

APLIKASI Pencarian Hadis Menggunakan *VECTOR SPACE MODEL* DENGAN PEMBOBOTAN TF-IDF DAN *CONFIX-STRIPPING STEMMER*

Novi Prisma Yunita^{*1}

¹Universitas AMIKOM Yogyakarta, Yogyakarta

Email: ¹novi@amikom.ac.id

^{*}Penulis Korespondensi

(Naskah masuk: 01 Desember 2022, diterima untuk diterbitkan: 21 Juni 2023)

Abstrak

Hadis adalah sumber ajaran Islam kedua setelah al-Qur'an. Kedudukannya yakni setelah Al-Qur'an, sebelum *Ijma'* dan *Qiyas*. Saat ini hadis sudah dapat diakses melalui berbagai platform digital, tetapi fitur pencarian yang disediakan masih sebatas *data retrieval* di mana hasil pencarian hanya didasarkan pada *keyword* pencarian tanpa memperhitungkan relevansi antara *keyword* dengan dokumen hasil pencarian. Penelitian ini bertujuan membangun aplikasi pencarian hadis yang dapat digunakan untuk menemukan hasil pencarian yang relevan dengan *keyword* pencarian. Aplikasi pencarian dibangun menggunakan metode *Information Retrieval* antara lain *Vector Space Model (VSM)* dan *Term Frequency – Invers Document Frequency (TF-IDF)* untuk membangun ruang vektor dan pembobotan *term*, dan *stemming* menggunakan algoritma *Confix-Stripping Stemmer (CS Stemmer)*. Selain itu, teknik *pre-processing* diterapkan menggunakan *stopwords removal*, dan pengukuran *similarity* menggunakan *Inner Product Similarity Measurement*. Penelitian ini menggunakan 162 dokumen hadis dari kitab *Bulughul Marom*. Dari sejumlah 6006 kata dalam dokumen, *stopwords removal* berhasil menghapus total 92 *stopwords*. Pengujian *CS Stemmer* terhadap 673 kosakata unik dalam dokumen, berhasil dilakukan dengan benar kepada 579 kata unik. Tingkat keberhasilan *CS Stemmer* adalah sebesar 78.6%. *Matrix of words* yang terbentuk dari VSM dan TF-IDF adalah matrik dengan ordo 673x 162. Pengujian aplikasi pencarian hadis dilakukan dengan memasukkan *keyword* yang berbeda ke dalam *form* pencarian. Hasil pengujian menunjukkan adanya relevansi antara *keyword* dengan dokumen hasil pencarian. Yakni dokumen dengan nilai *inner product* tertinggi adalah dokumen paling relevan dengan *keyword* pencarian. Semakin banyak kata yang muncul dalam dokumen, dan sesuai dengan *keyword* pencarian maka nilai relevansinya semakin tinggi.

Kata kunci: *hadis, information retrieval, stemming, confix-stripping, vector space model, stopwords, matrix of words, tf-idf, inner product*

HADITH SEARCH APPLICATION USING THE VECTOR SPACE MODEL WITH TF-IDF WEIGHTING AND CONFIX-STRIPPING STEMMER

Abstract

Hadith is the second source of Islamic teachings after the Al-Qur'an. Its position is after the Qur'an, before Ijma' and Qiyas. At present, the hadiths can be accessed through various digital platforms. But the search features are still limited to data retrieval, where search results are only based on search keywords without considering the relevance between the keywords and the search results document. This study aims to build a hadith search application that can be used to find search results relevant to the search keywords. The search application was created using the Information Retrieval method. Including Vector Space Model (VSM) and Term Frequency – Inverse Document Frequency (TF-IDF) to create vector space and term weighting and stemming using the Confix-Stripping Stemmer (CS Stemmer) algorithm. In addition, pre-processing techniques use stopwords removal, and similarity measurements use Inner Product Similarity Measurement. This study used 162 hadith documents from the Bulughul Marom book. Of the 6006 words in the document, stopword removal succeeded in removing 92 stopwords. The CS Stemmer test for 673 unique vocabularies in the document was successfully carried out correctly for 579 unique words. The CS Stemmer success rate is 78.6%. The matrix of words formed from VSM and TF-IDF is a matrix of the order 673x 162. The hadith search application is tested by entering different keywords in the search form. The test results show relevance between keywords and search results documents. That is, the document with the highest inner product value is the document most relevant to the search keyword—the more words that appear in the document and keyword, the higher the relevance value.

Keywords: *hadis, information retrieval, stemming, confix-stripping, vector space model, stopwords, matrix of words, tf-idf, inner product*

1. PENDAHULUAN

Hadis berasal dari bahasa Arab “*hadatsa*” yang berarti baru, lawan katanya adalah “*qodim*” berarti lama. Hadis adalah sesuatu hal baik sifat, perkataan, perbuatan, dan ketetapan yang disandarkan kepada Nabi Muhammad SAW (Nurdin & Shodik, 2019). Hadis merupakan sumber ajaran Islam kedua setelah Al-Qur’an. Hadis lebih utama dijadikan sumber hukum dibandingkan *Ijma* dan *Qiyas* karena. Hadis tidak bisa dipisahkan dari Al-Qur’an karena fungsinya sebagai penjelas terhadap ayat-ayat al-Qur’an bersifat global dan umum (Ummah, 2019).

Sebagai sumber ajaran kedua setelah Al-Qur’an, sudah semestinya mempelajari dan mengakses hadis bisa dilakukan dengan mudah. Di era digital ini, hadis banyak ditemukan dalam bentuk buku elektronik (*e-book*) (Ummah, 2019). Tersedia juga hadis dalam bentuk *software* (Ummah, 2019), aplikasi berbasis *mobile* seperti aplikasi “*Hadith Encyclopedia*”, dan website publik. Sejauh ini, pemanfaatan internet masih ada ditahap digitalisasi hadis. Fokusnya adalah bagaimana hadis bisa diakses dengan mudah melalui berbagai platform digital. Untuk menghindari mempelajari hadis yang keliru, fitur pencarian harus dibangun agar dapat memfasilitasi pencarian dan menemukan dokumen yang relevan. Fitur pencarian yang ada sejauh ini masih sebatas *data retrieval*, di mana relevansi antara *keyword* dengan dokumen hasil pencarian tidak dipertimbangkan.

Mesin pencari atau aplikasi pencarian adalah sebuah program komputer yang didesain khusus untuk menemukan dokumen dalam komputer (Umam & Negara, 2022). Aplikasi pencarian menerima masukan yang umumnya berupa frasa atau kata (Umam & Negara, 2022), meskipun dapat pula berupa data tidak terstruktur seperti gambar atau suara (Pratama, Darmalaksana, Maylawati, Sugilar, Mantoro, & Ramdhani, 2020). Aplikasi pencarian akan menemukan dokumen sesuai dengan masukan yang ditulis.

Saat ini, aplikasi pencarian bukan lagi sekedar kueri ke dalam basisdata (*data retrieval*). Pada *data retrieval*, kueri bekerja untuk menampilkan dokumen sesuai *keyword* pencarian, tetapi tidak mempertimbangkan relevansi antara *keyword* dengan dokumen. Untuk meningkatkan relevansi dokumen hasil pencarian, perlu penerapan *Information Retrieval* (IR) dalam penyusunan algoritma pencarian (Pratama, Darmalaksana, Maylawati, Sugilar, Mantoro, & Ramdhani, 2020). IR adalah konsep menemukan dokumen yang relevan yang berisi informasi tertentu, di antara suatu himpunan dokumen yang besar (Nie, 2022). IR yang digunakan untuk pencarian dokumen, di mana *keyword* dan dokumen menggunakan bahasa yang sama disebut dengan *Monolingual Information Retrieval* (MIR).

Sistem IR yang baik akan menghasilkan pencarian yang relevan dengan *keyword* masukan (Jelita, 2007).

