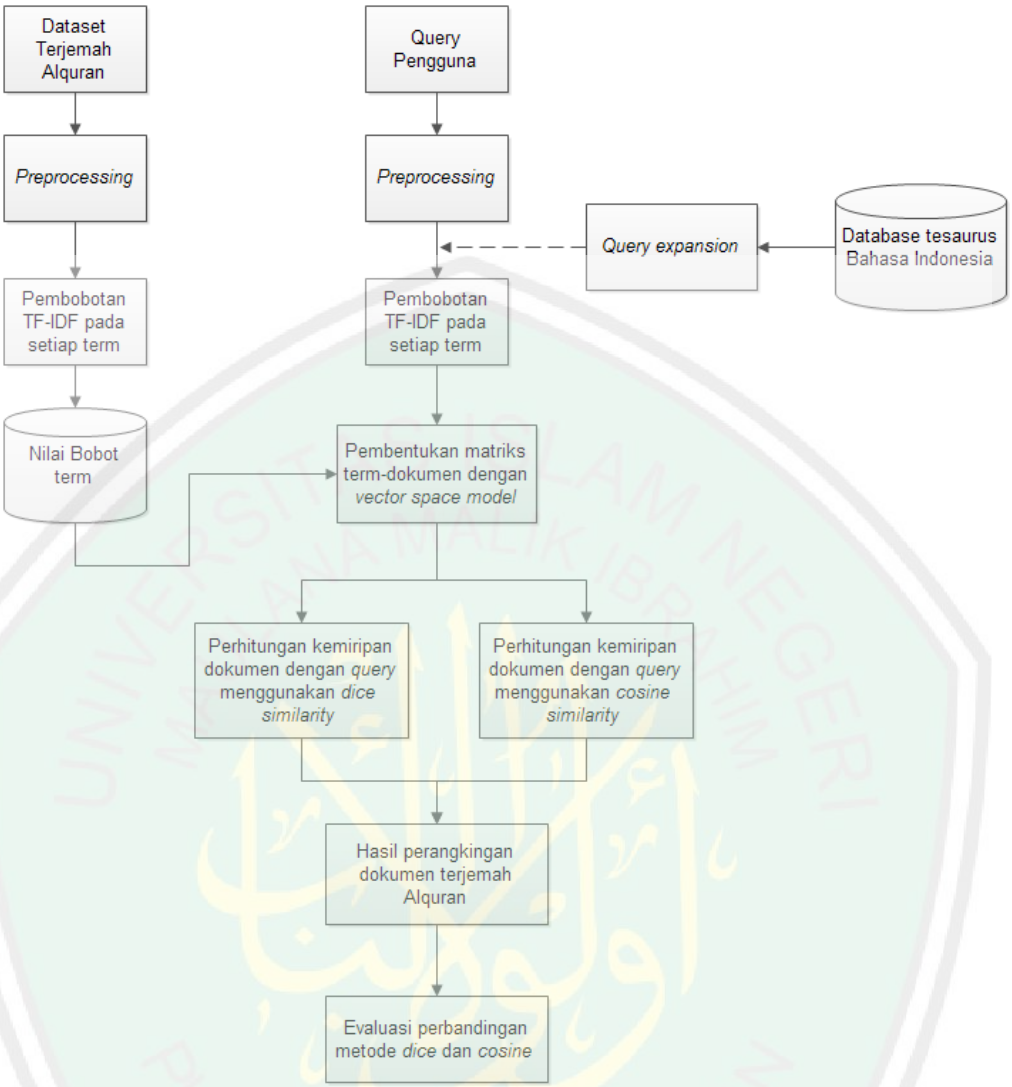
Pada setiap subtema yang telah disebutkan, terdapat beberapa bab dan subbab di dalamnya. *Ayatul ahkam* dikelompokkan secara kontekstual oleh Pusat Kajian Hadis Al-Mughni Center Jakarta berdasarkan subtema, bab dan subbab. Data subbab tersebut akan dijadikan kata kunci sebagai *input* pada penelitian ini.

Data tesaurus yang digunakan pada metode *query expansion* untuk mencari keterkaitan makna pada query diperoleh dari Kamus Tesaurus bahasa Indonesia terbitan Pusat Bahasa Departemen Pendidikan Nasional.

## 3.4 Perancangan Sistem

Bab ini menjelaskan mengenai bagaimana sistem dibuat untuk mempermudah peneliti dalam melakukan implementasi sistem, evaluasi dan analisa hasil yang didapatkan. Sistem ini akan dibangun berbasis web dengan menggunakan bahasa pemrograman *php* beserta *user interface* bagi pengguna. Perancangan sistem direpresentasikan seperti Gambar 3.2.

Terdapat beberapa proses yang dilakukan rancangan sistem yang akan dibuat. Pada bagian dokumen, proses pertama kali yang dilakukan adalah mempersiah dataset *Ayatul Ahkam*. Setelah itu *preprocessing* yang terdiri dari proses *case folding, tokenizing, filtering* dan *stemming*. *Preprocessing* akan memproses dokumen Alquran, yakni dokumen terjemah Alquran, yang menghasilkan *term-term* penting yang menjadi ciri dari masing-masing dokumen Alquran. Selanjutnya, pada setiap *term* tersebut dilakukan pembobotan dengan algoritma pembobotan TF-IDF yang menghasilkan indeks *term-term* beserta nilai bobot dalam setiap *term*. Hasil tersebut akan disimpan di dalam *database*.




Gambar 3.2 Perancangan sistem

Selanjutnya proses yang dilakukan terhadap *query* pengguna. Proses ini dimulai dari *query* masukan pengguna. Pada *query* masukan pengguna juga dilakukan *preprocessing* seperti pada bagian dokumen. Kemudian pada *query* pengguna diberikan pilihan untuk menggunakan *query expansion* ataupun tidak. Proses ini dilakukan dengan memilih gugus kata dari *database* tesaurus Bahasa Indonesia yang memiliki tingkat kemiripin yang tinggi dengan *query* tersebut. Selanjutnya dilakukan pembobotan pada setiap *term* yang dihasilkan dari *preprocessing* dan perluasan *query*. Hasilnya adalah *term* dan nilai bobot setiap

*term*. Kemudian, dibuat sebuah model ruang vektor dari hasil proses pada bagian dokumen yaitu nilai bobot *term* pada dokumen dan nilai bobot *term* dari *query* pengguna dengan atau tanpa *query expansion*. Dari model ruang vektor tersebut, *term* dokumen dapat dihitung kemiripannya dengan *term query* menggunakan dua metode perhitungan kemiripan dokumen dengan *query* yaitu *dice similarity* dan *cosine similiarity*. Sistem ini menggunakan dua metode adalah untuk mengetahui perbandingan antara kedua metode tersebut. Hasil dari proses tersebut adalah berupa dokumen Alquran yang relevan dengan *query* yang telah dimasukkan pengguna sesuai dengan urutannya. Setelah didapatkan hasil dari kedua metode tersebut, maka dilakukan sebuah perbandingan kinerja antara kedua metode tersebut menggunakan *precision, recall* dan *f-measure*.

### 3.4.1 *Preprocessing*

*Preprocessing* terdiri dari empat proses yaitu *case folding, tokenizing, filtering* dan *stemming*. Sebagai contoh terdapat dua buah dokumen yang diberi kode **D1**, **D2** dan **D3**. Seluruh dokumen berhubungan dengan hukum saja, dan difokuskan dalam perhitungan dan persamaan terhadap tesaurus. Contoh dokumen seperti pada Tabel 3.1 berikut ini :

Tabel 3.1 Contoh dokumen.

| Kode | Isi Dokumen |
|---|---|
| **D1** | Maka bersujudlah kepada Allah dan sembahlah (Dia). |
| **D2** | Sujudlah dan dekatkanlah (dirimu kepada Allah). |
| **D3** | Dan apabila Alquran dibacakan kepada mereka, mereka tidak bersujud. |

Dilakukan pencarian terhadap tiga dokumen tersebut dengan kata kunci/ *query* **Q** yaitu “Bersujud kepada Allah”, dapat dilihat pada tabel 3.2 berikut ini.

Tabel 3.2 Contoh *query* untuk pencarian.

| **Kode** | **Isi *Query*** |
|---|---|
| **Q** | Bersujud kepada Allah |

### 3.4.1.1 *Case Folding*

Pada proses ini dilakukan perubahan teks pada seluruh dokumen menjadi huruf kecil. Hasil pada proses dari tahap ini, dapat dilihat pada Tabel 3.3, dimana dari dua dokumen, keseluruhan dibuah menjadi huruf kecil sekaligus untuk mempermudah pencarian. Kemudian proses *case folding* pada kata kunci, ditunjukkan pada tabel 3.4 dan semua dirubah menjadi huruf kecil.

Tabel 3.3 Hasil *case folding* dokumen.

| **Kode** | **Isi Dokumen** |
|---|---|
| **D1** | maka bersujudlah kepada allah dan sembahlah (dia). |
| **D2** | sujudlah dan dekatkanlah (dirimu kepada allah). |
| **D3** | dan apabila alquran dibacakan kepada mereka, mereka tidak bersujud. |

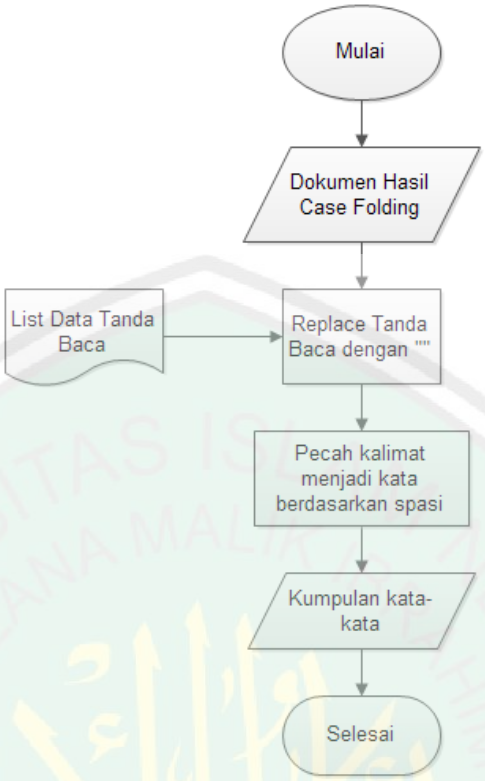
Tabel 3.4 Hasil *case folding query*.

| **Kode** | **Isi *Query*** |
|---|---|
| **Q** | bersujud kepada allah |

### 3.4.1.2 *Tokenizing*

*Tokenizing* merupakan proses menghilangkan simbol-simbol atau tanda baca pada dokumen teks, serta melakukan pemotongan data input berupa string menjadi kata-kata tunggal berdasarkan spasi, seperti pada gambar 3.3.

Setelah dilakukan perubahan menjadi huruf kecil pada proses sebelumnya, sekarang dokumen akan dipecah menjadi per kata serta dihilangkan tanda bacanya agar tidak mengganggu proses pencarian. Hasil *tokenizing* dokumen dilihat pada Tabel 3.5.




Gambar 3.3 Proses *tokenizing* dokumen.

