

Sistem Chatbot Informasi PPDB Berbasis TF-IDF dan Cosine Similarity Menggunakan Adaptive Threshold pada Web Sekolah

Raditya Mayesha Beldyq^{1*}, Rika Apriani²

^{1,2}Program Studi Teknik Informatika, Universitas Bina Insani, Indonesia

^{1*}radityabeldyq@gmail.com, ²rikaapriani@binainsani.ac.id

Abstrak: Layanan informasi Penerimaan Peserta Didik Baru (PPDB) di SMP Sejahtera 2 Cileungsi masih dilakukan secara manual sehingga menyebabkan keterlambatan respons dan penumpukan pertanyaan dari calon pendaftar. Penelitian ini bertujuan merancang dan mengembangkan sistem chatbot berbasis web yang mengimplementasikan pipeline Natural Language Processing (NLP) secara penuh, mencakup Text Preprocessing, pembobotan Term Frequency-Inverse Document Frequency (TF-IDF), dan pencocokan intent menggunakan Cosine Similarity. Tahapan Text Preprocessing meliputi case folding, cleansing, tokenization, normalisasi sinonim, stopword removal, stemming, dan ekstraksi bigram. Knowledge base disusun dari 32 intent, 655 pola pertanyaan, dan 46 jawaban yang diperoleh melalui wawancara dan observasi di SMP Sejahtera 2. Sistem dikembangkan menggunakan metode Rapid Application Development (RAD) dengan arsitektur serverless berbasis Firebase Firestore dan Vanilla JavaScript sehingga seluruh komputasi TF-IDF dan Cosine Similarity dijalankan pada sisi klien tanpa backend khusus. Adaptive threshold diterapkan secara dinamis berdasarkan keberadaan kata kunci domain PPDB untuk meningkatkan relevansi jawaban. Evaluasi akurasi model dilakukan terhadap 35 sampel pertanyaan uji yang dibagi dalam tiga kategori: in-pattern, out-of-pattern, dan out-of-domain, menghasilkan akurasi keseluruhan sebesar 82,9%. Pengujian fungsional menggunakan Black Box Testing terhadap 8 skenario seluruhnya dinyatakan valid. Evaluasi latensi menunjukkan rata-rata waktu respons chatbot sebesar 1,05 detik. User Acceptance Test (UAT) terhadap 25 responden memperoleh rata-rata kepuasan 87,4% dengan kategori Sangat Baik. Hasil penelitian membuktikan bahwa implementasi pipeline NLP berbasis TF-IDF dan Cosine Similarity mampu meningkatkan kualitas layanan informasi PPDB secara otomatis, cepat, dan dapat diakses kapan saja.

Kata Kunci: Chatbot; Cosine Similarity; Natural Language Processing; Term Frequency-Inverse Document Frequency; Text Preprocessing;

Abstract: The student admission (PPDB) information service at SMP Sejahtera 2 Cileungsi is still conducted manually, resulting in delayed responses and accumulation of inquiries from prospective students and parents. This study aims to design and develop a web-based chatbot system implementing a complete Natural Language Processing (NLP) pipeline, encompassing Text Preprocessing, Term Frequency-Inverse Document Frequency (TF-IDF)

weighting, and intent matching using Cosine Similarity. The Text Preprocessing pipeline includes case folding, cleansing, tokenization, synonym normalization, stopword removal, stemming, and bigram extraction. The knowledge base consists of 32 intents, 655 question patterns, and 46 responses collected through interviews and observations at SMP Sejahtera 2. The system was developed using the Rapid Application Development (RAD) method with a serverless architecture based on Firebase Firestore and Vanilla JavaScript, enabling all TF-IDF and Cosine Similarity computations to run client-side without a dedicated backend. An adaptive threshold mechanism was applied dynamically based on the presence of PPDB domain keywords to enhance answer relevance. Model accuracy was evaluated using 35 test question samples across three categories: in-pattern, out-of-pattern, and out-of-domain, achieving an overall accuracy of 82.9%. Functional testing using the Black Box Testing method across 8 scenarios was entirely declared valid. The latency evaluation demonstrated an average chatbot response time of 1.05 seconds. The User Acceptance Test (UAT) involving 25 respondents yielded a mean satisfaction rate of 87.4%, categorized as Very Good. The results demonstrate that the NLP pipeline implementation based on TF-IDF and Cosine Similarity significantly improves PPDB information services in an automated, fast, and anytime-accessible manner.

Keywords: Chatbot; Cosine Similarity; Natural Language Processing; Term Frequency-Inverse Document Frequency; Text Preprocessing;

1. PENDAHULUAN

Perkembangan Machine Learning sebagai bagian dari kecerdasan buatan (artificial intelligence) telah memberikan dampak signifikan dalam berbagai sektor, termasuk pendidikan. Teknologi ini memungkinkan sistem untuk mempelajari pola dari data dan memberikan respon secara otomatis tanpa intervensi manusia secara langsung. Salah satu cabang penting dari Machine Learning adalah Natural Language Processing (NLP), yang memfokuskan pada kemampuan sistem dalam memahami dan menghasilkan bahasa manusia [1]. NLP memungkinkan komputer berinteraksi dengan pengguna melalui bahasa alami, baik teks maupun suara. Teknologi ini diterapkan secara luas mulai dari layanan publik, kesehatan, hingga asisten virtual dan pencarian informasi [2], [3], [4].

