

IMPLEMENTASI METODE *HYBRID FUZZY JARO WINKLER* DAN *COSINE SIMILARITY* PADA SISTEM PENCARIAN AYAT AL-QURAN BERBASIS TRANSLITERASI LATIN

Gempar Perkasa Tahir^{*1)}, Emil Agusalim Habi Talib²⁾, Fahrirm Irhamna Rachman³⁾

1. Informatika, Universitas Muhammadiyah Makassar
105841105821@student.unismuh.ac.id
2. Informatika, Universitas Muhammadiyah Makassar
emil@unismuh.ac.id
3. Informatika, Universitas Muhammadiyah Makassar
fachrim141020@unismuh.ac.id

Abstract

This research addresses the challenge of retrieving Qur'anic verses in Latin transliteration, which is hindered by the absence of a standardized orthography, leading to diverse spelling variations. The study aims to design and implement a hybrid information retrieval system that integrates Fuzzy Jaro-Winkler for lexical similarity and Cosine Similarity on fine-tuned DistilBERT embeddings for semantic relevance. The system workflow begins with preprocessing and normalization of the dataset, followed by initial candidate selection using Jaro-Winkler, and final reranking through semantic similarity scoring. Evaluation was conducted using black-box testing across scenarios including ideal queries, spelling variations, incomplete queries, and varying query lengths. Results show high accuracy for ideal (96%) and varied spelling queries (92%), with performance improving as query length increases, reaching 96% for four-word queries. The hybrid approach effectively bridges lexical and semantic gaps, outperforming single-method baselines, and demonstrates robustness in handling non-standard transliteration in Qur'anic text retrieval.

Kata Kunci: Jaro Winkler, Cosine Similarity, Qur'an Retrieval, Transliteration, Semantic Search.

A. PENDAHULUAN

Pada era digital saat ini, ledakan volume informasi telah menjadi sebuah fenomena global yang mengubah cara manusia berinteraksi dengan pengetahuan. Kemampuan untuk mengelola, mencari, dan menemukan kembali informasi secara efisien telah menjadi pilar fundamental dalam berbagai sektor kehidupan. Konsekuensinya, pengembangan system pencarian informasi (*information retrieval*) yang canggih menjadi sebuah

keniscayaan. Sistem ini tidak hanya dituntut untuk mampu memproses data dalam skala besar, tetapi juga harus dapat memahami nuansa bahasa alami manusia untuk menyajikan hasil yang akurat dan relevan [1]. Salah satu bidang yang turut mengalami transformasi ini adalah akses terhadap teks-teks keagamaan, di mana platform digital kini menjadi medium atau media utama bagi miliaran orang untuk berinteraksi dengan kitab suci mereka [2].

Dalam konteks pencarian Al-Qur'an di Indonesia, tantangan utama muncul dari

ketergantungan pengguna pada transliterasi Latin. Problematika fundamentalnya adalah tidak adanya standar tunggal untuk alih aksara Arab-Latin, yang menyebabkan munculnya beragam variasi ejaan [3]. Variasi ini menjadi penghalang bagi sistem pencarian konvensional yang umumnya masih beroperasi berbasis pencocokan eksak dan memiliki keterbatasan dalam menangani kerumitan bahasa.

Penelitian terdahulu telah mengeksplorasi solusi untuk variasi ejaan. Inkonsistensi penulisan ini dapat dipandang sebagai *typographical error* dari perspektif komputasi [4]. Untuk menanganinya, algoritma fuzzy *string matching* seperti *Jaro Winkler* terbukti efektif dalam mengukur kemiripan *string* berdasarkan kecocokan karakter dan transposisi minimal [5]. Pada ranah semantik, metode berbasis ruang vektor seperti *Cosine Similarity* juga telah banyak diterapkan untuk mengukur kedekatan makna [6]. Pendekatan ini efektif dalam menilai relevansi kontekstual dengan merepresentasikan teks sebagai vektor [7]. Efektivitasnya telah terbukti dalam sistem penilaian otomatis hingga pencarian terjemahan Al-Qur'an [8].

Analisis literatur menunjukkan bahwa implementasi *Jaro-Winkler* dan *Cosine Similarity* cenderung dilakukan secara terpisah. *Jaro-Winkler* andal pada tingkat leksikal untuk mengatasi variasi ejaan [9], sementara *Cosine Similarity* unggul pada tingkat dokumen untuk menangani relevansi semantik [10]. Kondisi ini menyisakan sebuah celah penelitian (*research gap*) yang signifikan, yaitu belum adanya integrasi kedua metode ke dalam sebuah kerangka kerja hibrida yang solid. Sistem yang ada saat ini cenderung hanya unggul di satu aspek, namun jarang mampu mengatasi kedua tantangan leksikal dan semantik secara simultan.

