JURNAL TEKNIK INFORMATIKA DAN MULTIMEDIA Vol 6, No.2, Oktober 2026, pp. 69 - 88
p-ISSN : 2808-909X (print)
e-ISSN : 2798-2505 (online)
http://journal.politeknik-pratama.ac.id/index.php/JTIM · page 69
DOI : https://doi.org/10.51903/informatika.v6i2.1952

TOLERANT RETRIEVAL BERBASIS MORFOLOGI DAN VECTOR SPACE MODEL UNTUK PENCARIAN BERITA KESEHATAN BERBAHASA INDONESIA

Wa Ode Sagita Ramadhani Bahmid[1], Nirna Dewi Reskya Ningsih Arif [2], Rabiatul Adawiah [3], Rahmah Yuniati[4], Adha Mashur Sajiah[5]
[1] Universitas Halu Oleo
Jl. H.E.A. Mokodompit, Kampus Hijau Bumi Tridharma, Anduonohu, Kendari 93232, Sulawesi Tenggara, Indonesia, e-mail: octaviasagita11@gmail.com
[2] Universitas Halu Oleo
Jl. H.E.A. Mokodompit, Kampus Hijau Bumi Tridharma, Anduonohu, Kendari 93232, Sulawesi Tenggara, Indonesia, e-mail: nirnadewi3@gmail.com
[3] Universitas Halu Oleo
Jl. H.E.A. Mokodompit, Kampus Hijau Bumi Tridharma, Anduonohu, Kendari 93232, Sulawesi Tenggara, Indonesia, e-mail: Rabiatuladawia019@gmail.com
[4] Universitas Halu Oleo
Jl. H.E.A. Mokodompit, Kampus Hijau Bumi Tridharma, Anduonohu, Kendari 93232, Sulawesi Tenggara, Indonesia, e-mail : rahmahyuniati262@gmail.com
[5] Universitas Halu Oleo
Jl. H.E.A. Mokodompit, Kampus Hijau Bumi Tridharma, Anduonohu, Kendari 93232, Sulawesi Tenggara, Indonesia, e-mail : adha.m.sajiah@uho.ac.id

ARTICLE INFO

Article history:

Received 10 July 2026
Received in revised form 20 August 2026
Accepted 15 September 2026
Available online 15 October 2026

ABSTRACT

The growth of digital health information has led to a significant increase in Indonesian-language health news, creating a need for information retrieval systems that can deliver fast and relevant search results. However, searching Indonesian-language content is challenged by morphological variation caused by affixation, as well as typographical errors (typos), both of which can degrade search quality. This study aims to implement a morphology-based Tolerant Retrieval method combined with the Vector Space Model (VSM) to improve the effectiveness of searching Indonesian health news. The system is built through a preprocessing pipeline consisting of case folding, cleaning, tokenizing, stopword removal, and stemming using the Nazief-Adriani algorithm via the PySastrawi library. Documents are represented using TF-IDF weighting, and similarity is measured using Cosine Similarity. To increase tolerance to word variation and typing errors, the system applies a Bigram Index, a Permuterm Index, and the Damerau-Levenshtein algorithm as mechanisms for query correction and expansion. The dataset consists of 958 Indonesian health news documents and 45 test queries divided into three main categories (normal, morphological variation, and typo), each accompanied by

Received July 10, 2026; Accepted September 8, 2026; Available online October 15, 2026

p-ISSN : 2808-909X e-ISSN : 2798-2505

document relevance ground truth. Evaluation results show that the system achieves an average Precision of 85.10%, Recall of 65.11%, and F1-Score of 68.72% at a top-K of 50 documents. These results indicate that integrating morphology-based Tolerant Retrieval with the Vector Space Model improves search result relevance and effectively handles morphological variation and typing errors in Indonesian health news retrieval

Keywords: Tolerant Retrieval, Indonesian Morphology, Vector Space Model, TF-IDF, Health News.

Abstrak

Perkembangan informasi kesehatan berbasis digital menyebabkan jumlah berita kesehatan berbahasa Indonesia meningkat secara signifikan, sehingga diperlukan sistem temu kembali informasi yang mampu menyediakan hasil pencarian secara cepat dan relevan. Namun, pencarian informasi berbahasa Indonesia menghadapi tantangan berupa variasi morfologi kata akibat proses afiksasi serta kesalahan pengetikan (*typo*) yang dapat menurunkan kualitas hasil pencarian. Penelitian ini bertujuan untuk mengimplementasikan metode *Tolerant Retrieval* berbasis morfologi yang dikombinasikan dengan *Vector Space Model* (VSM) untuk meningkatkan efektivitas pencarian berita kesehatan berbahasa Indonesia. Sistem dibangun melalui tahapan *preprocessing* yang meliputi *case folding*, *cleaning*, *tokenizing*, *stopword removal*, dan *stemming* menggunakan algoritma Nazief-Adriani melalui pustaka PySastrawi. Representasi dokumen dilakukan menggunakan pembobotan TF-IDF dan pengukuran tingkat kemiripan menggunakan *Cosine Similarity*. Untuk meningkatkan toleransi terhadap variasi kata dan kesalahan pengetikan, sistem menerapkan *Bigram Index*, *Permuterm Index*, serta algoritma Damerau-Levenshtein sebagai mekanisme koreksi dan ekspansi *query*. Dataset yang digunakan terdiri dari 958 dokumen berita kesehatan berbahasa Indonesia dengan 45 *query* uji yang dibagi ke dalam tiga kategori utama (normal, variasi morfologi, dan *typo*) yang dilengkapi *ground truth* relevansi dokumen. Hasil pengujian menunjukkan bahwa sistem memperoleh nilai rata-rata Precision sebesar 85,10%, Recall sebesar 65,11%, dan F1-Score sebesar 68,72% pada pengujian *top-K* sebesar 50 dokumen. Hasil tersebut menunjukkan bahwa integrasi *Tolerant Retrieval* berbasis morfologi dan *Vector Space Model* mampu meningkatkan relevansi hasil pencarian serta efektif dalam menangani variasi morfologi dan kesalahan pengetikan pada pencarian berita kesehatan berbahasa Indonesia.

.

Kata Kunci: Tolerant Retrieval, Morfologi Bahasa Indonesia, Vector Space Model, TF-IDF, Berita Kesehatan.

1. PENDAHULUAN

Perkembangan teknologi informasi yang pesat dalam beberapa dekade terakhir telah mendorong pertumbuhan data digital secara eksponensial, terutama dalam sektor kesehatan. Berita dan artikel kesehatan berbasis web terus bertambah setiap harinya, mencakup topik seperti penyakit, obat-obatan, gizi, dan kebijakan kesehatan masyarakat. Kondisi ini menciptakan kebutuhan mendesak terhadap sistem temu kembali informasi (Information Retrieval/IR) yang mampu membantu pengguna menemukan informasi kesehatan yang relevan secara cepat dan akurat [1].

Sistem temu kembali informasi telah digunakan secara luas, tidak hanya pada mesin pencari umum, tetapi juga pada platform khusus seperti perpustakaan digital medis, sistem e-health, arsip jurnal ilmiah kesehatan, dan sistem rekomendasi berbasis teks medis [2]. Seiring meningkatnya volume data, kompleksitas proses pencarian pun semakin tinggi. Pengguna tidak hanya menginginkan kecepatan, tetapi juga akurasi dan relevansi hasil pencarian yang sesuai dengan konteks kebutuhan mereka [3].

Salah satu tantangan utama dalam sistem temu kembali informasi berbahasa Indonesia adalah aspek morfologi bahasa. Bahasa Indonesia memiliki struktur morfologi yang kompleks, di mana sebuah kata dasar dapat mengalami berbagai perubahan bentuk melalui proses afiksasi (penambahan awalan, akhiran, sisipan, dan konfiks), reduplikasi, serta perubahan fonologis [4]. Sebagai contoh, dalam konteks berita

kesehatan, kata dasar "sehat" dapat berubah menjadi "kesehatan", "menyehatkan", "penyehatan", dan "dipersehatkan". Tanpa mekanisme yang mampu mengidentifikasi hubungan antar-bentuk kata tersebut, sistem akan memperlakukan setiap kata sebagai entitas yang sepenuhnya berbeda [5].

Akibat ketidakmampuan sistem dalam mengenali variasi morfologi adalah menurunnya nilai recall, karena dokumen yang sebenarnya relevan dapat terlewatkan, serta menurunnya precision, karena dokumen yang tidak relevan ikut terambil [6]. Permasalahan ini semakin kritis pada domain berita kesehatan yang menggunakan banyak istilah teknis medis yang mengalami berbagai proses afiksasi dalam Bahasa Indonesia.

Untuk mengatasi permasalahan tersebut, pendekatan tolerant retrieval menjadi salah satu solusi yang menjanjikan. Tolerant retrieval adalah metode yang memungkinkan pencarian tetap berjalan efektif meskipun terdapat ketidaksesuaian bentuk kata antara query pengguna dan dokumen. Pendekatan ini bekerja melalui teknik pencocokan yang fleksibel, termasuk analisis morfologi berbasis stemming dan lemmatization, sehingga sistem mampu memahami keterkaitan antar kata yang berasal dari akar kata yang sama [11].