Proses Penerimaan Peserta Didik Baru (PPDB) merupakan aspek krusial dalam institusi pendidikan yang diatur berdasarkan kebijakan kementerian [5]. Sistem PPDB berbasis daring dirancang untuk mempermudah pendaftaran [6]. Namun, kompleksitas informasi yang disediakan seringkali menimbulkan kebingungan bagi calon peserta didik, sehingga meningkatkan kebutuhan akan layanan informasi yang responsif. Penelitian oleh Mursidah et al. [7] menunjukkan bahwa implementasi chatbot layanan informasi pendaftaran mampu mengurangi beban kerja staf administrasi dan mempercepat distribusi informasi. Oleh karena itu, diperlukan pengembangan solusi berbasis teknologi yang mampu mengotomatisasi layanan informasi PPDB agar berjalan lebih optimal.

Secara teknis, tahap awal dalam pipeline NLP adalah Text Preprocessing yang bertujuan membersihkan dan menyiapkan data teks agar dapat diproses secara optimal. Kombinasi teknik preprocessing terbukti meningkatkan kinerja model secara signifikan [8], [9]. Setelah teks diproses, sistem memerlukan representasi matematis menggunakan Term Frequency-Inverse Document Frequency (TF-IDF). TF-IDF bekerja dengan menggabungkan Term Frequency (TF) dan Inverse Document Frequency (IDF) untuk mengurangi bobot kata umum dan menyoroti kata relevan [10], [11]. Bobot ini kemudian diukur tingkat kemiripannya menggunakan Cosine Similarity [12], [13].

Beberapa penelitian terdahulu telah mengkaji metode serupa pada sistem chatbot. Prasetya dan Priyatno [14] mengembangkan chatbot PPDB menggunakan TF-IDF dan Dice Similarity, namun masih berbasis pencocokan kata dasar. Saputra dan Sucipto [15] mengimplementasikan chatbot menggunakan TF-IDF dan Cosine Similarity namun terbatas pada dataset kecil tanpa filter di luar konteks. Aprianto et al. [16] mengoptimalkan chatbot dengan kombinasi TF-IDF dan Word2Vec tetapi memerlukan komputasi tinggi. Holis et al. [17] menggunakan Sentence-BERT dengan pemahaman semantik lebih baik. Mohammed Salah Ibrahim et al. [18] mengembangkan chatbot FAQ dengan query expansion namun belum menerapkan adaptive threshold.

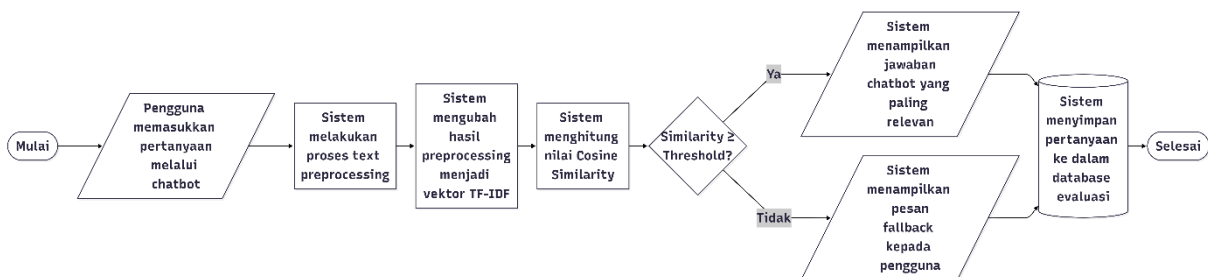
Berdasarkan kajian tersebut, terdapat dua permasalahan utama yang menjadi fokus penelitian ini. Pertama, bagaimana efektivitas pipeline NLP yang mengintegrasikan Text Preprocessing 7-tahap (case folding, cleansing, tokenization, normalisasi sinonim, stopword removal, stemming, dan bigram extraction) dengan pembobotan TF-IDF dan pencocokan intent menggunakan Cosine Similarity dalam mengenali maksud pertanyaan pengguna pada domain PPDB. Kedua, bagaimana mekanisme adaptive threshold berbasis domain keyword filtering dapat meningkatkan relevansi jawaban sekaligus menyaring pertanyaan di luar konteks PPDB. Penelitian ini bertujuan untuk: (1) merancang dan mengimplementasikan pipeline NLP berbasis TF-IDF dan Cosine Similarity yang efektif dalam mengenali intent pertanyaan seputar PPDB secara otomatis; dan (2) mengevaluasi efektivitas mekanisme adaptive threshold berbasis domain keyword filtering dalam menyaring pertanyaan di luar konteks domain PPDB pada sistem chatbot berbasis web serverless.

2. METODE PENELITIAN

Penelitian ini berfokus pada implementasi dan evaluasi pipeline NLP untuk sistem chatbot informasi PPDB pada SMP Sejahtera 2 Cileungsi. Data pertanyaan dan jawaban resmi dikumpulkan melalui wawancara dengan panitia PPDB dan observasi langsung terhadap pertanyaan yang sering diajukan calon pendaftar. Sistem dikembangkan menggunakan metode Rapid Application Development (RAD) [19], [20] dengan Vanilla JavaScript dan arsitektur serverless berbasis Firebase Firestore, sehingga seluruh komputasi NLP dijalankan pada sisi klien (client-side).

Berbeda dengan pendekatan deep learning seperti LSTM yang memerlukan data pelatihan besar dan proses training iteratif dengan teknik seperti Early Stopping, penelitian ini mengadopsi pendekatan retrieval based berbasis TF-IDF dan Cosine Similarity yang tidak memerlukan tahap pelatihan model, lebih ringan secara komputasi, serta dapat diimplementasikan dalam arsitektur serverless tanpa backend khusus.

Fokus utama penelitian ini adalah implementasi pipeline NLP yang terdiri dari tiga tahapan inti: (1) Text Preprocessing untuk membersihkan dan menyiapkan data teks; (2) pembobotan TF-IDF untuk representasi vektor; dan (3) pencocokan intent menggunakan Cosine Similarity. Alur proses pipeline NLP ditunjukkan pada Gambar 1.



