MIFORTEKH (Jurnal Manajemen Informatika & Teknologi)
p-ISSN : 2808-7550 (print) e-ISSN : 2798-0235 (online)
Vol. 6, No. 2, Oktober 2026
https://journal.stiestekom.ac.id/index.php/mifortekh

Penerapan Algoritma TF-IDF dan Cosine Similarity pada Fitur Chatbot untuk Layanan Pelanggan dan Operasional E-Commerce (Studi Kasus: PT Radhika Narya Daruna)

Mochamad Alvin Farrel Jizdan[1], Marsani Asfi[2], Arif Nursetyo[3]
[1]Teknik Informatika, Universitas Catur Insan Cendekia Kota Cirebon, Indonesia
[2,3]Sistem Informasi, Universitas Catur Insan Cendekia Kota Cirebon, Indonesia
Email author: jizdan.a14@gmail.com [1], marsani.asfi@cic.ac.id [2], arif.nursetyo@cic.ac.id [3]

Article Info

Article history:

Received July 30, 2026
Revised November 09, 2026
Accepted November 10, 2026

Keywords:

Chatbot,
Cosine Similarity,
Gemini API,
Interactive Sales Assistant,
Retrieval-Augmented Generation,
TF-IDF.

ABSTRACT

Responsive, accurate, and full-time customer service is essential for supporting e-commerce platform operations. PT Radhika Narya Daruna requires an update to its customer service system to facilitate product information communication and facilitate transactions more efficiently. This research aims to design and implement a web-based intelligent chatbot system that functions as an integrated Interactive Sales Assistant. The system development applies a hybrid method approach within the Retrieval-Augmented Generation (RAG) architecture framework. In the Natural Language Understanding (NLA) stage, the system implements the Term Frequency-Inverse Document Frequency (TF-IDF) algorithm for word weighting and Cosine Similarity to mathematically validate the relevance of questions against a corpus of Frequently Asked Questions (FAQs). Questions that meet the relevance threshold are then combined with actual product catalog data into a dynamic prompt.

Corresponding Author:

Mochamad Alvin Farrel Jizdan,
Universitas Catur Insan Cendekia
Jl. Kesambi No. 202 Kota Cirebon
Email : Jizdan.a14@gmail.com

OPEN ACCESS

1. PENDAHULUAN

Teknologi informasi telah merevolusi kegiatan bisnis, namun PT Radhika Narya Daruna yang bergerak di bidang ritel masih menghadapi kendala operasional. Mengandalkan aplikasi pesan pihak ketiga seperti WhatsApp untuk layanan pelanggan sering kali membuat proses belanja menjadi tidak praktis dan berisiko menggagalkan keputusan pembelian. Untuk mengatasi hal ini, implementasi chatbot berbasis kecerdasan buatan (AI) yang terintegrasi langsung di dalam sistem *e-commerce*

menjadi sebuah urgensi dan berfungsi efektif sebagai pengganti agen layanan pelanggan konvensional (D. Rusdianto and Y. Herdiana, 2023).

Chatbot pada platform *e-commerce* tidak sekadar menjawab pertanyaan rutin, tetapi berperan krusial dalam mempercepat proses transaksi, memberikan rekomendasi produk yang akurat, serta secara terukur meningkatkan konversi penjualan (Nayla Sifaul Qolbi, Genduk Nurul Aini, and Helen Risky Dwi Wahyuni, 2025). Dalam mengatasi keterbatasan *chatbot* berbasis aturan yang kaku, sistem harus mampu mengukur relevansi dan mengenali maksud (*intent*) pengguna meskipun terdapat variasi susunan kalimat atau kesalahan ketik. Penerapan algoritma *Term Frequency-Inverse Document Frequency* (TF-IDF) yang dipadukan dengan *Cosine Similarity* terbukti secara empiris sangat tangguh untuk mengekstraksi fitur teks, memberikan pembobotan kata secara proporsional, dan mengukur kemiripan kueri secara matematis (R. Ranveera, A. Kesharwani, and S. Kumari, 2021). Selain kendala pada aspek pelayanan, tata kelola operasional *e-commerce* di PT Radhika Narya Daruna juga masih menghadapi tantangan inefisiensi pada sisi *back-office*. Mengandalkan staf manusia untuk memproses setiap tahapan interaksi bisnis secara manual merupakan pekerjaan yang menyita waktu (R. Ranveera, A. Kesharwani, and S. Kumari, 2021). Ketiadaan sistem otomasi pada tugas manajerial, seperti sinkronisasi ketersediaan stok inventaris, menyebabkan alur kerja operasional menjadi lambat dan rentan terhadap ketidaksesuaian data (*human error*). Padahal, integrasi kecerdasan buatan seharusnya dioptimalkan tidak hanya untuk interaksi layanan pelanggan, tetapi juga untuk meningkatkan efisiensi operasional bisnis secara menyeluruh (A. P. Rizaldy, S. Riadi, and N. Wijaya, 2025).

