

Implementasi Sistem Temu Kembali Informasi Pencarian Hadis Menggunakan Metode Jaccard Similarity dan Algoritma Pembobotan TF-IDF

Sabriyanti¹, Syarifah Putri Agustini Alkadri², Sucipto³

¹ Program Studi Teknik Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Muhammadiyah Pontianak and sabriyanti26@gmail.com

² Program Studi Teknik Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Muhammadiyah Pontianak and agustini.putri@unmuhpnk.ac.id

³ Program Studi Teknik Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Muhammadiyah Pontianak and sucipto@unmuhpnk.ac.id

ABSTRAK

Pencarian hadis secara manual masih menjadi kendala di banyak lembaga pendidikan Islam karena jumlah hadis yang besar dan variasi redaksi yang menyulitkan pencarian berbasis kata kunci. Penelitian ini mengembangkan sistem pencarian hadis berbasis web yang mampu menyajikan hasil secara relevan dan efisien menggunakan metode *Jaccard Similarity* dengan dukungan pembobotan *TF-IDF*. Sistem bekerja melalui tahapan *preprocessing* yang meliputi *case folding*, *tokenisasi*, *filtering*, *stopword removal*, dan *stemming*, kemudian dilanjutkan dengan pembobotan *TF-IDF* serta perhitungan kemiripan antara *query* dan dokumen hadis. Pengujian dilakukan terhadap 1.150 data hadis di Pondok Pesantren Darussalam Sengkubang. Hasil evaluasi menggunakan *confusion matrix* menunjukkan *precision* 72%, *recall* 95%, dan *F1-score* 82%. Pengujian *black box* memastikan seluruh fitur berjalan dengan baik, sementara *UAT* yang melibatkan 30 pengguna dengan 10 pertanyaan menghasilkan nilai 82,33% dalam kategori sangat layak. Sistem ini dinilai mampu membantu pencarian hadis secara lebih efektif dan mendukung proses pembelajaran yang lebih praktis.

Kata Kunci: Temu Kembali Informasi Hadis, TF-IDF, Jaccard Similarity, User Acceptance Testing, Lembaga Pendidikan Islam

ABSTRACT

Manual hadith search remains a challenge in many Islamic educational institutions due to the large number of hadiths and the variety of wordings, which make keyword-based searches difficult. This study develops a web-based hadith search system capable of presenting relevant and efficient results using the Jaccard Similarity method with TF-IDF weighting. The system operates through preprocessing stages that include case folding, tokenization, filtering, stopword removal, and stemming, followed by TF-IDF weighting and similarity calculations between the query and hadith documents. Testing was conducted on 1,150 hadith data points at the Darussalam Sengkubang Islamic Boarding School. Evaluation results using a confusion matrix showed a precision of 72%, a recall of 95%, and an F1-score of 82%. Black-box testing confirmed that all features functioned properly, while a User Acceptance Test (UAT) involving 30 users with 10 questions yielded a score of 82.33% in the highly feasible category. This system is considered capable of facilitating more effective hadith searches and supporting a more practical learning process.

Keywords: Hadith Information Retrieval, TF-IDF, Jaccard Similarity, User Acceptance Testing, Islamic Educational Institution

PENDAHULUAN

Dalam praktik keilmuan Islam, hadis menjadi salah satu sumber utama dalam memahami ajaran Islam setelah Al-Qur'an. Hadis menjelaskan ayat-ayat Al-Qur'an secara lebih rinci karena Al-Qur'an sebagai kitab suci dan pedoman hidup umat Islam umumnya memuat ajaran yang perlu dipahami dan dijelaskan lebih lanjut agar mudah diamalkan (Syakhrani & Hidayah, 2022). Sebagai

pedoman hidup umat Islam, hadis mencakup berbagai aspek ajaran, mulai dari akidah, ibadah, akhlak, muamalah, hingga adab. Dengan kedudukan yang sangat penting tersebut, kemudahan dalam mempelajari dan mengakses hadis menjadi kebutuhan mendasar. Seiring perkembangan teknologi informasi, penyediaan hadis dalam bentuk digital juga semakin berkembang (Ridwan et al., 2021).

Meskipun demikian, jumlah hadis yang sangat banyak dan penyebarannya dalam berbagai kitab membuat proses pencarian hadis tidak selalu efisien. Pencarian melalui media digital seperti e-book pada umumnya masih bergantung pada kecocokan kata kunci secara langsung. Kondisi tersebut mengharuskan pengguna menelusuri isi kitab secara manual tanpa jaminan kecocokan yang optimal, sehingga diperlukan sistem yang mampu memberikan hasil pencarian berdasarkan tingkat relevansi secara lebih efektif.

Berdasarkan permasalahan tersebut, dibutuhkan sistem pencarian atau temu kembali informasi (*Information Retrieval*) yang mampu menyajikan hasil pencarian hadis secara cepat, relevan, dan tidak terbatas pada kesamaan kata secara literal (Elsadantia, 2023). Penelitian ini menggunakan metode *Jaccard Similarity* dengan dukungan pembobotan *TF-IDF* (*Term Frequency-Inverse Document Frequency*) sebagai tahap representasi *term* sebelum perhitungan kemiripan. Pemilihan *TF-IDF* didasarkan pada kemampuannya dalam memberikan bobot pada setiap kata dalam dokumen, sehingga kata yang sering muncul namun kurang penting, seperti kata hubung, tidak terlalu berpengaruh, sedangkan kata kunci yang lebih bermakna memperoleh bobot lebih tinggi (Mohammed & Rashid, 2023). Sementara itu, *Jaccard Similarity* dipilih karena kesederhanaannya dalam mengukur tingkat kemiripan antara dua himpunan kata, yaitu antara *query* pengguna dan dokumen hadis yang telah diproses. Kombinasi kedua metode ini memungkinkan proses pencarian tidak hanya bergantung pada kecocokan kata, tetapi juga mempertimbangkan relevansi *term* yang terbentuk agar hasil pencarian menjadi lebih relevan dan efisien (Apriani et al., 2021).