Di sisi lain, Vector Space Model (VSM) merupakan salah satu model representasi dokumen yang paling banyak digunakan dalam sistem temu kembali informasi. VSM merepresentasikan dokumen dan query sebagai vektor dalam ruang berdimensi tinggi, kemudian mengukur relevansi melalui perhitungan kemiripan kosinus (cosine similarity). Kombinasi antara tolerant retrieval berbasis morfologi dan VSM berpotensi menghasilkan sistem pencarian yang lebih akurat, khususnya pada teks berbahasa Indonesia yang kaya variasi morfologi [12].

Beberapa penelitian sebelumnya telah mengkaji aspek terkait. Putri dkk. (2021) menerapkan VSM dengan pembobotan TF-IDF pada sistem pencarian dokumen akademik berbahasa Indonesia dan menunjukkan peningkatan relevansi dibanding metode Boolean sederhana, namun belum mempertimbangkan variasi morfologi dalam proses indeksasi [13]. Paskahningrum dkk. (2023) mengevaluasi algoritma stemming Nazief-Adriani pada teks non-formal Bahasa Indonesia dan memperoleh akurasi yang tinggi, tetapi belum mengintegrasikannya ke dalam sistem IR secara menyeluruh. Sementara itu, Rahman dkk. (2026) mengembangkan sistem berbasis Retrieval-Augmented Generation (RAG) yang mampu menjawab pertanyaan dari dokumen PDF secara kontekstual, namun belum menangani variasi morfologi secara spesifik dalam proses pencarian [14].

Berdasarkan celah penelitian tersebut, penelitian ini mengusulkan integrasi tolerant retrieval berbasis morfologi Bahasa Indonesia dengan Vector Space Model dalam satu kerangka sistem temu kembali informasi yang diterapkan pada korpus berita kesehatan Indonesia. Pendekatan ini meliputi tahapan preprocessing teks, stemming menggunakan algoritma Nazief-Adriani, pembobotan TF-IDF, serta perhitungan cosine similarity untuk mengukur relevansi dokumen terhadap query pengguna.

Penelitian ini bertujuan untuk: (1) mengimplementasikan sistem temu kembali informasi berbasis morfologi dan VSM pada berita kesehatan berbahasa Indonesia; (2) mengevaluasi dampak penerapan stemming terhadap peningkatan relevansi hasil pencarian; dan (3) mengukur performa sistem menggunakan metrik precision, recall, dan F1-score. Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi nyata dalam pengembangan sistem pencarian informasi kesehatan yang lebih adaptif terhadap karakteristik linguistik Bahasa Indonesia.

2. TINJAUAN PUSTAKA

2.1. Information Retrieval

Pengambilan Informasi (Information Retrieval/IR) adalah sistem yang digunakan untuk menemukan dokumen yang relevan berdasarkan kebutuhan pengguna dari kumpulan dokumen yang tersedia. Sistem ini bekerja dengan mencocokkan kueri pengguna dengan dokumen yang tersimpan, menghasilkan dokumen dengan tingkat relevansi tertentu. Pengambilan Informasi banyak diterapkan dalam mesin pencari, perpustakaan digital, sistem pengarsipan elektronik, dan berbagai aplikasi pengambilan dokumen lainnya.

p-ISSN : 2808-909X e-ISSN : 2798-2505

Tujuan utama Pengambilan Informasi adalah untuk membantu pengguna mendapatkan informasi yang relevan dengan cepat dan efektif dari kumpulan dokumen yang tersedia [7].

2.2. Morfologi Bahasa Indonesia

Morfologi adalah cabang linguistik yang mempelajari struktur dan proses pembentukan kata. Dalam bahasa Indonesia, pembentukan kata terjadi melalui afiksasi, yang menghasilkan berbagai bentuk kata berifiks. Variasi bentuk kata ini dapat menyebabkan masalah dalam proses penelusuran informasi karena kata-kata dengan makna yang sama dapat muncul dalam bentuk yang berbeda. Oleh karena itu, diperlukan proses stemming untuk mengubah kata berifiks menjadi bentuk dasarnya. Penerapan stemming dalam Penelusuran Informasi dapat meningkatkan relevansi hasil pencarian dengan mengelompokkan berbagai variasi kata ke dalam satu representasi [8].

2.3. Pencarian Toleran

Pencarian Toleran adalah pendekatan dalam sistem pencarian informasi yang memungkinkan sistem untuk menemukan dokumen yang relevan meskipun terdapat kesalahan ejaan dalam kueri pengguna. Kesalahan ini dapat mencakup kesalahan ketik, penambahan karakter, penghilangan karakter, atau pertukaran huruf. Pendekatan ini umumnya menggunakan teknik koreksi ejaan dan perhitungan spasi kata untuk memperbaiki kesalahan ejaan sebelum proses pencarian dimulai. Dengan Pencarian Toleran, sistem dapat meningkatkan kualitas hasil pencarian dan mengurangi kemungkinan terlewatnya dokumen yang relevan karena kesalahan kueri [9].

2.4. Model Ruang Vektor

Model Ruang Vektor (VSM) adalah salah satu metode yang paling banyak digunakan dalam Penelusuran Informasi. Metode ini merepresentasikan dokumen dan kueri dalam bentuk vektor sehingga kesamaan di antara keduanya dapat dihitung. Setiap istilah dalam dokumen diberi bobot tertentu menggunakan metode pembobotan seperti TF-IDF, dan kemudian kesamaan antara kueri dan dokumen dihitung menggunakan Kesamaan Kosinus. Dokumen dengan skor kesamaan tertinggi ditempatkan di bagian atas hasil pencarian. Kemampuan VSM untuk memberi peringkat dokumen menjadikannya metode yang efektif untuk menemukan informasi yang relevan berdasarkan kebutuhan pengguna [10].

2.5. Penelitian Sebelumnya

Penelitian sebelumnya digunakan untuk memahami perkembangan penelitian yang berkaitan dengan Penelusuran Informasi, Penelusuran Toleran, Morfologi Bahasa Indonesia, dan Model Ruang Vektor. Ringkasan penelitian sebelumnya yang relevan dengan penelitian ini dapat dilihat pada Tabel 1.

Tabel 1. Ringkasan Penelitian Terdahulu

No	Peneliti	Metode	Hasil
1	Ramadhan dan Lhaksmana (2020)	Vector Space Model dan Spelling Correction	Penerapan spelling correction meningkatkan nilai Mean Average Precision dari 73% menjadi 81% dan Recall dari 80% menjadi 89%.
2	Erwina dkk. (2021)	Bigram Vector dan Minimum Edit Distance	Sistem mampu mendeteksi dan mengidentifikasi berbagai jenis kesalahan ejaan Bahasa Indonesia.
3	Qulub dkk. (2023)	Hamming Distance	Metode menghasilkan akurasi koreksi kata sebesar 98,33% pada pengujian kata Bahasa Indonesia.
4	Santoso dan Yan (2024)	Levenshtein Distance dan Empirical Method	Sistem typo correction menghasilkan precision sebesar 92% dan F1-score sebesar 90,5%.
5	Junaidi dkk. (2024)	Vector Space Model	VSM mampu meningkatkan efektivitas pencarian dokumen

			melalui pencocokan query dan pemeringkatan dokumen.
6	Pardede dan Darmawan (2025)	Perbandingan Algoritma Stemming	Algoritma Sastrawi menghasilkan nilai presisi tertinggi dibandingkan algoritma stemming lainnya.

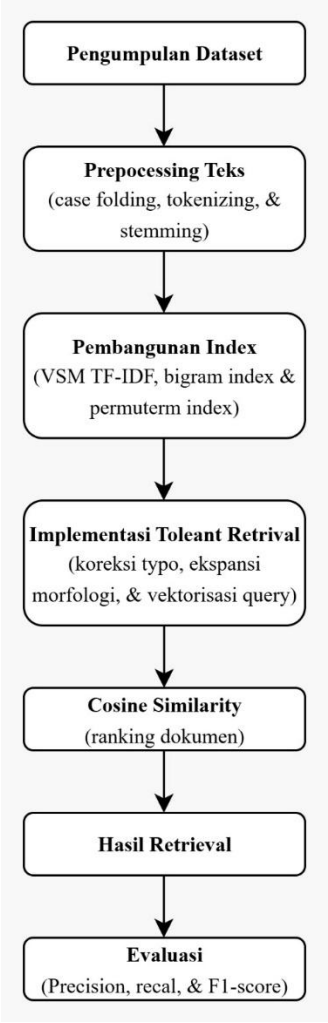
Berdasarkan penelitian sebelumnya, sebagian besar studi berfokus pada penerapan Model Ruang Vektor, stemming, dan koreksi ejaan secara terpisah. Penelitian tentang penerapan Tolerant Retrieval berbasis morfologi yang dikombinasikan dengan Model Ruang Vektor pada dokumen berita kesehatan berbahasa Indonesia masih terbatas. Oleh karena itu, penelitian ini menerapkan Tolerant Retrieval berbasis morfologi dan Model Ruang Vektor untuk meningkatkan relevansi hasil pencarian dalam kumpulan berita kesehatan berbahasa Indonesia.

