JISTech (Journal of Islamic Science and Technology)

ISSN: 2528-5718

Vol. 11, No. 2, 2026. Page: 158-163

http://jurnal.uinsu.ac.id/index.php/jistech

Implementasi Sistem Tanya Jawab Dokumen Regulasi Berbasis RAG dengan Pencarian Semantik dan Normalisasi Typo Levenshtein

Syahira Elfiandani Nasution[1*], M. Fakhriza[2]

[1,2] *Universitas Islam Negeri Sumatera Utara, Medan, Indonesia*

ABSTRAK

Penelitian ini mengimplementasikan sistem tanya jawab dokumen regulasi berbasis Retrieval-Augmented Generation (RAG) yang memadukan pencarian semantik berbasis cosine similarity dengan normalisasi kesalahan ketik menggunakan Levenshtein distance. Dokumen PDF diekstraksi per halaman, dipotong hingga 3.000 karakter, direpresentasikan menggunakan model gemini-embedding-001 dengan dimensi keluaran 768, dan disimpan sebagai vektor JSON pada MySQL. Kueri pengguna dinormalisasi menggunakan kamus dinamis dari korpus, kemudian kandidat halaman diperingkat berdasarkan cosine similarity, keyword boost 0,25, dan fallback pencarian LIKE ketika skor tertinggi kurang dari 0,2. Konteks terpilih diteruskan ke Gemini API untuk menghasilkan jawaban beserta nama dokumen dan nomor halaman. Pengujian fungsional menunjukkan bahwa autentikasi, unggah dan ekstraksi dokumen, pembentukan embedding, retrieval, generasi jawaban berbasis konteks, dan penyajian rujukan berjalan sesuai rancangan. Simulasi perhitungan memperlihatkan cara keyword boost mengubah urutan kandidat dan Levenshtein distance menormalkan "pasword" menjadi "password". Temuan ini menunjukkan kelayakan implementasi pipeline, tetapi belum membuktikan peningkatan akurasi retrieval karena belum tersedia dataset pertanyaan berlabel, baseline, ablation, dan evaluasi kuantitatif.

ABSTRACT

This study implements a regulatory document question-answering system based on Retrieval-Augmented Generation (RAG), combining cosine-similarity-based semantic retrieval with Levenshtein-distance-based spelling-error normalization. PDF documents are extracted at the page level, truncated to 3,000 characters, represented using gemini-embedding-001 with an output dimensionality of 768, and stored as JSON vectors in MySQL. User queries are normalized against a dynamic corpus-derived dictionary, after which candidate pages are ranked using cosine similarity, a 0.25 keyword boost, and a LIKE-search fallback when the highest score is below 0.2. The selected context is passed to the Gemini API to generate an answer accompanied by the document name and page number. Functional testing indicates that authentication, document upload and extraction, embedding generation, retrieval, context-grounded answer generation, and source presentation operate as designed. Illustrative calculations show how keyword boosting changes candidate ranking and how Levenshtein distance normalizes "pasword" to "password". These findings establish implementation feasibility but do not demonstrate improved retrieval accuracy because no labeled question set, baseline, ablation study, or quantitative evaluation is available.

Kata Kunci: Retrieval-Augmented Generation; Pencarian Semantik; Cosine Similarity; Levenshtein Distance; Normalisasi
Email: * syahiranst49@gmail.com

DOI: http://dx.doi.org/10.30829/jistech.v11i2.31523

Pendahuluan

Large Language Models (LLM) berbasis arsitektur Transformer telah memperluas kemampuan sistem dalam memahami dan menghasilkan bahasa alami [1], [2]. Namun, model generatif tetap berisiko menghasilkan informasi yang tidak didukung sumber, terutama ketika pertanyaan berkaitan dengan dokumen privat, pengetahuan setelah masa pelatihan, atau istilah yang sangat kontekstual [3], [4], [5]. Risiko tersebut menjadi penting pada dokumen regulasi karena jawaban perlu dapat ditelusuri kembali ke sumber dan halaman tertentu.