Tabel 3.5 Hasil tokenisasi dokumen.

| | | | |
|---|---|---|---|
| 1 | maka | 9 | dekatkanlah |
| 2 | bersujudlah | 10 | dirimu |
| 3 | kepada | 11 | apabila |
| 4 | allah | 12 | alquran |
| 5 | dan | 13 | dibacakan |
| 6 | sembahlah | 14 | mereka |
| 7 | dia | 15 | tidak |
| 8 | sujudlah | 16 | bersujud |

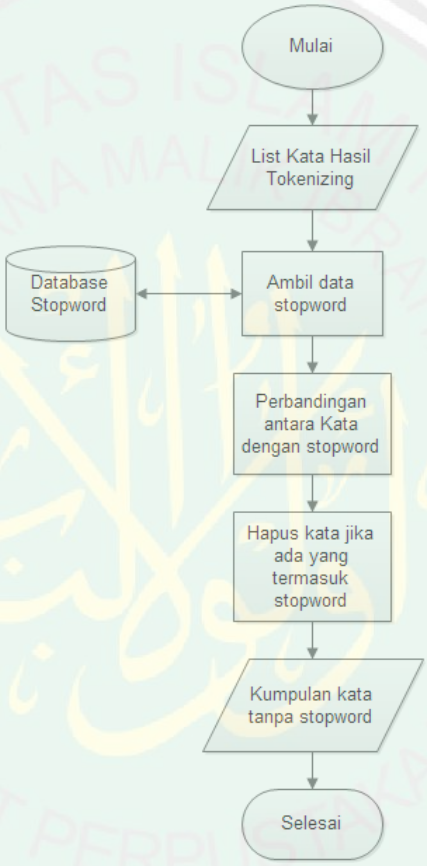
Tabel 3.6 Hasil *tokenizing query*.

| | | | |
|---|---|---|---|
| 1 | bersujud | 3 | allah |
| 2 | kepada | 4 | |

### 3.4.1.3 *Filtering*

Setelah proses tokenisasi, dilanjutkan proses *filtering* yaitu penghapusan kata-kata yang sering muncul dan tidak dipakai di dalam pemrosesan bahasa alami. Proses ini bertujuan untuk mengurangi volume kata sehingga hanya kata-kata

penting yang terdapat di dokumen. *Stopword* dapat berupa kata depan, kata penghubung, dan kata pengganti. Sehingga dalam kalimat yang tersisa hanya kata-kata penting yang siap diolah ke tahap selanjutnya, seperti pada gambar 3.4. Hasilnya dapat dilihat pada Tabel 3.7 untuk proses *filtering* dokumen dan Tabel 3.8 untuk proses *filtering* kueri.




Gambar 3.4 Proses *filtering* dokumen.

Tabel 3.7 Hasil *filtering* dokumen.

| | | | |
|---|---|---|---|
| 1 | bersujudlah | 6 | dirimu |
| 2 | allah | 7 | alquran |
| 3 | sembahlah | 8 | dibacakan |
| 4 | sujudlah | 9 | bersujud |
| 5 | dekatkanlah | 10 | |

Tabel 3.8 Hasil *filtering query.*

| | |
|---|---|
| 1 | bersujud |
| 2 | allah |

#### 3.4.1.4 *Stemming*

Proses *stemming* dilakukan setelah menghilangkan kata-kata dan simbol tidak penting. *Stemming* merupakan proses menjadikan kata-kata mengubahnya menjadi kata dasar, agar dapat diolah lebih mudah. Prosesnya adalah dengan menghilangkan imbuhan kata yang berada di depan dan berada di belakang. Imbuhan yang di depan seperti me, di, ber, men dan lain-lain. Imbuhan kata yang berada di belakang seperti kan, lah, nya, mu dan lain-lain. Proses ini menggunakan algoritma *stemming* Nazief. Proses *stemming* dilakukan seperti gambar 3.5.

Mengubah kata/*term* menjadi kata dasar agar mudah dilakukan perluasan kata kunci. Hasilnya bisa dilihat pada Tabel 3.9 untuk dokumen, dan Tabel 3.10 untuk hasil dari *query*.

Tabel 3.9 Hasil *stemming* dokumen.

| | | | |
|---|---|---|---|
| 1 | sujud | 6 | diri |
| 2 | allah | 7 | alquran |
| 3 | sembah | 8 | baca |
| 4 | sujud | 9 | sujud |
| 5 | dekat | 10 | |

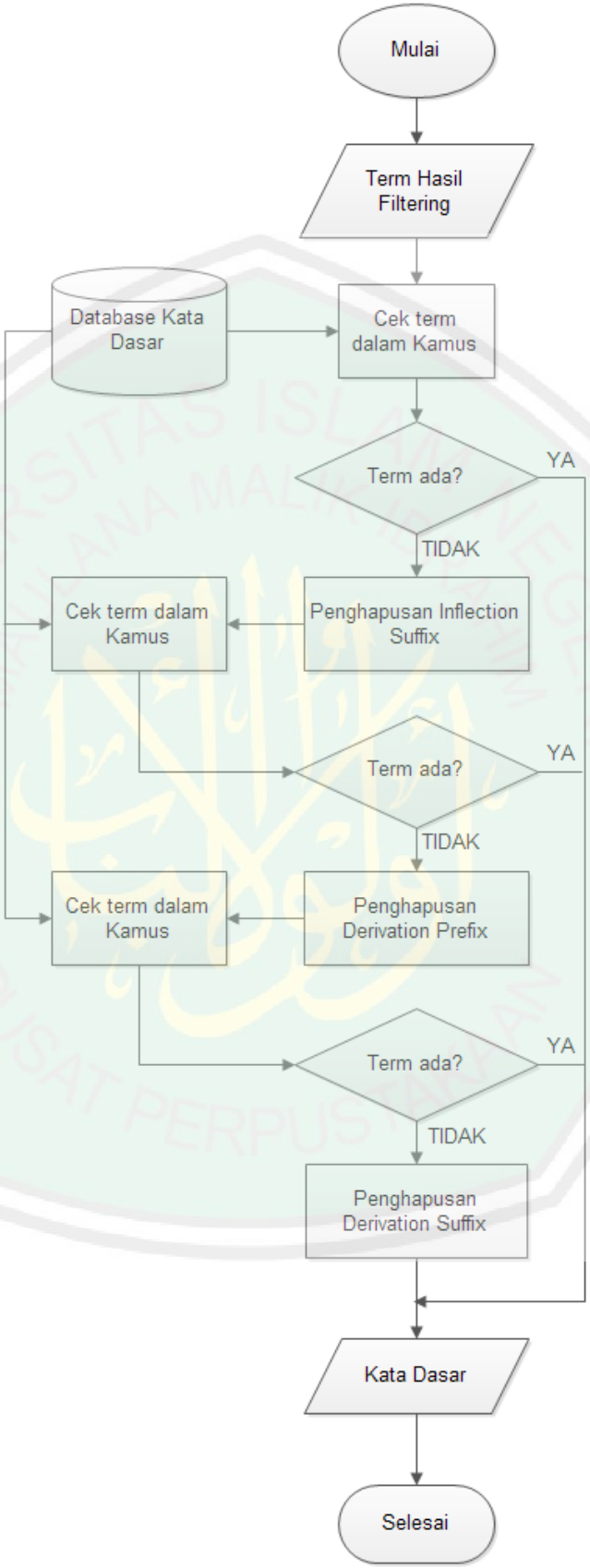
Tabel 3.10 Hasil *stemming* kata kunci.

| | |
|---|---|
| 1 | sujud |
| 2 | allah |

Hasil *preprocessing* yang diperoleh dari setiap dokumen adalah seperti ditunjukkan pada Tabel 3.11 berikut ini :

Tabel 3.11 Hasil *preprocessing* pada dokumen.

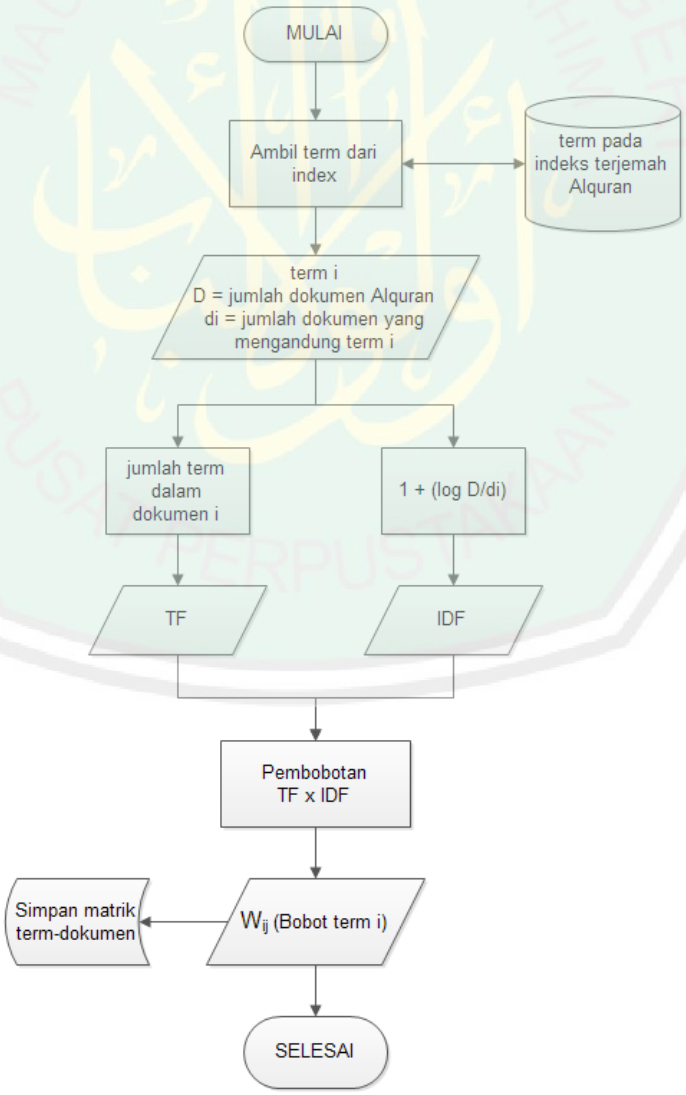
| **Kode** | **Isi Dokumen** |
|---|---|
| **D1** | sujud allah sembah |
| **D2** | sujud dekat diri allah |
| **D3** | alquran baca sujud |




Gambar 3.5 *Flowchart stemming* menggunakan algoritma Nazief.

### 3.4.3 Pembobotan TF-IDF

Setelah tahap *prepocessing*, dilakukan tahap pembobotan dengan algoritma pembobotan TF-IDF. **TF** (*term frequency*) menyatakan banyaknya suatu *term* muncul dalam sebuah dokumen. Dan DF (*document frequency*) menyatakan banyaknya dokumen yang mengandung suatu *term* dalam satu segmen. TF-IDF adalah nilai bobot dari suatu *term* yang diambil dari nilai TF dan Inverse dari DF(**IDF**). Proses pembobotan TF-IDF pada dokumen terjemah Alquran dapat dilihat pada Gambar 3.6 berikut ini. Dan Perhitungan TF-IDF terhadap hasil *preprocessing* dapat dilihat pada Tabel 3.12.




Gambar 3.6 Flowchart pembobotan TF-IDF

Tabel 3.12 Hasil perhitungan TF-IDF.

| term | Q(TF) | D1(TF) | D2(TF) | D3(TF) | DF | n/DF | IDF (log n/DF) + 1 |
|---|---|---|---|---|---|---|---|
| sujud | 1 | 1 | 1 | 1 | 3 | 1 | 1 |
| allah | 1 | 1 | 1 | 0 | 2 | 1.5 | 1.176 |
| sembah | 0 | 1 | 0 | 0 | 1 | 3 | 1.477 |
| dekat | 0 | 0 | 1 | 0 | 1 | 3 | 1.477 |
| diri | 0 | 0 | 1 | 0 | 1 | 3 | 1.477 |
| alquran | 0 | 0 | 0 | 1 | 1 | 3 | 1.477 |
| baca | 0 | 0 | 0 | 1 | 1 | 3 | 1.477 |

Dengan **n** merupakan jumlah koleksi dokumen. Lalu dihitung bobot pada masing-masing dokumen dan *query* dengan mengalikan nilai TF dan nilai IDF yang ditunjukkan pada tabel 3.13.

Tabel 3.13 Nilai bobot *query* dan masing-masing dokumen.

| term | IDF | WQ | WD1 | WD2 | WD3 |
|---|---|---|---|---|---|
| sujud | 1 | 1 | 1 | 1 | 1 |
| allah | 1.176 | 1.176 | 1.176 | 1.176 | 0 |
| sembah | 1.477 | 0 | 1.477 | 0 | 0 |
| dekat | 1.477 | 0 | 0 | 1.477 | 0 |
| diri | 1.477 | 0 | 0 | 1.477 | 0 |
| alquran | 1.477 | 0 | 0 | 0 | 1.477 |
| baca | 1.477 | 0 | 0 | 0 | 1.477 |

Dengan WQ merupakan bobot TF-IDF dari kata kunci. WD1, WD2 dan WD3 merupakan bobot dari masing-masing dokumen.