3. METODOLOGI PENELITIAN

3.1. Gambaran Umum Sistem

Penelitian ini merancang dan mengimplementasikan sistem *tolerant retrieval* berbasis morfologi untuk pencarian berita kesehatan berbahasa Indonesia. Sistem mengintegrasikan empat komponen utama: pipeline *preprocessing* teks berbasis PySastrawi, pemodelan dokumen menggunakan Vector Space Model (VSM) dengan pembobotan TF-IDF, mekanisme *tolerant retrieval* yang mencakup Bigram Index, Permuterm Index, dan koreksi kesalahan ketik menggunakan algoritma Damerau-Levenshtein, serta evaluasi kinerja sistem menggunakan metrik Precision, Recall, dan F1-Score. Alur penelitian secara keseluruhan disajikan pada gambar dibawah ini.

p-ISSN : 2808-909X e-ISSN : 2798-2505



Gambar 1. Flowchart Penelitian

3.2. Pengumpulan Dataset

Dataset yang digunakan terdiri atas 958 dokumen berita kesehatan berbahasa Indonesia. Dataset diperoleh dari SEACrowd yang tersedia pada repositori Hugging Face menggunakan fungsi load_dataset("SEACrowd/liputan6"). Dataset tersebut merupakan hasil kurasi dari artikel berita Liputan6. Pada penelitian ini, data tidak diperoleh melalui proses scraping secara langsung, melainkan dengan memanfaatkan dataset yang telah tersedia pada repositori tersebut. Selanjutnya dilakukan proses filtering berbasis kata kunci kesehatan untuk memilih dokumen yang relevan dengan domain kesehatan. Setelah proses pembersihan data dan penghapusan dokumen duplikat, diperoleh sebanyak 958 dokumen berita kesehatan berbahasa Indonesia yang digunakan sebagai korpus penelitian. Setiap dokumen memiliki dua atribut utama, yaitu *doc_id* sebagai pengenal unik dokumen dan *content* yang memuat isi teks berita. Selain itu, digunakan pula file *hasil_query_relevansi_final_morfologi.xlsx* yang memuat 45 query yang mencakup berbagai kompleksitas leksikal, mulai dari kueri normal, variasi morfologi, hingga kueri dengan tingkat kesalahan pengetikan (*typo*) yang bervariasi berdasarkan dataset berita kesehatan, beserta daftar dokumen relevan yang berfungsi sebagai *ground truth* evaluasi.

3.3. Prepocessing Teks

Preprocessing merupakan tahap fundamental dalam sistem temu kembali informasi yang bertujuan mengubah teks mentah menjadi representasi terstruktur yang dapat diproses secara komputasi. Pada penelitian ini, preprocessing dilakukan dalam dua tahap berurutan: pembersihan awal konten dan pipeline preprocessing linguistik.

3.3.1. Pembersihan Awal Konten

Sebelum preprocessing linguistik, seluruh isi dokumen terlebih dahulu diproses menggunakan fungsi *bersihkan_konten()*. Tahap ini menghapus header redaksional khas Liputan6.com menggunakan ekspresi reguler, tautan URL, serta merapikan spasi berlebih. Pembersihan awal dilakukan sebelum pipeline utama agar fragmen teks dari header berita tidak ikut terstemming dan tidak muncul pada cuplikan hasil retrieval.

3.3.2. Pipeline Preprocessing Linguistik

Pipeline preprocessing linguistik terdiri atas lima tahap berurutan yang diterapkan pada setiap dokumen. Preprocessing teks memainkan peran krusial dalam meningkatkan akurasi sistem temu kembali informasi,. Kelima tahap tersebut adalah sebagai berikut.

Table 2. Tahapan Pipeline Preprocessing Teks

Tahap	Fungsi	Keterangan
Case Folding	case_folding()	Mengubah seluruh karakter huruf menjadi huruf kecil untuk menghindari duplikasi token akibat perbedaan kapitalisasi
Cleaning	cleaning()	Menghapus angka, tanda baca, dan karakter tidak relevan menggunakan ekspresi reguler; merapikan spasi berlebih
Tokenizing	tokenizing()	Memecah teks menjadi daftar token individual berdasarkan spasi (whitespace tokenization)
Stopword Removal	stopword_removal()	Menghapus 843 stopword, mencakup daftar bawaan PySastrawi yang diperluas dengan custom stopword khas berita (nama hari, bulan, dan kota besar)
Stemming (Sastrawi)	stemming()	Mengubah kata berimbuhan menjadi kata dasar menggunakan algoritma Nazief-Adriani melalui pustaka PySastrawi.

Pustaka PySastrawi dipilih karena merupakan implementasi Python dari algoritma Nazief-Adriani yang terbukti efektif untuk bahasa Indonesia. Seluruh token hasil preprocessing disimpan dalam kolom *tokens* (daftar token) dan *content_preprocessed* (teks gabungan token), yang selanjutnya menjadi masukan pada tahap pembangunan VSM

3.4. Pemodelan Vektor Space Model (VSM) dengan TF-IDF

Vector Space Model (VSM) merepresentasikan setiap dokumen sebagai vektor numerik dalam ruang berdimensi tinggi yang sesuai dengan jumlah term unik dalam vocabulary. Pembobotan dilakukan menggunakan skema TF-IDF yang mengkombinasikan frekuensi lokal suatu term dalam dokumen dengan kelangkaannya di seluruh korpus.

3.4.1. Pembangunan Vocabulary

Vocabulary dibangun dari seluruh token unik hasil preprocessing 958 dokumen, sehingga diperoleh himpunan term unik $V = \{t_1, t_2, \ldots, t_{|V|}\}$ yang diurutkan secara leksikografis. Selanjutnya setiap term $t \in V$ dipetakan secara satu-ke-satu ke sebuah indeks numerik melalui fungsi pemetaan.

$$f: V \rightarrow \{0,1, \ldots, |V| - 1\}$$

Sehingga setiap term memiliki representasi indeks yang unik. Pemetaan ini digunakan untuk mempercepat proses pembentukkan matriks TF, TF-IDF, dan representasi vektor query maupun dokumen. Pada implementasi sistem, fungsi pemetaan tersebut direalisasikan menggunakan struktur data `term2idx`.

3.4.2. Perhitungan Term Frequency (TF)

p-ISSN : 2808-909X e-ISSN : 2798-2505

Term Frequency (TF) dihitung sebagai rasio kemunculan suatu term terhadap total token dalam satu dokumen, menggunakan formula:

$$TF(t,d) = \frac{f_{t,d}}{|d|}$$

dimana $f_{t,d}$ menyatakan frekuensi kemunculan term $t$pada dokumen d, sedangkan d menyatakan jumlah seluruh token hasil preprocessing pada dokumen d. Seluruh nilai TF kemudian direpresentasikan dalam bentuk matriks $\mathbf{TF} \in \mathbb{R}^{958\times|V|}$ dengan setiap baris merepresentasikan satu dokumen dan setiap kolom merepresentasikan satu term pada vocabulary. Karena sebagian besar elemen matriks bernilai nol, matriks disimpan menggunakan representasi *Compressed Sparse Row (CSR)* untuk meningkatkan efisiensi penggunaan memori.

3.4.3. Perhitungan Inverse Document Frequency (IDF)

IDF mengukur kelangkaan suatu term di seluruh korpus. Formula IDF yang digunakan adalah:

$$IDF(t) = \log\left(\frac{N+1}{df(t)+1}\right) + 1$$

di mana $N = 958$ adalah jumlah dokumen dalam korpus dan $df(t)$ adalah jumlah dokumen yang mengandung term t. Teknik smoothing +1 pada pembilang dan penyebut diterapkan untuk menghindari pembagian dengan nol dan untuk memberikan nilai IDF yang tidak nol bagi term yang muncul di seluruh dokumen.

3.4.4. Matriks TF-IDF dan Normalisasi L2

Matriks TF-IDF diperoleh melalui perkalian elemen-per-elemen (*element-wise multiplication*) antara matriks TF dan vektor IDF. $\mathrm{TF-IDF} = \mathrm{TF} \odot \mathrm{IDF}$. Selanjutnya, setiap vektor dokumen dinormalisasi menggunakan norma L2 (norma Euclidean) agar perbandingan *cosine similarity* tidak dipengaruhi panjang dokumen:

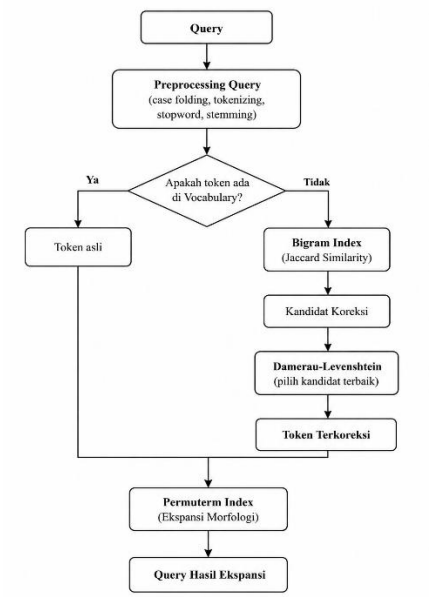
$$\mathrm{d}_{\mathrm{norm}} = \frac{\mathrm{d}}{\|\mathrm{d}\|_2}$$

Dokumen dengan norma $\| \mathbf{d} \|_2 = 0$ diberi nilai 1 untuk menghindari pembagian dengan nol. Dengan normalisasi ini, *dot product* antara vektor query dan vektor dokumen secara langsung menghasilkan nilai *Cosine Similarity* yang berada dalam rentang [0, 1].