Untuk mengatasi inefisiensi operasional tersebut, penelitian ini menginovasikan penerapan teknologi Agen Kecerdasan Buatan (*Autonomous Agents*). Agen otonom berbasis *Large Language Model* (LLM) merupakan sistem cerdas yang mampu merencanakan dan mengeksekusi instruksi kompleks dengan berinteraksi langsung terhadap lingkungan sistem eksternal (*Agent* AI). Fungsionalitas agen operasional ini diimplementasikan menggunakan teknologi *Function Calling*, di mana model kecerdasan buatan dilatih untuk secara mandiri mengeksekusi pemanggilan Antarmuka Pemrograman Aplikasi (API) ke dalam basis data (*Function Calling*). Melalui agen virtual yang dilindungi *Role-Based Access Control* (RBAC), admin dapat memperbarui ketersediaan stok produk dengan sangat efisien hanya melalui instruksi bahasa alami.

2. LANDASAN TEORI

Katalog produk (*Product Catalog*) dalam sistem *e-commerce* merupakan pilar informasi elektronik yang menyajikan spesifikasi, visual, dan harga barang secara komprehensif kepada pelanggan (K. G. Krisnanda, W. Gede, N. Rifky, and L. Rahardian, 2023). Dalam ekosistem belanja daring, ketersediaan e-katalog memfasilitasi konsumen untuk mencari informasi produk secara mandiri tanpa batasan ruang dan waktu (K. G. Krisnanda, W. Gede, N. Rifky, and L. Rahardian, 2023).

Chatbot adalah aplikasi berbasis AI yang memanfaatkan algoritma pemrosesan bahasa alami (*Natural Language Processing*/NLP) dan pembelajaran mesin (*Machine Learning*) untuk memahami dan merespons input pengguna secara otomatis. *Chatbot modern* dibagi menjadi dua jenis utama: *rule-based* (berbasis aturan) dan AI-*based* (berbasis pembelajaran) (A. P. Rizaldy, S. Riadi, and N. Wijaya, 2025). *Chatbot* dalam konteks penelitian ini tidak hanya berfungsi sebagai penjawab pesan otomatis biasa, tetapi juga untuk melayani proses pembelian melalui *website*. Dengan sistem ini, pelanggan dapat menanyakan informasi katalog produk, mengecek ketersediaan stok, sebelum diteruskan ke *dashboard* admin. Pendekatan ini memberikan keuntungan ganda, yaitu mempermudah pelanggan dalam mendapatkan informasi dan bertransaksi tanpa terhalang batasan jam operasional, serta membantu pihak perusahaan dalam menangani pertanyaan berulang dan pesanan secara lebih efisien dan terorganisir. Proses layanan pelanggan yang terintegrasi dengan sistem inventaris ini dirancang menggunakan algoritma *Term Frequency-Inverse Document Frequency* (TF-IDF) dan *Cosine Similarity* agar alur percakapannya dinamis dan sistem mampu memahami maksud pesan pelanggan dengan akurat, meskipun terdapat variasi susunan kalimat atau kesalahan ketik (*typo*). Dengan demikian, sistem *chatbot* yang dikembangkan tidak hanya meningkatkan efektivitas waktu respons layanan pelanggan, tetapi juga mempercepat proses pemesanan komoditas sehingga pengalaman bertransaksi pelanggan menjadi lebih praktis dan memuaskan.(L. Indahsari, K. Kusnadi, and T. E. Putri, 2021)

Algoritma Term Frequency-Inverse Document Frequency (TF-IDF) merupakan bidang information retrieval yang digunakan untuk menghitung seberapa relevan sebuah kata dalam rangkaian korpus dengan sebuah teks (V. Meida Hersianty, E. Larasati Amalia, D. Puspitasari, and D. Wahyu Wibowo, 2025). Algoritma ini menggabungkan dua konsep untuk perhitungan bobot, yaitu frekuensi kemunculan sebuah kata yang disebut term frequency (TF) di dalam sebuah dokumen tertentu dan inverse frequency dokumen yang mengandung kata yang disebut inverse document frequency (IDF) (V. Meida Hersianty, E. Larasati Amalia, D. Puspitasari, and D. Wahyu Wibowo, 2025).

Cosine Similarity adalah salah satu ukuran kemiripan paling popular yang diterapkan pada dokumen teks, seperti dalam berbagai aplikasi pengambilan dalam pembentukan dan pengelompokkan. Algoritma ini digunakan untuk menghitung *similarity* (tingkat kesamaan) antar dua buah objek. Secara umum perhitungan algoritma ini didasarkan pada *vector space similarity measure* (S. I. Adam and W. Grivin, 2025). Algoritma *Cosine Similarity* ini menghitung *similarity* antara dua buah objek yang dinyatakan dalam dua buah vector dengan menggunakan kata kunci dari sebuah dokumen sebagai ukuran (V. Meida Hersianty, E. Larasati Amalia, D. Puspitasari, and D. Wahyu Wibowo, 2025).

Natural Language Processing (NLP) adalah kombinasi dari ilmu komputer dan bidang kecerdasan buatan yang terkait dengan linguistik. NLP berkaitan dengan bagaimana mesin memahami bahasa manusia untuk saling berinteraksi. Dengan adanya NLP, komputer dapat belajar dan memahami bahasa manusia, sehingga komputer dapat berkomunikasi dengan manusia. Bahasa manusia adalah hal yang unik karena dibuat khusus untuk menyampaikan suatu makna. Untuk membuat komputer dapat mengerti bahasa manusia adalah tugas yang sulit, karena bahasa manusia memiliki struktur yang kompleks. Selain itu, setiap bahasa memiliki keunikannya sendiri dan mungkin memiliki makna ganda (K. Rizqy Putri Nugroho, C. M. Joy Sihombing, and F. Ihda Husni Husodo, 2025).