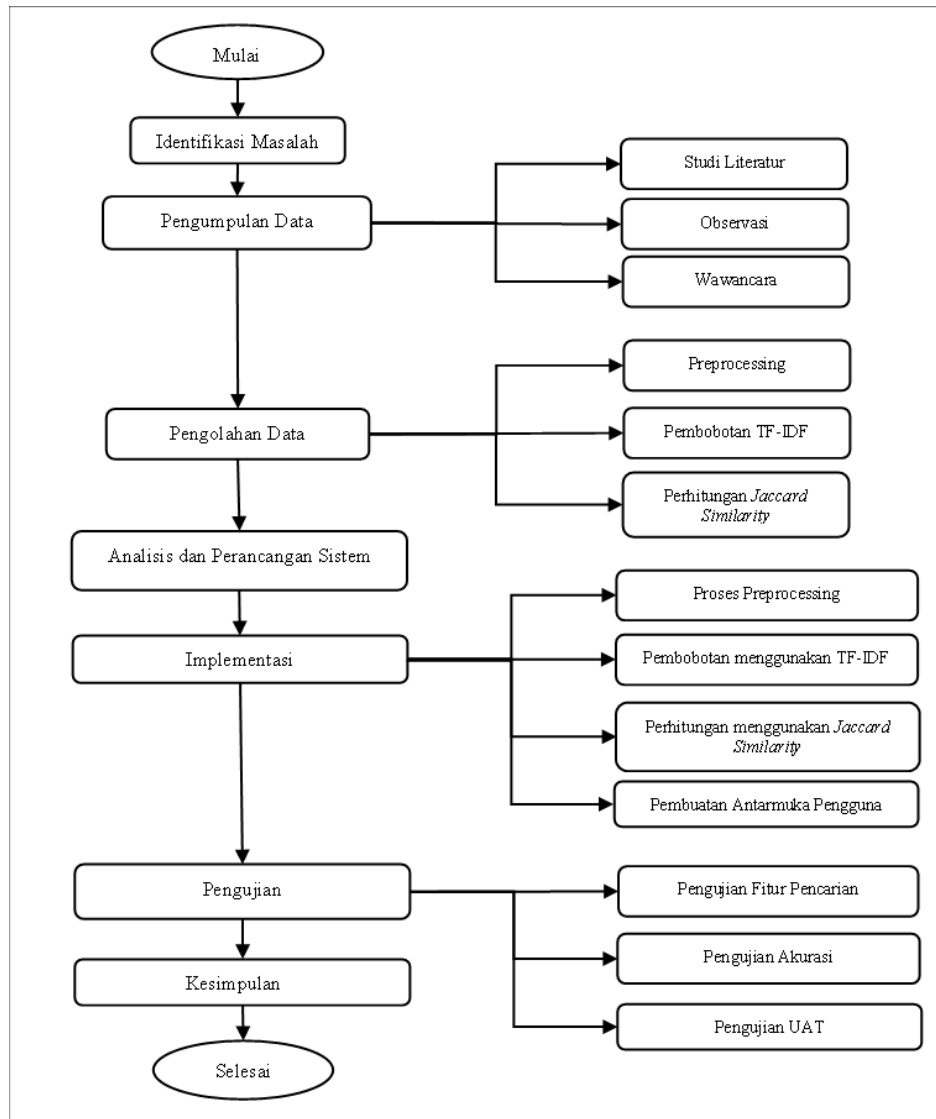
Proses pencarian hadis dilakukan melalui beberapa tahapan, yaitu *tokenisasi*, *filtering*, *stopword removal*, dan *stemming*. Setiap teks hadis maupun *query* dari pengguna diubah menjadi bentuk vektor yang berisi kata-kata penting dalam dokumen (Yunita, 2023). Melalui pembobotan *TF-IDF* dan perhitungan kemiripan menggunakan *Jaccard Similarity*, sistem dapat menyajikan daftar hadis yang paling mendekati *query* pengguna. Dengan sistem ini, pengguna tidak perlu membaca satu per satu isi kitab hadis untuk menemukan hadis yang relevan, sehingga proses pembelajaran, penelitian, maupun kebutuhan pribadi dalam menggali ajaran Islam melalui hadis dapat dilakukan secara lebih mudah dan modern. Dalam penelitian ini, *TF-IDF* tidak digunakan secara langsung dalam perhitungan kemiripan, tetapi dimanfaatkan sebagai tahap pembobotan untuk menyeleksi dan merepresentasikan *term* yang lebih penting sebelum perhitungan *Jaccard Similarity* dilakukan.

METODE PENELITIAN

Dalam penelitian ini, pembobotan *TF-IDF* digunakan untuk mengidentifikasi dan merepresentasikan *term* yang memiliki tingkat kepentingan tinggi dalam dokumen. Hasil pembobotan tersebut tidak secara langsung digunakan dalam perhitungan kemiripan, tetapi dimanfaatkan sebagai dasar dalam proses seleksi *term* sehingga hanya kata-kata yang lebih relevan yang dipertahankan pada tahap selanjutnya.

Selanjutnya, perhitungan tingkat kemiripan antara *query* dan dokumen dilakukan menggunakan metode *Jaccard Similarity*, yaitu dengan membandingkan kesamaan himpunan *term*

hasil *preprocessing* dan seleksi *term*. Hasil pengolahan tersebut digunakan sebagai dasar dalam perancangan dan pembangunan sistem. Sistem yang dikembangkan kemudian diimplementasikan dan diuji melalui pengujian fungsional, pengujian akurasi, serta pengujian penerimaan pengguna. Penelitian diakhiri dengan penarikan kesimpulan berdasarkan hasil pengujian yang telah dilakukan.