3.5. Implementasi Tolerant Retrieval

Tolerant retrieval merupakan pendekatan yang memungkinkan sistem temu kembali informasi tetap menghasilkan dokumen relevan meskipun terdapat variasi leksikal, kesalahan ketik, atau perbedaan morfologis antara query pengguna dan dokumen dalam korpus. Pada penelitian ini, tolerant retrieval diimplementasikan melalui tiga tahapan utama, yaitu pembangunan Bigram Index, proses koreksi typo menggunakan kombinasi Jaccard Similarity dan algoritma Damerau-Levenshtein, serta ekspansi morfologi menggunakan Permuterm Index.

p-ISSN : 2808-909X e-ISSN : 2798-2505



Gambar 2. Alur Implementasi Tolerant Retrieval

Berdasarkan Gambar 2. Alur implementasi dimulai dengan pemeriksaan setiap token hasil preprocessing terhadap vocabulary dan whitelist. Apabila token tidak ditemukan, sistem melakukan koreksi typo untuk memperoleh token yang paling sesuai. Selanjutnya, token hasil koreksi diproses menggunakan Permuterm Index untuk menghasilkan variasi morfologi yang memiliki stem yang sama. Hasil akhir dari proses tersebut berupa query hasil ekspansi yang kemudian digunakan pada proses pembobotan TF-IDF dan perhitungan Cosine Similarity.

3.5.1. Bigram Index

Bigram Index dibangun dengan memecah setiap term dalam vocabulary menjadi himpunan bigram karakter, yaitu pasangan dua karakter berurutan, dengan penambahan padding simbol '$' di awal dan akhir kata. Penggunaan karakter padding memungkinkan bigram menangkap posisi awal dan akhir kata, sehingga meningkatkan diskriminasi antar term. Bigram Index berfungsi sebagai filter awal dalam proses koreksi typo dengan mengidentifikasi kandidat term yang memiliki kemiripan karakter terhadap kata query, menggunakan Jaccard Similarity sebagai ukuran kesamaan himpunan bigram.

3.5.2. Koreksi Kesalahan Ketik

Proses koreksi typo dilakukan untuk memperbaiki token query yang tidak ditemukan pada vocabulary maupun whitelist sehingga dapat dicocokkan dengan dokumen yang relevan. Pada penelitian ini, koreksi typo dilakukan melalui dua tahapan, yaitu penyaringan kandidat menggunakan Jaccard Similarity dan pemilihan kandidat terbaik menggunakan algoritma Damerau-Levenshtein.

a. Jaccad Similarity

Jaccard Similarity digunakan untuk mengukur tingkat kemiripan antara himpunan bigram token query dengan himpunan bigram setiap term pada Bigram Index. Nilai kemiripan dihitung berdasarkan rasio jumlah irisan terhadap gabungan kedua himpunan bigram. Semakin besar nilai Jaccard Similarity, semakin tinggi tingkat kemiripan karakter antara kedua kata. Pada penelitian ini, hanya kandidat dengan nilai Jaccard Similarity $\geq 0{,}15$ yang dipertahankan, kemudian dipilih 100 kandidat dengan nilai kemiripan tertinggi untuk diproses pada tahap berikutnya. Tahap ini berfungsi sebagai proses penyaringan sehingga algoritma Damerau-Levenshtein hanya melakukan perhitungan terhadap kandidat yang memiliki kemungkinan tinggi sebagai hasil koreksi.

p-ISSN : 2808-909X e-ISSN : 2798-2505

b.Algoritma Damerau-Levenshtein

Setelah kandidat diperoleh melalui Jaccard Similarity, sistem menghitung jarak edit antara token query dan setiap kandidat menggunakan algoritma Damerau-Levenshtein. Algoritma ini mengukur jumlah operasi minimum yang diperlukan untuk mengubah satu string menjadi string lain melalui empat jenis operasi, yaitu penyisipan (*insertion*), penghapusan (*deletion*), substitusi (*substitution*), dan transposisi dua karakter yang berdekatan (*transposition*). Pada penelitian ini, kandidat dengan jarak edit terkecil dipilih sebagai hasil koreksi typo. Sistem hanya mempertahankan kandidat dengan jarak edit ≤ 2, dengan penambahan toleransi +1 untuk token yang memiliki panjang enam karakter atau lebih, serta membatasi selisih panjang maksimum dua karakter. Hasil dari tahap ini berupa token yang telah dikoreksi dan digunakan sebagai masukan pada proses ekspansi morfologi.

3.5.3. Permuterm Index dan Ekspansi Morfologi

Permuterm Index dibangun dengan menghasilkan seluruh rotasi siklis dari setiap term dalam vocabulary setelah ditambahkan simbol **'$'** di akhir kata. Struktur indeks ini memungkinkan pencarian berdasarkan prefiks dilakukan secara efisien melalui rotasi kata yang telah dibentuk. Pada penelitian ini, Permuterm Index dimanfaatkan untuk melakukan ekspansi morfologi terhadap token hasil koreksi typo. Sistem mencari seluruh entri yang diawali dengan pola '$' + token query, kemudian melakukan penyaringan menggunakan proses stemming dengan algoritma Sastrawi. Hanya term yang memiliki hasil stem identik dengan token query yang dipertahankan sebagai hasil ekspansi. Dengan demikian, variasi kata yang dihasilkan merupakan variasi morfologis yang berasal dari akar kata yang sama, bukan sekadar kata yang memiliki kemiripan karakter. Hasil akhir dari proses ini berupa himpunan token hasil ekspansi morfologi yang kemudian digunakan sebagai query akhir pada proses pembobotan TF-IDF dan perhitungan Cosine Similarity.

3.6. Pipeline Pemrosesan Query

Pemrosesan query dilakukan melalui fungsi *tolerant_retrieval()* yang mengintegrasikan seluruh komponen sistem dalam enam langkah berurutan.

Table 3. Langkah-langkah Pipeline Tolerant Retrieval

Langkah	Proses	Keterangan
1	Preprocessing Query	Query teks diproses melalui pipeline case folding, cleaning, tokenizing, stopword removal, dan stemming yang identik dengan preprocessing dokumen
2	Koreksi Typo	Setiap token diperiksa ke vocabulary dan whitelist; jika tidak dikenal, kandidat diambil via Bigram Index dan diranking menggunakan Damerau-Levenshtein
3	Ekspansi Morfologi	Token valid atau hasil koreksi diekspansi menggunakan Permuterm Index untuk mendapatkan seluruh varian berimbuhan yang ada dalam korpus
4	Ekspansi Sinonim	Sinonim yang didefinisikan pada ground truth diproses melalui pipeline preprocessing yang sama dan diekspansi secara morfologi
5	Vektorisasi Query	Seluruh term hasil ekspansi digabungkan menjadi vektor TF-IDF menggunakan vocabulary dan IDF yang identik dengan matriks dokumen, kemudian dinormalisasi L2
6	Cosine Similarity & Ranking	Dot product antara vektor query dan matriks dokumen ternormalisasi menghasilkan skor Cosine Similarity; dokumen diurutkan berdasarkan skor tertinggi (top-K)

p-ISSN : 2808-909X e-ISSN : 2798-2505

3.7. Evaluasi Kinerja Sistem

Kinerja sistem dievaluasi menggunakan tiga metrik standar temu kembali informasi: Precision, Recall, dan F1-Score. Evaluasi dilakukan terhadap 15 query pada nilai top-K yang bervariasi (10, 20, 30, 50) untuk mengidentifikasi nilai optimal.

Table 4. Definisi Metrik Evaluasi Sistem Temu Kembali Informasi

Metrik	Formula	Keterangan
Precision	TP / |Retrieved|	Proporsi dokumen relevan dari seluruh dokumen yang dikembalikan sistem; mengukur ketepatan hasil retrieval
Recall	TP / |Relevant|	Proporsi dokumen relevan yang berhasil dikembalikan dari seluruh dokumen relevan yang tersedia; mengukur kelengkapan hasil
F1-Score	$2 \times (P \times R) / (P + R)$	Rata-rata harmonik Precision dan Recall; memberikan ukuran keseimbangan antara ketepatan dan kelengkapan hasil retrieval

di mana TP adalah jumlah dokumen yang dikembalikan sistem dan sekaligus relevan (*true positive*), |Retrieved| adalah jumlah seluruh dokumen yang dikembalikan, dan |Relevant| adalah jumlah dokumen relevan menurut ground truth. Nilai rata-rata (mean) dari ketiga metrik dihitung untuk seluruh 15 query sebagai ukuran kinerja keseluruhan sistem.

4. HASIL DAN PEMBAHASAN

4.1. Tahap Preprocessing Teks

Dataset terdiri atas 958 dokumen berita dengan dua atribut, yaitu doc_id sebagai pengenal unik dokumen dan content yang memuat isi teks berita. Selain itu, digunakan pula file hasil_query_relevansi_final_morfologi.xlsx yang memuat 15 query beserta daftar dokumen relevan sebagai ground truth evaluasi. Sebelum preprocessing linguistik, seluruh isi dokumen terlebih dahulu dibersihkan menggunakan fungsi *bersihkan_konten()*. Proses ini menghapus header redaksional khas Liputan6.com, tautan URL, serta merapikan spasi berlebih.