Ada banyak algoritma yang dapat diterapkan dalam suatu IR (Pratama, Darmalaksana, Maylawati, Sugilar, Mantoro, & Ramdhani, 2020). *Vector Space Model* (VSM) mentransformasikan teks dengan panjang yang beragam ke dalam bentuk vektor numerik. Teks dalam hal ini dapat berupa kata, kalimat, paragraf, atau dokumen (Shahmirzadi, Lugowski, & Younge, 2019). Setiap kata (*term*) dalam dokumen direpresentasikan dalam vektor multidimensi (Pratama, Darmalaksana, Maylawati, Sugilar, Mantoro, & Ramdhani, 2020).

Term Frequency Invers Document Frequency (TF-IDF) adalah metode vektorisasi teks yang paling dasar (Shahmirzadi, Lugowski, & Younge, 2019) dan biasanya digunakan untuk *term weighting* pada sistem IR (Jalilifard, Caridá, Mansano, Cristo, & Fonseca, 2021). TF-IDF mengindikasikan seberapa penting sebuah *term* dalam dokumen (Pradana & Hayaty, 2019). Semakin banyak jumlah kemunculan *term* dalam dokumen, maka *term* tersebut dianggap semakin penting. TF menyimpan nilai kemunculan *term* dalam sebuah dokumen, IDF menyimpan nilai kemunculan *term* dari sebuah koleksi dokumen (Tripathy, Agrawal, & Rath, 2016).

Stemming adalah proses mengubah kata (*term*) menjadi kata dasarnya. *Stemming* biasanya digunakan dalam *Natural Language Processing* (NLP), salah satunya dalam *Information Retrieval* (Hidayatullah, Ratnasari, & Wisnugroho, 2016). Dalam IR, *stemming* dapat meningkatkan performa pemrosesan teks karena dalam prosesnya, kata yang memiliki akar yang sama akan digabungkan. Jika sebuah kata memiliki akar kata yang sama, maka dianggap memiliki kemiripan makna dan relevan (Rizki, Tjahyanto, & Trialih, 2019). Dalam kasus *stemming* bahasa Indonesia, ada beberapa algoritma *stemming*. Salah satunya adalah algoritma *Confix-Stripping Stemmer* (CS Stemmer). Algoritma ini adalah modifikasi dari algoritma Nazief & Adriani. Algoritma Algoritma Nazief dan Adriani berhasil melakukan *stemming* terhadap 92% kata unik bahasa Indonesia, sedangkan algoritma CS Stemmer dapat mencapai akurasi sebesar 95% (Adriani, Asian, Nazief, Tahaghoghi, & Williams, 2007).

Pada penelitian ini, aplikasi pencarian hadis dibangun menggunakan metode *Information Retrieval* yaitu VSM dan TF-IDF, serta *stemming* bahasa Indonesia menggunakan *Confix-Stripping Stemmer*. Dokumen yang digunakan adalah dokumen hadis dari kitab *Bulughul Marom* karangan Imam Ibnu Hajar Al-Asqalani. Tujuan penelitian ini adalah mengembangkan suatu aplikasi pencarian hadis yang dapat digunakan untuk menemukan hasil pencarian yang relevan dengan *keyword* pencarian.

2. STUDI LITERATUR

2.1. Penelitian Terdahulu

Beberapa penelitian terdahulu terkait untuk pencarian hadis antara lain: pengujian fitur bot telegram untuk pencarian hadis oleh (Wiguna, 2022). Hasil pengujian menggunakan UAT menunjukkan bahwa pengguna dapat menerima dan menggunakan bot sesuai kebutuhan. Pencarian hadis menggunakan metode *weighted inverse document frequency* (WIDF) dan *vector space model* (VSM) oleh (Pratama, Darmalaksana, Maylawati, Sugilar, Mantoro, & Ramdhani, 2020) dilakukan terhadap 380 hadis dari kitab Bulughul Marom. Hasilnya menunjukkan bahwa WIDF dan VSM yang diterapkan dalam aplikasi pencarian hadis sukses memberikan hasil pencarian yang sesuai, tetapi tidak relevan dengan *keyword* pencarian.

Penelitian dengan terapan TF-IDF untuk pencarian hadis juga dilakukan oleh (Al Ghofari, Rozi, Selmakaramy, & Ariansyah, 2021). *Confusion matrix* menghasilkan rata-rata nilai presisi sebesar 0.09, pengujian dilakukan dengan memasukkan 6 kata kunci yang dilakukan oleh 3 pengguna berbeda. (Agra, 2021) melakukan analisis VSM menggunakan filter *Linear Search* dan ORM Django pada pencarian hadis. Hasil pengujian menunjukkan bahwa dari 62.169 hadis dengan 20 contoh kata kunci, filter *Linear Search* mempercepat waktu eksekusi dokumen yakni hanya 7.93 detik, dan dengan ORM Django waktu eksekusi yang dibutuhkan adalah 8.41 detik. Waktu eksekusi ini jauh lebih cepat ketimbang pencarian hanya menggunakan VSM saja yakni selama 51 detik.

2.2. Algoritma *Stemming Confix-Stripping* (CS Stemmer)

Stemming berasal dari bahasa asing *stem* atau *root* yang berarti batang atau akar. *Stemming* adalah metode untuk mereduksi kata menjadi bentuk akarnya (Jelita, 2007). Kata “memasang”, “memasangkan”, “berpasangan”, “pemasangan” dapat direduksi menjadi akar katanya yaitu “pasang”. *Stemming* sangat penting dalam mendukung efektifitas *Information Retrieval* pada bahasa Indonesia. *Stemming* berguna dalam banyak hal, salah satunya untuk pencarian dokumen berbasis website (Adriani, Asian, Nazief, Tahaghoghi, & Williams, 2007).

Algoritma *stemming* berbeda untuk setiap bahasa. Ini disebabkan adanya karakteristik morfologi yang unik dari setiap bahasa sehingga perlu pendekatan tertentu untuk memotong suatu kata berimbuhan menjadi kata dasar. Ada beberapa algoritma *stemming* untuk bahasa Indonesia, antara lain: algoritma Arifin dan Setiono, algoritma Nazief dan Adriani, algoritma Ahmad, Yusoff, dan Sembok, dan algoritma Vega (Jelita, Williams, Tahaghoghi, 2007). Algoritma Confix-Stripping Stemmer (CS Stemmer) adalah modifikasi dari algoritma Nazief dan Adriani. Modifikasi yang dilakukan yakni

menyeleksi hasil *stemming* yang gagal kemudian menerapkan aturan-aturan tambahan terhadap kata-kata tersebut.

Aturan *stemming* Bahasa Indonesia:

- Bahwa kata yang berjumlah kurang dari 4 tidak perlu di *stemming* karena bisa dipastikan bahwa kata tersebut tidak berimbuhan
- Menghapus imbuhan hanya perlu dilakukan satu kali karena imbuhan tidak mungkin diulang
- Aturan *confix* bisa diterapkan, jika ditemukan kombinasi imbuhan yang tidak valid

Langkah algoritma CS Stemmer:

