PENERAPAN METODE *TERM FREQUENCY INVERSE DOCUMENT FREQUENCY* (TF-IDF) DAN *COSINE SIMILARITY* PADA SISTEM TEMU KEMBALI INFORMASI UNTUK MENGETAHUI *SYARAH HADITS* BERBASIS WEB (STUDI KASUS: *SYARAH UMDATIL AHKAM*)

Ria Melita[1], Victor Amrizal[2], Hendra Bayu Suseno[3], Taslimun Dirjam[4]

[1,2]Program Studi Teknik Informatika, Fakultas Sains dan Teknologi
Universitas Islam Negeri Syarif Hidayatullah Jakarta
ria.melita17@gmail.com, ersebros@gmail.com, hendra.bayu@uinjkt.ac.id,
taslimundirjam@uinjkt.ac.id

ABSTRAK

Hadits merupakan sumber ajaran Islam di samping *Al-Qur'an*. Tanpa *hadits*, syari'at Islam tidak dapat dimengerti secara utuh dan tidak dapat dilaksanakan. Namun dewasa ini, tidak sedikit orang yang keliru dalam memahaminya, hal tersebut disebabkan oleh banyaknya orang yang memahami *hadits* sebatas mengandalkan teks lahiriyah saja. Salah satu hal yang dapat kita tempuh untuk mengetahui makna yang terkandung dalam *hadits* adalah dengan mempelajari *syarah hadits* guna meminimalisir kesalahan penafsiran terhadap suatu *hadits*. Sejauh ini aplikasi *syarah hadits* yang ada masih terbatas, yaitu dalam bahasa *full* arab yang tidak semua orang dapat memahaminya. Sedangkan untuk bahasa indonesia hanya ada lidwa dan arbain, namun masih sangat luas jangkauannya. Oleh karena itu, diperlukan jangkauan pemahaman dalam ilmu hadits dan dengan adanya sistem yang akan dibangun, maka akan memberikan solusi permasalahan tersebut, yaitu Sistem Temu Kembali Informasi yang dapat dimanfaatkan untuk mengetahui syarah hadits, karena dapat memberikan alternatif berupa metode *similarity* yang dapat digunakan untuk melakukan pencarian dokumen relevan dengan yang kita inginkan. Metode *similiarity* yang digunakan adalah *cosine similarity* dengan pembobotan kata menggunakan metode TFIDF dan menerapkan *teks preprocessing* terlebih dahulu untuk memperkecil *term* sehingga bisa mempercepat proses perhitungan *term. Teks preprocessing* tersebut meliputi *tokenizing, stopword removal* atau *filtering,* dan *stemming.* Hasil uji coba dengan pengujian *confusion matrix* didapatkan: *recall* 88.7%, *precision* 100%, *accuracy* 88,73 %, dan *error rate* 11,27 %.

Kata Kunci: *syarah, hadits, cosine similarity, tf-idf*

ABSTRACT

Hadith is a source of Islamic teachings besides the Qur'an. Without using the hadith, the syari'at of Islam can not be fully understood and can not be implemented. But today, many people are mistaken in understanding it, it is caused by the many people who understand the hadith to rely on text lahiriyah only. One of the things that we can take to know the meaning contained in the hadith is to study syarah hadith in order to minimize misinterpretation of a hadith. So far the application of syarh hadith is still limited. Because so far the existing applications are still full Arab language that not everyone can understand it. As for the Indonesian language there are only lidwa and arbain, but still very wide reach. Therefore, we need a system for the solution of the problem, that is Information Retrieval System which can be utilized because it provides an alternative in the form of similarity method that can be used to search documents relevant to what we want. The similiarity method used is cosine similarity with word weighting using TFIDF method and applying preprocessing text first to minimize term so that it can speed up the term calculation process. The preprocessing text includes tokenizing, stopword removal or filtering, and stemming. The results of testing with confusion matrix test obtained: 88.7% recall, precision 100%, accuracy 88.73%, and error rate 11.27%.

Keywords: *syarah, hadith, cosine similarity, tf-idf*

http://dx.doi.org/10.15408/jti.v11i2.8623

I. PENDAHULUAN

1.1 Latar Belakang

Hadits merupakan sumber ajaran Islam disamping *Al-Qur'an*. Tanpa menggunakan *hadits,* syari'at Islam tidak dapat dimengerti secara utuh dan tidak dapat dilaksanakan. [1]. *Hadits* sebagai sumber hukum kedua setelah *Al-Qur'an*, memiliki otoritas dan posisi penting yang nampaknya sudah tidak perlu dipertanyakan lagi, karena selain sebagai penjelas (*bayan*) bagi otoritas wahyu Allah (*Al-Qur'an*) juga sebagai sumber hukum Islam kedua yang menjadi rujukan para *fuqaha* yaitu ahli fiqih atau hukum islam. [2].

Pentingnya posisi *hadits* tehadap *Al-Qur'an* turut menempatkan posisi *Syarah Hadits* sebagai sesuatu yang tidak dapat diabaikan. Teks-teks *hadits* yang ada menjadi penjelas atas teks-teks *Al-Qur'an* yang dianggap masih bersifat global. *Syarah* memegang peranan penting untuk menjelaskan hal-hal yang dianggap masih umum, sulit dipahami, terlihat bertentangan, maupun hal-hal yang menyimpan keganjilan dalam teks-teks *hadits*. *Syarah Hadits* adalah keterangan tentang suatu maksud dalam suatu hadits. Tanpa keberadaan *syarah hadits*, susah kiranya untuk memahami makna sesungguhnya dari suatu *hadits* [3].

Hasil wawancara yang telah penulis lakukan dengan Bapak Fiqri Muhammad Fatkhi, MA. Selaku Ketua Program Studi Hadits Fakultas Ushuluddin UIN Syarif Hidayatullah Jakarta, didapatkan bahwa "saat ini tidak sedikit orang yang keliru dalam memahami *hadits* dikarenakan oleh beberapa faktor yang mempengaruhinya seperti memahami *hadits* sebatas teks lahiriyah *hadits* saja".

Menurut Bapak Fiqri, hal yang dapat kita tempuh untuk mengetahui makna yang terkandung dalam *hadits* adalah salah satunya dengan mempelajari *syarah hadits*, dimana mempelajari *syarah hadits* untuk mengetahui makna yang terkandung didalamnya ini sangat penting untuk dilakukan sepanjang diperoleh dari orang yang memiliki otoritas dibidang penjelasan *hadits*.

Merujuk pada penjelasan beliau, sejauh ini aplikasi *syarah hadits* yang ada masih terbatas, yaitu dalam bahasa *full* arab yang tidak semua orang dapat memahaminya. Sedangkan untuk bahasa indonesia sendiri baru ada lidwa dan arbain, namun masih sangat luas jangkauannya yaitu dari berbagai perawi dan tidak ada dalam bentuk *softcopy* untuk memudahkan dalam memahami suatu hadits.

Berdasarkan pemaparan diatas, untuk mengetahui *syarah hadits* guna meminima-lisir kesalahan penafsiran terhadap suatu *hadits*, maka penulis ingin melakukan penelitian dengan merancang suatu sistem temu kembali informasi untuk mengetahui *syarah hadits* dengan bahasa indonesia yang berpedoman pada kitab *Taysirul 'Allam Syarah 'Umdatil Ahkam* yang akan dibuat dengan menerapkan metode *Term Frequency Inverse Document Frequency* (TF-IDF) dan *Cosine Similarity*.

Taysirul 'Allam Syarah 'Umdatil Ahkam adalah kitab *syarah* yang sangat populer di kalangan penuntut ilmu di seluruh dunia, termasuk pesantren-pesantren di Indonesia yang merupakan kumpulan *hadits-hadits* riwayat imam Bukhari-Muslim—keduanya sudah sangat *masyhur* di telinga kaum Muslimin seantero dunia sebagai jaminan keshahihan suatu *hadits*.[4].