Untuk mengisi celah tersebut, penelitian ini bertujuan merancang dan membangun sistem pencarian ayat Al-Qur'an berbasis transliterasi Latin menggunakan metode hibrida *Fuzzy Jaro-Winkler* dan *Cosine Similarity*. Solusi ini diusulkan untuk mengintegrasikan kekuatan kedua algoritma secara sinergis guna menciptakan sistem yang lebih komprehensif. Dalam arsitektur yang diusulkan, *Fuzzy Jaro-Winkler* berperan sebagai lapisan pertama untuk toleransi kesalahan ejaan. Hasilnya kemudian akan diperingkat oleh *Cosine Similarity* sebagai lapisan kedua untuk memastikan relevansi makna. Pendekatan hibrida ini diharapkan dapat menghasilkan pencarian yang akurat secara leksikal sekaligus relevan secara semantik.

B. METODE PENELITIAN

Tahapan perancangan sistem dalam penelitian ini berfokus pada pengembangan sebuah arsitektur hibrida yang mengintegrasikan dua pendekatan komputasi pencocokan leksikal berbasis *fuzzy string matching* dan pemahaman semantik kontekstual. Adapun alur perancangan sistem ini dibagi menjadi beberapa tahap utama. Proses ini divisualisasikan pada gambar 1.

1. Tahap pertama adalah pengumpulan dataset ayat Al-Qur'an dalam format *JSON*. Dataset yang bersumber dari situs Qur'an Kemenag ini diperoleh melalui platform Kaggle yang terdiri dari 564 ayat yaitu Juz 30, mencakup elemen utama berupa teks Arab, nama surah, no ayat, transliterasi Latin standar, dan terjemahan. Digunakan sebagai korpus data utama dalam penelitian.



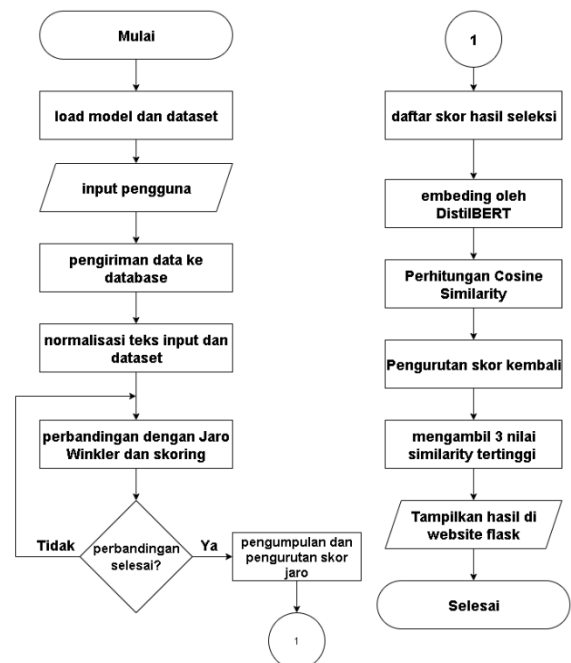
Gambar 1. Flowchart perancangan model *Hybrid*

2. Setelah pengumpulan dataset, dirancang sebuah sistem untuk menerima input berupa suara. Oleh karena itu, digunakan sebuah modul transkripsi yang memanfaatkan model *Whisper* dari *OpenAI*, yang dipilih karena kemampuannya yang andal dalam mengubah audio menjadi teks aksara Arab [11].
3. Setelah mendapatkan modul transkripsi, Teks Arab hasil transkripsi kemudian diubah menjadi format Latin. Untuk keperluan ini, dirancang sebuah fungsi transliterasi khusus yang menggunakan kamus pemetaan karakter (*ARABIC_TO_LATIN_QURAN*) [12]. Selanjutnya, teks Latin tersebut dinormalisasi dengan menghapus karakter *non* alfabet dan mengubahnya menjadi huruf kecil.
4. Setelah mendapatkan teks transliterasi, diterapkan metode *Jaro-Winkler* sebagai filter leksikal pada tahap awal. Sistem dirancang untuk membandingkan kueri input dari pengguna dengan seluruh data transliterasi di dalam korpus, menghitung skor kemiripan

ejaan, dan menyaring 50 kandidat teratas untuk diproses pada tahap berikutnya.

5. Tahap akhir adalah implementasi *Cosine Similarity* yang memerlukan representasi vektor. Model *embedding DistilBERT* dipilih karena kemampuannya dalam memproses multibahasa. Karena model ini belum terlatih untuk domain spesifik transliterasi, dilakukan proses *fine tuning* menggunakan dataset augmentasi dari Juz 30. Setelah model siap, *Cosine Similarity* diterapkan untuk memeringkat 50 kandidat dan memilih 3 hasil teratas sebagai *output* final.