Retrieval-Augmented Generation (RAG) mengurangi ketergantungan pada pengetahuan parametrik dengan mengambil konteks eksternal sebelum proses generasi [6]. Keandalan RAG sangat ditentukan oleh retriever karena konteks yang tidak relevan dapat menghasilkan jawaban yang tidak tepat meskipun generator memiliki kemampuan

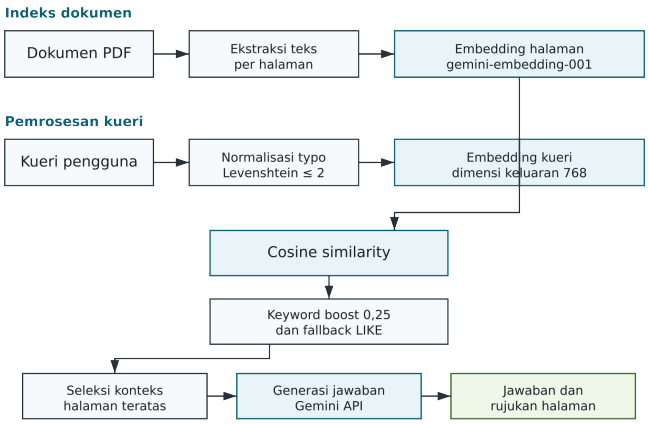
bahasa yang baik [3]. Representasi embedding kontekstual, termasuk pendekatan berbasis BERT dan Sentence-BERT, memungkinkan halaman dan kueri dibandingkan berdasarkan kedekatan semantik [1], [7].

Pencarian semantik dapat melewatkan istilah teknis, nomor pasal, singkatan, atau variasi ejaan yang penting dalam dokumen regulasi. Sebaliknya, Levenshtein distance mengukur jumlah minimum operasi penyisipan, penghapusan, dan penggantian karakter sehingga sesuai untuk menormalkan typo, tetapi tidak menangkap kesamaan makna [8]. Kombinasi beberapa sinyal retrieval berpotensi membantu pencarian [9], tetapi setiap komponen harus dijelaskan sesuai fungsi aktualnya dan diuji melalui evaluasi yang terkontrol.

Penelitian ini mengimplementasikan pipeline RAG pada aplikasi Laravel yang menggunakan unit retrieval per halaman, cosine similarity untuk pemeringkatan semantik, Levenshtein distance untuk normalisasi typo, keyword boost untuk istilah tertentu, serta fallback LIKE ketika skor semantik rendah. Kontribusi penelitian dibatasi pada perancangan pipeline, transparansi konfigurasi retrieval, dan verifikasi fungsional sistem. Evaluasi menggunakan dataset berlabel, baseline, ablation, serta metrik retrieval dan kualitas jawaban sebagaimana direkomendasikan dalam evaluasi RAG [10] belum dilakukan, sehingga penelitian ini tidak mengklaim superioritas kinerja terhadap metode lain.

Metodologi Penelitian

Penelitian menggunakan desain pengembangan sistem dengan verifikasi fungsional. Unit analisis retrieval adalah halaman dokumen, sedangkan unit masukan adalah kueri pengguna. Desain ini digunakan untuk memeriksa apakah seluruh komponen pipeline dapat dijalankan secara terintegrasi, bukan untuk menguji peningkatan akurasi terhadap baseline.



Gambar 1. Pipeline pencarian semantik, normalisasi typo, dan generasi jawaban berbasis RAG

Korpus berupa dokumen regulasi berformat PDF yang diunggah ke aplikasi sebagai knowledge base. Setiap dokumen diekstraksi per halaman, dibersihkan dari spasi berlebih, dan dipotong hingga 3.000 karakter sebelum dikirim ke endpoint embedding. Vektor disimpan sebagai JSON pada MySQL bersama nama dokumen dan nomor halaman. Jumlah dokumen, total halaman, dan jumlah kueri uji perlu dilaporkan pada versi akhir agar cakupan data dapat dinilai.

Pada tahap indeksasi, teks halaman direpresentasikan menggunakan gemini-embedding-001 dengan output dimensionality 768. Pada tahap kueri, setiap token dibandingkan dengan kamus dinamis yang dibentuk dari korpus; kandidat dengan Levenshtein distance 1 sampai 2 digunakan untuk normalisasi typo. Kueri hasil normalisasi kemudian diubah menjadi embedding dan dibandingkan dengan seluruh halaman menggunakan cosine similarity.

Skor kandidat ditetapkan sebagai Score(page) = CosineSim(Q,Dpage) + Boost(page), dengan Boost sebesar 0,25 ketika halaman memuat kata kunci yang telah dipetakan dan 0 untuk kondisi lainnya. Jika skor terbaik kurang dari 0,2, sistem menjalankan fallback LIKE terhadap teks halaman. Sistem memilih maksimum tiga halaman per dokumen dan lima dokumen teratas, lalu mengirim konteks tersebut bersama kueri ke gemini-3.6-flash dengan timeout 120 detik. Evaluasi dilakukan melalui skenario autentikasi, unggah dan ekstraksi dokumen, retrieval untuk kueri relevan, typo, dan tidak relevan, generasi jawaban, serta penyajian rujukan; evaluasi kuantitatif retrieval dan kualitas jawaban belum dilakukan.