Sistem temu kembali informasi ini merupakan suatu sistem yang menemukan (*retrieve*) informasi yang sesuai dengan kebutuhan *user* berdasarkan *query* yang dimasukkan. Dengan begitu, penelitian ini diharapkan dapat memberikan *output* berupa s*yarah hadits* yang sesuai ber-dasarkan *input* yang dimasukkan dengan menggunakan Metode *Term Frequency Inverse Document Frequency* (TF-IDF) dan *Cosine Similarity*. Metode *Term Frequency Inverse Document Frequency* (TF-IDF) ini merupakan suatu cara untuk memberikan bobot hubungan suatu kata (*term*) terhadap dokumen. Sedangkan, Metode *cosine similarity* merupakan metode untuk menghitung kesamaan antara dua buah objek yang dinyatakan dalam dua buah *vector* dengan menggunakan *keywords* (kata kunci) dari sebuah dokumen sebagai ukuran [5].

Metode tf-idf ini penulis gunakan, karena merupakan metode pembobotan kata yang terkenal efisien, mudah dan memiliki hasil yang akurat.[6]. Sedangkan Metode *Cosine Similarity* penulis gunakan dengan alasan bahwa, metode ini mempunyai nilai akurasi yang tinggi dimana kelebihan utama dari metode *cosine similarity* adalah tidak terpengaruh pada panjang pendeknya suatu dokumen. Sehingga, dengan melakukan perbandingan *keyword* yang dihasilkan, maka

kedekatan antara item-pun dapat dipastikan. [7].

Penelitian terkait dengan menggunakan metode tf-idf dan cosine similarity ini telah dilakukan sebelumnya oleh Rizki Tri Wahyuni, Dhidik Prastiyanto, dan Eko Supraptono dalam jurnalnya yang berjudul " Penerapan *Algoritma Cosine Similarity* dan Pembobotan *Term Frequency Inverse Document Frequency* pada Sistem Klasifikasi Dokumen Skripsi" mengenai perancangan sistem yang dapat meng-klasifikasikan dokumen secara otomatis kedalam folder berbeda pada *database* agar lebih mudah dalam mengelola dokumen yang ada.

Penelitian yang dilakukan oleh Dewa Ayu, Arie Lumenta, dan Agustinus Jacobus dalam jurnalnya yang berjudul "Pengukuran Kemiripan Dokumen Teks Bahasa Indonesia Menggunakan Metode *Cosine Similarity*" yaitu penelitian yang mengukur kemiripan dokumen untuk mengetahui adanya plagiarisme terhadap suatu karya tulis.

Penelitian dengan judul "Sistem Temu Kembali Informasi pada Dokumen Teks Menggunakan Metode *Term Frequency Inverse Document frequency*" yang dilakukan oleh Dhony Syafe'i, Sukmawati, dan Nurdin Bahtiar, mengenai pembuatan suatu perangkat lunak yang dapat mencari dokumen-dokumen penulisan ilmiah yang relevan sesuai tingkat pembobotannya.

Dari ketiga penelitian terkait yang telah penulis sebutkan diatas, terdapat beberapa irisan yang penulis tekankan dalam penelitian tersebut yaitu dalam penggunaan *stemming* pada *text preprocessing*. Dimana penelitian yang dilakukan oleh Rizki Tri Wahyuni, dkk dan Dewa Ayu, dkk tidak melakukan *stemming* pada *text preprocessing* dan penelitian yang dilakukan oleh Dhony Syafe'i menggunakan *Stemming* dengan jenis *Porter Stemmer* model Fadhillah Z.Tala. Oleh karena itu, penulis menggunakan *Stemming* model Nazief dan Adriani pada *text preprocessing* untuk meningkatkan performa sistem yang akan dibuat, dimana *stemming* model ini merupakan *stemming* yang memiliki tingkat akurasi (presisi) lebih tinggi dibandingkan dengan *stemming* model lain. Penelitian yang penulis lakukan menggunakan metode yang sama yaitu *term frequency inverse document frequency dan cosine similarity* pada sistem temu kembali informasi, namun terdapat perbedaan yang mendasar dari semua penelitian yang ada yaitu bahwa penulis melakukan penerapan metode tersebut pada sistem temu kembali informasi untuk mengetahui *syarah hadits* yang belum pernah dilakukan sebelumnya.

Sistem temu kembali informasi ini akan dibuat dengan menggunakan aplikasi berbasis web dengan kelebihan tidak memerlukan proses instalasi, bersifat terpusat, dapat di-jalankan di os manapun asalkan memiliki *browser* dan akses internet, dan tidak perlu spesifikasi komputer *client* yang tinggi.

Berdasarkan permasalahan diatas, maka penulis tertarik untuk melakukan penelitian dengan judul "Penerapan Metode *Term Frequency Inverse Document Frequency* (Tf-Idf) dan *Cosine Similarity* pada Sistem Temu Kembali Informasi untuk Mengetahui *Syarah Hadits* Berbasis Web (Studi Kasus: Syarah Umdatil Hakam)".

1.2 Rumusan Masalah

Berdasarkan latar belakang yang dikemukakan, maka rumusan masalah yang ada pada penelitian ini adalah:

1. Bagaimana menerapkan metode *term frequency inverse document frequency* (tf-idf) dan *cosine similarity* pada sistem temu kembali informasi untuk mengetahui syarah hadits berbasis web?
2. Bagaimana menghitung nilai akurasi pada *Stemming* Nazief dan Adriani yang digunakan dalam *text preprocessing*?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

1. Menerapkan metode *term frequency inverse document frequency* (tf-idf) dan *cosine similarity* pada sistem temu kembali informasi untuk mengetahui syarah hadits berbasis web (studi kasus: hadits shahih Bukhari-Muslim).
2. Menghitung nilai akurasi pada *Stemming* Nazief dan Adriani yang digunakan dalam *text preprocessing*.

II. TINJAUAN PUSTAKA

2.1 *Syarah Hadits*

Syarah hadits adalah suatu aktivitas penyingkapan atau penjelasan atas makna-

makna dan pemahaman atas sesuatu yang disandarkan pada Rasulullah SAW baik yang berupa perkataan, perbuatan, ketetapan, maupun sifat-sifatnya. [8].

2.2 *Taysirul 'Allam Syarah Umdatil Ahkam*

Kitab Umdatil Ahkam ini merupakan pilihan dari *atsar-atsar* Nabi SAW paling shahih bersumber dari dua kitab agung; Shahih al-Bukhari dan Shahih Muslim yang dapat dijadikan rujukan dalam pembelajaran hadits bagi siapapun yang sudah mencapai tingkatan mahir, sekaligus tangga bagi pemula menuju kitab-kitab Islam yang diriwayatkan dari manusia terbaik, Rasulullah SAW.

Dibandingkan dengan kitab syarah lain seperti syarah Al-Allamah ahli ijtihad, dan Ibnu Daqiq Id yang beredar luas dimana metode pembahasan yang digunakan penulis rumit dan kuat, jauh diatas pemahaman sebagian besar para penuntut ilmu dan mereka yang mencari informasi, keutamaan dari kitab ini adalah menggunakan bahasa yang mudah dipahami, keterangannya tidak bertele-tele, ringkas, dan jelas bagian-bagiannya, agar permasalahan-permasalahan yang ada beserta penjelasannya tidak tumpang tindih dan bercampur, yang tentu saja akan membingungkan [9].

2.3 *Text Mining*

Text mining dapat didefinisikan secara luas sebagai suatu proses menggali informasi dimana seorang user berinteraksi dengan sekumpulan dokumen menggunakan *tools* analisis yang merupakan komponen-komponen dalam *data mining* yang salah satunya adalah kategorisasi. [10]

2.4 Sistem Temu Kembali Informasi

Information Retrieval (IR) merupakan bagian dari computer science yang berhubungan dengan pengambilan informasi dari dokumen-dokumen yang didasarkan pada isi dan konteks dari dokumen-dokumen itu sendiri. *Information Retrieval* merupakan suatu pencarian informasi (biasanya berupa dokumen) yang didasarkan pada suatu *query* (inputan *user*) yang diharapkan dapat memenuhi keinginan *user* dari kumpulan dokumen yang ada [11].