Agentic AI merupakan sebuah sistem yang diharapkan mampu menyelesaikan tugas melalui perencanaan dan tindakan yang diarahkan secara mandiri untuk mencapai tingkat kecerdasan buatan umum (*Artificial General Intelligence*/AGI). Dalam perkembangan terkini, *Large Language Models* (LLMs) dimanfaatkan sebagai pengontrol pusat (*central controllers*) untuk membangun agen otonom guna mendapatkan kemampuan pengambilan keputusan yang setara dengan manusia. Arsitektur agen otonom berbasis LLM ini secara umum dibangun atas modul profil, modul memori, modul perencanaan (*planning*), dan modul aksi (*action*). Modul perencanaan berfungsi memberdayakan agen dengan kemampuan untuk memecah tugas yang kompleks menjadi beberapa subtugas yang lebih sederhana dan menyelesaikannya secara individual. Selanjutnya, modul aksi bertanggung jawab pada tahap paling hilir untuk menerjemahkan keputusan agen menjadi keluaran yang spesifik serta berinteraksi secara langsung dengan lingkungan eksternal (L. Wang et al, 2024).

Role-Based Access Control (RBAC) merupakan mekanisme pengamanan yang mengatur hak akses pengguna berdasarkan peran yang telah ditetapkan secara spesifik di dalam sistem, sehingga hanya pengguna yang berwenang yang dapat mengakses fitur tertentu. Implementasi RBAC terbukti dapat membantu meningkatkan keamanan, mempermudah manajemen akses, serta meminimalkan risiko kebocoran data dengan menerapkan prinsip hak akses minimum (*least privilege*). Keberagaman peran dalam suatu aplikasi menciptakan ekosistem interaksi yang kompleks, sehingga dapat memunculkan risiko *privilege escalation* atau penyalahgunaan akses lintas peran apabila tidak dikelola dengan mekanisme kontrol yang memadai. Di dalam arsitektur pengembangan web modern, perlindungan sistem tidak boleh hanya berfokus pada penyembunyian antarmuka di sisi *frontend* (*security by obscurity*). Mekanisme RBAC harus diintegrasikan pada lapisan *middleware* di sisi *backend* (seperti Express.js) untuk memverifikasi otoritas pengguna secara *real-time* dan memastikan adanya pemisahan hak akses yang ketat antara fitur administratif dan operasional (Rizal Firmansyah, A. Hamdi, and D. Riandini, 2026).

Function calling memungkinkan model untuk belajar memutuskan secara mandiri API mana yang akan dipanggil, kapan waktu yang tepat untuk memanggilnya, argumen apa yang harus diteruskan, dan bagaimana cara terbaik untuk memasukkan hasil tersebut ke dalam prediksi kata selanjutnya. Saat melakukan *generate* teks, model bahasa akan menghasilkan token panggilan API, menghentikan sementara proses *decoding* untuk mengeksekusi API yang sesuai, dan kemudian melanjutkan kembali proses *decoding* setelah menyisipkan respons dari API tersebut. Penggunaan alat atau API eksternal ini secara signifikan memperluas ruang aksi (*action space*) dari agen dan menjadi solusi krusial untuk memitigasi masalah halusinasi (*hallucination*) pada keluaran kecerdasan buatan (T. Schick et al, 2023).

API Gemini (Application Programming Interface Gemini) merupakan antarmuka pemrograman aplikasi yang dikembangkan oleh Google untuk memanfaatkan kemampuan AI generatif mereka,

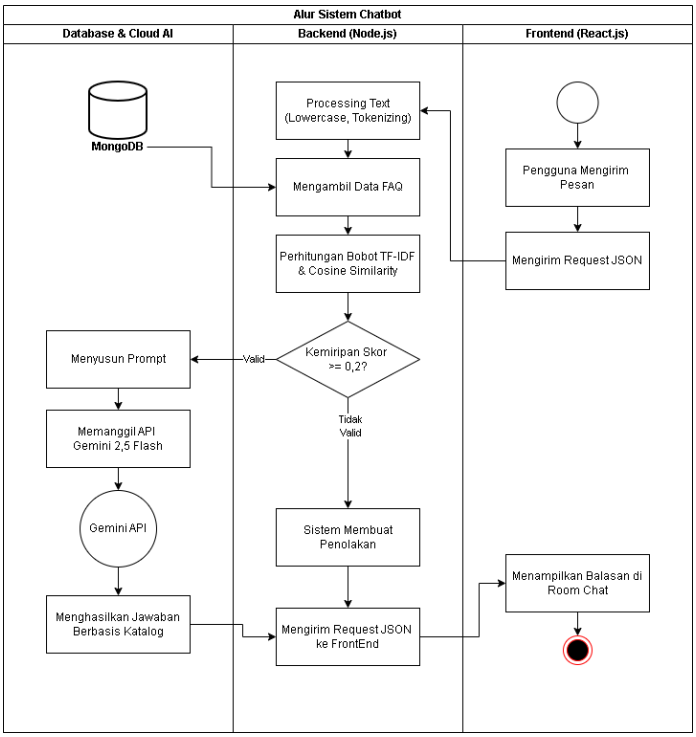
khususnya model bahasa generatif seperti Gemini. Gemini adalah model kecerdasan buatan yang dilatih untuk menghasilkan teks, memberikan tanggapan alami dan kontekstual dalam percakapan, serta melakukan berbagai tugas berbasis bahasa. LLM Gemini dapat digunakan untuk bebragai keperluan, seperti meringkas teks, menjawab pertanyaan, mengklasifikasikan data, melakukan analisis sentimen, mengekstrak data, membuat content, membuat kode program, dan melakukan percakapan (C. Lusia Kiareni et al, 2024).