Gambar 1. Diagram Alir (Flowchart) Alur Proses Sistem Chatbot PPDB

Raditya Mayesha Beldyq: *Penulis Korespondensi



Copyright © 2026, Raditya Mayesha Beldyq, Rika Apriani.

Gambar 1 mengilustrasikan alur proses sistem chatbot PPDB secara menyeluruh. Pipeline dimulai dari input pertanyaan pengguna yang melewati 7 tahapan text preprocessing (case folding, cleansing, tokenization, normalisasi sinonim, stopword removal, stemming, dan bigram extraction) untuk menghasilkan token bersih. Selanjutnya, sistem menghitung vektor TF-IDF dari query dan membandingkannya dengan seluruh 655 vektor dokumen dalam knowledge base menggunakan Cosine Similarity. Mekanisme adaptive threshold kemudian menentukan relevansi jawaban: jika pertanyaan mengandung kata kunci domain PPDB, threshold diturunkan ke 30% agar sistem lebih responsif. Sebaliknya, jika pertanyaan tidak mengandung kata kunci domain, threshold dinaikkan ke 65% untuk menyaring pertanyaan di luar konteks. Jika skor kemiripan tertinggi melampaui threshold yang berlaku, jawaban terbaik ditampilkan. Jika tidak, sistem menampilkan pesan fallback.

A. Penyusunan Knowledge Base

Knowledge base chatbot disusun berdasarkan hasil wawancara dengan panitia PPDB SMP Sejahtera 2 dan observasi terhadap pertanyaan yang sering diajukan calon peserta didik. Data disimpan dalam format JSON (JavaScript Object Notation) dengan tiga atribut: tag (label intent), patterns (variasi pola pertanyaan), dan responses (jawaban resmi). Knowledge base terdiri dari 32 intent, 655 pola pertanyaan, dan 46 jawaban yang mencakup topik biaya, jadwal, persyaratan, fasilitas, dan alur pendaftaran. Basis pengetahuan disimpan di Firebase Firestore sehingga dapat diperbarui admin tanpa mengubah kode program.

Tabel 1. Contoh Struktur Intent pada Knowledge Base

Atribut	Nilai Contoh	Keterangan
tag	tanya_biaya	Label kategori/intent pertanyaan
patterns	berapa biaya masuknya?	Variasi kalimat pertanyaan pengguna
patterns	total biaya ppdb	Variasi kalimat lainnya
patterns	berapa uang pangkalnya	Variasi dengan kata sinonim
responses	Total biaya PPDB Rp 2.700.000.	Jawaban resmi dari sekolah

B. Pipeline Text Preprocessing

Text Preprocessing diterapkan pada seluruh pola pertanyaan dalam knowledge base saat inialisasi sistem, serta pada setiap pertanyaan pengguna secara real-time. Pipeline terdiri dari tujuh tahapan berurutan. Tabel 2 menunjukkan hasil transformasi teks dari contoh kalimat input 'Info uang pangkal untuk pendaftaran SMP'.

Tabel 2. Pipeline Text Preprocessing dan Hasil Transformasi

No.	Tahapan	Proses	Hasil Transformasi
1	Case Folding	Ubah semua karakter ke huruf kecil	"info uang pangkal untuk pendaftaran smp"
2	Cleansing	Hapus tanda baca, angka, karakter khusus (regex)	"info uang pangkal untuk pendaftaran smp"
3	Tokenization	Pecah kalimat menjadi token kata	["info","uang","pangkal","untuk","pendaftaran","smp"]
4	Normalisasi Sinonim	Ganti sinonim ke kata standar (alias mapping)	["info","uang","biaya","untuk","pendaftaran","smp"]
5	Stopword Removal	Hapus kata umum tidak bermakna	["uang","biaya","pendaftaran","smp"]

Raditya Mayesha Beldyq: *Penulis Korespondensi



6	Stemming	Kembalikan ke bentuk dasar (afiks bahasa Indonesia)	["uang","biaya","daftar","smp"]
7	Bigram Extraction	Bentuk kombinasi 2 kata berurutan	["uang_biaya","biaya_daftar","daftar_smp"]
8	Token Final	Gabungan unigram + bigram	["uang","biaya","daftar","smp","uang_biaya","biaya_daftar","daftar_smp"]

Normalisasi sinonim menggunakan kamus alias mapping yang mendefinisikan kata-kata bersinonim ke satu bentuk standar. Contoh: kata 'pangkal', 'spp', 'iuran', dan 'ongkos' dinormalisasi menjadi 'biaya'. Hal ini mengatasi variasi kosakata pengguna yang beragam dalam menyampaikan pertanyaan yang sama.

Ekstraksi bigram dilakukan untuk mempertahankan konteks frasa tertentu. Selain menggunakan token tunggal (unigram), sistem membentuk kombinasi dua kata berurutan (bigram) sehingga hubungan antar kata dapat direpresentasikan dengan lebih baik. Teknik ini meningkatkan akurasi pencocokan kalimat dibandingkan hanya menggunakan token tunggal.

C. Pembobotan TF-IDF

Setelah preprocessing selesai, sistem menghitung bobot TF-IDF untuk setiap term dalam corpus. Terdapat tiga tahap perhitungan. Tahap pertama menghitung Term Frequency menggunakan (1), dilanjutkan perhitungan Inverse Document Frequency menggunakan (2), lalu dikombinasikan menjadi bobot TF-IDF pada (3).

Term Frequency (TF):

$$TF(i,j) = \frac{n(i,j)}{\sum_k n(k,j)} \quad (1)$$

Keterangan: $TF(i,j)$ = nilai term frequency term i pada dokumen j ; $n(i,j)$ = jumlah kemunculan term i pada dokumen j ; $\sum_k n(k,j)$ = jumlah seluruh term pada dokumen j .