2.5 *Text Preprocessing*

Text Preprocessing merupakan tahapan dari proses awal terhadap teks untuk mempersiapkan teks menjadi data yang akan diolah lebih lanjut. Suatu teks tidak dapat diproses langsung oleh algoritma pencarian, oleh karena itu dibutuhkan *preprocessing text* untuk mengubah teks menjadi data *numeric* [12]. Proses ini terdiri dari beberapa tahap pembersihan dokumen seperti berikut [13]:

a. Tokenizing

Tokenizing adalah proses memecah dokumen menjadi kumpulan kata. *Tokenization* dapat dilakukan dengan menghilangkan tanda baca dan memisahkannya per-spasi. Tahapan ini juga menghilangkan karakter-karakter tertentu seperti tanda baca dan mengubah semua token ke bentuk huruf kecil (*lower case*).

b. Stopword Removal* atau *Filtering

Stopwords removal merupakan proses penghilangan kata tidak penting pada deskripsi melalui pengecekan kata-kata hasil parsing deskripsi apakah termasuk di dalam daftar kata tidak penting (*stoplist)* atau tidak.

***c. Stemming* Nazief & Adriani**

Stemming merupakan bagian yang tidak terpisahkan dalam *Information Retrieval* (IR). Tidak banyak algoritma yang dikhususkan untuk *stemming* bahasa Indonesia dengan berbagai keterbatasan didalamnya. *Algoritma Porter* misalnya, algoritma ini membutuhkan waktu yang relatif lebih singkat dibandingkan dengan *stemming* menggunakan algoritma Nazief dan Adriani, namun proses *stemming* menggunakan *algoritma Porter* memiliki persentase keakuratan lebih kecil dibandingkan dengan *stemming* menggunakan algoritma Nazief dan Adriani. Algoritma Nazief dan Adriani sebagai algoritma *stemming* untuk teks berbahasa Indonesia yang memiliki kemampuan persentase keakuratan lebih baik dari algoritma lainnya [14].

Langkah-langkah pada Algoritma Nazief & Adriani adalah sebagai berikut[15]:

1. Kata yang belum di-*stemming* dicari pada kamus. Jika kata itu langsung ditemukan, berarti kata tersebut adalah kata dasar. Kata tersebut dikembalikan dan algoritma dihentikan.
2. *Inflection Suffixes* ("-lah", "-kah", "-ku", "-mu", atau "-nya") dibuang. Jika berupa partikel ("-lah", "-kah", "-tah",

atau “-pun”) maka langkah ini diulangi lagi untuk menghapus Possesive Pronouns (“-ku”, “-mu”, atau “- nya”), jika ada.

3. Hapus *Derivation Suffixes* (“-i”, “-an”, atau “-kan”). Jika kata ditemukan di kamus, maka algoritma berhenti. Jika tidak maka ke langkah 3a.
   a. Jika “-an”, telah dihapus dan huruf terakhir dari kata tersebut adalah “-k”, maka “-k” juga ikut dihapus. Jika kata tersebut ditemukan dalam kamus kata dasar maka algoritma berhenti. Jika tidak ditemukan maka lakukan langkah 3b.
   b. Akhiran yang dihapus (“-i”, “-an”, atau “-kan”) dikembalikan, lanjut ke langkah 4.
4. Hapus *Derivation Prefix* DP (“di-”, “ke-”, “se-”, “me-”, “be-”, “pe-”, “te”) dengan iterasi maksimum adalah 3 kali.
   a. Langkah 4 berhenti jika :
   - Terjadi kombinasi awalan dan akhiran yang terlarang seperti pada Tabel 1 di bawah ini.

Tabel 1. Kombinasi awalan akhiran yang tidak diizinkan

Awalan	Akhiran yang tidak diizinkan
be-	-i
di-	-an
ke-	-i, -kan
me-	-an
se-	-i, -kan

   - Tiga awalan telah dihilangkan.

   b.Tipe awalan ditentukan melalui langkah-langkah berikut:
   1. Jika awalannya adalah : “di-”, “ke-”, atau “se-”, maka tipe awalannya secara berturut-turut adalah “di-”, “ke-”, atau “se-”.
   2. Jika awalannya adalah “te-”, “me-”, “be-”, atau “pe-”, maka dibutuhkan sebuah proses tambahan untuk menentukan tipe awalannya.
   3. Cari kata yang telah dihilangkan awalannya ini di dalam kamus. Apabila tidak ditemukan, maka langkah 4 diulangi kembali. Apabila ditemukan, maka keseluruhan proses berhenti.
5. Setelah tidak ada lagi imbuhan yang tersisa, maka algoritma ini dihentikan kemudian kata dasar tersebut dicari pada kamus, jika kata dasar tersebut ketemu berarti algoritma ini berhasil tapi jika kata dasar tersebut tidak ketemu pada kamus, maka dilakukan *recoding*.
6. semua langkah telah dilakukan tetapi kata dasar tersebut tidak ditemukan pada kamus juga maka algoritma ini mengembalikan kata yang asli sebelum dilakukan *stemming*.

2.6 *Term Frequency Inverse Document Frequency* (Tf-Idf)

Metode TF-IDF merupakan metode untuk menghitung bobot setiap kata yang paling umum digunakan pada information retrieval. Metode ini juga terkenal efisien, mudah dan memiliki hasil yang akurat[16].

Metode *Term Frequency-Inverse Document Frequency* (TF-IDF) adalah cara pemberian bobot hubungan suatu kata (*term*) terhadap dokumen. TF-IDF ini adalah sebuah ukuran statistik yang digunakan untuk mengevaluasi seberapa penting sebuah kata di dalam sebuah dokumen atau dalam sekelompok kata. Untuk dokumen tunggal tiap kalimat dianggap sebagai dokumen. Frekuensi kemunculan kata di dalam dokumen yang diberikan menunjukkan seberapa penting kata itu di dalam dokumen tersebut. Frekuensi dokumen yang mengandung kata tersebut menunjukkan seberapa umum kata tersebut. Bobot kata semakin besar jika sering muncul dalam suatu dokumen dan semakin kecil jika muncul dalam banyak dokumen. [17].

Pada algoritma TF-IDF digunakan rumus untuk menghitung bobot (W) masing masing dokumen terhadap kata kunci dengan rumus yaitu :

$$Wdt = tfdt * Idft \qquad (1)$$

Dimana:
Wdt= bobot dokumen ke-d terhadap kata ke-t
tfdt = banyaknya kata yang dicari pada sebuah dokumen
Idft = *Inversed Document Frequency* (log (N/df))
N = total dokumen
df = banyak dokumen yang mengandung kata yang dicari.

2.7 *Cosine Similarity*

Cosine Similarity adalah ukuran kesamaan antara dua buah vektor dalam sebuah ruang dimensi yang didapat dari nilai

cosinus sudut dari perkalian dua buah vektor yang dibandingkan karena cosinus dari 0 adalah 1 dan kurang dari 1 untuk nilai sudut yang lain, maka nilai *similarity* dari dua buah vektor dikatakan mirip ketika nilai dari *cosine similarity* adalah 1 [18].

Berikut adalah rumus *cosine similarity*; Persamaan.....(2)

Similarity= cos*(θ)*=

$$\frac{Q \bullet D}{|Q||D|} = \frac{\sum_{i=1}^{n}(wqi \times wdij)}{\sqrt{\sum_{i=1}^{n}(wqi)^2} \times \sqrt{\sum_{i=1}^{n}(wdij)^2}}$$

Keterangan :

Q • D = *dot product* antara vektor Q dan vektor D

|Q| = panjang vektor Q

|D| = panjang vektor D

|Q||D| = *cross product* antara |Q| dan |D|

wqi = bobot *term* pada *query* ke- i,= tf x idf

wdij = bobot *term* pada dokumen ke-i istilah ke-j = tf x idf

i = jumlah *term* dalam kalimat.

N = jumlah vektor.

Penulis memilih menggunakan metode *cosine similarity* dikarenakan metode ini mempunyai nilai akurasi yang tinggi dimana menurut [19] kelebihan utama dari metode *cosine similarity* adalah tidak terpengaruh pada panjang pendeknya suatu dokumen. Sehingga, dengan melakukan perbandingan *keyword* yang dihasilkan, maka kedekatan antara item-pun dapat dipastikan.