3. METODE

Metode Retrieval-Augmented Generation (RAG) adalah teknologi arsitektur yang memadukan proses pencarian informasi dan pemrosesan bahasa alami (Natural Language Processing/NLP) untuk menghasilkan jawaban yang lebih relevan dan berbasis pada dokumen tertentu. Konsep ini diperkenalkan sebagai solusi untuk meningkatkan kualitas jawaban yang dihasilkan oleh model bahasa alami dengan memanfaatkan sumber daya informasi eksternal di dalam proses generasi teksnya (M. S. A. Setiawan, A. Pratama, and P. K. F. Ananto, 2026).

Dalam konteks pengembangan chatbot pada PT Radhika Narya Daruna, teknik ini memungkinkan sistem untuk mengambil informasi faktual yang relevan dari basis data lokal, seperti dokumen FAQ dan data katalog produk, menggunakan algoritma TF-IDF. Informasi hasil pencarian tersebut kemudian diinjeksikan ke dalam prompt sebelum menginstruksikan API Gemini untuk menyintesis jawaban. Pemilihan Gemini API (khususnya versi Flash) pada penelitian ini didasarkan pada keunggulan komputasinya dalam memproses penalaran (reasoning) secara luwes, terutama untuk menjaga stabilitas semantik pada bahasa non-Inggris dan tata bahasa Indonesia yang dinamis (M. S. A. Setiawan, A. Pratama, and P. K. F. Ananto, 2026). Selain itu, pengujian arsitektur pada lingkungan chatbot menunjukkan bahwa model Gemini mencatat latensi inferensi yang sangat rendah, sehingga mampu memberikan waktu tanggap (response time) yang real-time tanpa membebani server operasional (R. Fahreji and R. S. Wijaya, 2026). Model ini menjadikannya sangat ideal untuk arsitektur RAG karena mampu memproses gabungan dokumen korpus yang panjang menjadi respons berorientasi transaksi, serta secara empiris terbukti menekan fenomena halusinasi AI karena telah dipagari oleh data lokal perusahaan (M. S. A. Setiawan, A. Pratama, and P. K. F. Ananto, 2026).

Sistem *chatbot* pada PT Radhika Narya Daruna dibangun menggunakan arsitektur *hybrid* yang menggabungkan metode *Information Retrieval* klasik (TF-IDF) dengan teknologi *Generative AI* (Gemini 2.5 Flash). Berbeda dengan sistem layanan informasi pada umumnya yang menggunakan platform pihak ketiga, arsitektur ini mengintegrasikan *widget chat* secara langsung ke dalam ekosistem *website* perusahaan guna memastikan keamanan dan sinkronisasi data katalog secara *real-time*.



Gambar 1. Arsitektur Sistem

Gambar 1 diatas mengilustrasikan arsitektur sistem dari alur kerja sistem *chatbot* terintegrasi yang terbagi ke dalam tiga partisi (*swimlane*) arsitektur utama, yaitu *Database & Cloud AI*, *Backend* (Node.js), dan *Frontend* (React.js).

3.1. Data Collecting

Data collecting merupakan proses untuk mencari dan mengumpulkan data yang akan digunakan sebagai otak atau *Knowledge Base* dari *chatbot*. Pada penelitian ini, pengumpulan data dibagi menjadi dua sumber utama:

1. Data FAQ (Pertanyaan Umum): Disimpan di dalam MongoDB. Berisi daftar pertanyaan dan jawaban statis mengenai informasi umum perusahaan seperti lokasi gudang, cara pemesanan, dan syarat pengiriman.
2. Data Katalog Produk (*Real-time*): Ditarik langsung dari *database* MySQL sistem *e-commerce* perusahaan. Data ini berupa *array* produk (seperti Kopra Putih, Kelapa Cungkil) beserta harga terbarunya untuk disuntikkan ke dalam *prompt* AI.

3.2. Preprocessing Text

Sebelum dilakukan perhitungan algoritma, setiap pesan yang masuk melalui *server* akan melewati tahap pembersihan data untuk meningkatkan akurasi pencocokan. Tahapan *preprocessing* yang diimplementasikan meliputi:

1. *Case Folding*: Mengubah seluruh karakter menjadi huruf kecil (*lowercase*).
2. *Cleaning*: Menghapus simbol, tanda baca, dan karakter non-alfanumerik menggunakan *Regular Expression* (Regex).
3. *Tokenizing*: Memecah kalimat menjadi potongan kata tunggal (*token*) untuk dihitung frekuensi kemunculannya.

Tabel 1. Data Sampling Preprocessing Text

No	Tahapan Sistem	Hasil Pemrosesan Teks	Keterangan
1	Case Folding	"halo min, berapa harga kopra asongan hari ini!?"	Sistem menormalisasi seluruh karakter huruf menjadi huruf kecil (lowercase).