Setelah seluruh komponen dirancang, langkah selanjutnya adalah mengintegrasikannya ke dalam sebuah arsitektur sistem yang utuh. Arsitektur ini mendefinisikan bagaimana data mengalir dan diproses dari input pengguna hingga menjadi output yang relevan. Alur kerja sistem secara rinci diilustrasikan pada gambar 2.



Gambar 2. Flowchart alur kerja sistem *hybrid*

C. HASIL DAN PEMBAHASAN

1. Pengumpulan Dataset

Dataset yang digunakan dalam penelitian ini merupakan kumpulan ayat-ayat Al-Qur'an yang diperoleh dari Kaggle, dengan sumber asli berasal dari website resmi Kementerian Agama Republik Indonesia (Kemenag). Dataset ini disusun dalam format *JSON* dan berisi informasi lengkap mengenai seluruh isi Al-Qur'an, terdiri dari 114 surah dengan total 6.236 ayat. Setiap ayat memiliki beberapa atribut penting, seperti ID ayat, ID surah, nama surah dalam Latin, nomor ayat, teks Arab, transliterasi Latin, serta terjemahan dalam bahasa Indonesia.

Namun, pada penelitian ini hanya digunakan data dari Juz 30, yang mencakup 37 surah dengan total 564 ayat, penggunaan hanya Juz 30 juga terkait dengan kebutuhan *fine-tuning* model *embedding DistilBERT* yang digunakan dalam penelitian ini. Melatih model dengan seluruh isi Al Qur'an membutuhkan sumber daya komputasi yang besar. Oleh karena itu, Juz 30 dipilih sebagai representasi dataset yang lebih ringkas namun tetap memadai untuk mencapai tujuan penelitian.

2. Transkripsi audio menggunakan whisper

Modul transkripsi audio menggunakan model *Whisper* berfungsi untuk mengubah masukan berupa suara menjadi teks Arab secara otomatis. Proses ini diawali dengan penerimaan file audio dari pengguna, yang kemudian melalui tahap pra-pemrosesan untuk menyesuaikan format dan *sampling rate* sesuai kebutuhan model. Selanjutnya, model *Whisper* melakukan analisis sinyal suara dan mengenali fonem bahasa Arab, sehingga menghasilkan keluaran berupa teks Arab yang merepresentasikan bacaan pada audio.

Hasil transkripsi ini menjadi input bagi tahap berikutnya, yaitu proses transliterasi

dan normalisasi teks, sebelum dilakukan pencocokan dengan dataset ayat Al-Qur'an. Penggunaan *Whisper* pada sistem ini dipilih karena kemampuannya mengenali variasi pelafalan, mendukung bahasa Arab dengan tingkat akurasi tinggi, serta tetap memberikan hasil yang baik meskipun kualitas audio tidak sempurna.

3. Transliterasi dan Normalisasi

Tahap transliterasi dan normalisasi bertujuan untuk mengubah teks Arab hasil transkripsi menjadi bentuk teks Latin yang seragam dan siap diproses pada tahap pencocokan. Proses transliterasi dilakukan dengan mengonversi setiap karakter Arab ke padanan huruf Latin sesuai kaidah transliterasi yang digunakan, sehingga memudahkan proses pencarian dan perbandingan teks. Setelah itu, dilakukan tahap normalisasi untuk menyamakan format penulisan, antara lain dengan menghapus tanda baca dan diakritik, menghilangkan karakter non-alfabet, serta mengubah seluruh huruf menjadi huruf kecil (*lowercase*). Langkah ini penting untuk mengurangi variasi penulisan yang tidak relevan dengan makna, sehingga sistem dapat melakukan pencocokan secara konsisten dan akurat pada tahap berikutnya.

4. Perhitungan menggunakan Jaro Winkler

Pada tahap awal proses pencarian, sistem menggunakan metode *Jaro Winkler* untuk menghitung tingkat kemiripan antara teks masukan (*query*) dengan teks dataset ayat Al-Qur'an yang telah disimpan. Sebenarnya, metode *Jaro-Winkler* menyeleksi 50 kandidat ayat dengan skor kemiripan tertinggi, namun pada tabel ini hanya ditampilkan 10 kandidat teratas sebagai contoh hasil perhitungan. Berdasarkan input "*fasabbih bihamdirab*" dengan hasil yang diharapkan adalah "*Fasabbih bihamdi rabbika wastagfir h(u), innahū kāna*

tawwābā(n)”. Nilai skor *Jaro-Winkler* pada 10 kandidat tersebut berkisar antara 0.6104 hingga 0.6632, yang menunjukkan tingkat kedekatan string masukan dengan ayat-ayat dalam dataset.