### 3.4.4 *Dice Similarity*

Setelah proses pembobotan TF-IDF selesai, terdapat sejumlah n kata yang berbeda sebagai kamus kata(*vocabulary*) atau indeks kata(*terms index*). Setiap *term* i dalam dokumen Alquran akan membentuk representasi ruang vektor yang memliki dimensi sebesar n. Pembobotan juga dilakukan pada *query*, sehingga baik dokumen maupun *query* direpresentasikan sebagai vektor berdimensi n. Koleksi

dokumen Alquran juga direpresentasi dalam model ruang vektor sebagai matriks kata-dokumen (*terms-documents matrix*). Nilai dari elemen $W_{ij}$ adalah bobot kata i dalam dokumen j. Bobot w diperoleh dari proses sebelumnya yakni proses pembobotan TF-IDF. Misalkan terdapat sekumpulan kata T sejumlah n, yaitu T= ($T_1$, $T_2$, $T_3$, ..... ,$T_n$) dan sekumpulan dokumen D sejumlah m, yaitu D = ($D_1$,$D_2$,$D_3$, .... ,$D_m$) serta $W_{ij}$ adalah bobot kata i pada dokumen j. Maka representasi matriks kata-dokumen(*terms-documents matrix*) dapat dilihat pada Gambar 3.7 berikut ini.

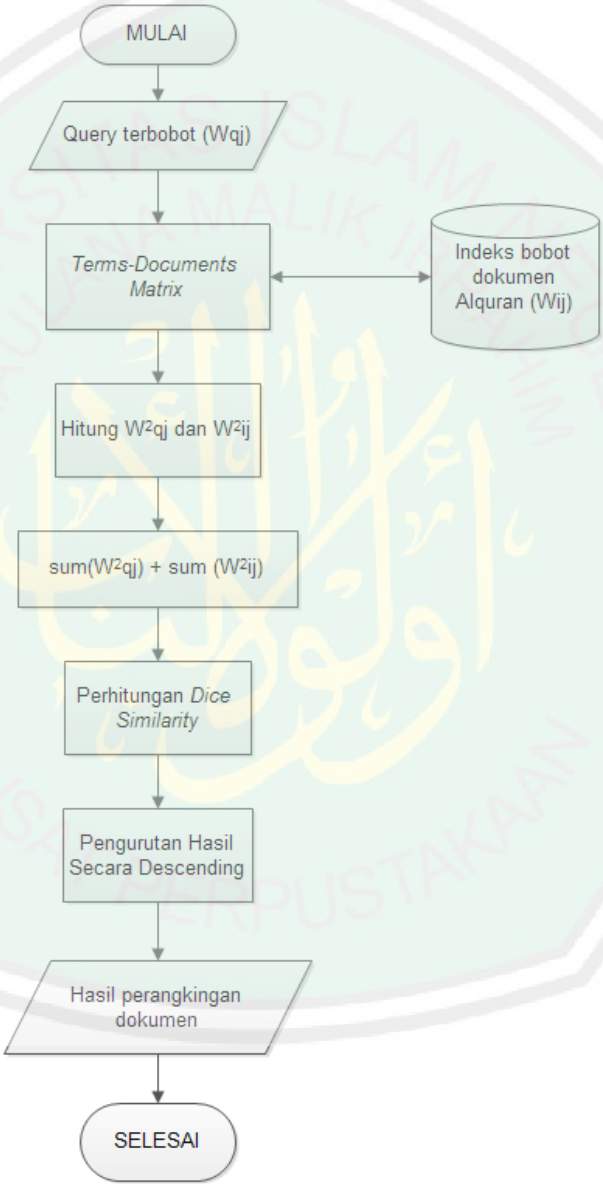
$$\begin{bmatrix} & T_1 & T_2 & \dots & T_n \\ D_1 & w_{11} & w_{21} & \dots & w_{n1} \\ D_2 & w_{12} & w_{22} & \dots & w_{n2} \\ \vdots & \vdots & & & \vdots \\ \vdots & \vdots & & & \vdots \\ D_m & w_{1m} & w_{2m} & \dots & w_{nm} \end{bmatrix}$$

Gambar 3.7 *Terms-documents matrix.*

Relevansi antara dokumen dengan *query* ditentukan sebagai pengukuran kesamaan representasi vektor dari matrik *query* dan matrik dokumen. Nilai kemiripan ditentukan dengan *dice similarity*, dengan ketentuan semakin besar nilai kemiripan yang diperoleh, maka dokumen tersebut dapat dikatakan relevan dengan *query* pengguna. Alur pengukuran kemiripan menggunakan *dice similarity* dapat dilihat pada Gambar 3.8.

*Query* terbobot adalah *query* yang telah melewati *preprocessing*, pembobotan TF-IDF dan perluasan *query* (akan dijelaskan pada sub bab selanjutnya). Indeks terbobot merupakan hasil pembobotan TF-IDF pada dokumen Alquran. Dari keduanya dibentuklah matriks *term-documents*. Setelah matriks

terbentuk, dihitung kuadrat dari setiap bobot dokumen dan bobot kueri. Kemudian dilakukan sebuah penjumlahan total kuadrat diantara kedua bobot tersebut. Setelah itu dihitung menggunakan *dice similarity*. Hasil yang diperoleh adalah dokumen yang telah terangking.




Gambar 3.8 *Flowchart* perhitungan *dice similarity*

Dari contoh dokumen sebelumnya, maka perhitungan yang diperoleh adalah seperti ditunjukkan pada Tabel 3.14.

Tabel 3.14 Hasil perhitungan *dice similarity*.

| | Q | D1 | D2 | D3 |
|---|---|---|---|---|
| Jumlah total kuadrat ($\sum W^2$) | 2.383190 | 4.565077 | 6.746965 | 5.363774 |
| Jumlah Perkalian bobot *query* dengan dokumen 2 x sim($W_q$ x $W_d$) | | 4.766381 | 4.766381 | 2 |
| Penjumlahan bobot kuadrat *query* dengan bobot kuadrat dokumen ($\sum W_q^2 + W_d^2$) | | 6.948268 | 9.130141 | 7.76965 |
| *Dice Similarity* | | 0.685981 | 0.552049 | 0.257416 |

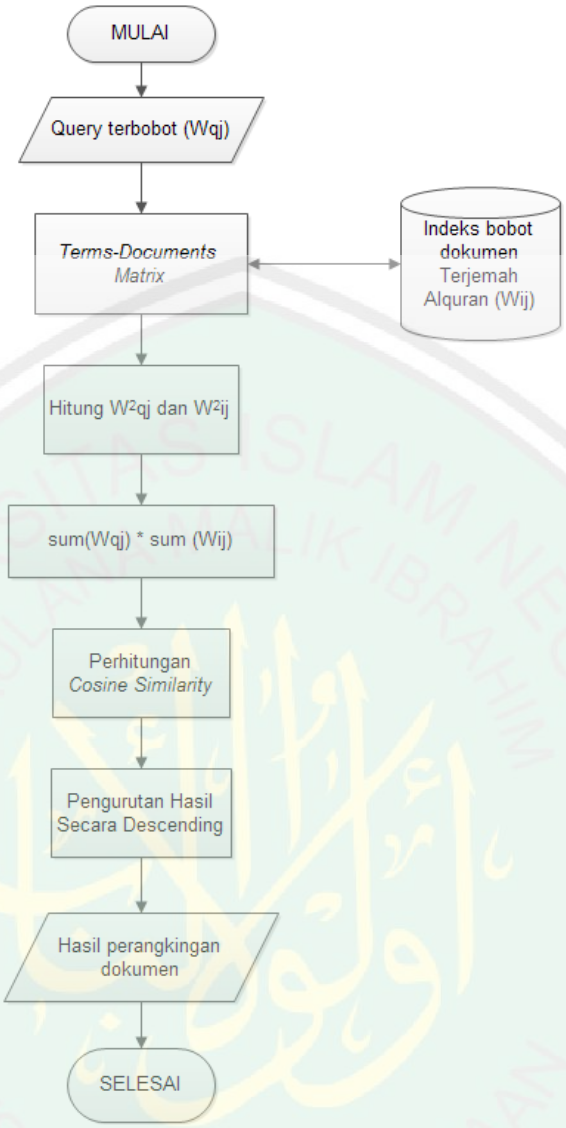
$$Dice\ similarity = \frac{2 \cdot \sum_{i=1}^{t}(w_{ij} \cdot w_{iq})}{\sum_{i=1}^{t} w_{ij}^2 + \sum_{i=1}^{t} w_{iq}^2} \quad (3.1)$$

- $Sim(d_1, q) = \frac{4.766381}{6.948268} = 0.685981$ (3.2)
- $Sim(d_2, q) = \frac{4.766381}{9.130141} = 0.552049$ (3.3)
- $Sim(d_3, q) = \frac{2}{7.76965} = 0.257416$ (3.4)

Berdasarkan hasil perhitungan tersebut, maka dokumen Alquran yang paling relevan adalah dokumen 1 dengan nilai perhitungan *dice similarity* paling besar yaitu 0.685981. Namun pada contoh diatas, *query* pengguna masih belum dilakukan perluasan kueri. *Query expansion* akan dijelaskan pada bab selanjutnya.

### 3.4.5 *Cosine Similarity*

Bobot yang didapatkan dari hasil perhitungan TF-IDF menjadi masukan untuk perhitungan kemiripan dokumen dengan *query* dengan menggunakan metode *dice similarity* dan *cosine similarity*. Pada subbab 3.2.4 telah dijelaskan mengenai perhitungan *dice similarity*, maka pada subbab ini dijelaskan mengenai perhitungan *cosine similiarity*. Proses perhitungan *cosine similarity* dijelaskan dengan *flowchart* pada gambar 3.9 berikut ini.




Gambar 3.9 *Flowchart* perhitungan *cosine similarity*.

Dari contoh dokumen yang diberikan sebelumnya, maka perhitungan *cosine similarity* menghasilkan seperti yang dijelaskan pada tabel 3.15 berikut ini.

Tabel 3.15 Hasil perhitungan *cosine similarity*.

| | **Q** | **D1** | **D2** | **D3** |
|---|---|---|---|---|
| Jumlah total kuadrat ($\sum W^2$) | 2.383190 | 4.565077 | 6.746965 | 5.363774 |
| Akar total kuadrat | 1.543758 | 2.136604 | 2.597492 | 2.315982 |
| Jumlah perkalian bobot *query* dengan bobot dokumen sum($W_q \times W_d$) | | 2,383190 | 2,383190 | 1 |
| *Cosine Similarity* | | 0,722529 | 0,594326 | 0,279695 |