2	Cleaning (Regex)	"halo min berapa harga kopra asongan hari ini"	Sistem menghapus seluruh tanda baca (?, !, ,) menggunakan Regular Expression sehingga hanya menyisakan karakter alfanumerik.
3	Tokenizing	["halo", "min", "berapa", "harga", "kopra", "asongan", "hari", "ini"]	Sistem memecah kalimat menjadi array of strings (kumpulan token) berdasarkan spasi. Token ini yang akan dihitung bobotnya dengan algoritma TF-IDF.

3.3. Model Implementation (TF-IDF dan Cosine Similarity)

Berbeda dengan model klasifikasi yang memerlukan proses *training* data yang lama, implementasi model pada penelitian ini menggunakan pendekatan pencarian informasi (*Information Retrieval*). Model ini membandingkan *token* dari pesan pengguna dengan korpus dokumen di *Knowledge Base*.

1. Perhitungan TF-IDF: Sistem menghitung *Term Frequency* (frekuensi kemunculan kata pada pesan) dan mengalikannya dengan *Inverse Document Frequency* (bobot keunikan kata dari seluruh dokumen di *database*).
2. *Cosine Similarity*: Mengukur sudut vektor antara kalimat pengguna dan kalimat di *database*. Semakin mendekati angka 1, semakin mirip makna kalimat tersebut.

Gatekeeper (Filter Keamanan): Hasil *Cosine Similarity* dievaluasi. Jika skor kemiripan tertinggi $\leq 0{,}2$, sistem akan langsung menolak pesan (dianggap *out of context*). Jika $\geq 0{,}2$, pesan dianggap valid untuk diteruskan ke tahap generasi.

Tabel 2. Data Sampling Model Implementation

NO	Teks *Input* Pengguna (*Preprocessed*)	*Knowledge Base* Terdekat	Skor Cosine Similarity	Keputusan Gatekeeper
1	"apakah tersedia komoditas kopra"	"Berapa harga kopra hari ini?"	0.73	(Valid) Skor ≥ 0.2.
2	"halo min selamat pagi"	"Halo, ada yang bisa dibantu terkait produk kami?"	0.12	(Tidak Valid) Skor ≤ 0.2.

Mengambil dari sampel data no. 1 pada Tabel 2.

1. Kueri Pengguna (A): "apakah tersedia komoditas kopra"
2. Referensi FAQ (B): "Berapa harga kopra hari ini?"

Setelah *preprocessing* (termasuk penghapusan kata imbuhan dan kata hubung), kata kunci inti yang diekstraksi beserta asumsi nilai bobot TF-IDF-nya direpresentasikan dalam matriks vektor berikut

Tabel 3. Total Nilai Bobot

Token Kata Kunci	Bobot Kueri (A)	Bobot FAQ (B)	A×B
komoditas	1.00	0.00	0.00
kopra	1.80	1.80	3.24

harga	0.00	1.20	0.00
Total			$\sum = 3.24$

$$|A| = \sqrt{1.00^2 + 1.80^2} = \sqrt{1.00 + 3.24} = \sqrt{4.24} = 2.059$$

$$|B| = \sqrt{0.00^2 + 1.80^2 + 1.20^2} = \sqrt{0 + 3.24 + 1.44} = \sqrt{4.68} = 2.163$$

Pada tahap akhir komputasi, tingkat kemiripan teks ditentukan dengan membagi nilai total dengan hasil kali antara panjang vektor A dan vektor B. Perhitungan ini akan merepresentasikan kedekatan sudut antar vektor dengan menghasilkan rasio kemiripan dalam rentang angka 0 hingga 1.

$$Cos\alpha = \frac{3.24}{2.059 \times 2.163} = \frac{3.24}{4.453} = 0.727 \approx 0.73$$

Berdasarkan komputasi matematis di atas, diperoleh skor kemiripan akhir sebesar 0.73. Dikarenakan skor tersebut bernilai lebih besar dari ambang batas toleransi (*threshold*) sistem yaitu 0,2, maka fungsi *Gatekeeper* secara valid mengklasifikasikan kueri pelanggan tersebut ke dalam kategori relevan dan langsung meneruskannya ke Gemini API.

Mengambil dari sampel data no. 2 pada Tabel 2.
1. Kueri Pengguna (A): "Halo min selamat pagi"
2. Referensi FAQ (B): " Halo, ada yang bisa dibantu terkait produk kami?"

Setelah *preprocessing* (termasuk penghapusan kata imbuhan dan kata hubung), kata kunci inti yang diekstraksi beserta asumsi nilai bobot TF-IDF-nya direpresentasikan dalam matriks vektor berikut:

Token Kata Kunci	Bobot Kueri (A)	Bobot FAQ (B)	A×B
halo	1.00	1.00	1.00
min	1.30	0.00	0.00
selamat	1.30	0.00	0.00
pagi	1.30	0.00	0.00
Total			$\sum = 1.00$

$$|A| = \sqrt{1.00^2 + 1.30^2 + 1.30^2 + 1.30^2} = \sqrt{1.00 + 1.69 + 1.69 + 1.69} = \sqrt{6.07}$$

$$= 2.463$$

$$|B| = \sqrt{1.00^2 + 1.85^2 + 1.85^2 + 1.85^2} = \sqrt{1.00 + 3.4225 + 3.4225 + 3.4225}$$

$$= \sqrt{11.2675} = 3.356$$

Pada tahap akhir komputasi, tingkat kemiripan teks ditentukan dengan membagi nilai total dengan hasil kali antara panjang vektor A dan vektor B. Perhitungan ini akan merepresentasikan kedekatan sudut antar vektor dengan menghasilkan rasio kemiripan dalam rentang angka 0 hingga 1.

$$Cos\alpha = \frac{1.00}{2.463 \times 3.356} = \frac{1.00}{8.265} = 0.1209 \approx 0.12$$

Berdasarkan komputasi matematis di atas, diperoleh skor kemiripan akhir sebesar 0.12, karena skor tersebut bernilai kurang dari ambang batas toleransi (*threshold*) sistem yaitu 0,2, maka fungsi *Gatekeeper* secara konsisten memutus rute komputasi dan menolak pesan tersebut.