Gambar 1. Diagram Alir Penelitian

A. Pengumpulan Data

Pengumpulan data dalam penelitian ini dilakukan melalui beberapa cara. Studi literatur digunakan dengan mempelajari jurnal, buku, dan penelitian sebelumnya yang berkaitan dengan sistem temu kembali informasi serta metode *TF-IDF* dan *Jaccard Similarity* sebagai dasar teori penelitian. Selain itu, observasi dilakukan secara langsung di Pondok Pesantren Darussalam Sengkubang untuk melihat proses pencarian hadis yang masih dilakukan secara manual. Penelitian ini juga dilengkapi dengan wawancara kepada pengelola perpustakaan pesantren guna memperoleh informasi mengenai jumlah koleksi hadis, cara pencarian yang digunakan, serta kendala yang sering dialami oleh pengguna.

B. Tahapan Pengolahan Data

Tahapan pengolahan data dilakukan untuk menyiapkan teks hadis dan kata kunci yang dimasukkan oleh pengguna agar dapat diproses secara optimal oleh sistem. Proses ini bertujuan menghasilkan data yang bersih, seragam, dan siap dianalisis sehingga hasil pencarian yang diperoleh memiliki tingkat relevansi yang lebih baik. Secara umum, pengolahan data dalam penelitian ini meliputi tahap *preprocessing* teks, pembobotan kata, dan perhitungan tingkat kemiripan antar dokumen.

1. Preprocessing

Tahap awal pengolahan data adalah *preprocessing* teks, yaitu proses membersihkan dan menyeragamkan data teks sebelum dilakukan perhitungan lebih lanjut (Hermawan & Ismiati, 2020). Pada tahap ini, seluruh teks hadis dan *query* pengguna diubah ke dalam bentuk huruf kecil (*case folding*) untuk menghindari perbedaan pengenalan kata akibat penggunaan huruf kapital (Ma'rifah et al., 2020). Selanjutnya dilakukan *tokenisasi*, yaitu memecah teks menjadi kata-kata berdasarkan spasi dan tanda baca (Rinjeni et al., 2024). Setelah itu, dilakukan proses *filtering* dan *stopword removal* dengan menghapus kata-kata umum yang tidak memiliki pengaruh signifikan terhadap makna pencarian (Budiman & Widjaja, 2020; Angelina et al., 2023). Tahap terakhir dalam *preprocessing* adalah *stemming*, yaitu mengubah kata berimbuhan menjadi kata dasar agar kata-kata yang memiliki makna sama dapat diperlakukan sebagai satu kesatuan (El-Khair, 2017).

2. Pembobotan Kata Menggunakan TF-IDF

Setelah tahap *preprocessing* selesai, setiap kata dalam dokumen hadis dan *query* diberikan bobot menggunakan metode *Term Frequency-Inverse Document Frequency* (TF-IDF) (Aziz, 2022). Metode ini digunakan untuk mengukur tingkat kepentingan suatu kata dalam dokumen. Nilai *Term Frequency* (TF) menunjukkan seberapa sering sebuah kata muncul dalam dokumen, sedangkan *Inverse Document Frequency* (IDF) berfungsi mengurangi pengaruh kata-kata yang sering muncul di banyak dokumen. Bobot TF-IDF diperoleh dari hasil perkalian nilai TF dan IDF, sehingga kata dengan bobot lebih tinggi dianggap lebih representatif terhadap isi dokumen. Nilai bobot yang dihasilkan selanjutnya dimanfaatkan sebagai dasar dalam proses seleksi *term*, sehingga hanya kata-kata yang memiliki tingkat kepentingan tinggi yang dipertahankan pada tahap perhitungan kemiripan.

$$TF(t, d) = f_{t,d} \quad (1)$$

$$TFW(t, d) = \begin{cases} 1 + \log_{10}(TF(t, d)), & \text{jika } TF(t, d) > 0 \\ 0, & \text{jika } TF(t, d) = 0 \end{cases} \quad (2)$$

$$IDF(t) = \log_{10} \left(\frac{N}{df_t} \right) \quad (3)$$

$$TF-IDF(t, d) = TFW(t, d) \times IDF(t) \quad (4)$$

3. Perhitungan Kemiripan dengan Jaccard Similarity

Perhitungan kemiripan antara kata kunci dan dokumen dilakukan menggunakan metode *Jaccard Similarity* dengan membandingkan kesamaan himpunan *term* hasil *preprocessing* dan seleksi *term* berbasis TF-IDF. Metode ini menghitung rasio antara jumlah *term* yang sama (irisan) dengan jumlah seluruh *term* unik (gabungan) dari *query* dan dokumen (Waskito et al., 2024). Nilai kemiripan

berada pada rentang 0 hingga 1, yaitu nilai yang lebih besar menunjukkan tingkat relevansi yang lebih tinggi. Dengan adanya proses seleksi *term* sebelumnya, perhitungan *Jaccard Similarity* menjadi lebih terfokus pada kata-kata yang relevan terhadap *query*. Hasil perhitungan digunakan untuk mengurutkan dokumen hadis berdasarkan tingkat kemiripan tertinggi.

$$Jaccard\ Similarity\ (A, B) = \frac{(A \cap B)}{(A \cup B)} \quad (5)$$

4. Perancangan dan Implementasi Sistem

Sistem temu kembali informasi hadis pada penelitian ini dikembangkan dalam bentuk aplikasi berbasis web untuk membantu pengguna menemukan hadis dengan lebih cepat dan sesuai kebutuhan. Dalam implementasinya, sistem dibangun menggunakan *framework Laravel* sebagai pengelola proses aplikasi, sedangkan *MySQL* digunakan sebagai media penyimpanan data hadis.

Proses dimulai saat pengguna memasukkan kata kunci pada halaman pencarian. Kata kunci tersebut tidak langsung digunakan, tetapi terlebih dahulu melalui tahapan pengolahan teks yang meliputi *case folding*, *tokenisasi*, *filtering*, *stopword removal*, dan *stemming*. Melalui tahapan ini, kata-kata yang tidak penting akan disaring sehingga tersisa kata yang lebih mewakili maksud pencarian.