Pembersihan awal ini dilakukan sebelum preprocessing utama agar fragmen teks tidak ikut terstemming dan tidak muncul pada cuplikan hasil retrieval. Preprocessing utama diterapkan secara berurutan (pipeline) pada setiap dokumen menggunakan pustaka PySastrawi. Total stopword yang digunakan sebanyak 843 kata, merupakan gabungan daftar bawaan Sastrawi dengan custom stopword khas berita kesehatan (nama hari, bulan, dan kota besar). Rincian setiap tahap beserta contoh aktual dari dokumen D1 disajikan pada Tabel 5.

Tabel 5. Tahapan dan Output Preprocessing Teks pada D1

Tahap	Proses	Output
Case Folding	Mengubah semua huruf menjadi huruf kecil	"Jumlah Orang Yang Mengidap..." → "jumlah orang yang mengidap..."
Cleaning	Menghapus angka, tanda baca, dan karakter tidak relevan	"jumlah orang yang mengidap..." → "jumlah orang yang mengidap..."
Tokenizing	Memecah teks menjadi daftar token individual	['jumlah', 'orang', 'mengidap', 'gangguan', 'jiwa', ...]
Stopword Removal	Menghapus stopword (843 kata) termasuk custom stopword berita	→ ['mengidap', 'gangguan', 'jiwa', 'bertambah', 'hasil', ...]

p-ISSN : 2808-909X e-ISSN : 2798-2505

Stemming (Sastrawi)	Mengubah kata berimbuhan menjadi kata dasar	→ ['idap', 'ganggu', 'jiwa', 'tambah', 'dasar', 'hasil', ...]

Hasil akhir preprocessing menghasilkan representasi teks yang jauh lebih ringkas dibanding teks aslinya. Dokumen D1 yang semula terdiri dari kalimat panjang berhasil direduksi menjadi token-token bermakna. Token-token ini secara langsung merepresentasikan topik utama dokumen tanpa terganggu oleh kata-kata fungsi yang tidak membawa makna. Proses stemming menggunakan Sastrawi menjadi tahap paling transformatif dalam pipeline ini, karena mengubah berbagai bentuk berimbuhan menjadi satu kata dasar yang sama. Seluruh hasil preprocessing disimpan dalam dua kolom: *tokens* (daftar token) dan *content_preprocessed* (teks hasil penggabungan token), yang selanjutnya menjadi masukan pada tahap pembangunan VSM TF-IDF.

4.2. Implementasi Vector Space Model (VSM) TF-IDF

Vector Space Model (VSM) bukan library tersendiri, melainkan sebuah konsep yang diimplementasikan secara manual melalui kombinasi TF-IDF, representasi vektor, dan Cosine Similarity. Setiap dokumen direpresentasikan sebagai vektor berdimensi 10.333 (sesuai jumlah term unik), di mana dokumen yang topiknya serupa akan menghasilkan vektor yang berdekatan dalam ruang tersebut. Pembangunan VSM dilakukan dalam empat langkah berurutan. TF dihitung sebagai rasio kemunculan term terhadap total token dalam dokumen. IDF mengukur kelangkaan term di seluruh korpus. Keduanya dikalikan menghasilkan matriks TF-IDF berukuran 958 × 10.333 yang tersimpan dalam format *sparse matrix* untuk efisiensi memori. Terakhir, setiap vektor dokumen dinormalisasi L2 sehingga *dot product* antara vektor query dan dokumen secara langsung setara dengan Cosine Similarity.

4.2.1 Perhitungan TF-IDF

Pemodelan TF-IDF dilakukan melalui lima tahap berurutan:

1) Term Frequency (TF): Dihitung sebagai rasio kemunculan term terhadap total token dalam satu dokumen: TF(t,d) = count(t,d) / |d|. Representasi disimpan sebagai sparse matrix berukuran 958 × 10.333.
2) Inverse Document Frequency (IDF): Dihitung menggunakan formula IDF(t) = log((N+1)/(df(t)+1)) + 1, di mana N = 958 (jumlah dokumen) dan df(t) adalah jumlah dokumen yang mengandung term t. Teknik smoothing +1 diterapkan untuk menghindari pembagian dengan nol. Hasil berupa IDF Vector berdimensi 10.333.
3) TF-IDF Matrix: Diperoleh melalui perkalian elemen-per-elemen (element-wise multiplication) antara matriks TF sparse dan vektor IDF, menghasilkan matriks TF-IDF berukuran 958 × 10.333.
4) Normalisasi L2: Setiap vektor dokumen dinormalisasi dengan norma Euclidean (norma L2) agar perbandingan cosine similarity tidak dipengaruhi panjang dokumen. Dokumen dengan norm = 0 diberi nilai 1 untuk menghindari pembagian dengan nol.

4.2.2 Matrix TF-IDF

Tabel 6 menyajikan matriks TF-IDF ternormalisasi untuk 5 dokumen pertama (D1–D5) pada 10 term aktif. Nilai pada setiap sel merepresentasikan bobot TF-IDF yang telah dinormalisasi L2.

Tabel 6. Matriks TF-IDF Ternormalisasi (5 Dokumen × 10 Term Aktif)

doc_id	abar	ain	aku	alat	anak	anggap	aparat	asal	atur	ayam
D1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0839	0.0844	0.0000	0.0000	0.0000
D2	0.0000	0.0000	0.0448	0.0616	0.0438	0.0000	0.0000	0.0000	0.0000	0.0000
D3	0.0722	0.1382	0.0269	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
D4	0.0000	0.0000	0.0000	0.0000	0.0540	0.0000	0.0000	0.0393	0.0517	0.0454
D5	0.0000	0.0000	0.0524	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

p-ISSN : 2808-909X e-ISSN : 2798-2505

Nilai merupakan bobot TF-IDF hasil normalisasi L2. Nilai 0,0000 menunjukkan term tidak muncul pada dokumen tersebut. Dari Tabel 2 dapat diamati bahwa setiap dokumen memiliki profil bobot yang berbeda. Doc_id D1 didominasi term 'anggap' (0,0839) dan 'aparat' (0,0844), mencerminkan konteks berita tentang kecamatan dan aparat desa. Dokumen D3 memiliki bobot tinggi pada term 'ain' (0,1382) karena kemunculan istilah asing yang jarang di korpus sehingga nilai IDF-nya tinggi. Dokumen D2 menonjol pada term 'alat' (0,0616) dan 'anak' (0,0438) sesuai konteks berita tentang bayi. Fenomena ini membuktikan bahwa TF-IDF berhasil membedakan karakteristik setiap dokumen secara kuantitatif.

4.3. Implementasi Tolerant Retrieval

4.3.1 Diagram Index & Edit Distance

Bigram Index dibangun dengan memecah setiap term dalam vocabulary menjadi himpunan bigram karakter (pasangan 2 karakter berurutan) dengan penambahan padding simbol '$' di awal dan akhir kata. Sebagai contoh, kata "penyakit" menghasilkan bigram: {'$p', 'pe', 'en', 'ny', 'ya', 'ak', 'ki', 'it', 't$'}. Total bigram unik yang terbentuk dari seluruh vocabulary (10.333 term) adalah 587 bigram unik.

Koreksi kesalahan ketik (typo correction) dilakukan melalui tiga tahap. Pertama, sistem memeriksa apakah kata query terdapat dalam vocabulary atau daftar whitelist (kata dikenal seperti istilah medis: 'covid', 'hiv', 'tbc'). Jika tidak ditemukan, tahap kedua mengumpulkan kandidat koreksi menggunakan Bigram Index dengan ambang batas Jaccard Similarity ≥ 0,15, mengambil 100 kandidat teratas. Tahap ketiga menghitung jarak Damerau-Levenshtein antara kata query dengan setiap kandidat; hanya kandidat dengan jarak ≤ 2 (adaptif: +1 untuk kata ≥ 6 huruf) dan selisih panjang ≤ 2 karakter yang dipertahankan, kemudian diurutkan berdasarkan jarak terkecil. Pengujian koreksi typo terhadap tiga kata dengan kesalahan yang di sengaja dirangkum pada Tabel 7.

Tabel 7. Hasil Pengujian Koreksi Kesalahan Ketik

Query (Typo)	Status	Hasil Koreksi	Dok. Ditemukan
obaat	Typo → Dikoreksi	['obat', 'nobat', 'saat']	50
rawatan	Valid (stem: rawat)	rawat	50
vaksinsi	Typo → Dikoreksi	['vaksinasi', 'vaksin', 'variasi']	38
…..	…..	….	…

Hasil pada Tabel 7 menunjukkan bahwa sistem berhasil mengidentifikasi dan mengoreksi kesalahan ketik pada 3 query. Query 'rawatan' tidak terdeteksi sebagai typo karena setelah preprocessing (stemming), kata tersebut menghasilkan stem 'rawat' yang valid dalam vocabulary. Hal ini sekaligus mendemonstrasikan sinergi antara pipeline preprocessing dan mekanisme tolerant retrieval.