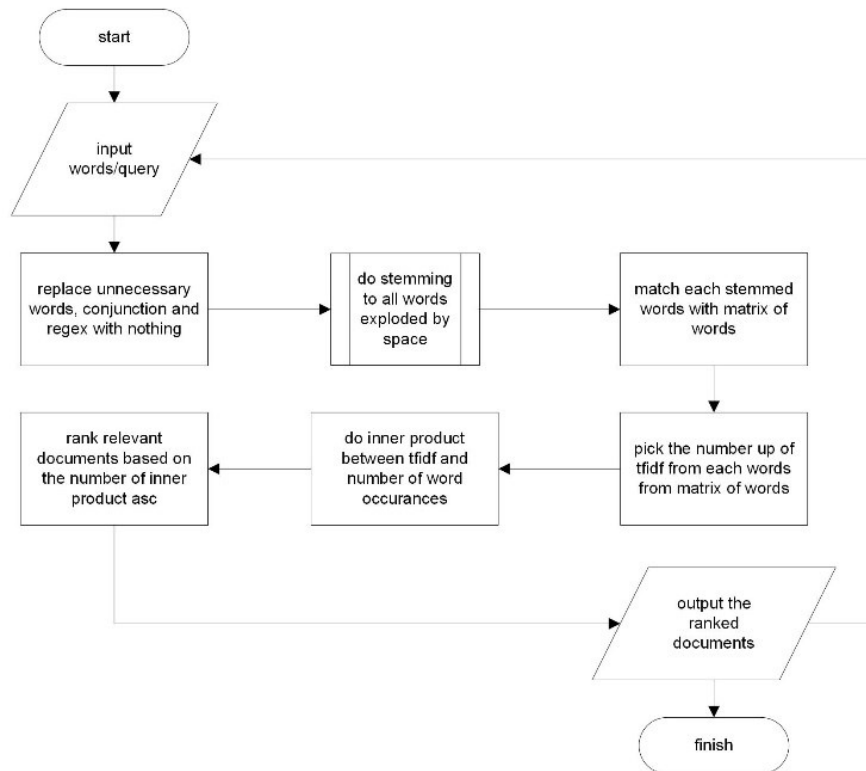
- Lakukan pengecekan terhadap korpus bahasa Indonesia setiap selesai 1 langkah *stemming*, jika kosakata ditemukan maka proses *stemming* berhenti
- Hapus akhiran infleksi, akhiran yang lebih dari 1 ditulis berdasarkan urutan. Hapus P, kemudian hapus PP. Contoh: “bajumulah” menjadi “bajumu”, kemudian menjadi “baju”. Pada tahap ini, susunan kata berimbuhan menjadi $[DP + [DP + [DP]]] \text{rootword} [+DS]$
- Hapus DS, yakni -i, -kan, dan -an. Contoh: “penempatan” menjadi “penempat”. Pada tahap ini, susunan kata berimbuhan menjadi $[DP + [DP + [DP]]] \text{rootword}$
- Hapus DP, yakni di-, be-, pe-, ke-, se-, te-, dan me-. Jika kata belum ditemukan setelah penghapusan DP, maka kenai aturan khusus
- Jika sampai tahap ini kata hasil *stemming* tidak ditemukan, maka kata tersebut dikembalikan ke bentuk awalnya.

2.3. Stop Words

Tidak semua kata memiliki bobot informasi yang sama (Jelita, 2007). *Stop words* adalah pendekatan yang digunakan untuk melakukan filter terhadap *keyword* yang tidak penting yang bertujuan untuk mengurangi ukuran dari istilah indeks dalam suatu dokumen (Zhang, 2008). *Keyword* yang termasuk dalam daftar *stop words* adalah kata umum, kata sambung, dan kata yang dianggap sebagai kata yang umum secara konteks. Dalam kasus bahasa Indonesia, *keyword* yang disepakati sebagai *stop words* yakni sebanyak 84 kata. Di antaranya: dan, dari, sedang, yang, dalam, wahai, bahwa, kepada, saat, dan lain-lain. Daftar *stopword* ini dapat bertambah sesuai konteks dokumen.

2.4. Vector Space Model dan Term Frequency and Invers Document Frequency (TF-IDF)

Vector space model (VSM) adalah pemodelan menggunakan vektor untuk merepresentasikan kueri dan dokumen. Ruang vektor dibentuk dari *term* yang berasal dari dokumen.



Gambar 1. Diagram Alir Aplikasi Pencarian

Pada vektor kueri dan vektor dokumen, masing-masing elemen merepresentasikan bobot dari *term* yang sesuai dengan kueri dan dokumen.

<i>Vector space</i>	< <i>t1</i>	<i>t2</i>	<i>t3</i>	...	<i>tn</i> >
<i>Document</i>	< <i>d1</i>	<i>d2</i>	<i>d3</i>	...	<i>dn</i> >
<i>Query</i>	< <i>q1</i>	<i>q2</i>	<i>q3</i>	...	<i>qn</i> >

TF-IDF adalah teknik pembobotan kata (*term weighting*) (Kadhim, 2019). *Term frequency* (TF) menyimpan frekuensi kemunculan kata dalam suatu dokumen. *Document frequency* (DF) menyimpan frekuensi kemunculan dokumen, dan *invers document frequency* (IDF) menyimpan nilai invers (kebalikan) dari DF. Perhitungan IDF yakni dengan menggunakan fungsi log sebagai ratio dari seluruh dokumen *N* terhadap jumlah dokumen yang mengandung suatu kata (Zhang, 2008). Mengacu pada Zhang (2008), berikut ini adalah rumus menghitung nilai IDF:

$$idf(this, D) = \log \frac{N}{|\{d \in D : t \in d\}|} \quad (1)$$

$$tfidf(t, d, D) = tf(t, d) \times idf(t, D) \quad (2)$$

Nilai TF-IDF didapatkan dari perkalian antara TF dan IDF. Sehingga nilai TF dan IDF harus diketahui terlebih dahulu. Pendekatan TF-IDF ini telah terbukti

sangat kuat dan tidak mudah dikalahkan, meski dengan sebagian besar teori pembobotan manapun (Zhang, 2008).

2.5. Inner Product Similarity Measurement

Pada pengukuran kemiripan *inner product*, fitur bobot yang dibagikan oleh dua vektor yang terlibat akan dipertimbangkan (Zhang, 2008 : 28). Mengacu pada Zhang (2008), rumus *inner product* adalah sebagai berikut:

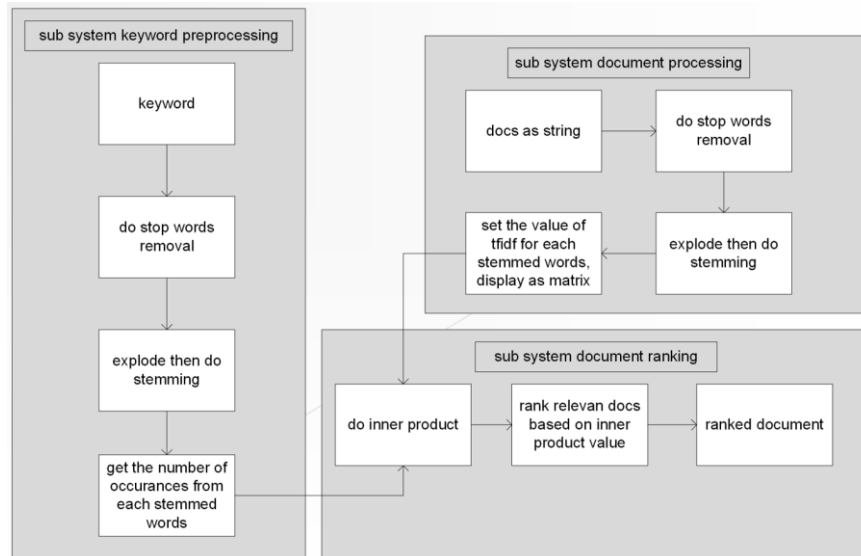
$$S(x, y) = \sum_{i=1}^n a_i \times b_i \quad (3)$$

Nilai *inner product* didapatkan dari perkalian jumlah kemunculan kueri dengan nilai TF-IDF dari dokumen, dan menjumlahkannya sebanyak *i* kali.

3. PERANCANGAN SISTEM

3.1. Studi kasus dan Data

Studi kasus yang digunakan dalam penelitian ini adalah kitab Bulughul Marom karangan Imam Ibnu Hajar Al-Asqalany, diambil dari website <http://www.alquran-sunnah.com/kitab.bulughul-maram/> (situs ini bersifat *freeware*) terdiri dari hadis dalam teks bahasa Arab dan terjemahannya dalam teks bahasa Indonesia. Jumlah dokumen yang digunakan adalah 162 dokumen (hadis). Data lain yang digunakan adalah korpus berisi kata dasar bahasa Indonesia dengan total kosakata yakni 26.275 kata bahasa Indonesia. Korpus ini didapatkan dari website KBBI online dengan format .sql.