Selanjutnya, setiap kata yang dihasilkan diberikan bobot menggunakan metode *TF-IDF*. Pembobotan ini membantu sistem mengenali kata yang memiliki pengaruh lebih besar terhadap isi dokumen maupun kata kunci yang dimasukkan. Kata dengan bobot lebih tinggi dianggap lebih relevan dalam proses pencarian. Setelah itu, sistem menghitung tingkat kemiripan antara kata kunci dan dokumen hadis menggunakan metode *Jaccard Similarity*. Perhitungan dilakukan dengan melihat kesamaan kata yang muncul pada *query* dan dokumen berdasarkan hasil pengolahan sebelumnya.

C. Pengujian Sistem

Pengujian sistem dilakukan untuk menilai kinerja aplikasi temu kembali informasi hadis yang telah dikembangkan. Evaluasi difokuskan pada dua aspek utama, yaitu tingkat akurasi hasil pencarian dan tingkat penerimaan pengguna terhadap sistem.

1. Pengujian Akurasi Sistem

Pengujian akurasi dilakukan untuk menilai kemampuan sistem dalam menampilkan hasil pencarian yang relevan terhadap kata kunci yang diberikan pengguna. Evaluasi menggunakan *Confusion Matrix* dengan parameter *Precision*, *Recall*, dan *F1-score*. Pada penelitian ini, dokumen dianggap relevan jika memiliki minimal dua kata yang sama dengan *query* (*minMatchCount* = 2) serta nilai kemiripan $\geq 0,2$.

Confusion Matrix digunakan untuk mengelompokkan hasil pencarian ke dalam kategori True Positive (TP), False Positive (FP), False Negative (FN), dan True Negative (TN). Kategori tersebut menjadi dasar dalam menghitung nilai *Precision*, *Recall*, dan *F1-score* (Normawati & Prayogi, 2021). Nilai *Precision* menunjukkan tingkat ketepatan hasil pencarian, yaitu proporsi hadis relevan dari seluruh hadis yang ditampilkan sistem (Prabowo, 2021). Nilai *Recall* mengukur kemampuan sistem dalam menemukan seluruh hadis relevan yang tersedia dalam basis data (Ikfa et al., 2024). Sementara itu, *F1-score* digunakan untuk menilai keseimbangan antara *Precision* dan *Recall* sehingga memberikan gambaran kinerja sistem secara lebih menyeluruh (Imran et al., 2024). Semakin tinggi nilai ketiga metrik tersebut, semakin baik kemampuan sistem dalam menemukan dan menampilkan hadis yang sesuai dengan kebutuhan pengguna.

Pengujian dilakukan menggunakan 10 *query* yang mewakili beberapa topik. Contoh hasil pengujian ditunjukkan pada Tabel 1.

Tabel 1. Hasil Pengujian Akurasi

No	Query	Hadis Relevan	Confusion Matrix	Recall	Precision	F1-score
1	Bagaimana niat mempengaruhi amal perbuatan manusia?	5	TP: 5; FP: 0; FN: 8; TN: 1127	0,3846	1	0,5555
2	Apakah berbohong termasuk tanda munafik?	2	TP: 2; FP: 0; FN: 9; TN: 1129	0,1818	1	0,3077
3	Apa keutamaan bersedekah dari harta yang halal?	10	TP: 10; FP: 0; FN: 6; TN: 1124	0,625	1	0,7692
4	Puasa Ramadan apa saja yang membuat dosanya diampuni?	2	TP: 2; FP: 0; FN: 8; TN: 1130	0,2	1	0,3333
5	Haji mabrur itu bagaimana dan apa balasannya?	3	TP: 3; FP: 0; FN: 0; TN: 1137	1	1	1

Tabel 1 menyajikan contoh hasil evaluasi pada beberapa *query* uji. Terlihat bahwa sistem cenderung menghasilkan nilai *precision* yang tinggi, yang menunjukkan bahwa sebagian besar hasil yang ditampilkan relevan terhadap *query*. Namun, nilai *recall* masih bervariasi pada setiap *query*, yang mengindikasikan bahwa tidak semua dokumen relevan berhasil ditemukan oleh sistem. Perbedaan nilai ini dipengaruhi oleh karakteristik *query* serta distribusi *term* pada dokumen. Oleh karena itu, untuk memperoleh gambaran kinerja sistem yang lebih representatif, dilakukan perhitungan rata-rata berdasarkan keseluruhan data pengujian, bukan hanya pada sampel *query* yang ditampilkan pada tabel.

Tabel 2. Nilai Rata-Rata Hasil Pengujian Akurasi

Parameter	Nilai Rata-Rata
Precision	72%
Recall	95%
F1-Score	82%

2. Pengujian Fungsional

Pengujian fungsional dilakukan untuk memastikan bahwa seluruh fitur dalam sistem dapat berjalan sesuai kebutuhan yang telah dirancang. Pengujian ini menggunakan pendekatan *black box*, yaitu menguji fungsi sistem berdasarkan masukan dan keluaran tanpa memperhatikan proses internalnya (Fahrezi et al., 2022). Beberapa skenario pengujian yang dilakukan meliputi proses pencarian hadis oleh pengguna, pengelolaan data oleh admin (*login*, tambah, edit, dan hapus data), serta tampilan sistem pada berbagai perangkat. Setiap skenario diuji untuk melihat apakah sistem mampu memberikan respons yang sesuai dengan harapan. Hasil pengujian fungsional disajikan pada Tabel 3.