Inverse Document Frequency (IDF):

$$IDF(i) = \log \left(\frac{N}{1 + df(i)} \right) \quad (2)$$

Keterangan: N = jumlah seluruh dokumen dalam corpus; $df(i)$ = jumlah dokumen yang mengandung term i ; Penambahan 1 pada penyebut untuk menghindari pembagian nol (smoothing).

Bobot TF-IDF:

$$TF-IDF(i,j) = TF(i,j) \times IDF(i) \quad (3)$$

Keterangan: $TF-IDF(i,j)$ = bobot akhir term i pada dokumen j ; nilai tinggi menandakan term sangat representatif untuk dokumen tersebut.

Proses implementasi TF-IDF: (1) seluruh pola pertanyaan dalam knowledge base diproses melalui text preprocessing membentuk corpus; (2) nilai IDF dihitung untuk seluruh kata unik dalam corpus; (3) setiap dokumen direpresentasikan sebagai vektor TF-IDF; (4) saat pengguna memasukkan pertanyaan, sistem memproses teks pertanyaan menggunakan preprocessing yang sama kemudian mengubahnya menjadi vektor TF-IDF menggunakan nilai IDF yang telah dihitung.

D. Pencocokan Intent dengan Cosine Similarity

Cosine Similarity digunakan untuk mengukur tingkat kemiripan antara vektor pertanyaan pengguna dengan seluruh vektor dokumen dalam corpus knowledge base. Kemiripan ini dihitung secara matematis menggunakan (4).

$$\text{Similarity}(A, B) = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n A_i^2} \times \sqrt{\sum_{i=1}^n B_i^2}} \quad (4)$$

Keterangan: A_i = bobot TF-IDF term ke- i pada vektor A (pertanyaan pengguna); B_i = bobot TF-IDF term ke- i pada vektor B (dokumen knowledge base); Nilai Similarity berkisar antara 0 hingga 1.

E. Adaptive Threshold

Sistem menerapkan mekanisme Adaptive Threshold menggunakan Domain Keyword Filtering untuk meningkatkan relevansi jawaban dan mencegah chatbot merespons pertanyaan di luar konteks PPDB. Mekanisme ini terdiri dari tiga komponen utama: (1) Threshold Rendah (30%) yang diterapkan jika pertanyaan mengandung kata kunci domain PPDB seperti 'sekolah', 'spp', 'pendaftaran', 'rapor', 'zonasi', guna memastikan pertanyaan relevan tetap terjawab meskipun formulasinya berbeda; (2) Threshold Tinggi (65%) yang diterapkan jika tidak ditemukan kata kunci domain PPDB, sehingga membantu menyaring pertanyaan spam atau di luar topik; dan (3) Continuous Evaluation, di mana setiap pertanyaan pengguna disimpan otomatis ke basis data log untuk ditinjau admin guna memperbarui knowledge base secara bertahap.

F. Metode Pengujian

Pengujian dilakukan dalam tiga tahap: (1) Evaluasi Akurasi Model menggunakan 35 sampel pertanyaan uji yang dibagi dalam tiga kategori—*in-pattern*, *out-of-pattern*, dan *out-of-domain*—dengan metrik Accuracy yang dievaluasi menggunakan (5); (2) Pengujian Kecepatan Respons (Latency Testing) menggunakan `API performance.now()` pada browser untuk mengukur durasi komputasi NLP di sisi klien; dan (3) Pengujian Fungsional serta User Acceptance Test (UAT).

$$\text{Accuracy} = \frac{\text{Jumlah Prediksi Benar}}{\text{Total Sampel Pengujian}} \times 100\% \quad (5)$$

Selain itu, User Acceptance Test (UAT) dilakukan terhadap 25 responden (24 calon peserta didik/orang tua dan 1 panitia PPDB) menggunakan Skala Likert 1-5 dengan 10 pernyataan yang mencakup aspek website dan chatbot. Persentase kepuasan akhir dievaluasi menggunakan (6).

$$P = \frac{T}{Y} \times 100\% \quad (6)$$

Keterangan: P = persentase kepuasan; T = total skor empiris; Y = total skor maksimal (jumlah responden $\times$ skor tertinggi Skala Likert).

Rincian pembagian data knowledge base dan sampel pengujian disajikan pada Tabel 2. Sistem berbasis retrieval tidak memerlukan pemisahan validation set formal seperti model deep learning, sehingga seluruh 655 pola pertanyaan digunakan sebagai corpus knowledge base. Sampel uji sebanyak 35 dipilih menggunakan dipilih secara acak berdasarkan kategori pengujian untuk memastikan keterwakilan setiap skenario uji.

Tabel 3. Pembagian Data Knowledge Base dan Sampel Pengujian

Jenis Data	Jumlah	Proporsi	Keterangan
Data Knowledge Base (Training)	655 pola	94,9%	Pola pertanyaan yang digunakan sebagai corpus retrieval
Sampel In-pattern	15 sampel	2,2%	Dipilih acak dari pola yang ada di knowledge base
Sampel Out-of-pattern	15 sampel	2,2%	Variasi pertanyaan baru di luar pola training
Sampel Out-of-domain	5 sampel	0,7%	Pertanyaan di luar konteks PPDB
Total Sampel Uji	35 sampel	5,1%	Dipilih secara acak berdasarkan kategori pengujian

3. HASIL DAN PEMBAHASAN

A. Hasil Perhitungan TF-IDF

Untuk mendemonstrasikan perhitungan TF-IDF secara konkret, digunakan contoh query pengguna 'Info uang pangkal untuk pendaftaran SMP' yang dibandingkan dengan pola pertanyaan pada knowledge base 'Berapa biaya pendaftaran SMP?'. Setelah melalui pipeline preprocessing (Tabel 2), token akhir adalah:

Vektor A (Query): [uang, biaya, daftar, smp, uang_biaya, biaya_daftar, daftar_smp]

Vektor B (Knowledge Base): [biaya, daftar, smp, biaya_daftar, daftar_smp]

Total dokumen dalam knowledge base N = 655 pola pertanyaan. Tabel 3 menampilkan hasil perhitungan TF, IDF, dan bobot TF-IDF untuk setiap term dari kedua vektor tersebut.