2.8 *Confusion Matrix*

Confusion Matrix adalah sebuah metode yang biasa digunakan untuk perhitungan akurasi. Dalam pengujian keakuratan hasil pencarian akan dievaluasi nilai *recall, precision, accuracy, dan error rate*. Dimana *precision* mengevaluasi kemampuan sistem untuk menemukan peringkat yang paling relevan, dan didefinisikan sebagai presentase dokumen yang di-*retrieve* dan benar-benar relevan terhadap *query*. *Recall* mengevaluasi kemampuan sistem untuk menemukan semua item yang relevan dari koleksi dokumen dan didefinisikan sebagai presentase dokumen yang relevan terhadap *query*. *Accuracy* merupakan perbandingan kasus yang diidentifikasi benar dengan jumlah seluruh kasus dan *error rate* merupakan kasus yang diidentifikasi salah dengan jumlah seluruh kasus [20].

Rumus *confusion matrix* adalah sebagai berikut:

Tabel 2. Rumus confusion matrix

Document	Nilai Sebenarnya	
	Relevant	**Non Relevant**
Retrieved	True Positive (tp) *Correct result*	False Positive (fp) *Unexpected result*
Not Retrieved	False Negative (fn) *Missing result*	True Negative (tn) *Corect absence of result*

Keterangan:

- TP (*True Positive*) = Jumlah prediksi yang benar dari data yang *relevant*.
- FP (*False Positive*) = Jumlah prediksi yang salah dari data yang tidak *relevant.*
- FN (*False Negative*) = Jumlah prediksi yang salah dari data yang tidak *relevant*.
- TN (*True Negative*) = Jumlah prediksi yang benar dari data yang *relevant*.

Sehingga, rumusnya adalah sebagai berikut:

Persamaan .....(3)

1. *Precision* = $\frac{tp}{(tp + fp)}$

2. *Recall* = $\frac{tp}{(tp + fn)}$

3. *Accuracy* = $\frac{(tp + tn)}{(tp + fp + tn + fn)}$

4. *Error rate* = $\frac{(fn + tn)}{(tp + fp + tn + fn)}$

Sistem yang dikatakan baik adalah sistem yang memiliki nilai *recall* dan *precision* tinggi.

III. METODOLOGI

3.1 Metode Pengumpulan Data

Dalam melaksanakan penelitian ini diperlukan data dan informasi terkait yang nantinya akan digunakan sebagai bahan rujukan untuk pembuatan penelitian serta mendukung keabsahan pembahasan pada laporan penelitian. Metode pengumpulan data yang digunakan yaitu:

1. Metode Studi Pustaka.

Dalam metode ini penulis melakukan studi pustaka dengan pencarian data secara manual juga melakukan pencarian data secara *online* melalui *browsing* internet dan melakukan pembelajaran terhadap jurnal dan skripsi yang relevan sesuai dengan topik penelitian yang

dijadikan sebagai bahan acuan dalam pembuatan penelitian.

Data yang didapatkan digunakan dalam penyusunan landasan teori, metodologi penelitian dan pengembangan sistem.

2. Metode Wawancara

Dalam metode wawancara ini penulis melakukan wawancara dengan Bapak Rifqi Muhammad Fatkhi, MA selaku Kepala Jurusan Program Studi Hadits Fakultas Ushuludin Universitas Islam Negeri Syarif Hidayatulllah Jakarta. Wawancara dilakukan 2 kali yaitu untuk mendukung permasalahan yang terkait, dan untuk pengujian sistem yang telah dibuat.

3.2 Metode Pengembangan Sistem

Dalam metode pengembangan system ini penulis menggunakan model *Rapid Aplication Development* (RAD) dengan tahapan sebagai berikut:

1. **Fase Perencanaan Syarat (*Requirement Planning)***

   Dalam fase ini penulis melakukan syarat-syarat perencanaan dengan langkah: Mengidentifikasi masalah, menganalisis solusi permasalahan, dan menganalisis kebutuhan sistem.

2. **Fase *Workshop* Desain (*Design Workshop)***

   Dalam fase ini penulis melakukan perancangan sistem, desain sistem dan pengkodean sistem.

3. **Fase Implementasi (*Implementation)***

   Fase ini merupakan fase terakhir setelah melalui fase-fase sebelumnya. Dimana dalam fase ini, hasil dari perancangan yang telah dilakukan akan diimplementasikan ke dalam sistem dengan langkah: Implementasi sistem, dan *testing*.

IV. HASIL DAN PEMBAHASAN

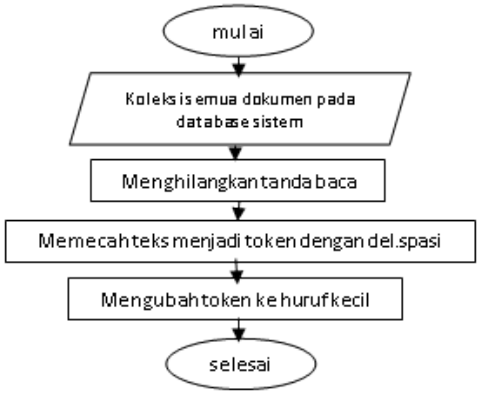
Perancangan sistem proses yang ada adalah teks *preprocessing* terhadap dokumen dan *query* dengan menerapkan metode yang digunakan yaitu dengan metode tf-idf untuk pembobotan kata dan *cosine similarity* untuk menemukan dokumen dengan menghitung kedekatan antar dokumen. Penjelasannya sebagai berikut:

4.1 Text Preprocessing

Sebelum melakukan tahap teks *preprocessing*, yang harus dilakukan adalah menyimpan semua dokumen yang akan dicari dalam sebuah koleksi dokumen. Dimana dokumen ini merupakan koleksi *hadits* bahasa indonesia yang akan digunakan sebagai *query* beserta *syarah*-nya (sebagai dokumen relevan) yang akan diproses dan disimpan dalam *database* MySQL. Selanjutnya tahap teks *preprocessing* dokumen, akan dijelaskan prosesnya sebagai berikut:

1. ***Tokenizing***

   Proses yang berjalan saat tahap *tokenizing,* akan digambarkan dalam *flowchart* berikut:



Gambar 1. *Flowchart tokenizing*

Gambar di atas menggambarkan proses yang dilakukan saat tahap *tokenizing*, dimana koleksi dokumen (*query* dan dokumen relevan) yang ada dalam *database* dilakukan penghilangan tanda baca, pemecahan teks menjadi token dengan delimeter spasi, dan pengubahan token yang terbentuk menjadi huruf kecil. Penerapan *tokenizing* penulis berikan pada salah satu contoh *hadits* dalam bab *thaharah*, dimana *hadits* yang diproses sebagai *query* harus memenuhi syarat untuk dapat diproses sistem, yaitu menggunakan *hadits* dengan struktur yang lengkap (terdapat *sanad* dan *matan*) karena tanpa salah satunya maka *hadits* tidak dapat diproses (untuk membedakan antara aplikasi *hadits* dengan aplikasi pengolahan teks yang lain). Hasil dari penerapan proses *tokenizing* adalah sebagai berikut:

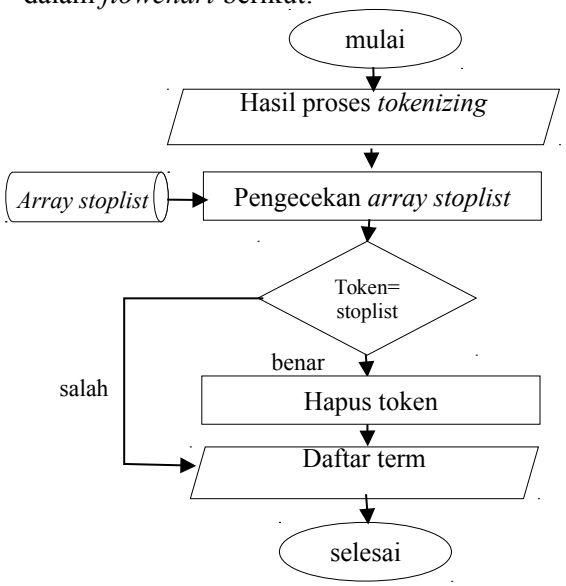
Tabel 3. Hasil tokenizing

Sebelum *Tokenizing*	**Sesudah *Tokenizing***
Dari Abu Hurairah ia berkata, “Rasulullah SAW bersabda: ‘Allah	dari abu hurairah

tidak menerima sholat seseorang di antara kallian ketika berhadats, hingga ia berwudhu'."	ia
	berkata
	rasulullah
	saw
	bersabda
	allah
	tidak
	menerima
	shalat
	seseorang
	diantara
	kalian
	ketika
	berhadats
	hingga
	ia
	berwudhu