Gambar 2. Gambaran Umum Aplikasi

3.2. Diagram Alir Aplikasi Pencarian

Gambar 1 adalah diagram alir yang menggambarkan alur pencarian. Berawal dari *keyword* dimasukkan sampai *output* dokumen ditampilkan. Alur kerja aplikasi pencarian adalah sebagai berikut:

Keyword pencarian dimasukkan ke dalam *form* pencarian, jumlah *keyword* yang dimasukkan boleh lebih dari satu. Aplikasi akan menerima *keyword* kemudian menghapus semua daftar *stopword* yang ada dalam *keyword* tersebut. Setelah itu algoritma *CS Stemmer* diaplikasikan kepada *keyword* yang sudah melewati proses *stopword removal*. Setelah mendapatkan hasil *stemming*, *keyword* ini akan dicocokkan dengan kata yang ada dalam *matrix of words*, jika kecocokan dikonfirmasi maka nilai TF-IDF diambil dari *matrix of words*.

Selanjutnya dilakukan perhitungan *inner product* (perkalian) antara nilai TF-IDF dengan frekuensi kemunculan kata. Dokumen diurutkan berdasarkan nilai *inner product* tertinggi sampai terendah. Terakhir, dokumen hasil pencarian ditampilkan. Jika *output* sudah muncul, pengguna dapat menggunakan ulang aplikasi pencarian ini untuk *keyword* yang lain.

3.3. Gambaran Umum Aplikasi

Aplikasi pencarian ini terdiri dari 3 sub-sistem, yakni: *sub-system keyword preprocessing*, *sub-system document processing*, dan *sub-system document ranking*. Pada *sub-system keyword preprocessing* dan *document processing*, tahapan yang dilakukan hampir sama. Keduanya sama-sama melewati tahap *stopword removal* dan menerapkan *CS Stemmer* terhadap *keyword* dan dokumen. Perbedaannya, pada *document processing* hasil *stemming* kemudian dipetakan dalam bentuk VSM, yakni *matrix of words* dengan pembobotan TF-IDF. Sementara pada *keyword processing*, dilakukan perhitungan kemunculan *term* yang sama terhadap

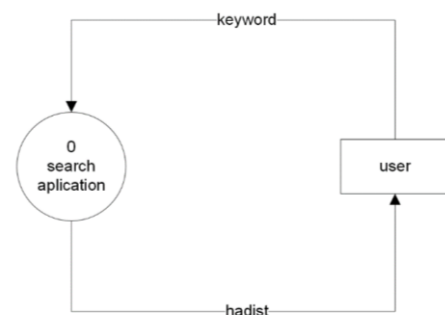
hasil *stemming*. Perbedaan lain adalah *sub-system keyword preprocessing* memproses *keyword* (kata kunci pencarian) sedangkan *sub-system document processing* memproses dokumen (dokumen yang akan dicari).

Sementara itu, *sub-system document ranking* bertanggung jawab terhadap pengurutan dokumen. Setelah nilai *inner product* dihitung, dokumen akan diurutkan berdasarkan nilai *inner product*, yakni dari nilai tertinggi ke terendah (*descending*), hasilnya berupa dokumen yang sudah diurutkan.

3.4. Data Flow Diagram

Data Flow Diagram (DFD) adalah grafik yang menunjukkan aliran data pada sistem. DFD digunakan untuk melakukan review terhadap sistem (Edy, 2010) serta menunjukkan bagaimana sebuah sistem dibagi atau dipecah ke dalam bentuk yang lebih kecil lagi sehingga dapat menyoroti aliran data di antara tiap-tiap bagiannya (Jimmy, 2008).

Rancangan arsitektur untuk aplikasi pencarian ini terdiri dari DFD level 0 dan DFD level 1. Gambar 3 adalah DFD level 0. DFD ini memiliki satu proses bernama "*search application*" dan satu entitas/terminator bernama "*user*". Entitas "*user*" mengalirkan data yaitu "*keyword*" masuk ke dalam proses "*search application*" kemudian proses tersebut memberikan keluaran yaitu "*hadist*".



Gambar 3. DFD Level 0

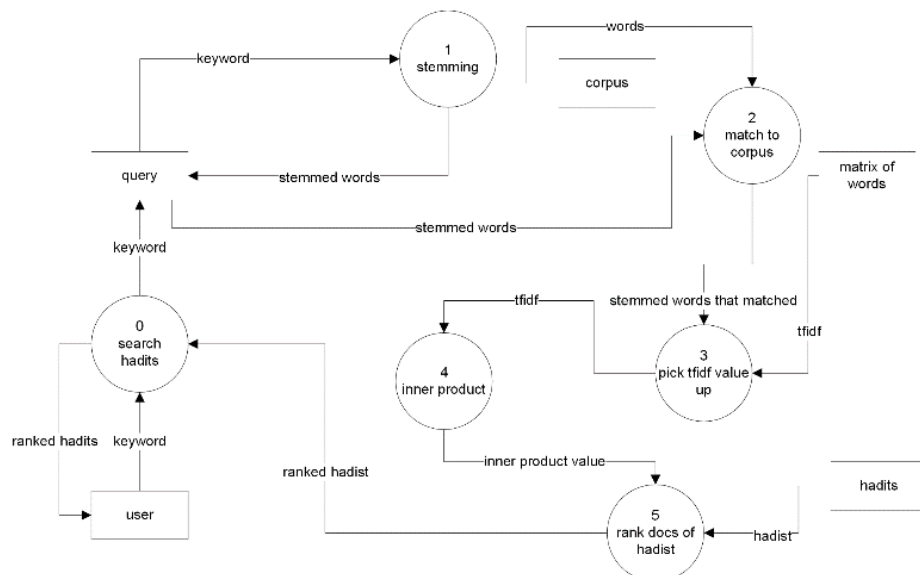
Gambar 4 adalah DFD level 1. DFD level 1 menggambarkan proses yang lebih detail berdasarkan DFD level 0. DFD level 1 memiliki 6 proses, 4 data store, dan satu entitas. Entitas ‘user’ memasukkan keyword ke proses 1 yakni “search hadist” kemudian keyword disimpan dalam data store ‘query’. Setelah disimpan, keyword akan melewati proses 2 yaitu “stemming”. Hasil dari proses 2 disimpan kembali ke data store “query”.

Selanjutnya dilakukan proses 3 yakni “match to corpus”, proses ini melibatkan data store “corpus”. Kemudian dilakukan pengambilan nilai TF-IDF yang melibatkan data store “matrix of words”. Setelah itu, proses 5 “inner product” dijalankan dan menghasilkan nilai inner product. Proses 6 adalah “rank docs of hadists” adalah pengurutan dokumen yang dilakukan berdasarkan output dari proses 5.

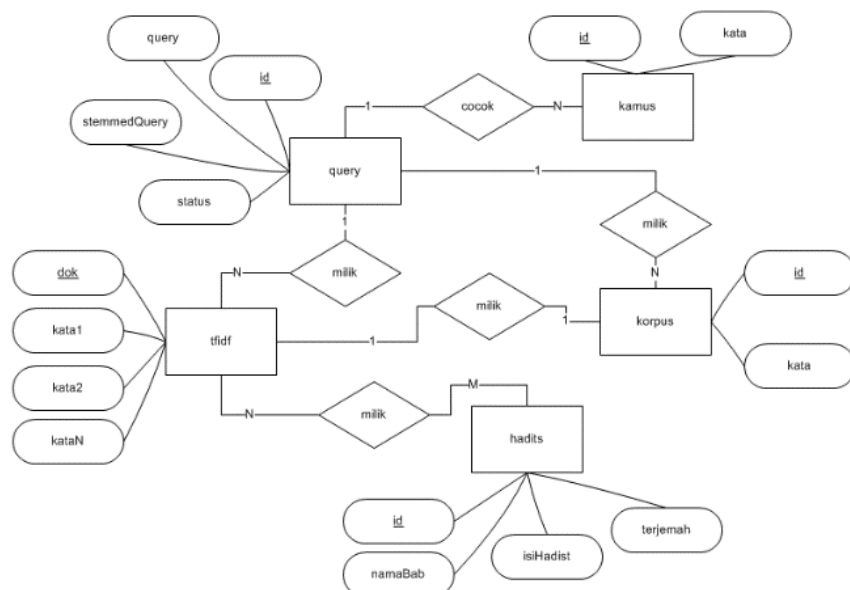
Hasil dari proses 6 adalah “ranked hadists”. “Ranked hadists” dikembalikan ke proses 1 dan diteruskan sebagai output kepada entitas “user”.