Tabel 4. Perhitungan Bobot TF-IDF Vektor A (Query) dan Vektor B (Knowledge Base)

Term	TF(A)	TF(B)	df	IDF	TF-IDF(A)	TF-IDF(B)	Ai×Bi
uang	0,143	0	20	1,494	0,214	0	0
biaya	0,143	0,200	50	1,109	0,159	0,222	0,035
daftar	0,143	0,200	150	0,637	0,091	0,127	0,012
smp	0,143	0,200	300	0,338	0,048	0,068	0,003
uang_biaya	0,143	0	5	2,038	0,291	0	0
biaya_daftar	0,143	0,200	10	1,775	0,254	0,355	0,090
daftar_smp	0,143	0,200	40	1,204	0,172	0,241	0,041
Σ (Total)					0,510 ²	0,504 ²	0,181

Berdasarkan Persamaan (1), (2), dan (3), contoh perhitungan TF untuk kata 'biaya' pada Vektor A dan Vektor B, serta perhitungan nilai IDF dan TF-IDF-nya secara berturut-turut adalah sebagai berikut:

$$TF(biaya, A) = \frac{1}{7} = 0,143$$

$$TF(biaya, B) = \frac{1}{5} = 0,200$$

$$IDF(biaya) = \log\left(\frac{655}{1+50}\right) = \log(12,843) = 1,109$$

$$TF - IDF(biaya, A) = 0,143 \times 1,109 = 0,159$$

$$TF - IDF(biaya, B) = 0,200 \times 1,109 = 0,222$$

Kata 'uang_biaya' memiliki nilai TF-IDF tertinggi (0,291) karena merupakan bigram yang sangat spesifik dengan df rendah (hanya muncul di 5 dokumen dari 655). Hal ini membuktikan bigram extraction efektif memberikan bobot lebih tinggi pada frasa-frasa yang membedakan intent satu dengan lainnya.

B. Hasil Perhitungan Cosine Similarity

Berdasarkan Persamaan (4) dan menggunakan nilai TF-IDF dari Tabel 3, Cosine Similarity dihitung sebagai berikut:

$$\begin{aligned} \sum (A_i \times B_i) &= 0 + 0,035 + 0,012 + 0,003 + 0 + 0,090 + 0,041 = 0,181 \\ |A| &= \sqrt{0,214^2 + 0,159^2 + 0,091^2 + 0,048^2 + 0,291^2 + 0,254^2 + 0,172^2} = \sqrt{0,260} = 0,510 \\ |B| &= \sqrt{0 + 0,222^2 + 0,127^2 + 0,068^2 + 0 + 0,355^2 + 0,241^2} = \sqrt{0,254} = 0,504 \\ \text{Similarity}(A,B) &= \frac{0,181}{0,510 \times 0,504} = \frac{0,181}{0,257} = 0,704 \end{aligned}$$

Nilai Cosine Similarity sebesar 0,704 menunjukkan tingkat kemiripan yang cukup tinggi antara query pengguna dan dokumen referensi. Sistem mendeteksi keberadaan kata kunci domain PPDB ('smp' dan 'pendaftaran'), sehingga adaptive threshold yang berlaku adalah 30% (0,300). Karena $0,704 > 0,300$, intent tanya_biaya dipilih sebagai jawaban yang valid. Proses ini dilakukan secara berulang terhadap seluruh 655 dokumen, dan intent dengan nilai Cosine Similarity tertinggi dipilih sebagai respons chatbot.

C. Evaluasi Akurasi Model

Evaluasi akurasi dilakukan terhadap 35 sampel pertanyaan uji yang disiapkan secara terpisah dari 655 pola pertanyaan training. Sampel uji dibagi dalam tiga kategori: (1) in-pattern: pertanyaan yang formulasinya tersimpan di knowledge base; (2) out-of-pattern: pertanyaan variasi baru yang tidak identik dengan pola tersimpan; (3) out-of-domain: pertanyaan di luar konteks PPDB yang harus difilter oleh adaptive threshold.

Tabel 5. Rekapitulasi Hasil Evaluasi Akurasi Model

Kategori Pengujian	Jml Sampel	Prediksi Benar	Prediksi Salah	Akurasi (%)
In-Pattern (sesuai pola teks)	15	15	0	100,0%
Out-of-Pattern (variasi baru)	15	10	5	66,7%
Out-of-Domain (luar konteks PPDB)	5	4	1	80,0%
Total Keseluruhan	35	29	6	82,9%

Akurasi total: $29/35 \times 100\% = 82,9\%$. Sistem sangat efektif pada kategori in-pattern (100%) karena formulasi pertanyaan sesuai dengan yang tersimpan. Pada out-of-pattern (66,7%), penurunan disebabkan variasi bahasa yang tidak sepenuhnya tercakup oleh kamus sinonim dan pola yang tersedia. Pada out-of-domain (80%), 1 dari 5 pertanyaan masih lolos threshold karena secara kebetulan mengandung kata yang terdaftar sebagai domain keyword. Peningkatan akurasi out-of-pattern dapat dilakukan melalui pembaruan kamus sinonim dan penambahan pola pertanyaan secara bertahap menggunakan fitur continuous evaluation.

D. Pengujian Kecepatan Respons

Pengujian kecepatan respons dilakukan untuk mengukur seberapa cepat sistem chatbot memberikan jawaban setelah pengguna mengirimkan pertanyaan. Tabel 5 menyajikan hasil pengujian waktu respons terhadap 10 sampel pertanyaan uji.