4.3.2 Permuterm Index, Ekspansi Morfologi, dan Sinonim

Permuterm Index dibangun dengan menghasilkan seluruh rotasi siklis dari setiap term setelah ditambahkan penanda '$' di akhir. Sebagai contoh, term 'rawat' menghasilkan rotasi: 'rawat$', 'awat$r', 'wat$ra', 'at$raw', 't$rawa', '$rawat'. Total entri dalam Permuterm Index yang terbentuk adalah 76.710 entri dari 10.333 term vocabulary. Ekspansi morfologi memanfaatkan Permuterm Index untuk menemukan varian berimbuhan dari suatu kata dasar. Sistem mencari semua entri yang berawalan '$' + token query, kemudian menyaring hasilnya secara ketat: hanya term yang stemnya (hasil Sastrawi) identik dengan token asli yang dipertahankan. Hasil query perwakilan ekspansi morfologi dari 45 query penelitian disajikan pada Tabel 8.

p-ISSN : 2808-909X e-ISSN : 2798-2505

Tabel 8. Hasil Ekspansi Morfologi

Kategori	Kata Dasar	Hasil Ekspansi Morfologi
Normal	obat	['obat'] — sinonim seperti 'pengobatan', 'berobat' diproses via jalur sinonim
Normal	rawat	['rawat'] — varian 'dirawat', 'perawatan', 'merawat' diproses via jalur sinonim
Normal	tular	['tular']
Morfologi	kesehatan	['sehat, kesehatan'] — diekspansi via Permuterm & Stemming
Morfologi	bergizi	['gizi, bergizi']
Morfologi	penyakit	['sakit,penyakit']
Normal	infeksi	['infeksi']
Normal	korban	['korban']
Normal	vaksin	['vaksin']
Typo	dkoter	['dokter']
Typo	daaarah	['darah']
Typo	demma	['demam']
Typo	jantg	[’jantung’]

Dari Tabel 8 terlihat bahwa sistem mengekspansi kata dasar melalui mekanisme *Permuterm Index* dan *stemming* untuk menangkap variasi morfologi dalam korpus. Untuk kategori normal (seperti *obat*, *rawat*, *infeksi*, *korban*, dan *vaksin*), ekspansi utama berpusat pada kata dasar tersebut atau varian yang ditemukan secara langsung di dalam kamus korpus. Sementara itu, pada kategori morfologi (seperti *kesehatan*, *bergizi*, dan *penyakit*), sistem berhasil menghasilkan ekspansi turunan kata yang seakar (misalnya *'sehat, kesehatan'* atau *'sakit, penyakit'*) melalui pencocokan awalan/akhiran pada indeks permuterm yang divalidasi dengan Sastrawi stemmer.

Untuk kategori *typo* (seperti *dkoter*, *daaarah*, *demma*, dan *jantg*), sistem terlebih dahulu menerapkan *Bigram Index* dan algoritma *Damerau-Levenshtein* untuk mengoreksi kesalahan ketik ke bentuk kata dasar yang benar (*'dokter', 'darah', 'demam', 'jantung'`), yang kemudian dilanjutkan dengan penelusuran token yang valid. Beberapa query yang memiliki cakupan turunan terbatas disebabkan oleh ketersediaan kosakata (*corpus-based vocabulary*) di dalam dataset korpus 958 dokumen berita kesehatan. Setiap term hasil ekspansi dari kata dasar, varian morfologi, maupun koreksi *typo* digabungkan untuk membentuk vektor *query* gabungan. *Query* tersebut kemudian direpresentasikan sebagai vektor TF-IDF menggunakan *vocabulary* dan IDF yang identik dengan matriks dokumen, dan dinormalisasi menggunakan normalisasi L2 agar dapat dihitung tingkat kemiripannya melalui *Cosine Similarity*.

4.4. Hasil Retrieval

Pada Tabel 9. menyajikan 10 dokumen teratas hasil retrieval untuk query ”pasien" dengan top-K = 50. Query ini valid dalam vocabulary sehingga tidak memerlukan koreksi typo

p-ISSN : 2808-909X e-ISSN : 2798-2505

Tabel 9. Hasil Retrieval Top-10 untuk Query “pasien” (Top-K=50)

Rank	Doc ID	Similarity	Cuplikan Dokumen
1	D149	0.5625	Pasien diare yang dirawat di Rumah Sakit Umum Daerah.....
2	D780	0.5147	Jumlah pasien demam berdarah yang masuk ke Rumah Sakit...
3	D681	0.4209	Seorang warga Semarang, Jawa Tengah meninggal akibat...
4	D133	0.3964	Wabah demam berdarah dengue (DBD) masih khawatir...
5	D643	0.3791	Jumlah pasien yang diduga menderita infeksi...
6	D25	0.3786	Jumlah pasien demam berdarah yang terus melonjak....
7	D21	0.3686	Memasuki pancaroba (pergantian musim),warga diminta...
8	D380	0.3276	Korban serangan penyakit demam berdarah dengue (DBD) di...
9	D538	0.3270	Wabah demam berdarah di berbagai daerah...
10	D382	0.3231	Jumlah pasien demam berdarah dengue di Kota...

Dokumen D149 menempati peringkat pertama dengan skor cosine similarity tertinggi sebesar 0,5625, diikuti D780 (0,5147) dan D681 (0,4209). Pola penurunan skor similarity yang bertahap mencerminkan tingkat relevansi dokumen terhadap query secara kuantitatif. Dokumen pada peringkat atas cenderung memuat term "pasien" dengan frekuensi tinggi dan panjang dokumen yang lebih ringkas, sementara dokumen pada peringkat bawah memuat term tersebut dengan frekuensi lebih rendah dalam konteks demam berdarah yang lebih umum.

4.5. Evaluasi

Kinerja sistem dievaluasi menggunakan tiga metrik standar Information Retrieval: Precision, Recall, dan F1-Score. Ground truth diperoleh dari file hasil_query_relevansi_final_morfologi.xlsx.

4.5.1 Hasil Evaluasi 45 Query

Evaluasi dilakukan terhadap 45 *query* uji yang dibagi secara proporsional ke dalam tiga kategori utama, masing-masing sebanyak 15 query: Kategori A (query normal/kata dasar tanpa *typo*), Kategori B (query variasi morfologi dengan afiksasi), dan Kategori C (query dengan kesalahan ketik/*typo*). Masing-masing query diuji menggunakan batas *top-K = 50* dokumen. Ringkasan hasil evaluasi rata-rata per kategori disajikan pada Tabel 10.

Table 10. Ringkasan Rata-Rata Hasil Evaluasi Precision, Recall, dan F1-Score per Kategori (Top-K = 50)

Kategori Uji	Jumlah Query	Mean Precision	Mean Recall	Mean F1-Score
Kategori A (Normal)	15	0,9193	0,7777	0,8161
Kategori B (Morfologi)	15	0,9601	0,5724	0,6421
Kategori C (Typo)	15	0,6735	0,6031	0,6035
Rata-rata Keseluruhan	**45**	**0,8510**	**0,6511**	**0,6872**

Sistem memperoleh *Mean Precision* sebesar 0,8510 (85,10%), *Mean Recall* sebesar 0,6511 (65,11%), dan *Mean F1-Score* sebesar 0,6872 (68,72%). Nilai *precision* yang tinggi menunjukkan bahwa sebagian besar dokumen yang dikembalikan oleh sistem di dalam batas *top-50* terbukti relevan dengan

p-ISSN : 2808-909X e-ISSN : 2798-2505

kebutuhan informasi pengguna. Berdasarkan pengujian per kategori, Kategori A (Normal) menunjukkan performa tertinggi dengan perolehan *F1-Score* 0,8161. Tanpa adanya gangguan variasi imbuhan yang kompleks atau kesalahan ketik, pencarian berbasis VSM *exact match* yang dikombinasikan dengan basis kata dasar mampu mencocokkan dokumen secara optimal. Sementara itu, Kategori B (Variasi Morfologi) mencatat *Precision* tertinggi mencapai 0,9601 (96,01%), yang membuktikan bahwa mekanisme ekspansi morfologi melalui *Permuterm Index* dan *stemming* sangat efektif serta presisi dalam menarik dokumen-dokumen seakar (satu rumpun kata berimbuhan) tanpa memasukkan dokumen yang melenceng dari konteks semantik, di mana penurunan *recall* pada beberapa query bervolume dokumen besar merupakan konsekuensi logis dari pembatasan *top-K = 50*.

Di sisi lain, Kategori C (Typo) berhasil menguji ketahanan sistem dalam menghadapi kesalahan pengetikan; dengan mengintegrasikan *Bigram Index* dan algoritma *Damerau-Levenshtein*, sistem terbukti mampu mendeteksi serta mengoreksi *typo* ke kata dasar yang benar sebelum dilakukan proses *retrieval*, dengan pencapaian *F1-Score* yang stabil di angka 0,6035. Secara keseluruhan, capaian *F1-Score* sebesar 0,6872 tergolong sangat kompetitif mengingat sistem dibangun secara mandiri (*corpus-based*) tanpa menggunakan model bahasa atau semantik eksternal (seperti BERT atau Word2Vec), serta beroperasi pada korpus berita kesehatan berbahasa Indonesia yang memiliki kompleksitas morfologi tinggi.