Tabel 1. Hasil perhitungan *Jaro Winkler*

No	Teks Inputan	Teks dataset	Skor Jaro Winkler
1	fasabbih bihamdirab	Sabbihisma rabbikal-a‘lā.	0.6266
2	fasabbih bihamdirab	Fasabbih bihamdi rabbika wastagfir h(u), innahū kāna tawwābā(n).	0.6243
3	fasabbih bihamdirab	Fa basysyirhum bi‘azābin alīm(in)	0.6210
4	fasabbih bihamdirab	Bi‘aidī safarah(tin).	0.6208
5	fasabbih bihamdirab	Fa šabba ‘alaihim rabbuka saṭa ‘azāb(in).	0.6205
6	fasabbih bihamdirab	Iqra' bismi rabbikal-laṣī khalaq(a).	0.6243
7	fasabbih bihamdirab	Famā yukaẓẓibuka ba‘du bid-dīn(i).	0.6210
8	fasabbih bihamdirab	Inna rabbaka labil-mirṣād(i).	0.6208
9	fasabbih bihamdirab	Faṣalli lirabbika wanḥar.	0.6205
10	fasabbih bihamdirab	Falya‘budū rabba hāẓal-bait(i).	0.6104

Dari hasil pengujian terlihat bahwa ayat dengan transliterasi “*Sabbihis ma rabbikal-a‘lā*” memperoleh skor tertinggi, yaitu 0,6632, sehingga sistem menempatkannya pada urutan pertama kandidat. Ayat ini memiliki kemiripan pola kata dengan masukan, khususnya pada penggunaan kata dasar “*sabbih*” yang terdapat pada kueri pengguna. Selanjutnya, hasil yang diharapkan terdapat pada urutan kedua yaitu ayat “*Fasabbih bihamdi rabbika wastagfirhu, innahū kāna tawwābā(n)*” mendapatkan

skor 0,6518, karena memiliki kesesuaian struktur frasa dengan masukan, meskipun terdapat tambahan kata pada bagian akhir.

Secara keseluruhan, metode *Jaro-Winkler* mampu mengurutkan kandidat ayat berdasarkan tingkat kemiripan string secara efektif. Ayat-ayat yang memiliki pola kata atau frasa yang mendekati masukan pengguna cenderung mendapatkan skor yang lebih tinggi, sedangkan ayat dengan perbedaan struktur yang lebih signifikan menghasilkan skor yang lebih rendah, seperti ayat ke-10 “*Falya‘budū rabba hāẓal-bait(i)*” dengan skor 0,6104.

Hasil ini menunjukkan bahwa penggunaan *Jaro-Winkler* sesuai untuk tahap pemilihan kandidat awal karena metode ini fokus pada perhitungan similaritas karakter dan pola kata. Namun, karena skor kemiripan pada kandidat teratas masih di bawah 0.7, maka hasil pencocokan ini belum sepenuhnya akurat untuk penentuan ayat akhir. Oleh karena itu, sistem melanjutkan proses pencarian menggunakan metode *BERT embedding* dan perhitungan *cosine similarity* untuk meningkatkan akurasi hasil akhir dengan mempertimbangkan kesamaan makna antar teks.

5. Reranking menggunakan Cosine Similarity

Setelah diperoleh 50 kandidat ayat dari tahap pertama yaitu *Jaro Winkler*, sistem menghitung nilai *Cosine Similarity* antara kueri dan kandidat menggunakan *DistilBERT embedding*. Hasil reranking menunjukkan 10 kandidat terbaik seperti pada Tabel 6 . Dari hasil tersebut, sistem menampilkan 3 ayat dengan skor tertinggi sebagai hasil pencarian akhir.

Tabel 2. Hasil Reranking Cosine Similarity

No	Teks Inputan	Teks Dataset	Skor Cosine Similarity
1	fasabbih bihamdirab	Fasabbih bihamdi rabbika wastag firh(u), innahū kāna tawwābā(n).	0.5602
2	fasabbih bihamdirab	Fa ṣabba ‘alaihim rabbuka sauṭa ‘azāb(in).	0.5275
3	fasabbih bihamdirab	Fa iżā hum bis-sāhirah(ti).	0.4932
4	fasabbih bihamdirab	Fa basasyirhum bi‘azābin alīm(in).	0.4870
5	fasabbih bihamdirab	Fa‘ambatnā fihā ḥabbā(n).	0.4486

Efektivitas pendekatan *hybrid* secara jelas tervalidasi saat membandingkan hasil pada Tabel 5 dan Tabel 6. Pada tahap awal yang hanya menggunakan *Jaro-Winkler* Tabel 5, ayat “*Sabbihisma rabbikal-a‘lā*” menduduki peringkat pertama dengan skor 0.6632. Hasil ini dapat dipahami karena kemiripan karakter leksikal yang tinggi dengan kueri ‘*fasabbih bihamdirab*’. Namun, ayat yang secara hubungan antar kueri paling relevan, yaitu “*Fasabbih bihamdi rabbika wastagfirh(u), innahū kāna tawwābā(n)*”, hanya berada di peringkat kedua.