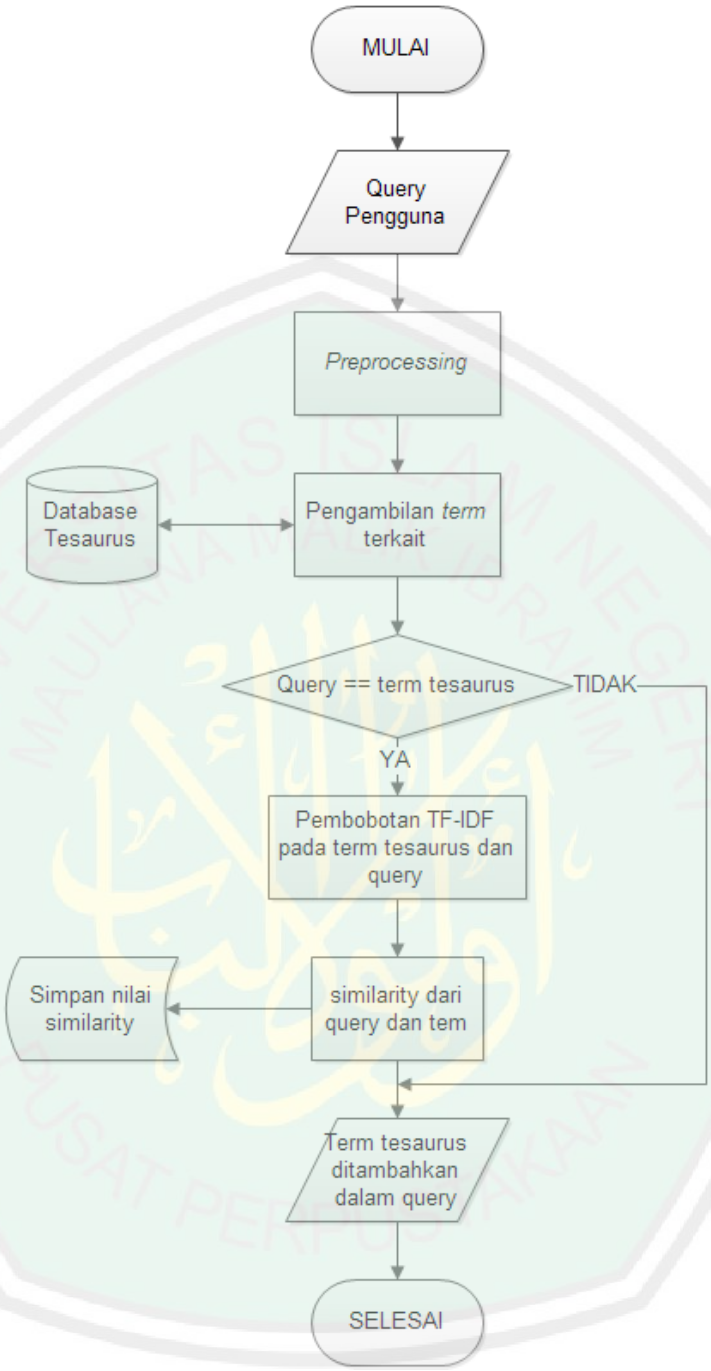
*Cosine* (Di) = sum($W_q \times W_d$) / sqrt($W_q^2$) × sqrt($W_d^2$). (3.5)

- Cos($d_1$, q) = $\frac{2.383190}{1.543758 \times 2.136604}$ = 0,722529 (3.6)
- Cos($d_2$, q) = $\frac{2,383190}{1.543758 \times 2.597492}$ = 0,594326 (3.7)
- Cos($d_3$, q) = $\frac{1}{1.543758 \times 2.315982}$ = 0,279695 (3.8)

Berdasarkan hasil perhitungan tersebut, maka dokumen Alquran yang paling relevan adalah dokumen 1 dengan nilai perhitungan *cosine similarity* paling besar yaitu 0,722529. Namun pada contoh diatas, *query* pengguna masih belum dilakukan perluasan kueri. *Query expansion* akan dijelaskan pada bab selanjutnya.

### 3.4.6 *Query Expansion*

Perluasan kueri pada tahap ini dilakukan dengan mencocokkan setiap kata kunci yang dimasukkan pengguna, dari kueri tersebut dilakukan *preprocessing* seperti halnya dilakukan pada dokumen terjemah Alquran dengan tabel tesaurus yang berada di database. Selain diambil kata yang berkaitan dengan kueri, dilakukan perhitungan pembobotan juga terhadap setiap kata tesaurus yang ada. Proses *query expansion* ditunjukkan pada Gambar 3.10. Setelah pengguna memasukkan *query*, dilakukan *preprocessing* pada *query* tersebut. Kemudian dilakukan pemanggilan *term* terkait pada *database* tesaurus. Kemudian dilakukan pencocokan antara *term* dengan *query*. Jika *term* tersebut sama dengan *query*, maka dilakukan pembobotan TF-IDF terhadap *term* tesaurus dan *query*. Kemudian dari hasil pembobotan tersebut, dilakukan perhitungan similaritas antara keduanya. Hasilnya akan disimpan ke dalam database, dan *term* tersebut dimasukkan ke dalam *query*.




Gambar 3.10 *Flowchart query expansion*

Contoh dari penerapan *query expansion* terhadap kata kunci "Perintah Melaksanakan Salat"dapat dilihat pada Tabel 3.16 berikut ini.

Tabel 3.16 Contoh *query expansion*.

| **Kueri asli** | ***Preprocessing*** | ***Query Expansion*** |
|---|---|---|
| Larangan Membunuh | larang bunuh | larang bunuh cegah haram mati binasa |

Perhitungan kemiripan gugus kata dalam tesaurus dimulai *preprocessing* kemudian proses perhitungan bobot dari setiap *term* menggunakan algoritma TF-IDF, kemudian dihitung kemiripannya menggunakan *similarity*. Term dengan nilai *similairity* terbesar nantinya akan ditambahkan pada kata kunci. Setiap satu *term* dalam kata kunci dihitung nilai *similarity*-nya. Pada Tabel 3.17 berikut ini adalah tesaurus dari kata **bunuh**. Kata bunuh memiliki tesaurus "mematikan, menewasakan, membinasakan".

Tabel 3.17 Tesaurus dari kata bunuh.

| Kode | Gugus Kata dalam Tesaurus | Kata |
|---|---|---|
| **D1** | Melenyapkan Membinasakan Bunuh Menewaskan Melumpuhkan | Mematikan |
| **D2** | Melenyapkan Membasmi Memusnahkan Bunuh | Menewaskan |
| **D3** | Mematikan Membinasakan Bunuh Melenyapkan Menghancurkan. | Membinasakan |

Kemudian dilakukan *preprocessing* pada setiap dokumen sama halnya pada contoh dokumen Alquran yaitu *case folding, tokenizing, filtering* dan *stemming*.

**a. *Case Folding***

Langkah awal adalah proses *case folding* pada dokumen tesaurus tersebut. Hasilnya dapat dilihat pada Tabel 3.18 berikut ini.

Tabel 3.18 *Case folding* pada dokumen tesaurus.

| Kode | Gugus Kata dalam Tesaurus | Kata |
|---|---|---|
| **D1** | melenyapkan membinasakan membunuh menewaskan melumpuhkan | mematikan |
| **D2** | melenyapkan membasmi memusnahkan membunuh | menewaskan |
| **D3** | mematikan membinasakan membunuh melenyapkan menghancurkan | membinasakan |

**b. *Tokenizing***

Kemudian proses *tokenizing* yaitu proses pemecahan setiap kata dan penghilangan tanda baca pada dokumen tesaurus yang ditunjukkan pada Tabel 3.19 berikut ini.

Tabel 3.19 *Tokenizing* pada dokumen tesaurus.

| | | | |
|---|---|---|---|
| 1 | melenyapkan | 6 | membasmi |
| 2 | membinasakan | 7 | memusnahkan |
| 3 | membunuh | 8 | mematikan |
| 4 | menewaskan | 9 | menghancurkan |
| 5 | melumpuhkan | | |

**c. *Filtering***

Setelah itu, dilakukan proses *filtering* untuk menghapus kata-kata tidak penting yang tidak dipakai pada pemrosesan bahasa alami. Hasilnya ditunjukkan pada Tabel 3.20 berikut ini.

Tabel 3.20 *Filtering* pada dokumen tesaurus.

| | | | |
|---|---|---|---|
| 1 | melenyapkan | 6 | membasmi |
| 2 | membinasakan | 7 | memusnahkan |
| 3 | membunuh | 8 | mematikan |
| 4 | menewaskan | 9 | menghancurkan |
| 5 | melumpuhkan | | |

**d. *Stemming***

Terakhir pada tahap preprocessing adalah proses stemming yakni mengubah kata menjadi kata dasar. Proses ini menggunakan algoritma stemming Nazief. Hasilnya ditunjukkan pada Tabel 3.21 berikut.

Tabel 3.21 *Stemming* pada dokumen tesaurus.

| | | | |
|---|---|---|---|
| 1 | lenyap | 6 | basmi |
| 2 | binasa | 7 | musnah |
| 3 | bunuh | 8 | mati |
| 4 | tewas | 9 | hancur |
| 5 | lumpuh | | |

**e. Pembobotan TF-IDF**

Dari contoh dokumen tesaurus yang digunakan, terdapat tiga dokumen. Tiga dokumen tersebut berasal dari kata kunci "membunuh" yang memiliki tesaurus "mematikan menewaskan membinasakan". Dilakukan proses pembobotan pada setiap *term* dalam tiga dokumen tersebut. Hasilnya ditunjukkan pada Tabel 3.22 berikut ini.

Tabel 3.22 Pembobotan TF-IDF pada dokumen tesaurus.

| term | Q(tf) | D1(tf) | D2(tf) | D3(tf) | df | n/df | Idf (log n/df) + 1 |
|---|---|---|---|---|---|---|---|
| lenyap | 0 | 1 | 1 | 1 | 3 | 1 | 1 |
| binasa | 1 | 1 | 0 | 1 | 2 | 1,5 | 1,176 |
| bunuh | 0 | 1 | 1 | 1 | 3 | 1 | 1 |
| tewas | 1 | 1 | 0 | 0 | 1 | 3 | 1,477 |
| lumpuh | 0 | 1 | 0 | 0 | 1 | 3 | 1,477 |
| basmi | 0 | 0 | 1 | 0 | 1 | 3 | 1,477 |
| musnah | 0 | 0 | 1 | 0 | 1 | 3 | 1,477 |
| mati | 1 | 0 | 0 | 1 | 1 | 3 | 1,477 |
| hancur | 0 | 0 | 0 | 1 | 1 | 3 | 1,477 |

Kemudian dihitung bobot antara dokumen tesaurus dan kata dengan mengalikan TF dan IDF. Hasilnya ditunjukkan pada Tabel 3.23 berikut.

Tabel 3.23 Perhitungan bobot dokumen dan kata pada tesaurus.

| term | IDF | WQ | WD1 | WD2 | WD3 |
|---|---|---|---|---|---|
| lenyap | 1 | 0 | 1 | 1 | 1 |
| binasa | 1,176 | 1,176 | 1,176 | 0 | 1,176 |
| bunuh | 1 | 0 | 1 | 1 | 1 |
| tewas | 1,477 | 1,477 | 1,477 | 0 | 0 |
| lumpuh | 1,477 | 0 | 1,477 | 0 | 0 |
| basmi | 1,477 | 0 | 0 | 1,477 | 0 |
| musnah | 1,477 | 0 | 0 | 1,477 | 0 |
| mati | 1,477 | 1,477 | 0 | 0 | 1,477 |
| hancur | 1,477 | 0 | 0 | 0 | 1,477 |

### f. Perhitungan *Similarity*

Setelah ditemukan hasil dari pembobotan dari setiap dokumen dan kata, maka dihitung besar *similarity*-nya. Hasil perhitungannya ditunjukkan pada Tabel 3.24 berikut ini.

Tabel 3.24 Hasil perhitungan *similarity* tesaurus.

| | **Q** | **D1** | **D2** | **D3** |
|---|---|---|---|---|
| Jumlah total kuadrat ($\sum W^2$) | 5,747 | 7,747 | 6,364 | 7,747 |
| Akar total kuadrat | 2,397 | 2,783 | 2,523 | 2,783 |
| Jumlah perkalian bobot *query* dengan bobot dokumen sum($W_q \times W_d$) | | 3,565 | 0 | 3,565 |
| *Cosine Similarity* | | 0,534 | 0 | 0,508 |

Berdasarkan hasil perhitungan *similarity* yang dilakukan terhadap tiga dokumen tesaurus tersebut, dokumen satu (D1) dan dokumen tiga(D3) merupakan dokumen dengan nilai *similarity* terbesar. Karena yang ditambahkan adalah dua *term* dengan kemiripan terbesar, maka *term* tesaurus yang ditambahkan ke kata kunci adalah *term* mematikan (D1) dan membinasakan (D3). Kata kuncinya menjadi "Larangan Membunuh Mematikan Membinasakan".

## 3.5 Metode Pengujian Sistem

Sistem yang akan dibangun ini termasuk dalam kategori *information retrieval system*. Diharapkan sistem ini dapat memberikan hasil berupa Ayatul Ahkam yang relevan, sesuai dengan *query* yang dimasukkan pengguna. terdapat dua kategori dokumen yang dihasilkan oleh sistem IR terkait pemrosesan *query*, yaitu *relevant document* (dokumen yang relevan dengan kueri pengguna) dan *retrieved document* (dokumen yang diterima pengguna).