Tabel 6. Hasil Pengujian Kecepatan Respons Chatbot

No	Teks Pertanyaan Pengguna (Sampel Uji)	Waktu Respons (Detik)
1	Berapa biaya pendaftaran di SMP ini?	1,24
2	Syarat masuknya apa saja?	0,88
3	Kapan jadwal pendaftaran gelombang 1 dibuka?	1,10
4	Di mana lokasi pendaftarannya?	1,06
5	Apakah ada ekstrakurikuler pramuka?	0,91
6	Bayarnya berapa?	1,18
7	Info beasiswa jalur prestasi	1,10
8	Jam berapa sekolah buka?	1,18
9	Tolong informasikan alur pendaftaran	0,95
10	Fasilitas laboratorium komputer ada atau tidak?	0,90
	Rata-rata Waktu Respons	1,05 detik

Berdasarkan Tabel 5, rata-rata waktu respons chatbot adalah 1,05 detik. Hal ini menunjukkan bahwa implementasi algoritma sangat responsif dan mampu secara signifikan meningkatkan efisiensi layanan informasi PPDB serta mengurangi beban antrian pertanyaan pengguna dibandingkan layanan manual.

E. Pengujian Fungsional

Pengujian fungsional dilakukan menggunakan metode Black Box Testing terhadap 8 skenario utama yang mencakup widget chatbot, manajemen dataset, Firebase Auth, perekaman log, serta fitur ekspor CSV. Hasil pengujian menunjukkan seluruh skenario berjalan dengan output sesuai harapan dan dinyatakan Valid, mengonfirmasi bahwa seluruh fungsionalitas sistem telah memenuhi spesifikasi.

F. Pengujian User Acceptance Test (UAT)

UAT dilakukan terhadap 25 responden yang terdiri dari 24 calon peserta didik/orang tua dan 1 panitia PPDB SMP Sejahtera 2. Penilaian menggunakan Skala Likert 1–5 terhadap 10 pernyataan yang mencakup aspek website (3 indikator) dan aspek chatbot (7 indikator). Persentase kepuasan dihitung menggunakan rumus $P = (T/Y) \times 100\%$, di mana T = total skor empiris dan Y = skor maksimal. Tabel 6 menampilkan rekapitulasi hasil UAT.

Tabel 7. Hasil User Acceptance Test (UAT) dengan Skala Likert 1-5 (n=25)

No.	Aspek	Indikator Penilaian	Skor	Persentase
1	Website	Tampilan website menarik dan profesional	107	85,6%
2	Website	Navigasi menu mudah dipahami dan digunakan	107	85,6%
3	Website	Informasi halaman PPDB lengkap dan jelas	112	89,6%
4	Chatbot	Widget chatbot mudah ditemukan dan dibuka	113	90,4%
5	Chatbot	Tampilan percakapan nyaman dan mudah dibaca	110	88,0%
6	Chatbot	Chatbot memberikan respons yang cepat	110	88,0%
7	Chatbot	Jawaban sesuai dengan pertanyaan yang diajukan	106	84,8%
8	Chatbot	Chatbot memahami variasi kata/kalimat	105	84,0%
9	Chatbot	Pesan jelas saat pertanyaan tidak dapat dijawab	108	86,4%
10	Chatbot	Chatbot membantu pencarian informasi PPDB	114	91,2%

Raditya Mayesha Beldyq: *Penulis Korespondensi



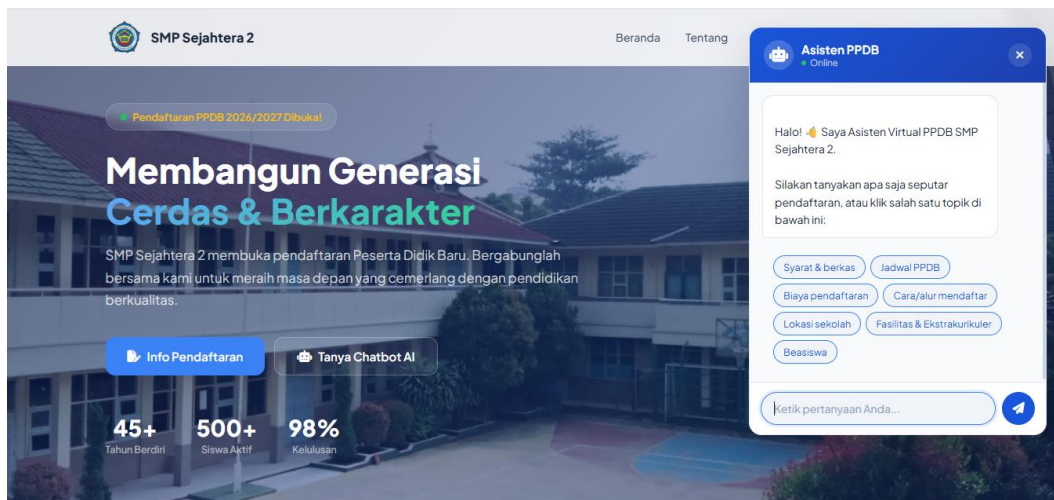
Copyright © 2026, Raditya Mayesha Beldyq, Rika Apriani.

No.	Aspek	Indikator Penilaian	Skor	Persentase
		Rata-rata Kepuasan Pengguna	108,2	87,4%

Rata-rata kepuasan pengguna sebesar 87,4% menunjukkan sistem berada pada kategori 'Sangat Baik'. Nilai tertinggi diraih pada indikator kebermanfaatan chatbot dalam membantu pencarian informasi PPDB (91,2%), diikuti kemudahan menemukan widget chatbot (90,4%). Nilai terendah pada kemampuan chatbot memahami variasi pertanyaan (84,0%), yang disebabkan keterbatasan kamus sinonim dan variasi pola pada knowledge base. Secara keseluruhan, hasil UAT membuktikan bahwa sistem chatbot berbasis pipeline NLP (TF-IDF + Cosine Similarity) berhasil diterima dengan baik oleh pengguna dan mampu meningkatkan kualitas layanan informasi PPDB.