2. *Stopword Removal* atau *Filtering*

Proses yang berjalan saat tahap *stopword removal* atau *filtering,* akan digambarkan dalam *flowchart* berikut:



Gambar 2. *Flowchart stopword removal*

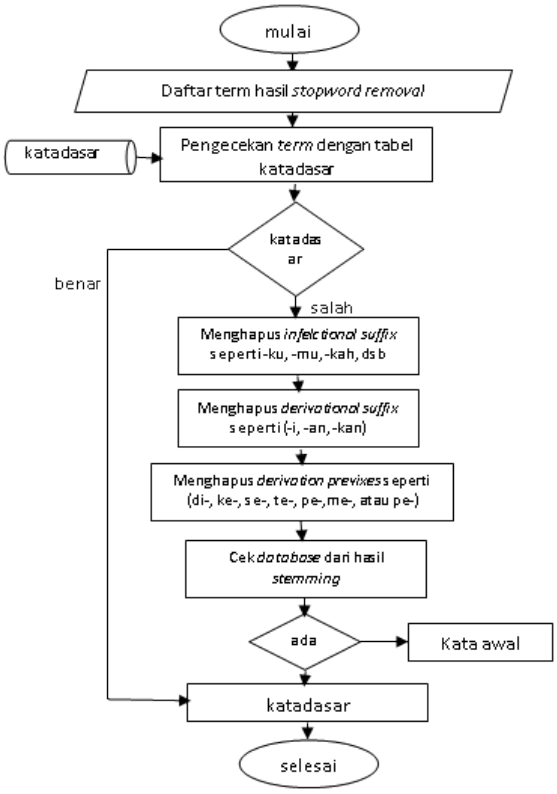
Gambar di atas menggambarkan proses yang dilakukan saat tahap *stopword removal* atau *filtering,* dimana hasil dari proses *tokenizing* yang dilakukan sebelumnya, akan di cocokan dengan *array stoplist* yang ada, apabila token yang dicek merupakan *stoplist* maka token akan dihapus, apabila token bukan termasuk *stoplist* maka *token* akan dibiarkan tetap ada. Contoh hasil penerapan *stopword removal* atau *filtering* adalah sebagai berikut:

Tabel 4. Hasil stopword removal

Sesudah *Tokenizing*	**Sesudah Proses *Stopword***
dari	-
abu	abu
hurairah	hurairah
ia	-
berkata	-
rasulullah	rasulullah
saw	saw
bersabda	bersabda
allah	allah
tidak	-
menerima	menerima
shalat	shalat
seseorang	seseorang
diantara	-
kalian	kalian
ketika	-
berhadats	berhadats
hingga	-
ia	-
berwudhu	berwudhu

3. *Stemming* Nazief Adriani

Adapun algoritma *stemming* yang akan digunakan yaitu *algoritma stemming* Nazief & Adriani, dimana *stemming* jenis ini merupakan *stemming* yang memiliki tingkat akurasi (presisi) lebih tinggi dari jenis *stemming* yang lain. Proses yang berjalan saat tahap *stemming*, akan digambarkan dalam *flowchart* berikut:



Gambar 3. *Flowchart stemming*

Gambar di atas menggambarkan proses yang dilakukan saat tahap *stemming*, dimana *term* hasil dari proses sebelumnya dilakukan pengecekan terhadap *database* kata dasar apakah *term* sudah kata dasar atau kata berimbuhan. Apabila term merupakan kata berimbuhan maka akan dilakukan *stemming* dengan melalui 3 tahapan yaitu menghapus *inflection suffix* (seperti -ku, -mu, -kah, dsb), menghapus *derivation suffix* (seperti -i, -an, atau -kan), dan menghapus *derivation prefix* (seperti di-, ke-,se-, dsb). Hasil dari proses stemming ini akan dilanjutkan dengan tahap berikutnya untuk dilakukan pembobotan kata menggunakan algoritma tf-idf.

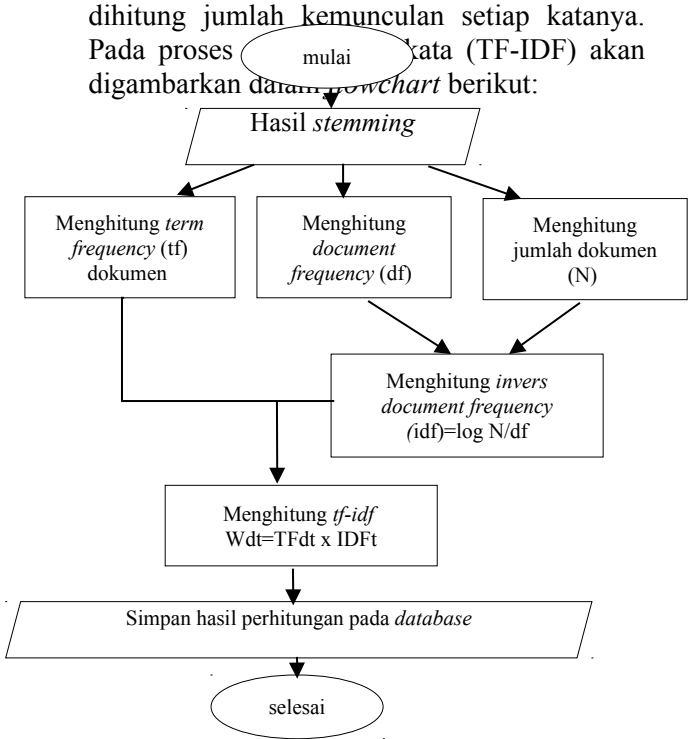
Hasil *stemming* dari algoritma Nazief Adriani adalah sebagai berikut:

Tabel 5. Hasil stemming

Sesudah Proses *Stopword*	**Hasil *Stemming***
abu	abu
hurairah	hurairah
rasulullah	rasulullah
saw	saw
bersabda	sabda
allah	allah
menerima	terima
shalat	shalat
seseorang	orang
kalian	kalian
berhadats	hadats
berwudhu	wudhu

4.2 Pembobotan Kata (Tf-Idf)

Setelah tahap teks *preprocessing* selesai, tahap selanjutnya adalah pembobotan kata (*term*). Dalam pembobotan kata (*term*) ini, setiap kata yang telah melewati proses *preprocessing*, akan di-*parsing* terlebih dahulu dan disimpan dalam *database*, kemudian dihitung jumlah kemunculan setiap katanya. Pada proses pembobotan kata (TF-IDF) akan digambarkan dalam *flowchart* berikut:



Gambar 4. *Flowchart Tf-Idf*

Gambar di atas menggambarkan tahap pembobotan kata dengan menggunakan metode *term frequency inverse document frequency* (TF-IDF), dimana daftar *term* hasil *stemming* dilakukan perhitungan untuk mengetahui bobot perkata dengan menghitung jumlah *term frequency* dokumen (tf) terlebih dahulu, kemudian menghitung nilai jumlah dokumen yang memiliki term (df), dan selanjutnya menghitung nilai idf dengan rumus log=N/df, dimana N merupakan jumlah seluruh dokumen yang ada. Setelah nilai TF dan IDF sudah didapat, maka langkah terakhir adalah menentukan bobot kata dengan mengalikan TF dan IDF dengan rumus Wdt=TFdt x IDFt. Hasil dari proses perhitungan ini disimpan dalam *database* dan akan dilanjutkan dengan tahap berikutnya untuk dilakukan perhitungan *cosine similarity* yang merupakan tahap akhir proses.

Contoh perhitungan pembobotan kata dalam penelitian ini menggunakan dokumen yang telah dilakukan teks *preprocessing* diatas (sebagai *query)* terhadap dua dokumen *output* untuk mengetahui kemiripannya, seperti berikut:

Diketahui : query => Dari Abu Hurairah ia berkata, “Rasulullah SAW bersabda: ‘Allah tidak menerima sholat seseorang di antara kallian ketika berhadats, hingga ia berwudhu’.”