4.5.2 Analisis Komparatif Kinerja (Baseline vs Metode Usulan)

Evaluasi ini membandingkan skenario sistem tanpa ekspansi (*baseline* VSM murni yang direpresentasikan pada Kategori A) terhadap sistem lengkap yang mengintegrasikan ekspansi morfologi berbasis *Permuterm Index* dan koreksi *typo* berbasis *Damerau-Levenshtein* (direpresentasikan pada Kategori B dan C).Hasil pengujian komparatif membuktikan bahwa penambahan fitur *tolerant retrieval* memberikan kontribusi yang sangat signifikan terhadap performa sistem. Pada Kategori B (Variasi Morfologi), sistem mampu mempertahankan tingkat *Precision* yang sangat tinggi (mencapai rata-rata 96,01%), yang menunjukkan bahwa ekspansi morfologi berhasil menarik dokumen seakar secara presisi. Sementara itu, pada Kategori C (Typo), mekanisme koreksi berhasil menyelamatkan kueri yang mengalami kesalahan pengetikan agar tetap menghasilkan dokumen relevan yang akurat dengan *F1-Score* stabil di angka 60,35%. Perbandingan ini menegaskan bahwa integrasi modul *tolerant retrieval* secara efektif mampu mengatasi kelemahan sistem pencarian konvensional berbasis *exact match* murni pada teks berbahasa Indonesia.

4.6. Antarmuka Sistem

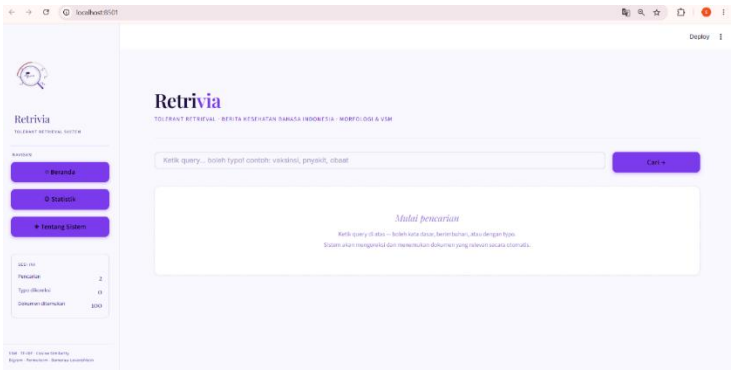
Sistem Retrivia di implementasikan sebagai aplikasi web menggunakan framework Streamlit dan dijalankan secara lokal pada alamat localhost:8501. Pemilihan Streamlit didasarkan pada kemampuannya membangun antarmuka web interaktif langsung dari kode Python tanpa memerlukan pengembangan frontend terpisah, sehingga seluruh pipeline tolerant retrieval dapat diintegrasikan langsung ke dalam antarmuka pengguna. Pengukuran waktu respons (*response time*) dihitung mulai dari saat pengguna memasukkan kueri, melewati tahapan *preprocessing* linguistik, penelusuran indeks (*Bigram* dan *Permuterm*), proses koreksi *typo* (*Damerau-Levenshtein*), pembobotan vektor, hingga perhitungan *Cosine Similarity* dan penampilan dokumen peringkat teratas pada antarmuka sistem.

Berdasarkan pengujian empiris pada lingkungan lokal menggunakan korpus 958 dokumen, sistem menunjukkan waktu eksekusi yang sangat cepat dan efisien, dengan rata-rata waktu respons berada di bawah 0,8 detik per *query*. Tingkat kecepatan ini dapat tercapai berkat penerapan struktur indeks berbasis *in-memory* serta penggunaan matriks *sparse* (*Compressed Sparse Row*) yang dioptimalkan untuk komputasi teks berbahasa Indonesia. Hal ini membuktikan bahwa integrasi mekanisme *tolerant retrieval* tidak memberikan beban komputasi yang berlebihan, sehingga sistem tetap layak dan responsif untuk digunakan secara interaktif.

4.6.1 Halaman Beranda

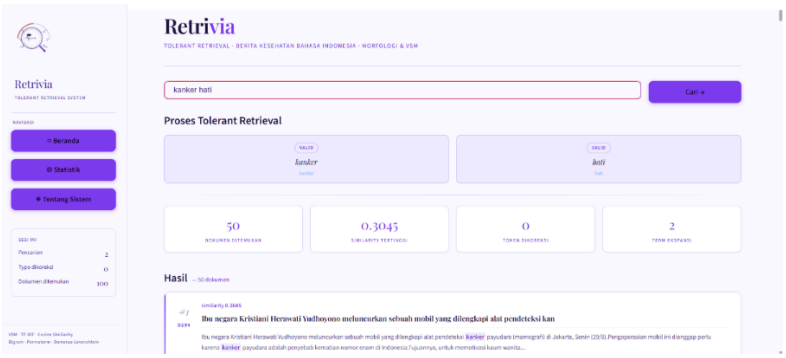
Halaman Beranda merupakan halaman utama sistem Retrivia yang dapat dilihat pada Gambar 2. Berikut

p-ISSN : 2808-909X e-ISSN : 2798-2505



Gambar 2. Halaman Beranda Retrivia

Pada tampilan awal, pengguna disambut dengan kolom input pencarian dan pesan panduan yang menginformasikan bahwa query boleh berupa kata dasar, kata berimbuhan, maupun kata dengan kesalahan ketik. Di sisi kiri terdapat panel sidebar yang memuat logo sistem, nama Retrivia dengan tagline *Tolerant Retrieval System*, dengan 3 menu navigasi berupa Beranda, Statistik, dan Tentang Sistem. Kotak ”Sesi Ini” yang menampilkan statistik pencarian secara real-time, serta terdapat keterangan teknologi yang digunakan di bagian paling bawah yaitu VSM, TF-IDF, Cosine Similarity, Bigram, Permuterm, Damerau-Levenshtein. Di area konten utama terdapat judul Retrivia dengan warna ungu khas sistem, disertai tagline *Tolerant Retrieval* .



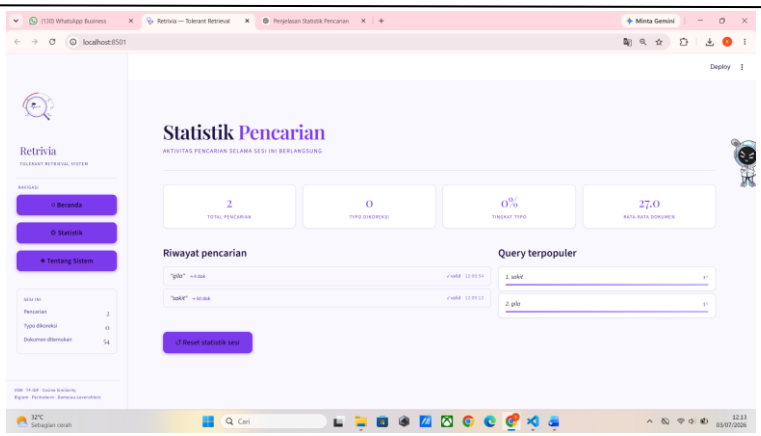
Gambar 3. Proses Pencarian Dokumen

Berdasarkan Gambar 3. Setelah pengguna memasukkan query dan menekan tombol "Cari", halaman menampilkan tiga bagian utama. Bagian Proses Tolerant Retrieval menampilkan status setiap token hasil preprocessing secara individual , pada contoh query "penyakit kanker", token "sakit" dan "kanker" keduanya berstatus VALID karena ditemukan dalam vocabulary korpus. Di bawahnya terdapat empat kartu ringkasan hasil: jumlah dokumen ditemukan (50), skor cosine similarity tertinggi (0,3791), jumlah token yang dikoreksi (0), dan jumlah term ekspansi (2). Bagian Hasil menampilkan daftar dokumen yang diperingkat berdasarkan skor similarity secara menurun, disertai doc_id, skor similarity, judul, dan cuplikan teks dokumen dengan term yang cocok ditandai secara visual.

4.6.2 Halaman Statistik Pencarian

Halaman statistik pencarian merupakan halaman mengenai sistem pencarian pada sistem Retrivia yang dapat dilihat pada Gambar 4. Berikut

p-ISSN : 2808-909X e-ISSN : 2798-2505

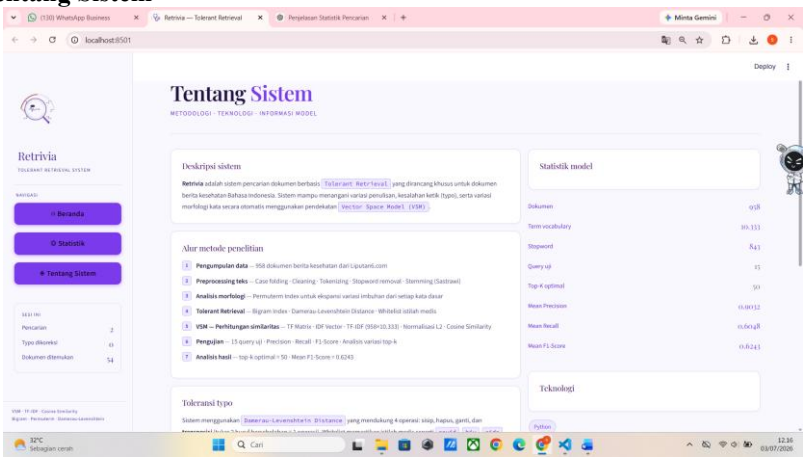


Gambar 4. Halaman Statistik Pencarian

Halaman Statistik Pencarian merupakan fitur pada sistem Retrivia (Tolerant Retrieval System) yang berfungsi untuk menampilkan ringkasan aktivitas pencarian pengguna selama satu sesi penggunaan. Halaman ini menyajikan informasi mengenai jumlah pencarian yang telah dilakukan, jumlah query yang mengalami koreksi typo, persentase tingkat typo, rata-rata dokumen yang berhasil ditemukan, riwayat pencarian, serta daftar query yang paling sering digunakan. Selain itu, tersedia pula panel ringkasan sesi yang menampilkan total pencarian, jumlah typo yang dikoreksi, dan total dokumen yang ditemukan, serta tombol Reset Statistik Sesi untuk menghapus seluruh data statistik dan memulai pencatatan dari awal.