Tabel 3. Hasil Pengujian Black Box Testing

No	Skenario Pengujian	Kasus Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Pencarian Hadis	Pengguna memasukkan kata kunci "puasa".	Sistem menampilkan hadis yang memuat kata "puasa".	Hadis terkait muncul.	Lulus
2	Login Admin	Admin memasukkan username dan password valid.	Sistem mengarahkan ke halaman dashboard admin.	Dashboard tampil.	Lulus
3	Tambah Data Hadis	Admin menambahkan data hadis baru.	Data hadis baru tersimpan dan muncul pada tabel.	Data tersimpan dan tampil.	Lulus
4	Edit Data Hadis	Admin mengedit data hadis yang sudah ada.	Data hadis diperbarui sesuai input terbaru.	Data terbaru.	Lulus
5	Hapus Data Hadis	Admin menghapus data hadis tertentu.	Data hadis terhapus dari database.	Data terhapus.	Lulus
6	Logout Admin	Admin melakukan logout dari sistem.	Sistem kembali ke halaman login.	Halaman login tampil.	Lulus
7	Responsive Layout	Sistem diakses dari HP dan laptop.	Tampilan sistem menyesuaikan ukuran layar perangkat.	Tampilan responsif.	Lulus

Berdasarkan Tabel 3, seluruh skenario pengujian menunjukkan hasil yang sesuai dengan yang diharapkan. Fitur pencarian mampu menampilkan hadis berdasarkan kata kunci yang dimasukkan, sementara fitur administrasi seperti *login*, pengelolaan data, dan *logout* berjalan dengan baik tanpa kendala. Selain itu, tampilan sistem juga dapat menyesuaikan dengan berbagai ukuran layar sehingga tetap nyaman digunakan pada perangkat yang berbeda. Secara keseluruhan, hasil pengujian ini menunjukkan bahwa sistem telah memenuhi kebutuhan fungsional yang dirancang dan dapat digunakan dengan baik oleh pengguna maupun admin.

3. User Acceptance Testing

Pengujian penerimaan pengguna dilakukan untuk mengetahui tingkat kelayakan sistem berdasarkan pengalaman langsung pengguna (Syaibah, 2021). Pengujian *User Acceptance Testing* (UAT) dilakukan dengan memberikan beberapa pertanyaan terkait sistem yang telah dibuat kepada pengguna melalui Google Form. Setiap pertanyaan dijawab menggunakan skala linear, mulai dari "tidak baik" hingga "sangat baik". Skala penilaian ini digunakan untuk melihat tingkat penerimaan pengguna terhadap sistem yang telah dikembangkan. Skala yang digunakan dapat dilihat pada Tabel 4.

Tabel 4. Skala Penilaian dan Klasifikasi Usability

Skor	Keterangan	Persentase (%)	Klasifikasi Usability
1	Tidak Baik	0%–20%	Sangat Tidak Layak
2	Kurang Baik	21%–40%	Tidak Layak
3	Cukup Baik	41%–60%	Cukup Layak
4	Baik	61%–80%	Layak
5	Sangat Baik	81%–100%	Sangat Layak

Pengujian ini dilakukan dengan memberikan pertanyaan kepada pengguna mengenai kemudahan penggunaan, tampilan sistem, relevansi hasil, kecepatan, dan kenyamanan penggunaan untuk mendukung kegiatan belajar di pesantren. Pengguna memberi penilaian dengan skala 1 (sangat tidak setuju) hingga 5 (sangat setuju) untuk memudahkan pengukuran tingkat kepuasan. Daftar pertanyaan yang digunakan dapat dilihat pada Tabel 5.

Tabel 5. Pertanyaan Pengujian UAT

No	Pertanyaan
1	Apakah Anda dapat menggunakan fitur pencarian hadis berdasarkan kata kunci dengan mudah?
2	Apakah hasil pencarian hadis yang ditampilkan sudah relevan dan sesuai dengan kata kunci yang Anda masukkan?
3	Apakah kecepatan sistem dalam menampilkan hasil pencarian hadis sudah sesuai dengan harapan Anda?
4	Apakah tampilan antarmuka website mudah dipahami dan tidak membingungkan saat digunakan untuk mencari hadis?
5	Apakah Anda dapat menemukan detail hadis (kitab, nomor hadis, perawi, isi hadis) dengan jelas setelah hasil pencarian ditampilkan?
6	Apakah sistem ini membantu Anda menemukan hadis lebih cepat dibandingkan cara manual di kitab cetak?
7	Apakah Anda merasa nyaman menggunakan sistem ini untuk kegiatan belajar atau diskusi?
8	Apakah fitur login admin dan manajemen hadis dapat digunakan tanpa kesulitan?
9	Apakah sistem berjalan dengan lancar tanpa terjadi error saat Anda menggunakan fitur pencarian hadis?
10	Apakah Anda merasa sistem ini bermanfaat untuk membantu belajar dan menambah pemahaman tentang hadis?

Setelah pengguna menjawab pertanyaan *UAT* yang telah diberikan, dilakukan rekapitulasi untuk melihat distribusi penilaian yang diberikan. Rekap ini membantu melihat seberapa baik sistem diterima dari sudut pandang pengguna berdasarkan aspek kemudahan penggunaan, tampilan, relevansi hasil pencarian, kecepatan sistem, serta manfaat penggunaan dalam kegiatan belajar di pesantren. Hasil rekap jawaban dari 25 siswa dan 5 pengurus perpustakaan dapat dilihat pada Tabel 6.

Tabel 6. Hasil Rekap Jawaban UAT

No	Pertanyaan	TB	KB	CB	B	SB
1	P1	0	0	9	13	8
2	P2	0	1	5	19	5
3	P3	0	1	6	10	13
4	P4	0	1	3	21	5
5	P5	0	1	2	12	15
6	P6	0	0	4	23	3
7	P7	0	1	5	9	15
8	P8	0	1	5	11	13
9	P9	0	0	4	15	11

No	Pertanyaan	TB	KB	CB	B	SB
10	P10	0	0	7	14	9
11	Jumlah	0	6	50	147	97

Setelah data *UAT* dari 30 responden dikumpulkan, dilakukan rekap untuk melihat jumlah jawaban pada setiap kategori. Masing-masing kategori diberi bobot nilai untuk memudahkan perhitungan skor total sistem. Hasil rekap jumlah, bobot, dan skor pada setiap kategori ditampilkan pada Tabel 7.