Pergeseran peringkat yang fundamental terjadi setelah tahap reranking dengan *Cosine Similarity* (Tabel 2). Ayat “*Fasabbih bihamdi rabbika wastagfirh(u), innahū kāna tawwābā(n)*” kini naik menjadi peringkat pertama dengan skor similaritas 0.5602, sementara ayat “*Sabbihisma rabbikal-a‘lā*” bahkan tidak lagi masuk dalam 10 besar kandidat. Fenomena ini secara empiris membuktikan nilai kebaruan penelitian ini: *Jaro-Winkler* berhasil bertindak sebagai penyaring awal initial filter untuk kandidat yang mirip secara ejaan,

sementara *Cosine Similarity* pada *embedding* yang telah di *fine-tuning* sukses bertindak sebagai peranking ulang reranker yang cerdas untuk memprioritaskan hasil yang paling relevan secara makna.

6. Pengujian Model inti Berdasarkan skenario

Pengujian skenario ejaan dilakukan dengan memberikan kueri transliterasi Latin yang tidak baku atau salah ketik. Tujuannya adalah menguji kemampuan sistem menemukan ayat yang benar meskipun ejaan kueri berbeda.

Tabel 3. Hasil Pengujian Skenario

No	Kueri inputan	Hasil yang di tampilkan	Hasil yang di harapkan (top 3)	Status
1	innahu ala rajihi	1) Innahū ‘al ā raj’ihī laqādir(un). 2) Innahū laqaulu rasūlin karīm(in). 3) Innahū zanna allay yahūr(a).	Innahū ‘alā raj’ihī laqādir(un).	Berhasil
2	kul audzu birabbinnas	1) Qul a‘ūzu birabbinnās(i). 2) Qul a‘ūzu birabbil-falaq(i). 3) Wa iżal-qubūru bu’sirat	Qul a‘ūzu birabbinnās(i).	Berhasil
3	inna attayna	1) Inna sa‘ya kum lasyattā. 2) Innā a‘ṭain ākal-kaūsar(a). 3) Inna ‘alainā lal-hudā.	Innā a‘ṭaināk al-kaūsar(a).	Berhasil
4	fasabbih	1) Fa ṣabba ‘alaihim rabbuka sauṭa ‘azāb(in). 2) Fa aina taḏḥabūn(a). 3) Fa sanuyassir	Fasabbih bihamdi rabbika wastagfirh(u), innahū kāna tawwābā(n).	Gagal

No	Kueri inputan	Hasil yang di tampilkan	Hasil yang di harapkan (top 3)	Status
		uhū lil-yusrā.		
5	wa min	1) Wa mizājuhū min tasnīm(in). 2) Wa min syarrin-naffāsāti fil-‘uqad(i). 3) Wa min syarri gāsiqin izā waqab(a).	Wa min syarri ḥāsīdin izā ḥasad(a).	Gagal

Dilakukan pengujian terhadap lima skenario kueri masukan untuk mengevaluasi performa sistem dalam menemukan ayat Al-Qur'an yang sesuai dengan transliterasi Latin yang diberikan. Pengujian ini menggunakan kombinasi metode *Jaro-Winkler* dan *Cosine Smilarity* untuk memperoleh hasil pencarian terbaik yaitu 3 teratas. Hasil pengujian menunjukkan bahwa dari lima skenario kueri yang diuji, tiga skenario berhasil menemukan ayat yang sesuai dengan harapan, sementara dua skenario lainnya mengalami kegagalan.

Pada skenario pertama dengan kueri “*innahu ala rajihī*”, sistem berhasil menampilkan ayat *Innahū ‘alā rajīhi laqādir(un)* di urutan teratas, sesuai dengan hasil yang diharapkan. Hal serupa terjadi pada skenario kedua dengan kueri “*kul audzu birabbinnas*”, di mana ayat yang ditemukan sesuai dengan ekspektasi, yaitu *Qul a‘ūzu birabbīn-nās(i)*. Skenario ketiga juga berhasil dengan baik, ketika kueri “*inna attayna*” menghasilkan ayat *Innā a‘tainākal-kausar(a)*. di urutan kedua.

Namun, pada skenario keempat dengan kueri “*fasabbih*”, sistem menampilkan ayat *Fa sabba ‘alaihim rabbuka sauta ‘azāb(in)* di urutan teratas, sedangkan ayat yang diharapkan adalah *Fasabbih biḥamdi rabbika wastagfir(u)*,

innahū kāna tawwābā(n) bahkan tidak terlihat di urutan ketiga. Hal ini menunjukkan bahwa meskipun metode hybrid mampu menemukan ayat yang relevan, ada kemungkinan hasil teratas tidak selalu tepat jika teks masukan terlalu singkat atau ambigu.