Ukuran umum yang digunakan untuk mengukur kualitas dari *text retrieval* adalah kombinasi *precision* dan *recall*. *Precision* mengevaluasi kemampuan sistem IR untuk menemukan kembali *top-ranked* yang paling relevan, dan didefinisikan sebagai presentase dokumen yang di-*retrieve* yang benar-benar relevan terhadap *query* pengguna. *Recall* mengevaluasi kemampuan sistem IR untuk menemukan semua *item* yang relevan dari dalam koleksi dokumen dan difenisikan sebagai presentase dokumen yang relevan terhadap *query* pengguna dan yang diterima.

Uji coba dilakukan dengan melakukan pencarian dan dilanjutkan dengan mengevaluasi hasil pencarian yang dihasilkan oleh sistem dengan menggunakan metode *dice similarity* dan *cosine similarity*. Pengujian dilakukan terhadap sejumlah input pengguna, mulai uji coba ke-1 sampai ke-n. Dari masing-masing uji coba dapat diperoleh nilai akurasi, *precision*, dan *recall* untuk nilai evaluasi kemampuan sistem. Pada Tabel 3.25 berikut ini ditunjukkan perumusan matriks terkenal oleh Lancaster (1979) sebagai ukuran *precision* dan *recall*.

Tabel 3.25 *Precision* dan *Recall.*

| | **Relevan** | **Tidak Relevan** | **Total** |
|---|---|---|---|
| Ditemukan | a | b | a + b |
| Tidak Ditemukan | c | d | c + d |
| Total | a + c | c + d | a + b + c + d |

Berdasarkan Tabel 3.16 diatas, perumusan *recall* dan *precision* menjadi :

*Recall* = {a / (a+c)} x 100% (3.5)

*Precision* = {a / (a+b)} x 100% (3.6)

Selanjutnya, f-*measure* juga digunakan pada penelitian ini sebagai perhitungan evaluasi pada sistem yang dibangun. F-*measure* merupakan salah satu

perhitungan evaluasi dalam *information retrieval system* yang mengkombinasikan *precision* dan *recall*. F-*measure* merupakan bobot *harmonic mean* dari *precision* dan *recall* yang merupakan ukuran timbal balik di antara keduanya. Berikut ini adalah persamaan yang digunakan untuk menghitung f-*measure.*

$$F - measure = 2 \text{ x } \frac{precision\ x\ recall}{precision+recall} \quad (3.7)$$

Setelah ditemukan hasil *precision* dan *recall* dari perangkingan dokumen yang diperoleh dan juga bobot harmonic *f-measure*. Maka dibandingkan hasil perangkingan dokumen terjemahan Alquran menggunakan metode *dice similarity* dengan hasil perangkingan dokumen terjemahan Alquran menggunakan metode *cosine similarity*. Uji coba sistem menggunakan 10 kata kunci yang digunakan pada setiap metode *dice similarity* atau *cosine similarity* dengan atau tanpa menggunakan *query expansion*. 10 kata kunci tersebut diperoleh dari subbab dari bab pada subtema *ayatul ahkam*. Jadi, setiap metode perhitungan kemiripan dokumen dengan *query* akan diketahui perbandingan keakuratannya dengan menggunakan perluasan kueri maupun tidak menggunakannya. Acuan pengujian yang dipakai pada penelitian ini adalah penelitian yang dilakukan oleh Thada(2013) tentang perbandingan koefisien *cosine, dice* dan *jaccard* dalam menemukan dokumen web menggunakan algoritma genetika. Penelitian tersebut menggunakan 10 kata kunci untuk menemukan dokumen web.

# BAB IV

# UJI COBA DAN PEMBAHASAN

Pada bab ini akan dibahas mengenai implementasi dari setiap langkah-langkah uji coba sistem sesuai dengan skenario pengujian yang telah dibahas pada bab sebelumnya. Kemudian, akan dijelaskan mengenai hasil pengujian serta pembahasan mengenai hasil pengujian tersebut. Kemudian akan dipaparkan integrasi antara penelitian ini dengan islam.

## 4.1 Langkah Uji Coba

Pengujian pada sistem yang dibangun pada penelitian ini dilakukan untuk mengetahui tingkat akurasi sistem. Sistem dibangun dengan menggunakan dua metode yakni *dice similarity* dan *cosine similarity* dengan atau tanpa metode *query expansion*. Langkah-langkah tersebut dapat diuraikan sebagai berikut :

### 4.1.1 *Preprocessing* Dokumen

Semua data yang digunakan meliputi dokumen Alquran dan *ayatul ahkam*, indeks dokumen, vektor dan cache disimpan dalam sebuah DBMS MySQL.

Sebelum dilakukan proses pengindeksan dokumen Alquran, terlebih dahulu dilakukan *preprocessing* dokumen. Tahapan ini terdiri dari *case folding*, *tokenizing*, *filtering* dan *stemming*. Untuk penjelasan dari masing-masing proses telah dipaparkan di bab sebelumnya. *Preprocessing* yang dilakukan terhadap 6238 ayat Alquran menghasilkan 50.330 *term*. Contoh sebagian dari hasil *preprocessing* dokumen dapat dilihat pada Tabel 4.1 berikut ini.

Tabel 4.1 Hasil *preprocessing*.

| id | term | docId |
|---|---|---|
| 1 | tunjuk | 5 |
| 2 | jalan | 5 |
| 3 | lurus | 5 |
| 4 | jalan | 6 |
| 5 | orang | 6 |
| 6 | engkau | 6 |
| 7 | nikmat | 6 |
| 8 | murka | 6 |
| 9 | sesat | 6 |

### 4.1.2 Pembobotan TF.IDF

Tahap selanjutnya adalah pembuatan indeks dokumen Alquran dan perhitungan bobot dari masing-masing term yang didapatkan menggunakan algoritma TF.IDF. Setiap *term* yang diperoleh dari *preprocessing* disimpan dalam sebuah tabel indeks. Kemudian dilakukan pembobotan pada masing-masing *term* tersebut. Sesuai dengan algoritma yang digunakan, proses pembobotan dimulai dengan menghitung TF (*term frequency*). Perhitungan TF dilakukan dengan menghitung jumlah frekuensi kemunculan setiap *term* terhadap masing-masing dokumen. Hasil dari proses perhitungan TF dapat dilihat pada Tabel 4.2 berikut.

Tabel 4.2 Hasil dari sebagian perhitungan TF.

| id | term | docId | tf |
|---|---|---|---|
| 1 | tunjuk | 5 | 1 |
| 2 | jalan | 5 | 1 |
| 3 | lurus | 5 | 1 |
| 4 | jalan | 6 | 3 |
| 5 | orang | 6 | 2 |
| 6 | engkau | 6 | 1 |
| 7 | nikmat | 6 | 1 |
| 8 | murka | 6 | 1 |
| 9 | sesat | 6 | 1 |

Perhitungan IDF dilakukan dengan menghitung persebaran tiap *term* dalam seluruh dokumen yang ada. Jadi, satu *term* akan dihitung persebarannya pada 6238 ayat Alquran. Kemudian dihitung antara TF IDF untuk menghasilkan bobot dari setiap *term*. Hasil perhitungan bobot setiap *term* dapat dilihat pada Tabel 4.3 berikut ini.

Tabel 4.3 Pembobotan *term* dalam dokumen.

| id | term | docId | tf | tf x idf |
|---|---|---|---|---|
| 1 | tunjuk | 5 | 1 | 4,190669232978 |
| 2 | jalan | 5 | 1 | 3,9551031616653 |
| 3 | lurus | 5 | 1 | 5,4591805584415 |
| 4 | jalan | 6 | 3 | 11,865309484996 |
| 5 | orang | 6 | 2 | 4,0166231339643 |
| 6 | engkau | 6 | 1 | 3,3306182194267 |
| 7 | nikmat | 6 | 1 | 4,6732516444306 |
| 8 | murka | 6 | 1 | 6,4777501394361 |
| 9 | sesat | 6 | 1 | 4,5101000037444 |

### 4.1.3 Perhitungan dengan *Dice Similarity* dan *Cosine Similarity*

Setelah dilakukan proses pemberian bobot terhadap masing-masing *term* dokumen, proses selanjutnya adalah menghitung kemiripan dengan dua metode yaitu *dice similarity* dan *cosine similarity*. Sebelum melakukan perhitungan kemiripan, Kata kunci yang dimasukkan oleh pengguna juga dilakukan proses seperti pada proses pengindeksan dokumen yakni *preprocessing* dan perhitungan bobot kata kunci. Dan juga dilakukan perhitungan panjang vektor dari setiap dokumen. Hasil dari perhitungan vektor setiap dokumen dapat dilihat pada Tabel 4.4 berikut ini.

Tabel 4.4 Panjang vektor setiap dokumen.

| docId | panjang |
|---|---|
| 0 | 9,34805 |
| 1 | 8,76857 |
| 2 | 7,36377 |
| 3 | 6,56349 |
| 4 | 10,6504 |
| 5 | 7,93771 |
| 6 | 15,8793 |
| 7 | 9,9533 |
| 8 | 9,54549 |
| 9 | 12,7707 |
| 10 | 14,4022 |

Kemudian dilakukan perhitungan kemiripan antara dokumen dengan kata kunci menggunakan *dice similarity* dan *cosine similarity*. Contoh kata kunci yang digunakan adalah "Perintah Melaksanakan Salat". Hasil yang diperoleh dari perhitungan *dice similarity* dapat dilihat pada Tabel 4.5 dan untuk perhitungan *cosine similarity* ditunjukkan pada Tabel 4.6 berikut ini.

Tabel 4.5 Hasil perhitungan *dice similarity*.

| id | query | docId | similarity |
|---|---|---|---|
| 1 | perintah laksana salat | 228 | 0,16266252820195 |
| 2 | perintah laksana salat | 244 | 1,2845939206836 |
| 3 | perintah laksana salat | 245 | 0,60662444154841 |
| 4 | perintah laksana salat | 283 | 0,68863355369654 |
| 5 | perintah laksana salat | 285 | 0,27597348791766 |
| 6 | perintah laksana salat | 331 | 0,44899123488255 |
| 7 | perintah laksana salat | 389 | 0,19731781315518 |
| 8 | perintah laksana salat | 405 | 0,32771284189158 |
| 9 | perintah laksana salat | 444 | 0,19991697269261 |
| 10 | perintah laksana salat | 464 | 0,30592260925725 |

Tabel 4.6 Hasil perhitungan *cosine similarity*.

| id | query | docId | similarity |
|---|---|---|---|
| 1 | perintah laksana salat | 228 | 0,1007922752391 |
| 2 | perintah laksana salat | 244 | 0,19831002394713 |
| 3 | perintah laksana salat | 283 | 0,21286130267513 |

| | | | |
|---|---|---|---|
| 4 | perintah laksana salat | 285 | 0,17100432447696 |
| 5 | perintah laksana salat | 331 | 0,13878623635718 |
| 6 | perintah laksana salat | 389 | 0,12226609012508 |
| 7 | perintah laksana salat | 444 | 0,12387663439968 |
| 8 | perintah laksana salat | 464 | 0,18956201022424 |
| 9 | perintah laksana salat | 475 | 0,12501497540701 |
| 10 | perintah laksana salat | 552 | 0,15090562756109 |

### 4.1.4 *Query Expansion* pada Kata Kunci

Proses ini adalah proses tambahan untuk membandingkan hasil perangkingan data dengan menggunakan *query expansion* atau tanpa menggunakannya. Proses ini berfungsi untuk menambahkan *term* yang terkait dengan kata kunci. *Term* yang terkait didapatkan dari tesaurus Bahasa Indonesia. Contoh dari tabel daftar tesaurus dapat dilihat pada Tabel 4.7 berikut ini.