Dimana terdapat 2 dokumen sebagai berikut:

Tabel 6. Dokumen output

d1	-Shalat orang yang berhadats tidak diterima, sampai ia bersuci dari hadats besar dan kecil. -Hadats membatalkan wudhu, dan membatalkan shalat jika terjadi di sela-selanya. -Maksud tidak diterima dalam hadits ini adalah shalatnya tidak sah. -Hadits ini menunjukkan, thaharah adalah syarat sah shalat.

d2	-Wajib memerhatikan anggota-anggota wudhu dan tidak boleh mengabaikan sedikitpun diantaranya. -Ancaman keras bagi yang tidak baik dalam berwudhu. -Kedua kaki wajib dibasuh saat berwudhu, seperti disebutkan dalam banyak dalil shahih dan ijmak umat.

Pembahasan: sebelum dilakukan pembobotan kata antara *query* dan dokumen yang harus dilakukan adalah melakukan teks *preprocessing* terhadap semua dokumen yang akan diproses terlebih dahulu agar kata siap untuk dilakukan pembobotan.

Setelah *preprocessing* maka hasil perhitungan tf-idf adalah sebagai berikut:

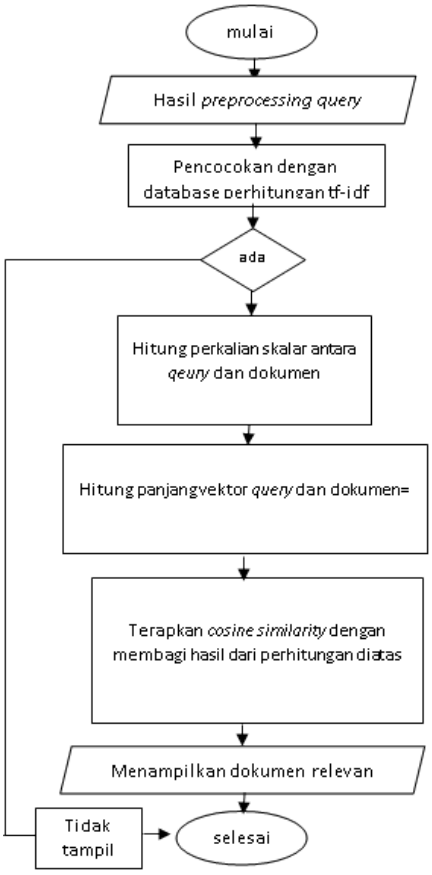
Tabel 7. Perhitungan Tf-Idf

term	**tf**			**df**	**Idf**	**Wdt= tf_{dt} x idf_t**		
	q	**d1**	**d2**		**Log (N/df)**	**q**	**d1**	**d2**
abai				1	1 0,477	0	0	0,477
abu	1				1 0,477	0,477	0	0
allah	1				1 0,477	0,477	0	0
ancaman				1	1 0,477	0	0	0,477
anggota				1	1 0,477	0	0	0,477
baik				1	1 0,477	0	0	0,477
banyak				1	1 0,477	0	0	0,477
basuh				1	1 0,477	0	0	0,477
batal		2			2 0,176	0	0,352	0
besar		1			1 0,477	0	0,477	0
dalil				1	1 0,477	0	0	0,477
dua				1	1 0,477	0	0	0,477
hadats	1	3			4 0,125	0,125	0,375	0
hadits		2			2 0,176	0	0,352	0
hati				1	1 0,477	0	0	0,477
hurairah	1				1 0,477	0,477	0	0
ijmak				1	1 0,477	0	0	0,477
kaki				1	1 0,477	0	0	0,477
kalian	1				1 0,477	0,477	0	0
kecil		1			1 0,477	0	0,477	0
keras				1	1 0,477	0	0	0,477
maksud		1			1 0,477	0	0,477	0
orang	1			1	2 0,176	0,176	0	0,176
rasulullah	1				1 0,477	0,477	0	0
sabda	1				1 0,477	0,477	0	0
sah		2			2 0,176	0	0,352	0
saw	1				1 0,477	0,477	0	0
sedikit				1	1 0,477	0	0	0.477
sela		1			1 0,477	0	0,477	0
shahih				1	1 0,477	0	0	0,477
shalat	1	4			5 0,221	0,221	0,884	0
suci		1			1 0,477	0	0,477	0
syarat		1			1 0,477	0	0,477	0
terima	1	2			3 0	0	0	0
thaharah		1			1 0,477	0	0,477	0
tunjuk		1			1 0,477	0	0,477	0
umat				1	1 0,477	0	0	0,477
wajib				1	1 0,477	0	0	0,477
wudhu	1	1		3	5 0,221	0,221	0,221	0,663

4.3 Cosine Similarity

Setelah melakukan pembobotan dokumen terhadap *term* dengan menggunakan perhitungan tf-idf, langkah terakhir yang dilakukan untuk menemukan dokumen yang relevan dengan *query* adalah menghitung

kemiripan antar dua dokumen dengan menggunakan rumus *cosine similarity* yang akan dijelaskan dalam *flowchart* berikut:



Gambar 5. *Flowchart cosine similarity*

Gambar di atas menggambarkan proses untuk menemukan dokumen yang relevan dengan *query user* menggunakan metode *cosine similarity*, dimana *query* yang dimasukkan *user* dilakukan tahap *preprocessing* yang hasilnya dicocokan dengan *database* bobot yaitu hasil perhitungan tf-idf, apabila term ditemukan maka akan dihitung perkalian skalar antara term *query* dengan dokumen dengan rumus wqi x wdij, selanjutnya yaitu menghitung nilai panjang setiap dokumen termasuk *query* dengan meng-kuadratkan bobot *query* dan bobot dokumen, jumlahkan nilai kuadrat dan selanjutnya diakarkan. Terakhir, membagi hasil dari perkalian skalar dan hasil panjang vektor yang sudah dihitung untuk menemukan hasil kemiripan antara *query* dengan dokumen, lalu sistem akan menampilkan dokumen yang relevan dengan *query* berdasarkan hasil perhitungan kemiripan dengan *cosine similarity* tersebut.

Contoh perhitungan berdasarkan hasil dari pembobotan kata yang telah dilakukan sebelumnya, untuk mengetahui kemiripan *query* dengan d1 yang nanti akan dibandingkan dengan d2, adalah sebagai berikut:

Tabel 8. Perhitungan cosine similarity

term	wq_i x wd_{ij}		Panjang vektor q dan d1, d2		
	Wqd1	Wqd2	wq^2	$wd1^2$	$wd2^2$
abai	0	0	0	0	0,227
abu	0	0	0,227	0	0
allah	0	0	0,227	0	0
ancaman	0	0	0	0	0,227
anggota	0	0	0	0	0,227
baik	0	0	0	0	0,227
banyak	0	0	0	0	0,227
basuh	0	0	0	0	0,227

batal	0	0	0	0,124	0
besar	0	0	0	0,227	0
dalil	0	0	0	0	0,227
dua	0	0	0	0	0,227
hadats	0,046	0	0,015	0,14	0
hadits	0	0	0	0,124	0
hati	0	0	0	0	0,227
hurairah	0	0	0,227	0	0
ijmak	0	0	0	0	0,227
kaki	0	0	0	0	0,227
kalian	0	0	0,227	0	0
kecil	0	0	0	0,227	0
keras	0	0	0	0	0,227
maksud	0	0	0	0,227	0
orang	0	0,03	0,03	0	0,03
rasulullah	0	0	0,227	0	0
sabda	0	0	0,227	0	0
sah	0	0	0	0,124	0
saw	0	0	0,227	0	0
sedikit	0	0	0	0	0.227
sela	0	0	0	0,227	0
shahih	0	0	0	0	0,227
shalat	0,195	0	0,048	0,781	0
suci	0	0	0	0,227	0
syarat	0	0	0	0,227	0
terima	0	0	0	0	0
thaharah	0	0	0	0,227	0
tunjuk	0	0	0	0,227	0
umat	0	0	0	0	0,227
wajib	0	0	0	0	0,227
wudhu	0,048	0,146	0,048	0,048	0,439

Jumlahkan hasil perkalian diatas		Jumlahkan hasil kuadrat diatas		
0,289	0,176	1.73	3.157	3.874

Akarkan hasil penjumlahan di atas		
1,315	1,776	1,968

Setelah diketahui nilai dari masing-masing *query* dan dokumen, selanjutnya adalah dengan menerapkan rumus *cosine similarity* sebagai berikut:

1. Cos (q, d1) = $$\frac{\sum_{i=1}^{n}(\mathbf{wqi\ x\ wd1})}{\sqrt{\sum_{i=1}^{n}(wqi)^2}\ x\ \sqrt{\sum_{i=1}^{n}(wd\mathbf{1})^2}}$$

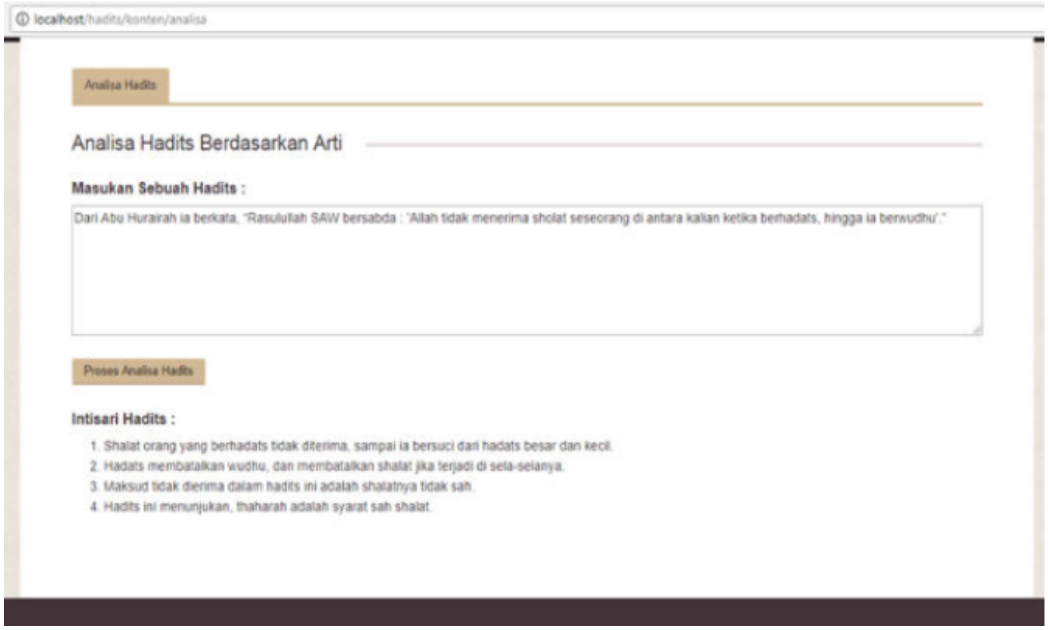
$$= \frac{0,28\mathbf{9}}{1,315\ x\ 1,776} = \frac{0,28\mathbf{9}}{2,335} = 0,123$$

2. Cos (q, d2) = $$\frac{\sum_{i=1}^{n}(\mathbf{wqi\ x\ wd2})}{\sqrt{\sum_{i=1}^{n}(wqi)^2}\ x\ \sqrt{\sum_{i=1}^{n}(wd\mathbf{2})^2}}$$

$$= \frac{0,17\mathbf{6}}{1,315\ x\ 1,968} = \frac{0,17\mathbf{6}}{2,58\mathbf{8}} = 0,068$$

Berdasarkan hasil dari perhitungan *cosine similarity* diatas, hasil dari persamaan antara *query* dan d1 adalah 0,123 sedangkan hasil dari persamaan antara *query* dan d2 adalah 0,068. Maka dapat disimpulkan bahwa dokumen yang terdekat dengan *query* adalah dokumen 1 (d1). Hal tersebut dikarenakan hukum *cosine similarity* adalah semakin besar nilai *cosinus* (maksimal 1) maka semakin mirip dokumen dengan yang dibandingkan. Oleh karena itu, yang lebih mirip dengan *query* dan menjadi *output* dari *query* adalah d1 karena lebih mendekati angka 1 dibandingkan dengan dokumen 2 (d2).

4.4 Hasil Penelitian



Gambar 6. Halaman analisis syarah

Gambar di atas merupakan tampilan dari halaman analisis *hadits* yang muncul ketika *user* mengklik tombol analisis hadits untuk melakukan proses analisis yaitu dengan mengisi *hadits* dalam Bahasa Indonesia dengan lengkap (sanad dan matan) pada kolom yang tersedia lalu setelah klik tombol proses analisis hadits maka akan muncul *syarah hadits* yang benar sesuai dengan hadits yang dimasukkan, sistem tidak akan memberikan *output* apabila *hadits* yang diinputkan tidak mengandung salah satu struktur *hadits* (*sanad* atau *matan*).

4.5 Pengujian

1. Pengujian *Stemming*

Adapun algoritma *stemming* yang digunakan dalam penelitian ini adalah algoritma Nazief & Adriani, dimana *stemming* jenis ini merupakan *stemming* yang memiliki tingkat akurasi (presisi) lebih tinggi dari jenis *stemming* yang lain. Namun, pada proses *stemming* algoritma ini, penulis menemukan beberapa kata yang tidak berhasil distemming pada saat melakukan analisis secara manual, kata yang gagal di-*stemming* tersebut terjadi karena proses penghilangan imbuhan (akhiran) yang dilakukan terlebih dahulu setelah itu baru penghilangan awalan sehingga terjadi *overstemming*, lalu tidak adanya beberapa aturan dalam *stemming* nazief adriani seperti pe- + –ng, dan ter- sehingga untuk beberapa kata dengan aturan tersebut tidak berhasil di-*stemming* (kata tidak berubah), juga beberapa aturan yang masih belum berhasil mengembalikan kata ke dalam bentuk kata dasarnya. Contoh beberapa kata yang gagal dilakukan *stemming* terlihat dalam Tabel 9 di bawah ini:

Tabel 9. Contoh hasil stemming yang kurang tepat

Id	Token Awal	Hasil *Stemming*	Hasil seharusnya
36	dinikahinya	nikahi	nikah
91	diampuni	mpuni	ampuni
236	memberkahi	ber	berkah
1897	sejumlah	sejum	jumlah
2226	setelah	sete	telah
2520	melakukan	laku	lakukan
2875	melangkah	lang	langkah
4691	memisah	misah	pisah
2128	memerintah	merintah	perintah
4066	terlimpah	terlimpah	limpah
4162	terpuji	terpuji	puji
4372	terakhir	terakhir	akhir
4753	termasuk	termasuk	masuk
3723	pengendara	pengendara	kendara
4062	pengagungan	pengagungan	agung

Selain itu, pada kata yang merupakan bahasa asing atau nomor, tidak dilakukan *stemming*, hal ini dikarenakan cara kerja yang pertama dari algoritma Nazief Adriani adalah dengan melakukan pengecekan apakah kata yang akan di-*stemming* terdapat pada kamus yang tersedia atau tidak, jika iya maka akan dilakukan *stemming*, tapi jika tidak ada maka kata akan dianggap sebagai *root word* atau sebagai kata asal, sehingga kata asing dan nomor seperti: *Musyik, qirath*, jihad, *mutalaffiat, khadaq,itikaf, shaum, qadha, haruriyah, istihadhah, istinja, madzi, khitan, junub, syafaat, subhanallah, jinabat, saw, rasulullah*, 1,2,3,4,5 dan sebagainya akan tetap menjadi seperti asalnya.

Berdasarkan analisis hasil *stemming* di atas, untuk mengetahui tingkat ke akuratan *Stemming* dengan Algoritma Nazief Adriani ini maka akan dihitung nilai persentase antara kata yang gagal di-*stemming* dengan yang berhasil. Total token yang ada pada *database* hasil *preprocessing* adalah 6.983, namun dalam 6.983 itu terdapat banyak pengulangan beberapa token, sehingga untuk pehitungan hasil *stemming* kata yang terbentuk hanya berjumlah 2.051. Untuk itu, hasil dari akurasi *stemming* Nazief Adriani adalah sebagai berikut:

1. Prosentase gagal = $\frac{186}{2051} x\ 100\% = 9{,}07\ \%$
2. Prosentase berhasil = $\frac{1865}{2051} x\ 100\% = 90{,}93\ \%$

2 Pengujian Sistem

Dalam fase *testing* ini untuk melakukan pengujian sistem, penulis menggunakan pengujian *confusion matrix* yang biasa digunakan dalam perhitungan akurasi pada suatu sistem temu kembali informasi untuk mengevaluasi seberapa baik kemampuan sistem dalam pencarian dokumen. Pengujian sistem ini adalah dengan melakukan percobaan sebanyak 204 kali menggunakan *query* terhadap 204 dokumen yang ada dalam *database. Query* yang digunakan adalah *query* yang memiliki jumlah dokumen relevan masing-masing 1 untuk masing-masing *query*. Artinya, satu *query* hanya relevan dengan satu dokumen dimana dokumen tersebut merupakan *syarah hadits* yang relevan berdasarkan *query* yang dimasukkan.