Skenario kelima dengan kueri “*wa min*” juga mengalami kegagalan karena sistem menampilkan ayat “*Wa mizājuhū min tasnīm(in)*” di urutan pertama dan “*Wa min syarrin-naffāsāti fil-‘uqad(i)*” di urutan kedua serta “*Wa min syarri gāsiqin izā waqab(a)*” di urutan ketiga. sementara ayat yang diharapkan adalah “*Wa min syarri ḥāsīdin izā ḥasad(a)*”. Kegagalan ini dapat disebabkan oleh variasi ejaan pada transliterasi atau karena kesamaan sebagian kata dengan ayat lain dalam dataset.

7. Hasil Kinerja Akurasi Agregat

Tabel ini menyajikan hasil perhitungan metrik performa utama dari seluruh pengujian model inti. Akurasi diukur sebagai persentase keberhasilan sistem menempatkan ayat yang benar *ground truth* pada peringkat pertama.

Berikut tabel hasil dari proses pengujian, tabel 4 merupakan hasil dari skenario kueri ideal, variasi ejaan dan kueri tidak lengkap, sedangkan tabel 5 merupakan hasil dari pengujian untuk kueri dengan jumlah kata beragam.

Tabel 4. Akurasi hasil uji skenario

Skenario	Total Data	Accuracy@ 1%	Accuracy@ 3%
Ideal	100	96.00	97.00
Variasi ejaan	100	92.00	97.00
Query Tidak lengkap	100	73.00	86.00

Analisis dari Tabel 4 menunjukkan beberapa temuan penting. Tingkat akurasi yang tinggi pada skenario 'Ideal' (96%)

dan 'Variasi ejaan' (92%) membuktikan ketangguhan kombinasi *Jaro-Winkler* dan *Cosine Similarity* dalam menangani kesalahan penulisan umum. Namun, terlihat penurunan akurasi yang cukup signifikan pada skenario kueri Tidak Lengkap menjadi 73% (Accuracy@1%). Hal ini mengindikasikan bahwa meskipun sistem mampu menangani kueri yang tidak utuh, ketiadaan kata kunci yang cukup dapat menciptakan ambiguitas semantik.

Tabel 5. Akurasi Hasil Uji Jumlah Kata

Jumlah kata	Total Data	Akurasi (%)
1	50	8.00
2	50	60.00
3	50	84.00
4	50	96.00

Tabel 5 menyajikan wawasan yang paling menarik mengenai perilaku sistem. Akurasi yang sangat rendah (8%) pada kueri satu kata mengonfirmasi batasan penelitian, di mana kata tunggal yang ambigu seperti “*wallahu*” tidak memberikan konteks yang cukup bagi model semantik. Namun, terjadi lonjakan performa yang eksponensial seiring bertambahnya jumlah kata dalam kueri, mencapai 96% pada kueri empat kata. Tren ini secara kuat memvalidasi bahwa model *embedding* yang digunakan sangat efektif dalam memanfaatkan konteks dari frasa untuk melakukan disambiguasi makna dan menemukan ayat yang paling relevan dengan tingkat presisi yang sangat tinggi.

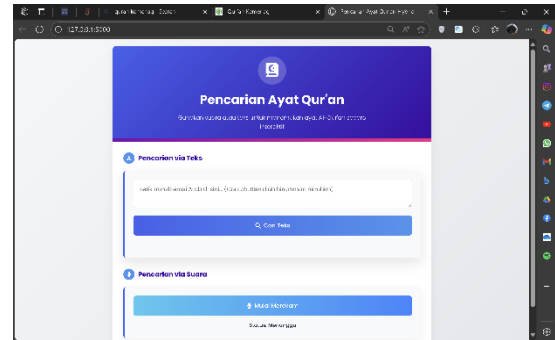
8. Pengujian audio berbasis flask

Bagian ini untuk menguji seluruh proses sistem secara *end-to-end*, Pengujian dilakukan dengan memberikan input audio bacaan ayat Al-Qur'an. Sistem kemudian mentranskripsi audio menggunakan *Whisper*, melakukan

transliterasi Latin, dan melakukan pencarian menggunakan metode *hybrid reranking*.

- Tampilan awal antarmuka *flask*

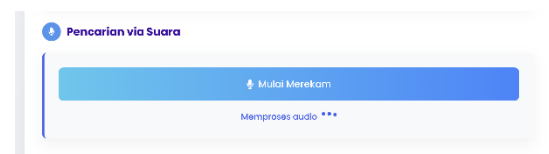
Tampilan awal aplikasi memiliki dua fitur utama, yaitu pencarian melalui teks dan pencarian melalui suara, yang dirancang sebagai instrument pendukung untuk keperluan pengujian sistem.