Tabel 4.7 Daftar sebagian tesaurus.

| id | term | tesaurus |
|---|---|---|
| 1 | jujur | benar bersih mukhlis tulus |
| 2 | benar | adil betul jujur lurus tepat |
| 3 | bersih | lurus mukhlis murni tulus jujur bening |
| 4 | mukhlis | bersih etis ikhlas jujur kredibel |
| 5 | tulus | ikhlas jujur rela sudi bersih mukhlis |
| 6 | takar | kadar ukuran timbangan |
| 7 | timbangan | ukuran takaran neraca |
| 8 | kadar | kemampuan kuasa ketentuan kodrat |
| 9 | ukuran | kadar takar ukuran |
| 10 | neraca | timbangan |

Langkah awal dalam proses ini adalah dengan menghitung kemiripan antar gugus kata dalam tiap *term* dalam daftar tesaurus. Proses ini berguna untuk mempercepat perluasan kata kunci. Ketika pengguna memasukkan kata kunci, kata kunci tersebut akan langsung dicocokkan dengan daftar gugus kata yang memiliki kemiripan terbesar. Proses perhitungan kemiripan tersebut hampir sama dengan proses perhitungan kemiripan antara kata kunci dengan dokumen. Dimulai dengan

proses perhitungan bobot setiap *term* dan tesaurusnya, kemudian menghitung panjang vektor setiap gugus kata dalam tesaurus dan dihitung kemiripannya menggunakan perhitungan *cosine similarity*. *Term* tesaurus yang ditambahkan adalah 2 *term* dari gugus tesaurus dengan kemiripan terbesar. Hasil perhitungan tersebut dapat dilihat pada Tabel 4.8 berikut ini.

Tabel 4.8 Hasil perhitungan *similarity* tesaurus.

| id | query | tesaurus | similarity |
|---|---|---|---|
| 1 | jujur | jujur | 0,999999 |
| 2 | jujur | benar | 0,249308 |
| 3 | jujur | bersih | 0,130285 |
| 4 | jujur | mukhlis | 0,0367468 |
| 5 | jujur | tulus | 0,130285 |
| 6 | takar | takar | 1 |
| 7 | takar | kadar | 0,0279747 |
| 8 | takar | ukuran | 0,726084 |
| 9 | takar | timbangan | 0,25279 |
| 10 | timbangan | timbangan | 0,999998 |
| 11 | timbangan | ukuran | 0,143886 |
| 12 | timbangan | takaran | 0 |
| 13 | timbangan | neraca | 0,439769 |

Misalkan kata kunci yang dimasukkan adalah “Perintah untuk jujur dalam takaran” maka menjadi “perintah jujur takar suruh bersih timbang ukur”.

## 4.2 Hasil Uji Coba

Berdasarkan langkah uji coba yang telah dijelaskan pada subbab sebelumnya, pada subbab ini akan dipaparkan lingkup uji coba dan hasil uji coba.

### 4.2.1 Lingkup Uji Coba

Proses uji coba sistem dilakukan pada komputer dengan spesifikasi intel *Pentium Inside* dengan RAM DDR3 2 GB dan sistem operasi Windows 10. Sistem dibangun menggunakan bahasa pemrograman PHP 7.2.1, DBMS MySQL dan *web server* Apache. Dengan spesifikasi komputer tersebut dapat melakukan proses

pengindeksan dokumen Alquran selama kurang lebih 14 jam tanpa henti dan proses perhitungan kemiripan antara kata kunci dengan dokumen selama kurang lebih setengah jam tiap kata kunci. Spesifikasi komputer yang lebih bagus dapat mempercepat proses tersebut.

### 4.2.2 Hasil

Proses pengujian sistem dilakukan dengan 10 kata kunci berdasarkan subbab pada dokumen *ayatul ahkam*. Berdasarkan kata kunci tersebut, dapat dihitung dan dievaluasi tingkat akurasi sistem dengan menggunakan metode *dice similarity* dan *cosine similarity* dengan atau tanpa menggunakan *query expansion*. Evaluasi sistem didapatkan dari nilai *precision, recall* dan akurasi. Daftar kata kunci dapat dilihat pada Tabel 4.9 berikut ini .

Tabel 4.9 Daftar kata kunci yang digunakan.

| NO | Kata Kunci |
|---|---|
| 1 | Manusia Diperintahkan untuk Taat kepada Hukum Allah |
| 2 | Perintah Melaksanakan Salat |
| 3 | Perintah untuk Jujur dalam Takaran |
| 4 | Perkawinan Dua Jenis (Laki dan Perempuan) Membuahkan Kelanjutan Kehidupan |
| 5 | Perintah Berbakti Kepada Kedua Orang Tua |
| 6 | Larangan Membunuh |
| 7 | Celaan Buat Mereka yang Mengingkari Janji |
| 8 | Larangan Berzina |
| 9 | Perintah Bertasbih |
| 10 | Jual Beli yang paling Merugi |

Data *ayatul ahkam* yang digunakan sebagai acuan kerelevanan dokumen yang diambil dari situ http://alquranalhadi.com berdasarkan kata kunci yang digunakan dapat dilihat pada Tabel 4.10 berikut ini.

Tabel 4.10 Data *ayatul ahkam* yang dijadikan acuan.

| **Kata Kunci** | **Nomor Surat : Ayat** |
|---|---|
| Manusia Diperintahkan untuk Taat kepada Hukum Allah | 24:54, 3:32, 4:59, 47:33, 8:20 |
| Perintah Melaksanakan Salat | 2:43, 2:110, 2:83, 4:77, 4:103, 6:72, 10:87, 11:114, 14:31, 17:78, 20:14, 22:78, 24:56, 29:45, 30:31, 31:17, 33:33, 58:13, 73:20, 98:5 |
| Perintah untuk Jujur dalam Takaran | 6:152, 7:85, 11:84, 11:85, 17:35, 55:8, 55:9 |
| Perkawinan Dua Jenis (Laki dan Perempuan) Membuahkan Kelanjutan Kehidupan | 4:1, 11:40, 13:3, 16:72, 23:27, 49:13, 51:49, 53:45, 75:39 |
| Perintah Berbakti Kepada Kedua Orang Tua | 29:8, 31:14, 46:15 |
| Larangan Membunuh | 4:29, 6:151, 17:33, 25:68, 40:28 |
| Celaan Buat Mereka yang Mengingkari Janji | 2:100, 3:77, 7:102, 8:56, 9:77, 13:25, 20:86 |
| Larangan Berzina | 17:32, 60:12 |
| Perintah Bertasbih | 110:3, 15:98, 20:130, 25:5, 33:42, 48:9, 50:40, 52:48, 52:49, 56:74, 56:96, 69:52 |
| Jual Beli yang paling Merugi | 2:16, 2:41, 2:79, 2:90, 2:174, 2:175, 3:77, 3:187, 5:44, 16:95, 31:6 |

Proses untuk menampilkan rangking dokumen dilakukan berdasarkan nilai *similarity* yang lebih dari nilai *threshold* atau nilai ambang batas. Dokumen Alquran yang nilainya lebih dari *threshold* dinyatakan mirip sedangkan yang kurang dari *threshold* dinyatak tidak mirip. Nilai *threshold* yang dipakai adalah 0.20. Hasil dari perhitungan menggunakan metode *dice similarity* tanpa *query*

*expansion* ditunjukkan pada Tabel 4.11 dan Tabel 4.12 untuk *cosine similarity* tanpa *query expansion*.

Tabel 4.11 Hasil perhitungan ranking data menggunakan *dice similarity*.

| NO | JR | R | TR | TD | T | P | R | F |
|---|---|---|---|---|---|---|---|---|
| 1 | 384 | 4 | 380 | 1 | 5 | 0,01041667 | 0,8 | 0,0205655 |
| 2 | 206 | 20 | 186 | 0 | 20 | 0,0970874 | 1 | 0,1769911 |
| 3 | 68 | 5 | 63 | 2 | 7 | 0,0735294 | 0,7142857 | 0,1333333 |
| 4 | 224 | 6 | 218 | 3 | 9 | 0,0267857 | 0,6666667 | 0,0515021 |
| 5 | 119 | 3 | 116 | 0 | 3 | 0,0252101 | 1 | 0,0491803 |
| 6 | 95 | 5 | 90 | 0 | 5 | 0,0526316 | 1 | 0,1 |
| 7 | 250 | 5 | 245 | 2 | 7 | 0,02 | 0,7142857 | 0,0389105 |
| 8 | 44 | 1 | 43 | 1 | 2 | 0,0227273 | 0,5 | 0,0434783 |
| 9 | 157 | 11 | 146 | 2 | 13 | 0,0700637 | 0,8461538 | 0,1294118 |
| 10 | 98 | 9 | 89 | 2 | 11 | 0,0918367 | 0,8181818 | 0,1651376 |
| **Total** | | | | | | 0,4902885 | 8,059573 | 0,9085106 |
| **Rata-rata** | | | | | | 0,04902885 | 0,8059573 | 0,09085106 |

Tabel 4.12 Hasil perhitungan ranking data menggunakan *cosine similarity*.

| NO | JR | R | TR | TD | T | P | R | F |
|---|---|---|---|---|---|---|---|---|
| 1 | 121 | 4 | 117 | 1 | 5 | 0,0330578 | 0,8 | 0,0634921 |
| 2 | 105 | 17 | 88 | 3 | 20 | 0,1619048 | 0,85 | 0,272 |
| 3 | 21 | 5 | 16 | 2 | 7 | 0,2380952 | 0,7142857 | 0,3571428 |
| 4 | 59 | 3 | 56 | 6 | 9 | 0,0508474 | 0,3333333 | 0,0882353 |
| 5 | 36 | 2 | 34 | 1 | 3 | 0,0555556 | 0,6666667 | 0,1025641 |
| 6 | 46 | 2 | 44 | 3 | 5 | 0,0434782 | 0,4 | 0,0784314 |
| 7 | 79 | 3 | 76 | 4 | 7 | 0,0379747 | 0,4285714 | 0,0697674 |
| 8 | 23 | 1 | 22 | 1 | 2 | 0,0434783 | 0,5 | 0,08 |
| 9 | 66 | 11 | 55 | 2 | 13 | 0,1666667 | 0,8461538 | 0,2784810 |
| 10 | 31 | 5 | 26 | 6 | 11 | 0,1612903 | 0,4545454 | 0,2380952 |
| **Total** | | | | | | 0,9923491 | 5,9935564 | 1,6282093 |
| **Rata-rata** | | | | | | 0,09923491 | 0,59935564 | 0,16282093 |

Keterangan *header* tabel : JR = Jumlah dokumen yang didapatkan
R = Dokumen yang relevan
TR = Dokumen yang tidak relevan
TD = Dokumen yang tidak ditemukan.
T = Total relevan
P = *Precision*
R = *Recall*
F = *F-measure*.

Hasil perhitungan menggunakan metode *dice similarity* adalah nilai *precision* sebesar 4,902%, nilai *recall* sebesar 80,596% dan nilai *f-measure* sebesar 9,085%. Sedangkan pada perhitungan menggunakan *cosine similarity*, nilai *precision*,*recall* dan *f-mesaure* masing-masing sebesar 9,923%, 59,935% dan 16,282%.

Kemudian untuk hasil perhitungan menggunakan metode *dice similarity* tanpa *query expansion* dapat dilihat pada Tabel 4.13 dan *cosine similarity* dengan *query expansion* pada Tabel 4.14.