3.4. System Integration

Tahap implementasi sistem pada penelitian ini dilakukan dengan membangun sebuah *backend service* menggunakan Node.js sebagai penghubung antara pengguna, algoritma TF-IDF, dan layanan API Gemini. Antarmuka pengguna dikembangkan menggunakan React.js dalam bentuk *widget chat* di dalam *website* utama perusahaan.

Implementasi dirancang agar ketika pertanyaan pengguna lolos filter *Gatekeeper* (skor ≥ 0.2), Node.js akan menyusun *prompt* dinamis. *Prompt* ini menggabungkan pesan pengguna, jawaban FAQ dari hasil TF-IDF, serta data katalog produk secara *real-time*. Keseluruhan *prompt* ini dikirim ke model Gemini 2.5 Flash untuk diolah menjadi jawaban yang memiliki gaya bahasa natural layaknya seorang *Customer Service* manusia, namun tetap terikat pada fakta data katalog perusahaan.

Untuk meningkatkan nilai guna sistem, diimplementasikan fitur Interactive Sales Assistant. Mekanisme ini bekerja dengan menyisipkan instruksi khusus pada *prompt* Gemini AI agar menyertakan token identitas produk (misalnya [product.id]) saat memberikan rekomendasi. *Backend* Node.js kemudian melakukan proses *parsing* untuk memisahkan teks jawaban dengan ID produk tersebut, sehingga *frontend* dapat merender komponen tombol interaktif secara dinamis.

Tabel 4 Data Sample System Integration

NO	Komponen	Data Sample	Fungsi bagi Model
1	Identitas Sistem	"Anda adalah AI Customer Service PT Radhika Narya Daruna. Tugas Anda membantu penjualan produk"	Mengatur gaya bahasa agar profesional dan membatasi fokus AI hanya pada tujuan bisnis.
2	Contextual FAQ (Cara Pemesanan)	"Informasi: Pemesanan dapat dilakukan dengan klik tombol 'Tambah ke Keranjang' atau melalui WhatsApp admin di nomor 0812-XXXX."	Memberikan Standar Operasional Prosedur (SOP) resmi perusahaan agar AI tidak menyarankan metode pemesanan yang keliru atau fiktif.
3	Live Product Data (Katalog)	"[{"id": "KPR-01", "nama": "Kopra Putih Grade A", "harga": 13500}, {"id": "KLC-02", "nama": "Kelapa Cungkil", "harga": 11000}]"	Menyediakan data metrik (stok dan harga) terbaru yang diambil langsung dari *database* secara *real-time*.
4	User Query	"Saya tertarik beli kopra putih, cara pesannya gimana ya?"	Pesan atau pertanyaan aktual yang diajukan oleh pengguna melalui *widget chat*.

3.5. Interactive Sales Assistant Integration

Tahap integrasi asisten penjualan interaktif bertujuan untuk menghubungkan kemampuan generatif Gemini AI dengan fungsionalitas keranjang belanja (*shopping cart*) pada sistem *e-commerce* perusahaan. Mekanisme ini memastikan bahwa setiap rekomendasi produk yang diberikan oleh *chatbot* dapat langsung ditindaklanjuti oleh pengguna melalui aksi transaksional. Proses integrasi ini dibagi menjadi tiga tahapan teknis utama:

1. Mekanisme *Prompt* Engineering dan Injeksi Metadata

Pada tahap awal, *backend* Node.js menyusun instruksi (*prompt*) khusus yang mewajibkan Gemini AI untuk menyisipkan token identitas produk setiap kali memberikan rekomendasi barang dari katalog. Data katalog yang dikirimkan ke AI bersumber dari *database* MySQL yang memuat atribut product_id, nama, dan harga terkini.

2. Pemrosesan dan *Parsing Token* pada *Backend*

Setelah Gemini AI menghasilkan teks jawaban, server Node.js tidak langsung mengirimkan teks tersebut ke pengguna. Sistem melakukan proses penyaringan (parsing) menggunakan metode Regular Expression (Regex) untuk mendeteksi keberadaan token produk.

Jika token ditemukan, sistem akan mengekstraksi nilai ID produk tersebut dan menghapus kode token dari teks jawaban agar antarmuka chat tetap terlihat bersih dan profesional bagi pengguna. Hasil ekstraksi ini kemudian dibungkus ke dalam objek JSON response sebagai atribut productId.

3. Render Komponen Transaksional pada Frontend

Pada sisi *Client* (React.js), sistem menerima paket data JSON yang berisi teks jawaban dan metadata productId. Komponen *widget chat* diinstruksikan untuk melakukan pemindaian terhadap atribut tersebut:

a) Jika productId bernilai *null*, sistem hanya menampilkan *bubble chat* berisi teks biasa.
b) Jika productId berisi nilai ID tertentu, React secara dinamis akan me-*render* tombol "Tambah ke Keranjang" tepat di bawah teks jawaban tersebut.