Tabel 7. Rekap Skor UAT

Kategori	Jumlah	Bobot	Skor
TB	0	1	0
KB	6	2	12
CB	50	3	150
B	147	4	588
SB	97	5	485
Total	300		1.235

Nilai maksimum diperoleh dari 30 responden $\times$ 5 nilai maksimal $\times$ 10 pertanyaan = 1.500. Berdasarkan perhitungan tersebut, persentase kelayakan sistem adalah sebagai berikut:

$$\begin{aligned} \text{Persentase} &= (\text{Jumlah skor total} / \text{Nilai maksimum}) \times 100\% \\ &= (1.235 / 1.500) \times 100\% = 82,33\% \end{aligned}$$

Berdasarkan hasil pengujian, diperoleh nilai sebesar 82,33%. Nilai tersebut termasuk dalam kategori “sangat layak”, yang menunjukkan bahwa sistem dapat diterima dengan baik oleh pengguna. Hal ini mengindikasikan bahwa sistem tidak hanya berjalan dengan baik secara teknis, tetapi juga mampu memberikan pengalaman penggunaan yang cukup nyaman dan mudah dipahami.

HASIL DAN PEMBAHASAN

Bagian ini memaparkan hasil implementasi sistem temu kembali informasi pencarian hadis serta analisis kinerjanya berdasarkan tahapan pengolahan data yang dilakukan.

A. Hasil Implementasi Sistem

Sistem temu kembali informasi pencarian hadis telah diimplementasikan dalam bentuk aplikasi berbasis web yang memungkinkan pengguna melakukan pencarian berdasarkan kata kunci. Proses pencarian dilakukan melalui tahapan *preprocessing*, pembobotan *TF-IDF*, dan perhitungan kemiripan menggunakan *Jaccard Similarity*. Dataset yang digunakan berjumlah 1.150 hadis dari berbagai kitab dan telah dinormalisasi sebelum digunakan dalam sistem. Hasil implementasi menunjukkan bahwa sistem mampu menampilkan dokumen hadis berdasarkan tingkat kemiripan terhadap *query*, dengan urutan dari nilai tertinggi ke terendah sehingga memudahkan pengguna dalam menemukan hadis yang relevan.

B. Hasil Preprocessing

Tahap *preprocessing* dilakukan melalui *case folding*, *tokenisasi*, *stopword removal*, dan *stemming* untuk menyeragamkan representasi teks. Proses ini mengubah teks dari bentuk kalimat menjadi *token* kata dasar yang lebih homogen. Normalisasi yang dilakukan mampu mengurangi noise serta variasi kata yang tidak relevan, sehingga meningkatkan kualitas representasi data pada tahap pembobotan dan perhitungan kemiripan. Contoh hasil *preprocessing* data hadis ditunjukkan pada Tabel 8.

Tabel 8. Hasil Preprocessing

No	Kalimat Hadis	Case Folding	Tokenisasi	Filtering & Stopword Removal	Stemming
1	Barang siapa berpuasa enam hari di bulan Syawal maka seperti puasa setahun	barang siapa berpuasa enam hari di bulan syawal maka seperti puasa setahun	barang, siapa, berpuasa, enam, hari, di, bulan, syawal, maka, seperti, puasa, setahun	puasa, enam, hari, bulan, syawal, puasa, setahun	puasa, enam, hari, bulan, syawal, puasa, tahun

Tabel 8 memperlihatkan bahwa proses *preprocessing* berhasil menyederhanakan teks menjadi kata dasar dengan menghilangkan kata tidak penting dan variasi morfologis. Hal ini menghasilkan representasi teks yang lebih konsisten sehingga mendukung proses pembobotan dan perhitungan kemiripan secara lebih akurat.

C. Hasil Pembobotan TF-IDF

Hasil pembobotan *TF-IDF* merepresentasikan tingkat kepentingan setiap *term* pada dokumen berdasarkan frekuensi kemunculan dan distribusinya dalam koleksi data. Tabel 9 menunjukkan nilai bobot *term* pada masing-masing dokumen (D1–D10) serta *query*.

Tabel 9. Hasil Pembobotan TF-IDF

Term	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	Query
sungguh	1,301	0	0	0	0	0	0	0	0	0	0
amal	0,5229	0	0	0	0	0,5229	0	0	0,5229	0	0,5229
gantung	1	0	0	0	0	0	0	0	0	0	0
niat	1,301	0	0	0	0	0	0	0	0	0	1
orang	0,5229	0	0	0	0,5229	0	0	0,5229	0	0	0,5229
dapat	1	0	0	0	0	0	0	0	0	0	0
sesuai	1	0	0	0	0	0	0	0	0	0	0

Berdasarkan Tabel 9, terlihat bahwa beberapa *term* seperti “sungguh”, “niat”, “gantung”, “dapat”, dan “sesuai” memiliki bobot tinggi pada dokumen tertentu, khususnya pada D1, serta bernilai nol pada dokumen lainnya. Hal ini menunjukkan bahwa *term* tersebut bersifat spesifik dan memiliki daya pembeda yang kuat terhadap dokumen. Sebaliknya, *term* seperti “amal” dan “orang” memiliki bobot yang lebih rendah karena muncul pada beberapa dokumen sehingga kontribusinya dalam membedakan dokumen menjadi lebih kecil.