Gambar 3. Tampilan awal flask

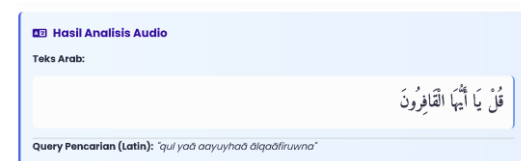
- Melakukan input audio pada tampilan antarmuka *flask*

Ketika pengguna menekan tombol “Mulai Merekam” pada fitur pencarian suara, aplikasi secara otomatis memulai proses perekaman audio melalui perangkat yang digunakan. Setelah proses perekaman selesai, audio tersebut dikirim ke sistem untuk ditranskripsi secara otomatis menggunakan model *Whisper*. Hasil transkripsi kemudian diubah menjadi teks dalam bentuk transliterasi Latin. Selama proses ini berlangsung, aplikasi menampilkan indikator status seperti “Memproses Audio” untuk memberikan informasi kepada pengguna bahwa audio sedang dalam tahap pemrosesan.



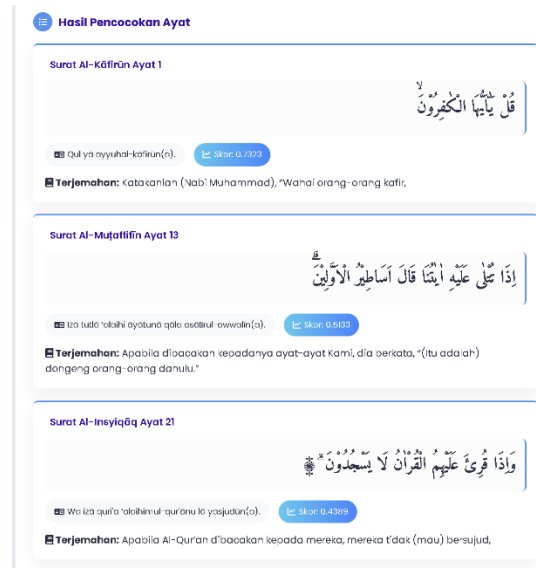
Gambar 4. Tombol inputan audio

- Hasil transkripsi oleh *whisper* yang ditampilkan pada antarmuka *flask*
- Setelah proses transkripsi audio selesai, sistem akan menampilkan hasil analisis berupa dua informasi utama yang ditujukan untuk memberikan transparansi kepada pengguna sebelum pencarian dilakukan. Pertama, sistem menampilkan *Teks Arab*, yaitu hasil transkripsi dari audio yang telah diubah ke dalam bentuk tulisan Arab sesuai dengan pengucapan pengguna. Kedua, sistem menyajikan *Query Pencarian* dalam huruf Latin, yang merupakan hasil transliterasi dari teks Arab tersebut. Teks Latin inilah yang kemudian digunakan sebagai *query* untuk menjalankan proses pencarian dalam sistem. Dengan menampilkan kedua bentuk informasi ini, pengguna dapat memahami dan memverifikasi hasil konversi audio secara lebih jelas sebelum sistem melanjutkan ke tahap pencarian.



Gambar 5. Tampilan Hasil Transkripsi Audio

- Hasil pencarian yang dilakukan oleh metode *hybrid Fuzy Jaro Winkler dan Cosine Similarity*
- Gambar 6 menunjukkan hasil pencarian ayat *Al-Qur'an* oleh sistem, yang menampilkan nama surat dan nomor ayat, teks Arab, transliterasi Latin, terjemahan Bahasa Indonesia, serta skor kecocokan. Tiga ayat ditampilkan berdasarkan tingkat relevansi, dengan ayat berskor tertinggi di urutan pertama. Ini membuktikan bahwa metode pencarian *hybrid (Jaro-Winkler dan BERT)* bekerja secara efektif.



Gambar 6. Tampilan hasil pencarian

D. KESIMPULAN DAN SARAN

Penelitian ini berhasil merancang dan mengevaluasi sistem pencarian ayat *Al-Qur'an* berbasis transliterasi Latin dengan pendekatan arsitektur hibrida dua-tahap. Sistem ini terbukti efektif dalam menjembatani kesenjangan antara variasi leksikal dan makna semantik pada korpus transliterasi non-standar. Tahap pertama menggunakan algoritma *Fuzzy Jaro-Winkler* untuk menyaring kandidat secara leksikal, sementara tahap kedua memanfaatkan *embedding DistilBERT* yang telah *di-fine-tune* untuk melakukan perankingan ulang berdasarkan kedekatan makna. Kombinasi ini memungkinkan sistem mengidentifikasi ayat yang relevan secara semantik meskipun tidak selalu cocok secara string.