Tombol ini terhubung langsung dengan fungsi handleAddToCart() pada aplikasi utama, sehingga ketika diklik, produk yang direkomendasikan oleh AI Customer Service akan otomatis masuk ke dalam antrean belanja pengguna tanpa harus meninggalkan halaman *chat*.

4. HASIL DAN PEMBAHASAN

4.1. Hasil Implementasi Sistem

Sistem chatbot dan AI Agent berhasil diimplementasikan pada platform e-commerce PT Radhika Narya Daruna. Implementasi antarmuka (User Interface) dibagi menjadi dua segmen operasional utama:

a) Antarmuka Layanan Pelanggan (Front-Office): Widget chatbot dirancang dengan antarmuka yang intuitif untuk menerima instruksi bahasa alami dari pelanggan. Hasil eksekusi backend (menggunakan NLP dan Gemini API) berhasil divisualisasikan dengan baik, di mana sistem mampu merender komponen keranjang belanja secara interaktif di layar obrolan untuk memfasilitasi transaksi secara langsung.

b) Antarmuka AI Agent (Back-Office): Dasbor administrator dilengkapi dengan widget AI Agent otonom berbasis teks. Fitur ini terbukti mampu mengeksekusi payload perintah bahasa alami menggunakan teknologi Function Calling untuk memodifikasi basis data (seperti ketersediaan stok) secara real-time.

4.2. Hasil Pengujian White Box

Untuk memvalidasi kebenaran logika struktural dari peladen *backend* (Node.js), dilakukan pengujian *White-Box* menggunakan metode *Basis Path Testing*. Evaluasi dilakukan dengan menghitung nilai *Cyclomatic Complexity* ($V(G) = E - N + 2$) pada tiga modul krusial guna memastikan seluruh percabangan komputasi tereksekusi tanpa cacat logika (*bug*).

1. Pengujian Modul *Gatekeeper (Cosine Similarity)* Berdasarkan ekstraksi *flowgraph*, diperoleh 5 *Edge* dan 5 *Node*, sehingga menghasilkan kompleksitas $V(G)$ = 5 - 5 + 2 = 2. Hasil eksekusi 2 jalur independen membuktikan bahwa sistem secara valid memanggil Gemini API pada kueri dengan skor $\geq 0{,}2$ dan sukses memotong jalur API untuk langsung memberikan balasan penolakan statis pada skor $< 0{,}2$.
2. Pengujian Modul *RBAC Middleware* Modul keamanan otorisasi *AI Agent* ini memiliki 8 *Edge* dan 7 *Node*, menghasilkan nilai $V(G)$ = 8 - 7 + 2 = 3. Hasil dari 3 skenario uji mengonfirmasi bahwa peladen berhasil menolak *request* tanpa *token* (HTTP 401), memblokir *token* dengan peran selain Admin (HTTP 403), dan secara sah meneruskan eksekusi eksklusif untuk *token* khusus Admin.
3. Pengujian Modul Ekstraksi Token Produk (*Regex*) Modul yang bertugas memindai token produk dari balasan Gemini AI ini memiliki 5 *Edge* dan 5 *Node*, menghasilkan nilai $V(G)$ = 5 - 5 + 2 = 3 Hasil pengujian menunjukkan bahwa sistem secara valid mampu mengekstrak *ID Product* ke dalam variabel terpisah, serta terbukti aman (tidak *crash*) dengan mengembalikan nilai *null* saat AI memberikan balasan percakapan biasa tanpa token.

5. KESIMPULAN

Berdasarkan hasil analisis, perancangan, implementasi, hingga pengujian sistem *chatbot* pada PT Radhika Narya Daruna yang telah diuraikan pada bab-bab sebelumnya, maka dapat ditarik beberapa kesimpulan sebagai berikut:

1. Efisiensi Layanan Pelanggan melalui Arsitektur RAG: Penerapan algoritma *Term Frequency-Inverse Document Frequency* (TF-IDF) dan *Cosine Similarity* yang dipadukan dengan Gemini API dalam kerangka *Retrieval-Augmented Generation* (RAG) terbukti berhasil mengotomatisasi layanan pelanggan. Sistem *Gatekeeper* dengan ambang batas (*threshold*) kemiripan 0,2 mampu secara presisi memfilter pertanyaan di luar konteks, sehingga respons kecerdasan buatan selalu relevan, bebas dari halusinasi (*hallucination*), dan secara signifikan mengurangi beban kerja manual pada bagian *Customer Service*.
2. Optimalisasi Transaksi via *Interactive Sales Assistant*: Integrasi *chatbot* dengan basis data inventaris MySQL secara *real-time* berhasil mewujudkan asisten penjualan yang interaktif. Analisis ekstraktif menggunakan metode *Regular Expression* (Regex) pada peladen (*backend*) terbukti efektif dalam

memicu kemunculan tombol "Tambah ke Keranjang" secara dinamis di dalam ruang obrolan, sehingga memperpendek alur transaksi dan memudahkan pelanggan dalam melakukan pembelian komoditas.

3. Otomatisasi Operasional melalui *AI Agent*: Implementasi teknologi *Function Calling* dari model bahasa besar (LLM) sukses menciptakan Agen Otonom (*AI Agent*) pada sisi *back-office*. Fitur ini memungkinkan administrator untuk memperbarui ketersediaan stok produk dan mengubah status pesanan hanya melalui instruksi bahasa alami. Penggunaan *middleware Role-Based Access Control* (RBAC) juga memastikan bahwa agen cerdas ini beroperasi dalam lingkungan otorisasi yang sangat aman.