Pada kolom *query*, *term* seperti “niat”, “amal”, dan “orang” memiliki bobot yang menunjukkan keterkaitannya dengan kata kunci pencarian. Kesesuaian antara *term* pada *query* dan

dokumen dengan bobot tinggi akan meningkatkan nilai kemiripan sehingga dokumen yang relevan dapat diprioritaskan dalam hasil pencarian. Dengan demikian, pembobotan *TF-IDF* berperan penting dalam meningkatkan kemampuan sistem membedakan dokumen relevan dan tidak relevan sebelum dilakukan perhitungan kemiripan menggunakan metode *Jaccard Similarity*.

D. Hasil Perhitungan Jaccard Similarity

Perhitungan *Jaccard Similarity* digunakan untuk mengukur tingkat kemiripan antara *query* dan dokumen berdasarkan kesamaan *term* hasil *preprocessing*. Dalam penelitian ini, dokumen dikategorikan sebagai relevan apabila memiliki nilai kemiripan $\geq 0,2$. Hasil perhitungan *Jaccard Similarity* ditunjukkan pada Tabel 10.

Tabel 10. Hasil Perhitungan Jaccard Similarity

Dokumen	Nilai Jaccard Similarity
D1	0,4286
D2	0,25
D3	0
D4	0
D5	0,25
D6	0,25
D7	0
D8	0,25
D9	0,25
D10	0

Berdasarkan Tabel 10, dokumen D1 memiliki nilai kemiripan tertinggi sebesar 0,4286, yang menunjukkan bahwa dokumen tersebut memiliki kesesuaian *term* paling dominan terhadap *query*. Hal ini mengindikasikan bahwa distribusi *term* pada D1 lebih representatif dibandingkan dokumen lainnya. Sementara itu, dokumen seperti D2, D5, D6, D8, dan D9 memiliki nilai kemiripan sebesar 0,25 dan masih termasuk dalam kategori relevan berdasarkan *threshold* yang digunakan. Sebaliknya, dokumen dengan nilai 0 tidak memiliki kesamaan *term* dengan *query* sehingga dikategorikan sebagai tidak relevan.

Nilai kemiripan dipengaruhi oleh proporsi kesamaan *term* terhadap keseluruhan *term*. Oleh karena itu, dokumen dengan distribusi *term* yang lebih relevan terhadap *query* akan memperoleh nilai similarity yang lebih tinggi. Penerapan *threshold* memungkinkan sistem menyaring dokumen secara lebih selektif sehingga hasil pencarian menjadi lebih terfokus. Dengan demikian, metode *Jaccard Similarity* mampu mendukung proses identifikasi dokumen relevan secara efektif dalam sistem temu kembali informasi hadis.

E. Pembahasan

Penelitian ini berhasil mengembangkan sistem pencarian hadis berbasis web yang mampu menyajikan hasil pencarian secara relevan berdasarkan kata kunci pengguna. Penerapan *preprocessing* serta pembobotan *TF-IDF* membantu meningkatkan kualitas representasi *term* sebelum dilakukan perhitungan kemiripan menggunakan *Jaccard Similarity*.

Hasil evaluasi menunjukkan bahwa sistem memiliki kinerja yang cukup baik dengan nilai *precision* sebesar 72%, *recall* sebesar 95%, dan *F1-score* sebesar 82%. Selain itu, pengujian fungsional menunjukkan seluruh fitur berjalan sesuai rancangan, dan hasil *User Acceptance Testing* (UAT) sebesar 82,33% mengindikasikan bahwa sistem dapat diterima dengan baik oleh pengguna.

Secara keseluruhan, sistem mampu mendukung proses pencarian hadis secara lebih efektif dibandingkan metode pencarian konvensional. Namun, pengembangan lebih lanjut masih diperlukan untuk meningkatkan ketepatan hasil pada *query* yang bersifat umum.

KESIMPULAN

Sistem pencarian hadis berbasis web yang dikembangkan dalam penelitian ini mampu membantu proses penelusuran hadis menjadi lebih cepat dan terarah. Pengguna dapat menemukan hadis berdasarkan kata kunci tanpa perlu membuka hadis secara manual. Tahapan *preprocessing* seperti *case folding*, *tokenisasi*, *filtering*, *stopword removal*, dan *stemming* memberikan kontribusi dalam menyederhanakan teks sehingga proses pencocokan menjadi lebih efektif.

Berdasarkan hasil pengujian menggunakan *confusion matrix*, sistem menunjukkan performa yang cukup baik dengan nilai *precision* sebesar 72%, *recall* sebesar 95%, dan *F1-score* sebesar 82%. Nilai tersebut mengindikasikan bahwa sebagian besar dokumen relevan berhasil ditemukan, meskipun masih terdapat beberapa hasil yang kurang sesuai dengan kebutuhan pencarian. Dari sisi fungsional, seluruh fitur berjalan sesuai rancangan, dan hasil pencarian ditampilkan berdasarkan urutan tingkat kemiripan untuk memudahkan pengguna dalam memilih informasi yang dibutuhkan.

Penilaian pengguna melalui *User Acceptance Testing* (UAT) memperoleh nilai sebesar 82,33%, yang menunjukkan bahwa sistem dapat diterima dengan baik. Kemudahan penggunaan, tampilan antarmuka, serta manfaat yang dirasakan menjadi faktor utama dalam mendukung tingkat penerimaan tersebut. Secara umum, sistem yang dibangun sudah mampu menjalankan fungsinya dengan baik dalam mendukung proses temu kembali informasi hadis.

SARAN

Pengembangan lebih lanjut masih diperlukan untuk meningkatkan kualitas hasil pencarian, terutama dalam menangani *query* yang bersifat umum agar hasil yang ditampilkan menjadi lebih tepat. Penyesuaian nilai *threshold* kemiripan dapat dipertimbangkan untuk mendapatkan keseimbangan yang lebih baik antara *precision* dan *recall*.

Selain itu, pendekatan pembobotan *term* dapat dikembangkan lebih lanjut, misalnya dengan memanfaatkan metode yang mampu mempertimbangkan konteks kata sehingga hasil pencarian tidak hanya bergantung pada kesamaan *term*. Penambahan fitur seperti rekomendasi hadis atau pengelompokan berdasarkan topik juga dapat menjadi alternatif untuk meningkatkan pengalaman pengguna dalam menggunakan sistem.