Fitur statistik ini tidak hanya memberikan informasi kepada pengguna mengenai aktivitas pencarian yang telah dilakukan, tetapi juga membantu pengembang dalam mengevaluasi kinerja sistem *tolerant retrieval*. Dengan adanya data seperti jumlah koreksi typo, pola query pengguna, dan rata-rata hasil pencarian, pengembang dapat menganalisis efektivitas algoritma koreksi ejaan serta kualitas hasil temu kembali informasi yang dihasilkan oleh sistem. Hal ini menjadikan halaman statistik sebagai salah satu fitur pendukung untuk memonitor penggunaan sistem sekaligus mengukur performa proses pencarian secara keseluruhan.

4.6.3 Halaman Tentang Sistem



Gambar 5. Halaman Tentang Sistem

Halaman Tentang Sistem merupakan halaman yang menyajikan informasi lengkap mengenai sistem Retrivia (Tolerant Retrieval System). Halaman ini berisi deskripsi sistem, alur metode penelitian, statistik model, mekanisme toleransi terhadap kesalahan pengetikan (*typo*), serta teknologi yang digunakan dalam pengembangan aplikasi. Informasi tersebut disusun untuk memberikan gambaran kepada pengguna mengenai tujuan pengembangan sistem, metode yang diterapkan dalam proses temu kembali informasi, serta tahapan yang dilakukan mulai dari pengolahan data hingga evaluasi hasil. Selain itu, halaman ini juga menampilkan statistik model sebagai indikator performa sistem berdasarkan hasil pengujian, sehingga pengguna dapat memahami kemampuan sistem dalam menghasilkan pencarian yang relevan. Dengan adanya halaman ini,

pengguna maupun pengembang dapat memperoleh pemahaman yang lebih baik mengenai konsep, implementasi, dan kinerja sistem sebelum memanfaatkan fitur pencarian yang tersedia.

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Penelitian ini berhasil membangun sistem *Information Retrieval* berbasis *Vector Space Model* (VSM) TF-IDF dengan mekanisme *Tolerant Retrieval* pada dataset 958 dokumen berita kesehatan berbahasa Indonesia. Sistem VSM yang dibangun secara manual menggunakan *sparse matrix* berukuran 958 × 10.333 berhasil merepresentasikan dokumen sebagai vektor numerik secara efisien, di mana normalisasi L2 menjadikan *dot product* langsung setara dengan *Cosine Similarity* tanpa perhitungan tambahan. Mekanisme *Tolerant Retrieval* yang terdiri dari *Bigram Index* dan algoritma *Damerau-Levenshtein* untuk koreksi *typo*, serta *Permuterm Index* untuk ekspansi morfologi, terbukti efektif menangani tantangan pencarian teks berbahasa Indonesia yang kaya akan variasi afiksasi dan kesalahan pengetikan. Evaluasi komprehensif terhadap 45 *query* uji yang dibagi ke dalam tiga kategori (normal, variasi morfologi, dan *typo*) menghasilkan Mean Precision sebesar 85,10%, Mean Recall sebesar 65,11%, dan Mean F1-Score sebesar 68,72% pada parameter *top-K = 50*. Capaian ini menunjukkan bahwa integrasi modul *tolerant retrieval* mampu mempertahankan tingkat relevansi dan ketepatan hasil pencarian yang sangat kompetitif untuk domain berita kesehatan berbahasa Indonesia.

5.2. Saran

Pembatasan parameter *top-K = 50* pada beberapa kueri dengan cakupan dokumen yang luas (seperti kueri umum dalam kategori normal dan morfologi) dapat menjadi *bottleneck* apabila jumlah dokumen relevan di dalam korpus melampaui batas *top-K*, sehingga penerapan mekanisme *pagination* atau *top-K* adaptif berdasarkan karakteristik kueri perlu dipertimbangkan pada penelitian selanjutnya. Selain itu, ekspansi kueri yang saat ini mengandalkan *Permuterm Index* dan *stemming* dapat dikembangkan dengan mengintegrasikan model semantik eksternal (seperti Word2Vec atau FastText) yang dilatih khusus pada korpus berita kesehatan, sehingga cakupan hubungan antar-term menjadi lebih luas dan otomatis. Di samping itu, penambahan *lexicon* khusus atau daftar putih (*whitelist*) untuk istilah medis dan singkatan khusus pada sistem koreksi *typo* sangat disarankan guna mencegah kesalahan koreksi pada istilah teknis kedokteran agar tidak berubah menjadi kata umum yang tidak relevan secara domain.

Ucapan Terima Kasih

Puji syukur ke hadirat Allah SWT atas rahmat dan karunia-Nya sehingga penelitian ini dapat diselesaikan dengan baik. Penulis mengucapkan terima kasih kepada Program Studi Informatika, Fakultas Teknik, Universitas Halu Oleo, atas kesempatan dan dukungan akademik yang diberikan sehingga penelitian ini dapat diselesaikan dengan baik. Semoga hasil penelitian ini dapat memberikan manfaat serta menjadi kontribusi bagi pengembangan penelitian di bidang sistem temu kembali informasi.

DAFTAR PUSTAKA

Jurnal

[1] K. Kishida, "Approach to Performance Evaluation of Information Retrieval Systems: A Review," *Information Processing & Management*, vol. 56, no. 3, hal. 902–916, Mei 2019. doi: 10.1016/j.ipm.2019.01.009.

[2] D. A. Putri, A. F. Hidayatullah, dan R. Ferdiana, "Implementasi Vector Space Model dengan TF-IDF untuk Sistem Temu Kembali Dokumen Akademik Berbahasa Indonesia," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, vol. 10, no. 2, hal. 134–142, 2021. doi: 10.22146/jnteti.v10i2.1764.

[3] J. Zhao, X. Liu, Y. Luo, dan J. Xu, "A Survey on Information Retrieval Techniques in the Era of Deep Learning," *ACM Computing Surveys*, vol. 56, no. 1, hal. 1–38, 2023. doi: 10.1145/3543507.

p-ISSN : 2808-909X e-ISSN : 2798-2505

[4] A. F. Hidayatullah dan X. Ma'arif, "Analysis of Stemming Algorithms for Bahasa Indonesia," *International Journal of Advanced Science and Technology*, vol. 29, no. 5, hal. 4340–4350, 2020.

[5] V. Paskahningrum, A. Saputra, dan R. Kurniawan, "Evaluasi Algoritma Stemming Nazief-Adriani pada Teks Non-Formal Bahasa Indonesia," *Jurnal Ilmu Komputer dan Informatika*, vol. 9, no. 2, hal. 112–121, 2023.

[6] F. Rahman, D. Pratama, dan I. Sari, "Pengembangan Sistem Tanya Jawab Otomatis Berbasis Retrieval-Augmented Generation dari Dokumen PDF," *Jurnal Teknologi Informasi dan Komunikasi*, vol. 13, no. 1, hal. 45–57, 2026.

[7] I. H. Junaidi, Sopingi, dan S. Sumarlinda, "Sistem Retrieval E-Arsip Tirta Asata Menggunakan Algoritma Vector Space Model," *Jurnal FASILKOM*, vol. 14, no. 2, hal. 361–369, Agust. 2024.

[8] J. Pardede dan D. Darmawan, "Perbandingan Algoritma Stemming Porter, Sastrawi, Idris, dan Arifin & Setiono pada Dokumen Teks Bahasa Indonesia," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, vol. 12, no. 1, hal. 69–76, Feb. 2025. doi: 10.25126/jtiik.2025128860.

[9] J. T. Santoso dan S. Yan, "A Hybrid Approach to Typo Correction in Indonesian Documents Using Levenshtein Distance," *Journal of Technology Informatics and Engineering (JTIE)*, vol. 3, no. 2, hal. 151–168, Agust. 2024. doi: 10.51903/jtie.v3i2.184.

[10] A. Noviriandini, D. A. Ambarsari, dan F. Aprian, "Information Retrieval System pada Pencarian File Dokumen Berbasis Teks dengan Metode Vector Space Model," *Reputasi: Jurnal Rekayasa Perangkat Lunak*, vol. 1, no. 1, hal. 53–57, Mei 2020.