Tabel 4.13 Hasil *dice similarity* dengan menggunakan *query expansion*.

| NO | JR | R | TR | TD | T | P | R | F |
|---|---|---|---|---|---|---|---|---|
| 1 | 291 | 4 | 287 | 1 | 5 | 0,0137457 | 0,8 | 0,0270270 |
| 2 | 183 | 20 | 163 | 0 | 20 | 0,1092896 | 1 | 0,1970443 |
| 3 | 83 | 5 | 78 | 2 | 7 | 0,0602409 | 0,7142857 | 0,1111111 |
| 4 | 185 | 9 | 176 | 0 | 9 | 0,0486486 | 1 | 0,0927835 |
| 5 | 102 | 3 | 99 | 0 | 3 | 0,0294118 | 1 | 0,0571428 |
| 6 | 95 | 5 | 90 | 0 | 5 | 0,0526316 | 1 | 0,1 |
| 7 | 209 | 6 | 203 | 1 | 7 | 0,0287081 | 0,8571428 | 0,0555556 |
| 8 | 93 | 1 | 92 | 1 | 2 | 0,0107527 | 0,5 | 0,0210526 |
| 9 | 164 | 11 | 153 | 2 | 13 | 0,0670732 | 0,8461538 | 0,1242938 |
| 10 | 92 | 9 | 82 | 2 | 11 | 0,0978261 | 0,8181818 | 0,1747572 |
| **Total** | | | | | | 0,5183283 | 8,5357642 | 0,9607681 |
| **Rata-rata** | | | | | | 0,05183283 | 0,85357642 | 0,09607681 |

Tabel 4.14 Hasil *cosine similarity* dengan *query expansion*.

| NO | JR | R | TR | TD | T | P | R | F |
|---|---|---|---|---|---|---|---|---|
| 1 | 71 | 4 | 67 | 1 | 3 | 0,0563380 | 1,3333333 | 0,1081081 |
| 2 | 105 | 20 | 85 | 0 | 20 | 0,1904761 | 1 | 0,32 |
| 3 | 30 | 5 | 25 | 2 | 7 | 0,1666667 | 0,7142857 | 0,2702703 |
| 4 | 33 | 3 | 30 | 6 | 9 | 0,0909091 | 0,3333333 | 0,1428571 |
| 5 | 26 | 1 | 25 | 2 | 3 | 0,0384615 | 0,3333333 | 0,0689655 |
| 6 | 73 | 3 | 70 | 2 | 5 | 0,0410959 | 0,6 | 0,0769231 |
| 7 | 47 | 6 | 41 | 1 | 7 | 0,1276596 | 0,8571428 | 0,2222222 |
| 8 | 31 | 1 | 22 | 1 | 2 | 0,0322581 | 0,5 | 0,0606061 |
| 9 | 60 | 11 | 49 | 2 | 13 | 0,1833333 | 0,8461538 | 0,3013699 |
| 10 | 78 | 6 | 66 | 5 | 11 | 0,0769231 | 0,5454545 | 0,1348315 |
| **Total** | | | | | | 1,0041214 | 7,0630369 | 1,7061537 |
| **Rata-rata** | | | | | | 0,10041214 | 0,70630369 | 0,17061537 |

Hasil perhitungan menggunakan metode *dice similarity* dengan *query expansion* adalah nilai *precision* sebesar 5,183%, nilai *recall* sebesar 85, 357 % dan nilai *f-measure* sebesar 9,607%. Sedangkan pada perhitungan menggunakan *cosine similarity* dengan *query expansion*, nilai *precision,recall* dan *f-mesaure* masing-masing sebesar 10,041%, 70,630% dan 17,061%.

### 4.3 Pembahasan

Berdasarkan hasil uji coba yang telah dikemukakan pada bab sebelumnya, diketahui bahwa dengan menambahkan *query expansion* pada setiap metode dapat meningkatkan hasil *retrieve* ayat Alquran yang sesuai dengan kata kunci pengguna. Hal ini ditunjukkan pada persentase nilai *recall* dari kedua metode yaitu sebesar 85,357% untuk metode *dice similairity* dan 70,630% untuk metode *cosine similiraty*. Nilai *recall* menunjukkan bahwa sistem mampu me-*retrieve* ayat Alquran yang relevan dengan kata kunci masukan.

Metode *dice similarity* memiliki keunggulan dalam pengembalian dokumen yang relevan dengan kata kunci masukan (*recall*) dibandingkan dengan metode *cosine similarity* baik dengan menggunkan *query expansion* maupun tidak dengan menggunakananya. Hal ini dikarenakan terdapat nilai *threshold* sebesar 0.2 yang digunakan sebagai batas nilai *similarity* sehingga pada metode *cosine similarity* terdapat beberapa dokumen yang relevan dengan kata kunci, namun nilai similarity dibawah dari nilai *threshold*. Perbandingan nilai *recall* kedua metode dengan atau tanpa menggunakan *query expansion* dapat dilihat pada Tabel 4.15 berikut ini.

Tabel 4.15 Perbandingan nilai persentase *recall* setiap metode.

| **Metode** | ***Recall*** |
|---|---|
| *Dice* | 80,596% |
| *Cosine* | 59,935% |
| *Dice dengan Query Expansion* | 85,357% |
| *Cosine dengan Query Expansion* | 70,630% |

Persentase nilai *recall* dari kedua metode tersebut, *dice* dan *cosine*, menunjukkan peningkatan jika menggunakan *query expansion*. Pada metode *dice* meningkat sebesar 4,761% , dari 80,596% menjadi 85,357% dan pada metode meningkat sebesar 10,695% , dari 59,935% menjadi 70,630%. Pada setiap percobaan kata kunci, nilai *recall*-nya sebagian besar mengalami kenaikan. Namun pada kata kunci ke-9 yaitu "Perintah Bertasbih" tidak terdapat peningkatan sama sekali, baik pada metode *dice* maupun *cosine*. Karena pada gugus kata tesaurus bahasa Indonesia belum ada arti yang cocok dengan makna tasbih sendiri.

Dari percobaan yang telah dilakukan, metode *dice similarity* selalu me-*retrieve* dokumen Alquran dengan jumlah yang lebih besar jika dibandingkan dengan metode *cosine similarity*. Dengan tambahan *query exapansion*, jumlah dokumen yang di-*retrieve* oleh kedua metode menjadi lebih menurun. Hal ini ditunjukkan dengan nilai persentase *precision* pada masing-masing metode. Nilai *precision* menunjukkan kemampuan sistem dalam ketepatan me-*retrieve* ayat Alquran yang relevan dari semua ayat yang tar-*retrieve* oleh sistem. Perbandingan nilai *precision* kedua metode dengan atau tanpa menggunakan *query expansion* dapat dilihat pada Tabel 4.16 berikut ini.

Tabel 4.16 Perbandingan nilai *precision* setiap metode.

| **Metode** | ***Precision*** |
|---|---|
| *Dice* | 4,902% |
| *Cosine* | 9,923% |
| *Dice dengan Query Expansion* | 5,183% |
| *Cosine dengan Query Expansion* | 10,041% |

Dengan menggunakan *query expansion*, nilai persentase dari kedua metode selalu mengalami peningkatan. Pada metode *dice similarity* meningkat sebesar 0,281% dari 4,902% menjadi 5,183% dan pada metode *cosine similariy* meningkat sebesar 0,118% dari 9,923% menjadi 10,041%. Metode *cosine similarity* memliki kelebihan dalam hal *precision*, baik dengan menggunakan *query expansion* mapan tanpa menggunakannya. Karena metode *cosine similarity* pada setiap pengujiannya selalu me-*retrieve* dokumen ayat Alquran yang lebih sedikit dibandingkan dengan metode *cosine similarity*.

Kemudian untuk perbandingan nilai bobot harmonik atau *f-measure* antara kedua metode ditunjukkan pada Tabel 4.17 berikut ini.

Tabel 4.17 Perbandingan nilai *f-measure* setiap metode.

| **Metode** | ***f-measure*** |
|---|---|
| *Dice* | 9,085% |
| *Cosine* | 16,282% |
| *Dice dengan Query Expansion* | 9,607% |
| *Cosine dengan Query Expansion* | 17,061%. |

Metode *cosine similarity* memiliki keunggulan dalam nilai persentase bobot harmonik atau *f-measure* dibandingkan dengan metode *dice similarity*. Hal ini disebabkan adanya jarak nilai persentase yang cukup besar pada kemampuan sistem kemampuan sistem dalam ketepatan me-*retrieve* ayat Alquran yang relevan dari semua ayat yang ter-*retrieve* oleh sistem.

Dengan demikian, perbandingan antara nilai *recall, precision* dan *f-measure* dari setiap pengujian kata kunci di antara beberapa metode tersebut ditunjukkan pada Tabel 4.18 berikut ini.

Tabel 4.18 Perbandingan nilai *recall, precision* dan *f-measure*.

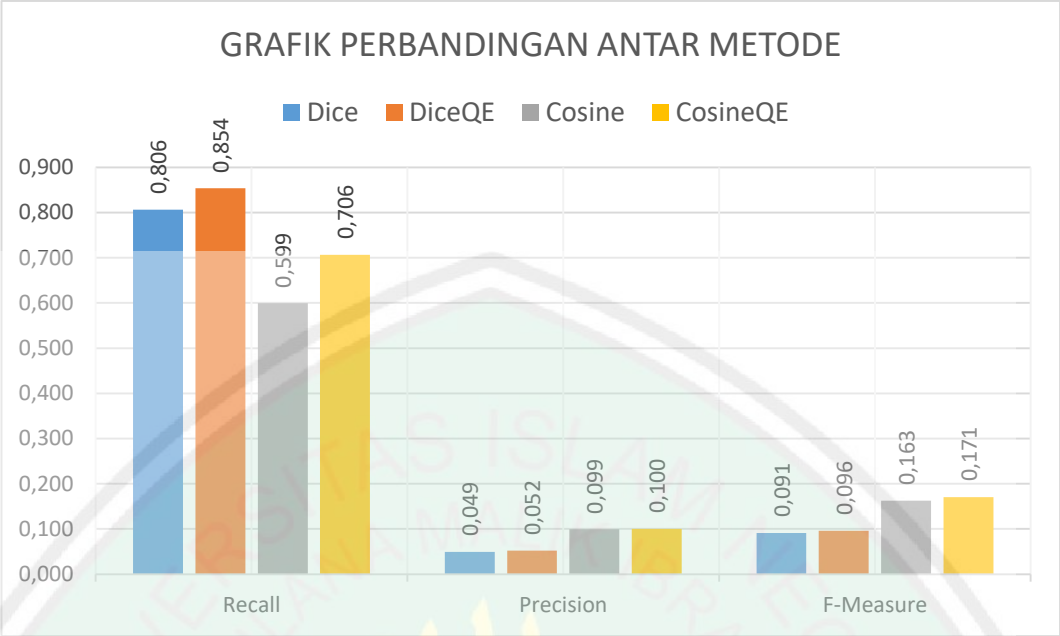
| NO | Recall | | | | Precision | | | | F-Measure | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Dice | Cosine | DiceQE | CosineQE | Dice | Cosine | DiceQE | CosineQE | Dice | Cosine | DiceQE | CosineQE |
| 1 | 0,800 | 0,800 | 0,800 | 1,333 | 0,010 | 0,033 | 0,014 | 0,056 | 0,021 | 0,063 | 0,027 | 0,108 |
| 2 | 1,000 | 0,850 | 1,000 | 1,000 | 0,097 | 0,162 | 0,109 | 0,190 | 0,177 | 0,272 | 0,197 | 0,320 |
| 3 | 0,714 | 0,714 | 0,714 | 0,714 | 0,074 | 0,238 | 0,060 | 0,167 | 0,133 | 0,357 | 0,111 | 0,270 |
| 4 | 0,667 | 0,333 | 1,000 | 0,333 | 0,027 | 0,051 | 0,049 | 0,091 | 0,052 | 0,088 | 0,093 | 0,143 |
| 5 | 1,000 | 0,667 | 1,000 | 0,333 | 0,025 | 0,056 | 0,029 | 0,038 | 0,049 | 0,103 | 0,057 | 0,069 |
| 6 | 1,000 | 0,400 | 1,000 | 0,600 | 0,053 | 0,043 | 0,053 | 0,041 | 0,100 | 0,078 | 0,100 | 0,077 |
| 7 | 0,714 | 0,429 | 0,857 | 0,857 | 0,020 | 0,038 | 0,029 | 0,128 | 0,039 | 0,070 | 0,056 | 0,222 |
| 8 | 0,500 | 0,500 | 0,500 | 0,500 | 0,023 | 0,043 | 0,011 | 0,032 | 0,043 | 0,080 | 0,021 | 0,061 |
| 9 | 0,846 | 0,846 | 0,846 | 0,846 | 0,070 | 0,167 | 0,067 | 0,183 | 0,129 | 0,278 | 0,124 | 0,301 |
| 10 | 0,818 | 0,455 | 0,818 | 0,545 | 0,092 | 0,161 | 0,098 | 0,077 | 0,165 | 0,238 | 0,175 | 0,135 |
| Rerata | 0,806 | 0,599 | 0,854 | 0,706 | 0,049 | 0,099 | 0,052 | 0,100 | 0,091 | 0,163 | 0,096 | 0,171 |

Kemudian pada setiap pengujian berdasarkan kata kunci, dihitung nilai rata-rata dari setiap metode. Tabel perbandingan rata-rata dari nilai *recall, precision* dan *f-measure* dapat dilihat pada Tabel 4.19.