G. Implementasi Sistem

Sistem chatbot berhasil diimplementasikan sebagai web serverless yang dapat diakses secara online. Sistem terdiri dari dua antarmuka utama: (1) antarmuka publik yang mencakup halaman utama website, widget chatbot AI, dan halaman informasi PPDB; serta (2) antarmuka admin yang mencakup dashboard kelola dataset, evaluasi log dengan fitur filter dan ekspor CSV, serta halaman pengaturan website (CMS dinamis). CMS dinamis memungkinkan admin mengubah informasi seperti tahun ajaran dan rincian biaya tanpa mengubah kode program, sehingga sistem dapat beroperasi secara mandiri dan berkelanjutan.



Gambar 2. Tampilan Antarmuka Widget Chatbot pada Halaman PPDB

Penggunaan Vanilla JavaScript tanpa library machine learning eksternal memastikan seluruh komputasi TF-IDF dan Cosine Similarity dapat dijalankan di sisi klien (browser pengguna). Pendekatan ini meminimalkan latensi respons karena tidak ada proses roundtrip ke server untuk perhitungan NLP. Hasil pengujian menunjukkan chatbot mampu memberikan respons dengan waktu rata-rata sekitar 1 detik untuk query pengguna.

4. KESIMPULAN

Berdasarkan hasil penelitian, diperoleh kesimpulan yang menjawab dua permasalahan utama yang diangkat. Pertama, pipeline NLP yang mengintegrasikan Text Preprocessing 7-tahap (case folding, cleansing, tokenization, normalisasi sinonim, stopword removal, stemming, dan bigram extraction) dengan pembobotan TF-IDF dan pencocokan intent menggunakan Cosine Similarity terbukti efektif dalam mengenali maksud pertanyaan pengguna pada domain PPDB. Evaluasi terhadap 35 sampel pengujian menghasilkan akurasi 100% pada kategori in-pattern, 66,7% pada out-of-pattern, dan 80% pada out-

of-domain, dengan akurasi keseluruhan sebesar 82,9%. Normalisasi sinonim dan bigram extraction berkontribusi signifikan dalam meningkatkan pencocokan intent, dibuktikan oleh bobot TF-IDF tertinggi pada bigram spesifik (0,291 untuk 'uang_biaya') yang mampu membedakan intent secara lebih akurat dibandingkan unigram saja.

Kedua, mekanisme adaptive threshold berbasis domain keyword filtering terbukti efektif meningkatkan relevansi jawaban. Threshold rendah (30%) diterapkan saat pertanyaan mengandung kata kunci domain PPDB, sementara threshold tinggi (65%) diterapkan untuk pertanyaan tanpa kata kunci domain, sehingga 80% pertanyaan di luar konteks PPDB berhasil disaring. Rata-rata waktu respons chatbot sebesar 1,05 detik membuktikan bahwa arsitektur client-side mampu memberikan respons secara efisien. User Acceptance Test terhadap 25 responden menghasilkan rata-rata kepuasan 87,4% (Sangat Baik), mengonfirmasi bahwa sistem dapat diterima dan digunakan dengan baik oleh pengguna akhir.

Untuk pengembangan selanjutnya, disarankan penambahan variasi pola pertanyaan secara berkala melalui fitur continuous evaluation guna meningkatkan akurasi pada kategori out-of-pattern, serta penerapan metode NLP yang lebih kompleks seperti word embedding atau model Transformer untuk meningkatkan pemahaman semantik pada pertanyaan dengan formulasi yang lebih beragam.

Secara keseluruhan, penelitian ini memberikan kontribusi berupa implementasi arsitektur chatbot informasi PPDB yang sepenuhnya client-side dan serverless dengan pipeline NLP 7-tahap yang terintegrasi, tanpa ketergantungan pada backend khusus maupun model deep learning. Pendekatan ini memiliki implikasi praktis yang signifikan: sekolah-sekolah lain dapat mengadopsi sistem serupa hanya dengan memperbarui file knowledge base (JSON) tanpa memerlukan infrastruktur server ataupun keahlian machine learning khusus. Untuk penelitian selanjutnya, direkomendasikan secara spesifik: (1) integrasi dengan platform pesan instan seperti WhatsApp atau Telegram melalui API webhook untuk memperluas jangkauan pengguna; (2) penerapan teknik active learning untuk pembaruan knowledge base secara semi-otomatis berdasarkan pertanyaan yang tidak terjawab; dan (3) eksplorasi model word embedding kontekstual seperti IndoBERT untuk meningkatkan pemahaman semantik pada pertanyaan dengan variasi linguistik yang lebih kompleks.

5. REFERENCES

- [1] S.-C. Necula, F. Dumitriu, and V. Greavu-Şerban, "A Systematic Literature Review on Using Natural Language Processing in Software Requirements Engineering," *Electronics*, vol. 13, no. 11, p. 2055, May 2024, doi: 10.3390/electronics13112055.
- [2] E. Grassini, M. Buzzi, B. Leporini, and A. Vozna, "A systematic review of chatbots in inclusive healthcare: insights from the last 5 years," *Univers. Access Inf. Soc.*, vol. 24, no. 1, pp. 195–203, Mar. 2025, doi: 10.1007/s10209-024-01118-x.
- [3] S. Shah, H. Ghomeshi, E. Vakaj, E. Cooper, and S. Fouad, "A review of natural language processing in contact centre automation," *Pattern Anal. Appl.*, vol. 26, no. 3, pp. 823–846, Aug. 2023, doi: 10.1007/s10044-023-01182-8.
- [4] W. S. Nsaif, H. M. Salih, H. H. Saleh, and B. T. Al-Nuaim, "Conversational Agents: An Exploration into Chatbot Evolution, Architecture, and Important Techniques," *Eurasia Proc. Sci. Technol. Eng. Math.*, vol. 27, pp. 246–262, Aug. 2024, doi: 10.55549/epstem.1518795.
- [5] Kementerian Pendidikan dan Kebudayaan Republik Indonesia, "Peraturan Menteri Pendidikan dan Kebudayaan Nomor 1 Tahun 2021 tentang Penerimaan Peserta Didik Baru," Kementerian Pendidikan dan Kebudayaan, 2021. Accessed: Apr. 28, 2026.