REFERENSI

- Angelina, S. J., Bijaksana, A., Negara, P., & Muhandi, H. (2023). Analisis pengaruh penerapan *stopword removal* pada performa klasifikasi sentimen tweet bahasa Indonesia. JUARA (Jurnal Aplikasi dan Riset Informatika), 2(1), 165–173. <https://doi.org/10.26418/juara.v2i1.69680>
- Apriani, A., Zakiyudin, H., & Marzuki, K. (2021). Penerapan algoritma cosine similarity dan pembobotan *TF-IDF* pada sistem penerimaan mahasiswa baru pada kampus swasta. Jurnal Bumigora Information Technology, 3(1), 19–27. <https://doi.org/10.30812/bite.v3i1.1110>
- Aziz, A. (2022). Analisis sentimen identifikasi opini terhadap produk, layanan dan kebijakan perusahaan menggunakan algoritma *TF-IDF* dan SentiStrength. Jurnal Sains Komputer & Informatika (J-SAKTI), 6(1), 115.
- Budiman, A. E., & Widjaja, A. (2020). Analisis pengaruh teks *preprocessing* terhadap deteksi plagiarisme pada dokumen tugas akhir. Jurnal Teknik Informatika dan Sistem Informasi, 6(3), 475–488. <https://doi.org/10.28932/jutisi.v6i3.2892>
- El-Khair, I. A. (2017). Effects of stop words elimination for Arabic information retrieval: A comparative study. 1–15.
- Elsadantia, B. A. (2023). Perkembangan dan peran OPAC pada aplikasi CIP (Cerah Informasi Pustaka) untuk temu kembali informasi di Perpustakaan Universitas Tridinanti Palembang. 2(4).
- Fahrezi, A., Salam, F. N., Ibrahim, G. M., & Rahman, R. (2022). Pengujian *black box* testing pada aplikasi inventori barang berbasis web di PT AINO Indonesia. 1(1), 1–5.
- Hermawan, L., & Ismiati, M. B. (2020). Pembelajaran text *preprocessing* berbasis simulator untuk mata kuliah information retrieval. Jurnal Transformatika, 17(2), 188. <https://doi.org/10.26623/transformatika.v17i2.1705>
- Ikfa, M. Z., dkk. (2024). Evaluasi fungsi pencarian di React.js untuk pengambilan data dalam sistem Wiki-Inf. 4(1), 930–937.
- Imran, B., Karim, M. N., & Ningsih, N. I. (2024). Klasifikasi berita hoax terkait pemilihan umum presiden Republik Indonesia tahun 2024 menggunakan Naïve Bayes dan SVM. Dinamika Rekayasa, 20(1), 1–9. <https://doi.org/10.20884/1.dinarek.2024.20.1.27>
- Ma'rifah, H., Wibawa, A. P., & Akbar, M. I. (2020). Klasifikasi artikel ilmiah dengan berbagai skenario *preprocessing*. Sains, Aplikasi, Komputasi dan Teknologi Informasi, 2(2), 70. <https://doi.org/10.30872/jsakti.v2i2.2681>
- Mohammed, M. T., & Rashid, O. F. (2023). Document retrieval using *term* frequency inverse sentence frequency weighting scheme. Indonesian Journal of Electrical Engineering and Computer Science, 31(3), 1478–1485. <https://doi.org/10.11591/ijeecs.v31.i3.pp1478-1485>
- Normawati, D., & Prayogi, S. A. (2021). Implementasi Naïve Bayes classifier dan *confusion matrix* pada analisis sentimen berbasis teks pada Twitter. Jurnal Sains Komputer & Informatika (J-SAKTI), 5(2), 697–711.
- Prabowo, T. T. (2021). Efektivitas sistem temu kembali informasi Perpustakaan Digital Institut Seni Indonesia (ISI) Yogyakarta dalam tinjauan *recall* dan *precision*. Media Pustakawan, 28(1), 37–48. <https://doi.org/10.37014/medpus.v28i1.1087>
- Ridwan, M., Umar, M. H., & Ghafar, A. (2021). Sumber-sumber hukum Islam dan implementasinya: Kajian deskriptif kualitatif tentang Al-Qur'an, Sunnah, dan Ijma'. Borneo Journal of Islamic Studies, 1(2), 28–41.
- Rinjeni, T. P., Indriawan, A., & Rakhmawati, N. A. (2024). Matching scientific article titles using cosine similarity and *Jaccard Similarity* algorithm. Procedia Computer Science, 234, 553–560. <https://doi.org/10.1016/j.procs.2024.03.039>
- Syaibah, S. (2021). Sistem pendukung keputusan rekomendasi judul skripsi Prodi Teknik Informatika menggunakan metode ELECTRE. Digital Intelligence, 2(1), 1. <https://doi.org/10.29406/diligent.v2i1.2743>
- Syahrani, A. W., & Hidayah, H. (2022). Kedudukan hadist dalam pembentukan hukum. MUSHAF Journal: Jurnal Ilmu Al Quran dan Hadis, 3(1), 24–31. <https://doi.org/10.54443/mushaf.v3i1.85>
- Waskito, M. R., Rahajoe, A. D., Nurlaili, A. L., & Timur, J. (2024). Collaborative *filtering* menggunakan algoritma cosine similarity dan *Jaccard Similarity* pada sistem e-commerce. 12(3), 4307–4316.
- Yunita, N. P. (2023). Aplikasi pencarian hadis menggunakan vector space model dengan pembobotan *TF-IDF* dan Confix-Stripping Stemmer. Jurnal Teknologi Informasi dan Ilmu Komputer, 10(3), 665–676. <https://doi.org/10.25126/jtiik.2023106736>