Tabel 4.19 Perbandingan nilai rata-rata *recall, precision* dan *f-measure*.

| | Dice | Cosine | DiceQE | CosineQE |
|---|---|---|---|---|
| Recall | 0,806 | 0,599 | 0,854 | 0,706 |
| Precision | 0,049 | 0,099 | 0,052 | 0,100 |
| F-Measure | 0,091 | 0,163 | 0,096 | 0,171 |

Dan grafik perbandingan antara beberapa metode berdasarkan nilai recall, precision dan f-measure dapat dilihat pada Gambar 4.1 berikut ini.




Gambar 4.1 Grafik perbandingan nilai evaluasi setiap metode.

## 4.4 Integrasi dengan Islam

Umat Islam diperintahkan oleh Allah SWT untuk selektif dan kritis ketika mendapatkan informasi. Hal ini sesuai dengan firman Allah SWT pada surat *Al-Hujurat* ayat 6.

يَـٰٓأَيُّهَا ٱلَّذِينَ ءَامَنُوٓاْ إِن جَآءَكُمْ فَاسِقٌۢ بِنَبَإٍ فَتَبَيَّنُوٓاْ أَن تُصِيبُواْ قَوْمًۢا بِجَهَـٰلَةٍ فَتُصْبِحُواْ عَلَىٰ مَا فَعَلْتُمْ نَـٰدِمِينَ ٦

*Artinya : Hai orang-orang yang beriman, jika datang kepadamu orang fasik membawa suatu berita, maka periksalah dengan teliti agar kamu tidak menimpakan suatu musibah kepada suatu kaum tanpa mengetahui keadaannya yang menyebabkan kamu menyesal atas perbuatanmu itu.*

Pada ayat diatas, disebutkan bahwasanya manusia diperintahkan untuk memeriksa satu informasi dengan teliti agar terhindar dari musibah yang menyebabkan manusia menyesal dengan perbuatannya. Sesuai dengan penelitian ini, dilakukan perbandingan dua metode untuk diukur tingkat relevansi dan

keakuratannya. Kedua metode tersebut diteliti agar didapatkan metode yang terbaik untuk pencarian data dalam kumpulan dokumen.

Segala sesuatu di dunia ini memiliki takaran-takaran atau ukuran-ukurannya sendiri. Begitu juga metode *similarity* yang memiliki ukuran tersendiri dalam perhitungan kemiripannya. Sesuai dengan firman Allah SWT dalam surat Al-Furqon ayat 2, yang berbunyi :

ٱلَّذِى لَهُۥ مُلْكُ ٱلسَّمَٰوَٰتِ وَٱلْأَرْضِ وَلَمْ يَتَّخِذْ وَلَدًا وَلَمْ يَكُن لَّهُۥ شَرِيكٌ فِى ٱلْمُلْكِ وَخَلَقَ كُلَّ شَىْءٍ فَقَدَّرَهُۥ تَقْدِيرًا ۝

*Artinya : yang kepunyaan-Nya-lah kerajaan langit dan bumi, dan Dia tidak mempunyai anak, dan tidak ada sekutu bagi-Nya dalam kekuasaan(Nya), dan dia telah menciptakan segala sesuatu, dan Dia menetapkan ukuran-ukurannya dengan serapi-rapinya*.

Metode *dice similarity* memiliki ukuran tersendiri dalam perhitungan kemiripan, begitu juga metode *cosine similarity*. Karena itu pada penelitian ini dilakukan perbandingan antara kedua metode tersebut untuk dihitung tingkat akurasi serta relevansi yang dihasilkan. Kemudian diperoleh hasil mana yang terbaik diantara kedua metode tersebut. Setelah diperoleh hasilnya, maka hasil penelitian akan menunjukkan metode yang terbaik sebagai pembelajaran dalam hal perangkingan data.

# BAB V

# PENUTUP

Pada bab ini akan dijelaskan mengenai penarikan kesimpulan berdasarkan hasil yang diperoleh dari penelitian ini dan saran terhadap penelitian yang dilakukan untuk pengembangan penelitian ini.

## 5.1 Kesimpulan

Berdasarkan hasil uji coba penelitian dan pembahasan mengenai perbandingan metode *dice similarity* dengan *cosine similarity* dengan atau tanpa menggunakan *query expansion*, dapat ditarik kesimpulan bahwa pencarian *ayatul ahkam* yang dilakukan secara tekstual dengan metode *dice similarity* dengan menggunakan *query expansion* menghasilkan nilai persentase keberhasilan sistem yaitu nilai *recall* sebesar 85,357% terhadap 10 kata kunci yang diujikan. 10 kata kunci tersebut mewakili setiap *ayatul ahkam* yang dikelompokkan secara kontekstual. Perbandingan antara metode *dice similarity* dengan *cosine similarity* tanpa menggunakan *query expansion*, didapatkan bahwa metode *dice similarity* memiliki hasil nilai *recall* sebesar 80,596%, nilai *precision* sebesar 4,902% dan nilai *f-measure* 9,085%, sedangkan perhitungan metode *cosine similarity* menghasilkan nilai *recall* 59,935%, nilai *precision* sebesar 9,923% dan nilai *f-measure* sebesar 16,282%. Perbadingan antara metode *dice similarity* dan *cosine* dengan menggunakan *query expansion*, didapatkan bahwa metode *dice* memiliki nilai *recall* sebesar 85,357%, nilai *precision* sebesar 5,183% dan nilai *f-measure* sebesar 9,607%, sedangkan metode *cosine* memiliki hasil nilai *recall* 70,630%,

nilai *precision* sebesar 10,041% dan nilai *f-measure* sebesar 17,061%. Dengan demikian, metode *dice similarity* dengan *query expansion* memiliki keunggulan dalam *retrieve* dokumen yang relevan dengan kata kunci masukan (*recall*) dibandingkan dengan metode lainnya, sedangkan metode *cosine similarity* dengan *query expansion* memiliki keunggulan dalam hal ketepatan me-*retrieve* ayat Alquran yang relevan dari semua ayat yang tar-*retrieve* oleh sistem*(precision)* dan keunggulan dalam nilai bobot harmonik atau *f-measure*. Berdasarkan uji evaluasi sistem yang telah dijelaskan pada bab sebelumnya, maka metode *cosine similarity* dengan menggunakan *query expansion* merupakan metode terbaik diantara metode lainnya karena unggul dalam dua faktor penilaian evaluasi kinerja sistem.

## 5.2 Saran

Setelah dilakukan berbagai kegiatan dalam penelitian ini, terdapat beberapa saran yang mungkin berguna untuk dapat mengembangkan penelitian ini. Berikut ini adalah saran-saran tersebut :

1. Menggunakan beberapa metode untuk mengetahui keterkaitan kata selain *query expansion* untuk mengetahui keterkaitan istilah-istilah dalam agama islam.
2. Menggunakan metode *similarity* lain untuk perbandingan.
3. Data yang digunakan bukan hanya terjemah saja namun Tafsir Alquran.
4. Menggunakan metode komputasi paralel pada saat pengindeksan dokumen tidak memakan terlalu lama.

## DAFTAR PUSTAKA


Afuan, L. (2013). Stemming Dokumen Teks Bahasa Indonesia Menggunkan Algoritma Porter. *Jurnal Telematika*, Vol. 6 No. 2.

Alkautsar, A. (2012). *Perbandingan Efisiensi Model Ruang Vektor pada Sistem Temu Kembali Informasi.* Bogor: Institut Pertanian Bogor.

Aziz, A. R. (2015). Implementasi Vector Space Model dalam Pembangkitan Frequently Asked Question Otomatis dan Solusi yang Relevan untuk Keluhan Pelanggan. *Scientific Journal of Informatics*, Vol.2, hlm.111-122.

Baiti, N. A. (2017). *Query Answering System Hadis Muttafaqun 'Alaih Menggunakan Metode Dice Similarity dan Thesaurus Based Query Expansion.* Malang: UIN Maulana Malik Ibrahim Malang.

Bunyamin, H. (2008). *Aplikasi Information Retrieval (IR) CATA Dengan Metode Generalized Vector Space Model.* Bandung: Universitas Kristen Maranatha.

Chahal, M. (2016). Information Retrieval using Dice Similarity Coefficient. *International Journal of Advanced Research in Computer Sciencne and Software Engineering*, Vol.6, hlm.72-75.

Evana, A. N. (2014). *Rancang Bangun Seacrh Engine Terjemahan Tafsir Ayat-ayat Al Quran pada Dokumen Teks Berbahasa Indonesia Menggunakan Dice Similarity.* Malang: Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim Malang.

Feldman, R. &. (2007). *The Text Mining Handbook.* New York: Cambridge University.

Indranandita, A. d. (2008). *Sistem klasifikasi dan pencarian jurnal dengan mbnggunakan metode naive bayes dan vector space model.* Yogyakarta: Universitas Kristen Duta Wacana.

Kaltsum, L. U., & Moqsith, A. (2015). *Tafsir Ayat-Ayat Ahkam.* Jakarta: UIN Press Syarif Hidayatullah Jakarta.

Lancaster, F. (1979). *Information Retrieval Systems: Characteristics, Testing, and Evaluation, 2nd Edition.* New York: John Willey.

Mandala, R. d. (2002). *Improving Information Retrieval System Performance by Automatic Query Expansion.* Bandung: Institut Teknologi Bandung.

Manning, C. D. (2008). *Introduction to Information Retrieval.* Cambridge: Cambridge University Press.

Maulana, D. (2014). *Question Answering System Berbasis Clustering pada Buku Pedoman PTIIK Dengan Menggunakan Algoritma Levenshtein Distance.* Malang: Repositori Jurnal Mahasiswa PTIIK UB.

Muharrom, M. A. (2017). *Pencarian Ayat Berdasarkan Tematik Dalam Terjemahan Alquran Menggunakan Metode Vector Space Model Dan Ekspansi Query*. Malang: UIN Maulana Malik Ibrahim Malang.

Murad, A. M. (2007). Word Similarity for Document Grouping using Soft Computing. *IJCSNS (International Journal of Computer Science and Network Security) Vol.7*, 20-28.

Qiu, Y. d. (1993). Concept-based query expansion. *SIGIR*.

Raymond, J. M. (2006). *Machine Learning Text Categorization.* Austin: University of Texas.

Selberg, E. (1997). *Information retrieval advances using relevance feedback.* Washington: University of Washington Department of Computer Science and Engineering General Exam.

Thada, V. (2013). Comparison of Jaccard, Dice, Cosine Similarity Coefficient to Find Best Fitness Value for Web Retrieve Documents using Genetic Algorithm. *Interational Journal of Innovations in Engineering and Technology*, Vol.2, hlm.202-205.

Weiss, S. I. (2005). *Text Mining : Predictive Methods fo Analyzing Unstructered Information.* New York: Springer.

Wira, P. B. (2009). *Pengklasifikasian Artikel.* Depok: Ilmu Komunikasi Universitas Indonesia.

Wisnu, D., & Anindita, H. (2015). Perancangan information retrieval (ir) untuk pencarian ide pokok teks artikel berbahasa inggris dengan pembobotan vector space model. *Jurnal Ilmiah Teknologi dan Informasi ASIA*, Vol. 9 No 1.

Zafikri, A. (2010). *Implementasi Metode Term Frequency Inverse Document* . Medan: Universitas Sumatera Utara.