Dari sisi performa, sistem menunjukkan ketangguhan yang tinggi terhadap variasi ejaan. Pengujian kuantitatif menghasilkan akurasi sebesar 92% untuk kueri dengan ejaan tidak standar dan 96% untuk skenario ideal, membuktikan bahwa pendekatan hibrida

ini mampu menangani kondisi dunia nyata secara andal.

Sebagai saran untuk pengembangan selanjutnya, penelitian ini merekomendasikan perluasan dataset ke seluruh Al-Qur'an (30 Juz) agar sistem lebih general dan aplikatif. Selain itu, perlu ada mekanisme penanganan kueri ambigu, seperti saran kueri atau pemanfaatan riwayat pencarian. Pengembangan antarmuka pengguna juga penting untuk meningkatkan pengalaman pengguna, misalnya dengan fitur audio murottal dan terjemahan multibahasa. Terakhir, eksplorasi model *end-to-end* yang mampu mempelajari fitur leksikal dan semantik secara bersamaan dapat menjadi arah baru untuk meningkatkan efisiensi dan akurasi sistem secara keseluruhan.

E. REFERENSI

- [1] Ather, M. M. (2023). The fusion of multilingual semantic search and large language models: A new paradigm for enhanced topic exploration and contextual search. [Preprint].
- [2] Poerbaningtyas, E. (2023). Penerapan metode cosine similarity pada sistem informasi retrieval pencarian terjemahan ayat-ayat suci Al-Qur'an. [Artikel ilmiah].
- [3] Rosyid, A. (2021). Problematika transliterasi Al-Qur'an. *Al-Fanar: Jurnal Ilmu Al-Qur'an dan Tafsir*, 4(2), 191–215. <https://doi.org/10.33511/alfanar.v4n2.191-215>.
- [4] Sujaini, H., Muhardi, H., & Simanjuntak, J. H. (2022). Aplikasi pengoreksi ejaan (spelling correction) pada naskah jurnal bidang informatika dengan N-gram dan Jaro-Winkler distance. *Jurnal Edukasi dan Penelitian Informatika (JEPIN)*, 8(2). <https://doi.org/10.26418/jp.v8i2.48092>.
- [5] Jayadianti, H., Santosa, B., Cahyaning, J., Saifullah, S., & Drezewski, R. (2023). Essay auto-scoring using N-gram and Jaro-Winkler based Indonesian typos. *MATRIK: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, 22(2). <https://doi.org/10.30812/matrik.v22i2.2473>.
- [6] Sanjaya, A., Setiawan, A. B., Mahdiyah, U., Farida, I. N., Prasetyo, A. R., & Universitas Nusantara PGRI Kediri. (2023). Pengukuran kemiripan makna menggunakan cosine similarity dan basis data sinonim kata (Measurement of meaning similarity using cosine similarity and word synonyms database). *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, 10(4).
- [7] Prismadana, T. A. (2023). Aplikasi ruang tugas dengan deteksi kemiripan teks pada dokumen tugas menggunakan cosine similarity. *Jurnal Informatika dan Multimedia*, 15(1). <https://doi.org/10.33795/jtim.v15i1.4405>.
- [8] Kurniadi, D., Gernowo, R., Surarso, B., Wibowo, A., & Warsito, B. (2023). Sistem penilaian jawaban singkat otomatis pada ujian online berbasis komputer menggunakan algoritma

- cosine similarity. *Jurnal Edukasi dan Penelitian Informatika (JEPIN)*, 9(2).
<https://doi.org/10.26418/jp.v9i2.66934>.
- [9] Pitchandi, P., & Balakrishnan, M. (2023). Document clustering analysis with aid of adaptive Jaro-Winkler with Jellyfish search clustering algorithm. *Advances in Engineering Software*, 175, 103322.
<https://doi.org/10.1016/j.advengsoft.2022.103322>.
- [10] Ilyasa, M. D. H., & Yamasari, Y. (2023). Perbandingan cosine similarity dan Euclidean distance pada model rekomendasi buku dengan metode item-based collaborative filtering. *Journal of Informatics and Computer Science (JINACS)*, 4(3), 264–274.
<https://doi.org/10.26740/jinacs.v4n03.p264-274>.
- [11] Amorese, T., Greco, C., Cuciniello, M., Milo, R., Sheveleva, O., & Glackin, N. (2023). Automatic speech recognition (ASR) with Whisper: Testing performances in different languages. In *CEUR Workshop Proceedings*.
- [12] Rasool, S., Mushtaq, T., & Rahat, N. (2022). Multidisciplinary status of translation sciences. *Al-Aijaz Research Journal of Islamic Studies & Humanities*, 6(2), 165–170.
[https://doi.org/10.53575/u15.v6.02\(22\).165-170](https://doi.org/10.53575/u15.v6.02(22).165-170).