Setelah melakukan percobaan terhadap 204 dokumen dengan *query* tersebut, didapatkan hasil benar yang ditemukan berjumlah 181 dokumen yang ditemukan relevan (sesuai dengan *query*), dan ditemukan 23 dokumen yang ditemukan tidak relevan (tidak sesuai dengan *query*). Oleh karena itu, maka hasil pengujian *confusion matrix* sesuai dengan rumus yang telah dibahas pada *point* 2.8 Tabel 2 adalah sebagai berikut:

1. *Precision* $= \frac{181}{(181+0)} x\ 100\% = 100\%$
2. *Recall* $= \frac{181}{(181+23)}$ x 100% = 88,7%

3. *Accuracy* = $\frac{(181+0)}{(181+0+0+23)}$ x 100% = 88,73%
4. *Error rate* = $\frac{(23+0)}{(181+0+0+23)}$ x 100% = 11,27 %

V. PENUTUP

5.1 Kesimpulan

Berdasarkan rumusan masalah yang telah dikemukakan, maka kesimpulan dalam penelitian ini adalah bahwa metode *term frequency inverse document frequency* (tf-idf) dan *cosine similarity* telah berhasil diterapkan dalam sistem dengan baik dimana sistem dapat memberikan *output* berupa dokumen yang relevan yaitu *syarah hadits* sesuai dengan *query* yang di-input kan, dengan melalui 3 tahapan teks *preprocessing* yaitu *tokenizing, stopword removal* atau *filtering*, dan *stemming*. Hasil pengujian stemming model Nazief Adriani yang telah dilakukan menunjukkan hasil akurasi sebesar 90,93% yang menyatakan bahwa *stemming* model ini memiliki tingkat akurasi yang tinggi. Kemudian, pengujian sistem yang dilakukan dengan menggunakan *confusion matrix* dalam penelitian ini didapatkan nilai *precision* 100%, *recall* 88,7%, *accuracy* 88,73 %, dan *error rate* 11,27 %. Sehingga, sistem dapat dikatakan baik, dikarenakan sistem yang baik adalah sistem yang memiliki nilai *recall* dan *precision* tinggi serta tingkat akurasi yang tinggi pula**.**

5.2 Saran

Untuk pengembangan selanjutnya, maka saran-saran yang dapat penulis berikan adalah sebagai berikut:

1. Sistem dapat dilengkapi menjadi *full* satu kitab dan dapat dikembangkan dengan menggunakan kitab *syarah hadits* yang lebih spesifik lainnya seperti kitab *syarah Fatkhul Barri*, *syarah Imam An-Nawawi* atau kitab *syarah* dari *hadits shahih* periwayat lain.
2. Sistem dapat dikembangkan dengan *user interface* yang lebih menarik dan *user friendly.*
3. Penerapan metode pembobotan kata dan metode *similarity* yang memiliki tingkat akurasi lebih tinggi.
4. Sistem dapat dikembangkan selain menggunakan manajemen *database* MySql, karena dengan menggunakan MySql sistem berjalan sedikit lebih lambat dikarenakan sistem harus menelusuri data yang ada sehingga waktu untuk menemukan dokumen yang relevan relatif lama.

DAFTAR PUSTAKA

[1] Zuhri, Muhammad. 2011. Hadis Nabi Telaah Historis dan Metodologis. Yogyakarta: Tiara Wacana Yogya.

[2] Rosyid, Khoirul. 2016. Kepemimpinan Menurut Hadits Nabi Saw. Skripsi. Jurusan Tafsir Hadits. Fakultas Ushuluddin. Institut Agama Islam Negeri (IAIN) Raden Intan Lampung.

[3] Mukaromah, Kholila. 2015. Kajian Syarah Hadits Subul Al-Salam. Tesis. Jurusan Studi Agama dan Filsafat. Fakultas Humaniora. Universitas Islam Negeri Sunan Kalijaga. Yogyakarta.

[4] Katsir, Ibnu. 2013. Tafsirul ‘Allam Syarh ‘Umdatul Ahkam. Jakarta: Ummul-Qura.

[5] Rizki, Dhidik, dan Eko Supraptono. 2017. Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen. Skripsi. Jurusan Teknik Elektro. Fakultas Teknik. Universitas Negeri Semarang Kampus Sekaran, Gunungpati, Semarang.

[6] Ma’arif, Abdul Aziz. 2015. Penerapan Algoritma Tf-Idf Untuk Pencarian Karya Ilmiah. Jurnal. Jurusan Teknik Informatika. Fakultas Ilmu Komputer. Universitas Dian Nuswantoro Semarang.

[7] Rozas, I. R. dan Sarno, R. 2011. Sistem Pemilihan Kontrol Keamanan Informasi Berbasis ISO 27001. Seminar Nasional Pascasarjana XI. IT. Surabaya.

[8] Mukaromah, Kholila. 2015. Kajian Syarah Hadits Subul Al-Salam. Tesis. Jurusan Studi Agama dan Filsafat. Fakultas Humaniora. Universitas Islam Negeri Sunan Kalijaga. Yogyakarta.

[9] Katsir, Ibnu. 2013. Tafsirul ‘Allam Syarh ‘Umdatul Ahkam. Jakarta: Ummul-Qura.

[10] Feldman, Ronen, dan Sanger, James. 2007. The Text Mining Handbook Advanced Approaches in Analyzing Unstructured Data. New York: Cambridge University Press.

[11] Safitri, Rima Noer. 2013. Temu Kembali Informasi Pada Pencarian Jurnal Skripsi Menggunakan Metode Single Pass Clustering. Skripsi. Universitas Muhammadiyah Gresik.
[12] Feldman, Ronen, dan Sanger, James. 2007. The Text Mining Handbook Advanced Approaches in Analyzing Unstructured Data. New York: Cambridge University Press.
[13] Rizki, Dhidik, dan Eko Supraptono. 2017. Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen. Skripsi. Jurusan Teknik Elektro. Fakultas Teknik. Universitas Negeri Semarang Kampus Sekaran, Gunungpati, Semarang.
[14] Arwanda, Ivan. 2013. Penerapan Metode Text Mining pada Aplikasi Chatbot. Skripsi. Jurusan Teknik Informatika Universitas Ilmu Komputer. Bandung.
[15] Simorangkir, Manase Sahat. 2017. Studi Perbandingan Algoritma-Algoritma Stemming Untuk Dokumen Teks Bahasa Indonesia. Jurnal. Teknik Informatika. Universitas Presiden Jababeka Education.
[16] Ma'arif, Abdul Aziz. 2015. Penerapan Algoritma Tf-Idf Untuk Pencarian Karya Ilmiah. Jurnal. Jurusan Teknik Informatika. Fakultas Ilmu Komputer. Universitas Dian Nuswantoro Semarang.
[17] Putra, Agung Auliaguntary Arif. 2016. Implementasi Text Summarization Menggunakan Metode Vector Space Model pada Artikel Berita Bahasa Indonesia. Skripsi. Jurusan Teknik Informatika. Fakultas Teknik dan Ilmu Komputer. Universitas Komputer Indonesia.
[18] Dewa, Arie, dan Agustinus. 2016. Pengukuran Kemiripan Dokumen Teks Bahasa Indonesia Menggunakan Metode Cosine Similarity. Jurnal. Teknik Informatika Universitas Sam Ratulangi Manado.
[19] Rozas, I. R. dan Sarno, R. 2011. Sistem Pemilihan Kontrol Keamanan Informasi Berbasis ISO 27001. Seminar Nasional Pascasarjana XI. IT. Surabaya.
[20] Ferdinandus, Subari. 2015. Sistem Information Retrieval Layanan Kesehatan Untuk Berobat dengan Metode Vector Space Model berbasis WebGis. Jurnal. Teknik Informatika. STIKI Malang.