3.5. Entity Relationship Diagram

Entity Relationship Diagram (ERD) adalah diagram yang menunjukkan bagaimana informasi dibuat, disimpan, dan digunakan dalam sistem bisnis (Fatta, 2007). ERD digunakan untuk memodelkan struktur data dan hubungan antar data. Notasi dalam ERD terdiri dari entitas dan atribut, serta relasi dari satu entitas ke entitas yang lain.



Gambar 4. DFD Level 1



Gambar 5. ERD Aplikasi Pencarian

Berikut ini adalah identifikasi entitas, atribut, dan relasi untuk aplikasi pencarian (sisi kiri adalah nama entitas, sisi kanan adalah daftar atribut):

kamus : id, kata

query : id, query, stemmedQuery, status

korpus : id, kata

tfidf : dok, kata1, kata2, kataN

hadist : id, namaBab, isiHadist, terjemah

Berdasarkan Gambar 5, ERD terdiri dari 5 buah entitas dengan masing-masing atribut. Kardinalitas yang ada pada entitas antara lain: *one-to-many* terjadi antara entitas query dengan kamus, entitas tfidf dengan korpus, entitas query dengan tfidf, dan entitas query dengan korpus; *many-to-many* terjadi antara entitas tfidf dengan hadist.

4. HASIL DAN PEMBAHASAN

4.1. Antarmuka Aplikasi Pencarian

Aplikasi pencarian hadis ini dibuat berbasis website. Aplikasi pencarian terdiri dari dua halaman. Halaman utama yakni halaman pencarian, halaman ini terdiri dari satu buah *form* untuk menuliskan *keyword* pencarian. *Form* pencarian dapat menerima *keyword* lebih dari satu kata, dan tidak terbatas jumlahnya. Gambar 6 adalah halaman utama aplikasi pencarian.

Gambar 7 adalah tampilan halaman hasil pencarian. Halaman ini menampilkan daftar dokumen hasil pencarian yang relevan dengan *keyword* pencarian. Baris pertama hasil pencarian adalah dokumen paling relevan, nilai relevansinya semakin ke bawah semakin menurun. Jumlah baris hasil pencarian yang ditampilkan tergantung dengan *keyword* pencarian, dan minimal memiliki satu kata yang sama dengan dokumen yang dicari. Jika tidak ada dokumen yang relevan, maka aplikasi akan menampilkan halaman kosong.



Gambar 6. Antarmuka Aplikasi Pencarian

Aplikasi ini dibangun menggunakan gabungan teknik MIR yang fokus untuk pencarian dokumen yang relevan. Oleh karenanya, aplikasi ini tidak memperhitungkan waktu eksekusi (*execution time*) terhadap jumlah kata/karakter pada *keyword* pencarian. Aplikasi ini juga tidak menyediakan fitur *advanced* seperti pembatasan jumlah dokumen yang ditampilkan dan/atau *error handling* (penanganan jika terjadi galat). Jika terjadi kesalahan saat menulis *keyword* sehingga karakter tersebut tidak dikenali

oleh KBBI, maka *keyword* tersebut tidak akan terdeteksi sebagai salah ketik, tetapi tetap diproses sesuai alur prosesnya.

4.2. Implementasi Stop Words Removal dan Algoritma CS Stemmer

Algoritma CS Stemmer diimplementasikan dua kali. Pertama saat membangun *Vector Space Model* (VSM) dari dokumen (*subsystem document processing*), kedua saat melakukan pencarian dokumen (*subsystem keyword preprocessing*). Algoritma CS Stemmer pada saat pencarian dokumen (*subsystem keyword preprocessing*) akan bekerja berdasarkan *keyword* yang dimasukkan. Sub-bab ini berisi paparan implementasi CS Stemmer untuk *subsystem document processing*.

Tabel 1. Hasil Analisis Kata dalam Dokumen

Keterangan	Jumlah (kata)
Kata dari 162 dokumen	6006
Kata unik (tidak berulang)	765
Kata unik selain <i>stopwords</i>	673
Hasil <i>stemming</i> ada di KBBI	529
Gagal <i>stemming</i>	144

Tingkat kesuksesan algoritma CS Stemmer adalah 78.1%, rincian kata dan jumlahnya dipaparkan pada Tabel 2. Dari total 6006 kata pada dokumen, ada sejumlah 765 kata unik. Setelah dikurangi dengan daftar *stopwords* maka total kata adalah 673. Dari total tersebut, 529 kata berhasil di-*stemming*, sisanya 144 kata gagal. Berhasil artinya kata tersebut bisa direduksi menjadi kata dasar dan ditemukan di KBBI, gagal artinya kata dasar hasil *stemming* tidak ditemukan di KBBI.

Adapun alasan gagal *stemming* yang terjadi terhadap 144 kata antara lain: kata tersebut adalah nama orang, kata tersebut adalah bahasa Arab yang ditulis latin (bukan terjemahan bahasa Indonesia yang diakui KBBI), kata tersebut adalah angka, kata tersebut tidak di-*cover* dalam algoritma dan aturan khusus CS Stemmer. Klasifikasi gagal *stemming* dipaparkan pada Tabel 2.

Tabel 2. Rincian Gagal Stemming

Klasifikasi kesalahan	Kata Berimbuhan	Kata dasar	Hasil Stemming
Nama	Utsman	-	Uts
orang	Rasulullah	-	-
Kata dasar	Demikian	Demikian	Demiki
Kata Ulang	Bagian-bagian	Bagian	Bagian-bagi
	Sebagai	Bagai	Baga
	Memabukkan	Mabuk	Pabuk
Kata berimbuhan	Menyatakan	Nyata	Sata
	Pergelangan	Gelang	Pergelang
	Kebanyakan	Banyak	Banya
	Mengandung	Kandung	Andung
Angka	20	-	-

Pada *subsystem document processing*, kata selain *stopwords* akan melalui proses *stemming*. Sehingga nama orang dan gelar yang sering muncul dalam terjemahan hadis seperti “Ibnu”, “Abbas”, “Utsman”, “Rasulullah”, “alaih”, akan melewati proses *stemming*.