- [Online]. Available: <https://peraturan.bpk.go.id/Details/163568/permendikbud-no-1-tahun-2021>
- [6] R. Suci, Ismarmiaty, and R. Rismayati, "SISTEM INFORMASI PENERIMAAN SISWA BARU BERBASIS WEBSITE SDN 3 GERUNG UTARA," *JATISKOM J. Apl. Teknol. Inf. Dan Sains Komput.*, vol. 1, no. 2, pp. 134–146, Dec. 2024, doi: 10.20414/jatiskom.v1i2.12689.
- [7] E. Mursidah, L. Ambarwati, and F. A. Karima, "IMPLEMENTASI CHATBOT LAYANAN INFORMASI PENDAFTARAN MAHASISWA BARU PROGRAM PASCASARJANA DEPARTEMEN TEKNIK INFORMATIKA ITS," *Netw. Eng. Res. Oper.*, vol. 7, no. 1, p. 43, Apr. 2022, doi: 10.21107/nero.v7i1.276.
- [8] A. Jalili, H. Tabrizchi, A. Mosavi, and A. R. Varkonyi-Koczy, "Enhancing Language Model Performance with a Novel Text Preprocessing Method," *Acta Phys. Pol. A*, vol. 146, no. 4, pp. 542–552, Oct. 2024, doi: 10.12693/APhysPolA.146.542.
- [9] A. A. Nafea *et al.*, "A Brief Review on Preprocessing Text in Arabic Language Dataset: Techniques and Challenges," *Babylon. J. Artif. Intell.*, vol. 2024, pp. 46–53, May 2024, doi: 10.58496/BJAI/2024/007.
- [10] M. Das, Selvakumar Kamalanathan, and Pja Alphonse, "A Comparative Study on TF-IDF feature Weighting Method and its Analysis using Unstructured Dataset," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 5, Aug. 2023, doi: <https://doi.org/10.48550/arXiv.2308.04037>.
- [11] L. Zhang, "Features extraction based on Naive Bayes algorithm and TF-IDF for news classification," *PLOS One*, vol. 20, no. 7, p. e0327347, Jul. 2025, doi: <https://doi.org/10.1371/journal.pone.0327347>.
- [12] J. Halim and D. Lasut, "Document Plagiarism Detection Application Using Web-Based TF-IDF and Cosine Similarity Methods: English," *Bit-Tech*, vol. 7, no. 2, pp. 202–213, Dec. 2024, doi: 10.32877/bt.v7i2.1697.
- [13] M. B. Arrisalah, M. M. A. Haromainy, and A. Junaidi, "Design of Thesis Topic Recommendation System Using TF-IDF and Cosine Similarity," *J. Inf. Syst. Technol.*, vol. 8, no. 2, Apr. 2026.
- [14] M. R. A. Prasetya and A. M. Priyatno, "Dice Similarity and TF-IDF for New Student Admissions Chatbot," *RIGGS J. Artif. Intell. Digit. Bus.*, vol. 1, no. 1, pp. 13–18, Jul. 2022, doi: 10.31004/riggs.v1i1.5.
- [15] R. H. Saputra and H. Sucipto, "Implementation of a WhatsApp-based Canteen Menu Chatbot using TF-IDF and cosine similarity: A case study of employees at Matahari Department Store, Sunrise Mall and Mojokerto," vol. 9, no. 1, 2026.
- [16] E. Aprianto, D. Mahdiana, and A. Wibowo, "Optimizing Bag of Words and Word2Vec with Vocabulary Pruning and TF-IDF Weighted Embeddings for Accurate Chatbot Responses in Indonesian Treasury Services," *J. Tek. Inform. Jutif*, vol. 7, no. 1, pp. 587–605, Feb. 2026, doi: 10.52436/1.jutif.2026.7.1.5370.
- [17] R. M. Holis, P. E. P. Utomo, and B. F. Hutabarat, "Semantic FAQ Chatbot Using SBERT (Sentence-BERT) and Cosine Similarity for Academic Services," *Brill. Res. Artif. Intell.*, vol. 5, no. 2, pp. 915–922, Oct. 2025, doi: 10.47709/brilliance.v5i2.7027.
- [18] Mohammed Salah Ibrahim, Anas Diab Sallibi, Ahmed Adnan Hadi, and Ahmed Adil Nafea, "A Chatbot for Frequently Asked Questions using TF-IDF and Query Expansion Techniques," *Passer J. Basic Appl. Sci.*, no. 1, Jun. 2025, doi: 10.24271/psr.2025.487515.1805.
- [19] D. Muazarah, B. Fachri, and A. Khaliq, "Web-Based New Student Admission System using the Rapid Application Development (RAD) Method at Penraujan Private Junior High School," vol. 16, no. 02, 2026.
- [20] W. J. T. Subroto, Lady Silk Moonlight, and Hadi Prayitno, "Implementation of the Rapid Application Development (RAD) Method for the Information System of Sultan Babullah Ternate Airport Facilities," *J. Syst. Inf. Technol.*, pp. 27–34, 2025.