Validitas Logika Sistem: Berdasarkan hasil pengujian *White-Box* menggunakan metode *Basis Path Testing* dan perhitungan *Cyclomatic Complexity*, terbukti bahwa struktur logika internal peladen (termasuk gerbang logika *Cosine Similarity*, validasi token *middleware* RBAC, dan pemindaian teks Regex) telah tereksekusi dengan baik tanpa cacat fungsional (*bug*).

DAFTAR PUSTAKA

D. Rusdianto and Y. Herdiana, "RANCANG BANGUN SISTEM INFORMASI INVENTORI MENGGUNAKAN FRAMEWORK LARAVEL 8 DI PT. CEMARA KWANGJIN TEKSTIL," 2023.

Nayla Sifaul Qolbi, Genduk Nurul Aini, and Helen Risky Dwi Wahyuni, "CHATBOT PADA ECOMMERCE: PERAN CHATBOT DALAM MENINGKATKAN KONVERSI PENJUALAN, REKOMENDASI PRODUK, DAN LOYALITAS PELANGGAN," 2025.

R. Ranveera, A. Kesharwani, and S. Kumari, "NOVATEUR PUBLICATIONS INTERNATIONAL JOURNAL OF INNOVATIONS IN ENGINEERING RESEARCH AND TECHNOLOGY [IJIERT] CUSTOMER SUPPORT CHATBOT USING NATURAL LANGUAGE PROCESSING," *Website: ijiert.org*, vol. 8, 2021.

A. P. Rizaldy, S. Riadi, and N. Wijaya, "Peran Chatbot Ai Dalam Mengotomatiskan Layanan Pelanggan Dan Meningkatkan Efisiensi Operasional E-Commerce," vol. 6, no. 1, 2025.

K. G. Krisnanda, W. Gede, N. [2] Rifky, and L. Rahardian[3], "Penerapan E-Katalog Sebagai Media Promosi Pada PT. Satria Pangan Sejati," *MAVIB Journal*, vol. 4, no. 2, p. 2023.

L. Indahsari, K. Kusnadi, and T. E. Putri, "Rancang Bangun LINE Chatbot Informasi dan Edukasi Kesehatan Mental Menggunakan Algoritma Jaro Winkler," *Jurnal Eksplora Informatika*, vol. 10, no. 2, pp. 68–79, Mar. 2021, doi: 10.30864/eksplora.v10i2.428.

V. Meida Hersianty, E. Larasati Amalia, D. Puspitasari, and D. Wahyu Wibowo, "PENERAPAN ALGORITMA TF-IDF DAN COSINE SIMILARITY DALAM SISTEM REKOMENDASI LOWONGAN PEKERJAAN," 2025.

S. I. Adam and W. Grivin, "IMPLEMENTASI SISTEM REKOMENDASI PRODUK E-COMMERCE MENGGUNAKAN CONTENT-BASED FILTERING BERBASIS COSINE SIMILARITY," vol. 10, no. 2, 2025.

K. Rizqy Putri Nugroho, C. M. Joy Sihombing, and F. Ihda Husni Husodo, "Meninjau Perkembangan Teknologi NLP dan LLM dalam Meningkatkan Kualitas Percakapan serta Pemahaman Konteks oleh Chatbot," *Jurnal Profesi Insinyur Universitas Lampung (JPI)*, vol. 6, no. 2, pp. 2722–5771, doi: 10.23960/jpi.v6n2.274.

L. Wang *et al.*, "A survey on large language model based autonomous agents," Dec. 01, 2024, *Higher Education Press Limited Company*. doi: 10.1007/s11704-024-40231-1.

Rizal Firmansyah, A. Hamdi, and D. Riandini, "PERANCANGAN SISTEM AUTENTIKASI MULTI-ROLE BERBASIS RBAC PADA PLATFORM E-LEARNING PEMBERDAYAAN EKONOMI PEREMPUAN," *Rabit : Jurnal Teknologi dan Sistem Informasi Univrab*, vol. 11, no. 1, pp. 1094–1104, Jan. 2026, doi: 10.36341/rabit.v11i1.7226.

T. Schick *et al.*, "Toolformer: Language Models Can Teach Themselves to Use Tools," Feb. 2023, [Online]. Available: http://arxiv.org/abs/2302.04761

C. Lusia Kiareni *et al.*, "IMPLEMENTASI FITUR DESKRIPSI PRODUK BERBASIS API GEMINI DALAM PENGEMBANGAN E-COMMERCE BERBASIS MOBILE MENGGUNAKAN FRAMEWORK FLUTTER," 2024.

M. S. A. Setiawan, A. Pratama, and P. K. F. Ananto, "Performance Analysis of Reasoning Models in RAG-Based Question Answering System for University Admission Services," *bit-Tech*, vol. 8, no. 3, pp. 3789–3798, Apr. 2026, doi: 10.32877/bt.v8i3.3707.

R. Fahreji and R. S. Wijaya, "Development of BAPOLAIC: AI chatbot for optical character recognition based-document extraction and voice assistant," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 16, no. 2, p. 1002, Apr. 2026, doi: 10.11591/ijece.v16i2.pp1002-1009.