Monolingual Information Retrieval for Indonesian Document			
<div> <div>type your word....</div> <div>search</div> </div>			
Arab	Indonesia		Prosentase
وَعَلَى الْأَيْمَنِ: (يُجَادِي إِبْرَاهِيمَ يُبَيِّنُ)	Menurut riwayat Imam Empat: Kulit binatang apapun yang telah disamak (ia menjadi suci).		100%
وَعَنِ ابْنِ عَبَّاسٍ رَضِيَ اللَّهُ عَنْهُمَا قَالَ: قَالَ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ يَا بُنَيَّ الْإِبْرَاهِيمَ قَدْ طَهَّرَ الْفَرْجَةَ شَيْئًا	Dari Ibnu Abbas Radliyallaahu 'anhu bahwa Rasulullah Shallallaahu 'alaihi wa Sallam bersabda: Jika kulit binatang telah disamak maka ia menjadi suci. Diriwayatkan oleh Muslim.		100%
وَعَنِ ثَعْلَبَةَ بْنِ قُثَيْبٍ رَضِيَ اللَّهُ عَنْهُ قَالَ: قَالَ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ يَا بُنَيَّ الْإِبْرَاهِيمَ قَدْ طَهَّرَ الْفَرْجَةَ شَيْئًا مِمَّا سَمِعْتُهُ مِنْ أَبِي جَدِّانَ	Dari Salamah Ibnu al-Muhabbig Radliyallaahu 'anhu bahwa Rasulullah Shallallaahu 'alaihi wa Sallam bersabda: Menyamak kulit bangkai adalah menyucikannya. Hadits shahih menurut Ibnu Hibban.		70.33%
وَعَنِ جَزْأَنَ بْنِ خُسَيْبٍ رَضِيَ اللَّهُ عَنْهُ قَالَ قَالَ أَبِي هُرَيْرَةَ رَضِيَ اللَّهُ عَنْهُ وَشِعْبَةَ بْنِ قُثَيْبٍ: مَا سَمِعْتُ أَبَا جَدِّانَ إِلاَّ أَنَّهُ قَالَ: طَهَّرَ الْفَرْجَةَ شَيْئًا مِمَّا سَمِعْتُهُ مِنْ أَبِي جَدِّانَ	Dari Imran Ibnu Hushoin Radliyallaahu 'anhu bahwa Nabi Shallallaahu 'alaihi wa Sallam dan para sahabatnya berwudlu di mazadah (tempat air yang terbuat dari kulit binatang) milik seorang perempuan musyrik. Muttafaq Alaihi dalam hadits yang panjang.		60.65%
وَعَنِ عُمَرَ بْنِ الْخَارِجَةِ رَضِيَ اللَّهُ عَنْهُ قَالَ: سَمِعْتُ أَبِي هُرَيْرَةَ رَضِيَ اللَّهُ عَنْهُ وَشِعْبَةَ بْنَ قُثَيْبٍ: مَا سَمِعْتُ أَبَا جَدِّانَ إِلاَّ أَنَّهُ قَالَ: طَهَّرَ الْفَرْجَةَ شَيْئًا مِمَّا سَمِعْتُهُ مِنْ أَبِي جَدِّانَ	Dari Amru Ibnu Khorijah Radliyallaahu 'anhu berkata: Nabi saw berkhotbah pada waktu kami di Mina sedang beliau di atas binatang kendaraannya, dan air liur binatang tersebut mengalir di atas pundakku. Dikeluarkan oleh Ahmad dan Tirmidzi, dan dinilainya hadits shahih.		59.36%
وَعَنِ ابْنِ خُزَيْمَةَ رَضِيَ اللَّهُ عَنْهُ قَالَ: قَالَ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ يَا بُنَيَّ الْإِبْرَاهِيمَ قَدْ طَهَّرَ الْفَرْجَةَ شَيْئًا مِمَّا سَمِعْتُهُ مِنْ أَبِي جَدِّانَ	Ibnu Mas'ud Radliyallaahu 'anhu berkata: "Nabi Shallallaahu 'alaihi wa Sallam hendak buang air besar lalu beliau menyuruhku untuk mengambilkan tiga biji batu kemudian saya hanya mendapatkan dua biji dan tidak menemukan yang ketiga. Lalu saya membawakan kotoran binatang. Beliau mengambil dua biji batu tersebut dan membuang kotoran binatang seraya bersabda: "Ini kotoran menjijikkan." Diriwayatkan oleh Bukhari, Ahmad dan Daruquthni sebagai "Ambikan air yang lain."		59.36%
وَعَنِ ابْنِ كُرَيْزَةَ رَضِيَ اللَّهُ عَنْهُ قَالَ: قَالَ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ يَا بُنَيَّ الْإِبْرَاهِيمَ قَدْ طَهَّرَ الْفَرْجَةَ شَيْئًا مِمَّا سَمِعْتُهُ مِنْ أَبِي جَدِّانَ	Dari Abu Hurairah Radliyallaahu 'anhu bahwa Rasulullah Shallallaahu 'alaihi wa Sallam bersabda: "Sesungguhnya di bawah setiap helai rambut terdapat jinabat. Oleh karena itu cucilah rambut dan bersihkanlah kulitnya." Riwayat Abu Dawud dan Tirmidzi dan keduanya menganggap hadits ini lemah.		30.99%
وَعَنِ ابْنِ كُرَيْزَةَ رَضِيَ اللَّهُ عَنْهُ قَالَ: قَالَ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ يَا بُنَيَّ الْإِبْرَاهِيمَ قَدْ طَهَّرَ الْفَرْجَةَ شَيْئًا مِمَّا سَمِعْتُهُ مِنْ أَبِي جَدِّانَ	Dari Abu Hurairah Radliyallaahu 'anhu bahwa Rasulullah Shallallaahu 'alaihi wa Sallam bersabda: "Tanah itu merupakan alat berwudlu bagi orang Islam meskipun ia tidak menjumpai air hingga sepuluh tahun. Maka jika ia telah mendapatkan air hendaklah ia bertakwa kepada Allah dan menggunakan air itu untuk mengusap kulitnya." Diriwayatkan oleh al-Bazzar. Shahih menurut Ibnu Athththan dan mursal menurut Daruquthni.		30.99%
وَعَنِ ابْنِ كُرَيْزَةَ رَضِيَ اللَّهُ عَنْهُ قَالَ: قَالَ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ يَا بُنَيَّ الْإِبْرَاهِيمَ قَدْ طَهَّرَ الْفَرْجَةَ شَيْئًا مِمَّا سَمِعْتُهُ مِنْ أَبِي جَدِّانَ	Maimunah Radliyallaahu 'anhu berkata bahwa Rasulullah Shallallaahu 'alaihi wa Sallam melewati seekor kambing yang sedang diseret orang-orang. Beliau		30.99%

Gambar 7. Tampilan Hasil Pencarian

Pada terjemahan hadis, kadang ditemukan pula kalimat yang mengandung angka. Pada “20 hari” angka 20 dianggap satu buah kata yang juga melewati proses *stemming*. Dalam konteks terjemahan hadis, angka tidak bisa digolongkan sebagai *stopwords* karena akan merubah konteks dari terjemahan hadis. Oleh karenanya, kehadiran angka pada daftar *stemming* cukup menyumbang jumlah kata yang gagal *stemming*.

Selain karena konteks studi kasus yang merupakan teks terjemahan, ditemukan beberapa kata gagal *stemming* di alasan konteks. Misalnya kata berulang “bagian-bagian” gagal di-*stemming* menjadi “bagi” karena aturan yang belum tercakup dalam algoritma. *CS Stemmer* tidak bisa menyelesaikan semua kasus kata berimbuhan bahasa Indonesia, karena adanya ambiguitas dalam bahasa itu sendiri pemahaman. Untuk menyelesaikan permasalahan gagal *stemming* karena adanya ambiguitas ini diperlukan adanya lebih mendalam terkait bahasa Indonesia (Jelita, 2007).

4.3. Implementasi VSM dan TF-IDF

Implementasi pemodelan VSM merangkai suatu matriks dua dimensi dengan ordo 673×162 . Yakni 673 kolom sesuai jumlah kata unik yang terdapat dalam dokumen (setelah *stop words removal*), dan 120 baris sesuai jumlah dokumen studi kasus. Pertemuan antara kolom dan baris pada matriks ini diisi dengan nilai pembobotan TF-IDF.

Penggunaan TF-IDF pada VSM memungkinkan adanya bobot untuk suatu *term* bernilai lebih dari satu, tergantung dari nilai IDF dan jumlah dokumen yang ada. Nilai TF-IDF yang besar mengindikasikan besarnya frekuensi kemunculan *term* dalam suatu dokumen. Artinya, semakin besar nilai TF-IDF maka

semakin besar pula frekuensi kemunculan *term* dalam dokumen.

Pada pemodelan VSM, pertemuan antara kolom dan baris pada matriks dapat bernilai 0. Nilai 0 diberikan jika tidak ada *term* yang cocok pada dokumen tersebut sehingga nilai bobotnya tidak ada.

4.4. Pengujian

Pengujian dilakukan menggunakan aplikasi pencarian. *Keyword* ditulis di kolom pencarian kemudian klik tombol “*search*” untuk melihat hasilnya. Pada sesi pengujian, hasil pencarian disajikan detail dengan nilai TF-IDF dari masing-masing *keyword*, dan *inner product* dari masing-masing dokumen. Penyajiannya diurutkan berdasarkan nilai *inner product* secara *descending*.

Pengujian dilakukan sebanyak 3 kali. Berikut ini adalah paparan dari pengujian aplikasi:

Pengujian 1

Keyword : kulit binatang yang disamak

Jumlah kata : 4

Stop words removal : kulit binatang disamak

Hasil *stemming* : kulit, binatang, samak

Jumlah dokumen relevan: 12 dokumen

Tabel 3. Rincian Hasil Pengujian 1

No	No Dokumen	Nilai tdfidf			Inner Product
		kulit	binatang	samak	
1	21	1.364	1.306	1.732	4.402
2	20	1.364	1.306	1.732	4.402
3	22	1.364	-	1.732	3.096
4	25	1.364	1.306	-	2.67
5	29	-	2.613	-	2.613
6	107	-	2.613	-	2.613

No	No Dokumen	Nilai tfidf			Inner Product
		kulit	binatang	samak	
7	134	1.364	-	-	1.364
8	142	1.364	-	-	1.364
9	23	1.364	-	-	1.364
10	139	-	1.306	-	1.306
11	108	-	1.306	-	1.306
12	17	-	1.306	-	1.306

Berdasarkan Tabel 3, dua dokumen teratas nomor 21 dan 20 memiliki nilai *inner product* yang sama. Dokumen nomor 21 bertuliskan, “Menurut riwayat Imam Empat: Kulit binatang apapun yang telah disamak (ia menjadi suci).” Dokumen nomor 20 bertuliskan, “Dari Ibnu Abbas Radliyallaahu ‘anhu bahwa Rasulullah Shallallaahu ‘alaihi wa Sallam bersabda: Jika kulit binatang telah disamak maka ia menjadi suci. Diriwayatkan oleh Muslim.” Kedua dokumen ini masing-masing mengandung kata “kulit”, “binatang”, dan “disamak” dengan kemunculan satu kali.

Sementara itu, dokumen di baris ke-3 sampai baris ke-12 memiliki nilai *inner product* dengan nilai semakin kecil. Dokumen nomor 22 di baris ke-3, mengandung kata “kulit” dan “samak” tetapi tidak mengandung kata “binatang”. Seterusnya hingga dokumen nomor 17 di baris ke-12, mengandung kata “binatang” tetapi tidak mengandung kata “kulit” dan “samak”. Sehingga, dapat disimpulkan bahwa dokumen nomor 21 dan 20 adalah dokumen yang paling relevan dengan *keyword* pencarian karena mengandung kata yang sama dengan *keyword*.

Pengujian 2

Keyword : hukum bagi wanita yang sedang haid

Jumlah kata : 6

Stop words removal : hukum wanita haid

Hasil *stemming* : hukum, wanita, haid

Jumlah dokumen relevan : 13 dokumen

Berdasarkan Tabel 4, dokumen teratas nomor 158 dan 131 memiliki nilai *inner product* yang sama. Adapun, redaksi dokumen nomor 158 bertuliskan, “Dari Abu Said Al-Khudry bahwa Rasulullah Shallallaahu ‘alaihi wa Sallam bersabda: Bukankah wanita itu jika datang haid tidak boleh shalat dan berpuasa. Muttafaq Alaihi dalam hadits yang panjang.” Dokumen nomor 131 bertuliskan, “Ummu Salamah Radliyallaahu ‘anhu berkata: Aku bertanya wahai Rasulullah sungguh aku ini wanita yang mengikat rambut kepalaku. Apakah aku harus membukanya untuk mandi jinabat? Dalam riwayat lain disebutkan: Dan mandi dari haid? Nabi menjawab: “Tidak tetapi kamu cukup mengguyur air di atas kepalamu tiga kali.” Riwayat Muslim.” Kedua dokumen tersebut mengandung kata “wanita” dan “haid” dengan kemunculan satu kali.

Tabel 4. Rincian Hasil Pengujian 2

No	No Dokumen	nilai tfidf			Inner Product
		hukum	wanita	haid	
1	158	-	1.732	1.168	2.9
2	131	-	1.732	1.168	2.9
3	151	-	-	2.336	2.336
4	73	-	-	2.336	2.336
5	56	2.21	-	-	2.21
6	116	-	1.732	-	1.732
7	157	-	-	1.168	1.168
8	156	-	-	1.168	1.168
9	160	-	-	1.168	1.168
10	149	-	-	1.168	1.168
11	132	-	-	1.168	1.168
12	152	-	-	1.168	1.168
13	155	-	-	1.168	1.168

Dokumen di baris ke-3 sampai baris ke-13 memiliki nilai *inner product* dengan nilai semakin kecil. Dokumen nomor 151 dan 73 mengandung kata “haid” sebanyak 2 buah. Dokumen nomor 56 mengandung kata “hukum” sebanyak 1 buah. Dokumen nomor 116 mengandung kata “wanita” sebanyak 1 buah. Sisanya dokumen di baris ke-7 sampai baris 13 seluruhnya mengandung kata “haid” sebanyak 1 buah. Sehingga dapat disimpulkan bahwa, dokumen nomor 158 dan 131 adalah dua dokumen teratas yang paling relevan dengan *keyword* pencarian, karena mengandung kata yang sama dengan *keyword* dengan jumlah kata paling banyak dan total *inner product* paling tinggi.

Pengujian 3

Keyword : makan dalam suatu bejana

Jumlah kata : 4

Stop words removal : makan bejana

Hasil *stemming* : makan, bejana

Jumlah dokumen relevan : 9 dokumen

Berdasarkan Tabel 5, dokumen 24 sebagai dokumen teratas bertuliskan, “Dari Abu Tsa’labah al-Khusny berkata: Saya bertanya, wahai Rasulullah, kami tinggal di daerah Ahlul Kitab, bolehkah kami makan dengan bejana mereka? Beliau menjawab: Janganlah engkau makan dengan bejana mereka kecuali jika engkau tidak mendapatkan yang lain. Oleh karena itu bersihkanlah dahulu dan makanlah dengan bejana tersebut. Muttafaq Alaihi.”

Tabel 5. Rincian hasil pengujian 3

No	No Dokumen	nilai tfidf		Inner Product
		makan	bejana	
1	24	4.822	4.093	8.915
2	18	1.607	1.364	2.971
3	155	1.607	-	1.607

No	No Dokumen	nilai tfidf		Inner Product
		makan	bejana	
4	81	1.607	-	1.607
5	150	-	1.364	1.364
6	96	-	1.364	1.364
7	19	-	1.364	1.364
8	26	-	1.364	1.364
9	95	-	1.364	1.364

Berdasarkan tabel 6, dokumen nomor 24 adalah dokumen dengan nilai *inner product* tertinggi. Dokumen tersebut bertuliskan, “*Dari Abu Tsa’labah al-Khusny berkata: Saya bertanya, wahai Rasulullah, kami tinggal di daerah Ahlul Kitab, bolehkah kami makan dengan bejana mereka? Beliau menjawab: Janganlah engkau makan dengan bejana mereka kecuali jika engkau tidak mendapatkan yang lain. Oleh karena itu bersihkanlah dahulu dan makanlah dengan bejana tersebut. Muttafaq Alaihi.*”. Dokumen ini mengandung kata “makan” sebanyak 3 dan “bejana” sebanyak 3.

Dokumen baris ke-2 sampai baris ke-9 nilai *inner product*-nya semakin menurun. Dokumen nomor 18 mengandung kata “makan” sebanyak 1 buah dan “bejana” sebanyak 1 buah. Dokumen nomor 155 dan 81 mengandung kata “makan” sebanyak 1 buah. Sisanya dokumen 150 sampai dokumen 95 seluruhnya mengandung kata “bejana” sebanyak 1 buah. Sehingga dapat disimpulkan bahwa dokumen teratas nomor 24 adalah dokumen yang paling relevan dengan *keyword* pencarian karena mengandung kata yang sama dengan *keyword* dengan jumlah paling banyak, sehingga nilai *inner product*-nya paling tinggi.

5. KESIMPULAN

Aplikasi pencarian dibangun berbasis web dengan menerapkan pendekatan dalam *Monolingual Information Retrieval* (MIR) dan memberikan hasil pencarian yang baik dan relevan berdasarkan pengujian yang sudah dilakukan. Parameter baik dan relevan diperoleh dari kecocokan kata (*term*) antara *keyword* dengan dokumen. Implementasi *Vector Space Model* (VSM) menghasilkan matriks dengan ordo 673x162, matriks ini dibangun dari 673 jumlah kata unik dalam 162 dokumen studi kasus. Pertemuan antara baris dan kolom diisi dengan pembobotan TF-IDF. Persentase keberhasilan implementasi *CS Stemmer* yakni sebesar 78.6% terhadap 673 *term* unik dalam dokumen.

Hasil pengujiannya menunjukkan relevansi antara *keyword* pencarian dengan dokumen hasil pencarian. Relevansi ini ditunjukkan dengan adanya kesamaan *term* yang muncul di *keyword* dengan dokumen. Semakin banyak kesamaan antara *term* pada *keyword* dan dokumen, semakin besar nilai TF-IDF untuk kata tersebut sehingga perkalian *inner*

product juga menghasilkan nilai yang semakin tinggi. Maka semakin besar relevansi antara *keyword* dan dokumen.

DAFTAR PUSTAKA

- ADAWIYAH, R., dan RULDEVIYANI, Y., 2020. Evaluasi Maturity Level Pada Data Operations Management Menggunakan Capability Maturity Model Integration (CMMI): Studi Kasus Lembaga Penelaah Transaksi Keuangan. Masyarakat Telematika Dan Informasi: Jurnal Penelitian Teknologi Informasi Dan Komunikasi, 11(1), 29. <https://doi.org/10.17933/mti.v11i1.168>
- ASIH, S. N., FITRIANI, W. R., NABILA, R., HIDAYANTO, A. N., ISMED, I. H., & YUDHOATMOJO, S. B. 2019. Evaluation of Data Operations Management Maturity Level using CMMI in a State-Owned Enterprise. 2019 5th International Conference on Computing Engineering and Design (ICCED).
- ATRINAWATI, L. H., RAMADHANI, E., FIQAR, T. P., WIRANTI, Y. T., ABDULLAH, A. I. N. F., SAPUTRA, H. M. J., & TANDIRAU, D. B. 2021. Assessment of Process Capability Level in University XYZ Based on COBIT 2019. Journal of Physics: Conference Series, 1803(1). <https://doi.org/10.1088/1742-6596/1803/1/012033>
- Badan Pusat Statistik. Retrieved November 26, 2021, from https://www.bps.go.id/indikator/indikator/view_data_pub/0000/api_pub/UFpWMmJZOVZIZTJnc1pXaHhDV1hPQT09/da_01/1
- CHAUDHARY, M., & CHOPRA, A., 2017. CMMI for Development. In CMMI for Development. Apress. <https://doi.org/10.1007/978-1-4842-2529-5>
- CMMI Institute - CMMI Levels of Capability and Performance. Retrieved December 24, 2021, from <https://stage.cmmiinstitute.com/learning/appraisals/levels>
- DAMA. (2017). DAMA-DMBOK Data Management Body of Knowledge.
- ISACA. (2018a). COBIT® 2019 Framework : introduction and methodology.
- ISACA. (2018b). Governance and Management JELITA, A., 2007. Effective Technique for Indonesian Text Retrieval. Tesis, tidak diterbitkan. Australia : School of Computer Science and Information Technology, Science, Engineering,

- and Technology Portfolio, RMIT University
- FATTA, H., 2007. Analisis dan Perancangan Sistem Informasi untuk Keunggulan Bersaing Perusahaan dan Organisasi Modern. Yogyakarta : Andi.
- NIE, J., 2010. Cross-Language Information Retrieval. Toronto : Morgan & Claypool Publishers
- ZHANG, J., 2008. Visualization for Information Retrieval. Heidelberg : Springer.
- IRWANSYAH, E., & V. MONTIAGA, J., 2014. Pengantar Teknologi Informasi. Yogyakarta : Deepublish.
- JIMMY L GAOL, CHR., 2008. Sistem Informasi Manajemen Pemahaman dan Aplikasi. Jakarta : Grasindo.
- ADRIANI, M., ASIAN, J., NAZIEF, B., TAHAGHOGHI, S.M. AND WILLIAMS, H.E., 2007. Stemming Indonesian: A confix-stripping approach. ACM Transactions on Asian Language Information Processing (TALIP), 6(4), pp.1-33.
- JELITA, A., WILLIAMS, H., TAHAGHOGHI, S.M.M., 2007. Stemming Indonesian. School of Computer Science and Information Technology RMIT University, GPO Box 2476V, Melbourne 3001, Australia
- PRATAMA, S.E., DARMALAKSANA, W., MAYLAWATI, D.S.A., SUGILAR, H., MANTORO, T. AND RAMDHANI, M.A., 2020. Weighted inverse document frequency and vector space model for hadith search engine. Indones. J. Electr. Eng. Comput. Sci, 18(2), pp.1004-1014.
- KADHIM, A.I., 2019. Term weighting for feature extraction on Twitter: A comparison between BM25 and TF-IDF. In 2019 international conference on advanced science and engineering (ICOASE) (pp. 124-128). IEEE.
- NURDIN, A., SHODIK, A.F., 2019. Studi Hadis Teori dan Aplikasi. Bantul: Ladang Kata.
- UMAM, K. AND NEGARA, Y.D.P., 2022. Design And Implementation Search Engine Using Method VSM (Vector Space Model). International Journal of Science, Engineering and Information Technology, 7(1), pp.341-347.
- SHAHMIRZADI, O., LUGOWSKI, A. AND YOUNGE, K., 2019, December. Text similarity in vector space models: a comparative study. In 2019 18th IEEE international conference on machine learning and applications (ICMLA) (pp. 659-666). IEEE.
- PRADANA, A.W. AND HAYATY, M., 2019. The effect of stemming and removal of stopwords on the accuracy of sentiment analysis on indonesian-language texts. Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, pp.375-380.
- JALILIFARD, A., CARIDÁ, V.F., MANSANO, A.F., CRISTO, R.S., DA FONSECA, F.P.C., 2021. Semantic Sensitive TF-IDF to Determine Word Relevance in Documents. In: Thampi, S.M., Gelenbe, E., Atiquzzaman, M., Chaudhary, V., Li, KC. (eds) Advances in Computing and Network Communications. Lecture Notes in Electrical Engineering, vol 736. Springer, Singapore.
- TRIPATHY, A., AGRAWAL, A. AND RATH, S.K., 2016. Classification of sentiment reviews using n-gram machine learning approach. Expert Systems with Applications, 57, pp.117-126.
- RIZKI, A.S., TJAHYANTO, A. AND TRIALIH, R., 2019. Comparison of stemming algorithms on Indonesian text processing. TELKOMNIKA (Telecommunication Computing Electronics and Control), 17(1), pp.95-102.
- AL GHOFARI, K., ROZI, N.F., SELMAKARAMY, L. AND ARIANSYAH, F.Y., 2021, June. Pembuatan Sistem Pencarian Hadis dengan menggunakan Metode Pembobotan TF-IDF. In Prosiding Seminar Nasional Teknik Elektro, Sistem Informasi, dan Teknik Informatika (SNESTIK) (Vol. 1, No. 1, pp. 207-212).
- UMMAH, S.S., 2019. Digitalisasi Hadis (Studi Hadis Di Era Digital). Diroyah: Jurnal Studi Ilmu Hadis, 4(1).
- WIGUNA, G.S., 2022. Pembuatan Bot Telegram Untuk Layanan Pencarian Hadis Riwayat Sahih Bukhari. Jurnal Informatika Terpadu, 8(2), pp.109-116.
- AGRA, A., 2021. Analisis Vector Space Model (VSM) TF-IDF dengan Linear Search dan ORM Django pada pencarian data hadis (Doctoral dissertation, UIN Sunan Gunung Djati Bandung).