Hasil dan Pembahasan

1. Implementasi Sistem

Sistem dikembangkan menggunakan Laravel dan MySQL dengan modul autentikasi, pengelolaan dokumen PDF, tanya jawab berbasis RAG, riwayat percakapan, dan compliance ISO 27001. Dokumen diekstraksi per halaman dan vektor embedding disimpan dalam kolom JSON pada tabel document_pages dan documents. Implementasi ini menunjukkan bahwa pipeline indeksasi, retrieval, dan generasi dapat diintegrasikan dalam aplikasi web.

2. Implementasi Algoritma

Mekanisme retrieval terdiri atas embedding, cosine similarity, normalisasi typo, keyword boost, fallback LIKE, dan seleksi konteks.

a. Vector Embedding

Embedding dihasilkan melalui endpoint gemini-embedding-001:embedContent dengan output dimensionality 768. Teks halaman dipotong hingga 3.000 karakter, sedangkan pemanggilan API menggunakan maksimum tiga percobaan dengan jeda satu detik. Nilai 768 merupakan konfigurasi keluaran yang dipilih sistem, bukan satu-satunya dimensi yang didukung model.

$$\text{CosineSim}(Q,D) = (Q \cdot D) / (||Q|| \times ||D||)$$

Cosine similarity secara umum berada pada rentang -1 sampai 1; nilai yang lebih tinggi menunjukkan arah vektor yang lebih serupa. Dalam implementasi, nilai digunakan untuk mengurutkan halaman, bukan sebagai probabilitas relevansi.

b. Levenshtein Distance untuk Normalisasi Typo

Levenshtein distance dihitung antara token kueri dan kata dalam kamus dinamis korpus. Token diganti dengan kandidat terdekat ketika jarak edit bernilai 1 atau 2. Aturan ini membantu menangani typo, tetapi berisiko mengubah istilah pendek atau nama khusus secara keliru sehingga memerlukan pengujian tambahan dan aturan pengecualian.

$$M[i,j] = \min\{M[i-1,j]+1; M[i,j-1]+1; M[i-1,j-1]+\text{cost}(i,j)\},$$
dengan $\text{cost}(i,j)=0$ jika karakter sama dan 1 jika berbeda.

Untuk contoh “pasword” dan “password”, jarak edit minimum adalah 1 melalui penyisipan satu huruf “s”; karena nilai tersebut tidak melebihi ambang 2, sistem menormalkan token menjadi “password”.

c. Strategi Hybrid Retrieval

Setelah normalisasi typo, sistem memeringkat halaman berdasarkan cosine similarity dan keyword boost. Fallback LIKE merupakan jalur kondisional ketika skor terbaik kurang dari 0,2, sehingga Levenshtein dan LIKE tidak menjadi komponen aditif dalam rumus skor.

Boost(page) bernilai 0,25 ketika halaman mengandung kata kunci yang dipetakan dan 0 pada kondisi lainnya. Ambang dan bobot ini bersifat heuristik karena belum ditentukan melalui validasi pada development set.

$$\text{Score}(page) = \text{CosineSim}(Q, D\_page) + \text{Boost}(page)$$

Tabel 1 menyajikan simulasi tiga kandidat untuk memperlihatkan cara keyword boost mengubah urutan, bukan hasil evaluasi pada dataset berlabel.

Kueri ilustratif yang digunakan adalah “Bagaimana cara mengganti password pegawai?”.

Halaman A (Halaman 5 dokumen SK PNS): CosineSim = 0,72 ; mengandung kata 'sandi' → Boost = 0,25.

Halaman B (Halaman 8 dokumen SK PNS): CosineSim = 0,68 ; tidak mengandung kata kunci → Boost = 0.

Halaman C (Halaman 2 dokumen Pedoman CUTI): CosineSim = 0,55; mengandung kata “login”; Boost = 0,25.

Perhitungan skor akhir masing-masing kandidat:

Score(A) = 0,72 + 0,25 = 0,97

Score(B) = 0,68 + 0 = 0,68

Score(C) = 0,55 + 0,25 = 0,80

Urutan kandidat dari skor tertinggi adalah A (0,97), C (0,80), B (0,68). Karena skor terbaik (0,97) lebih besar dari ambang batas fallback 0,2, mekanisme LIKE tidak diaktifkan. Hasil urutan ini kemudian dikelompokkan per dokumen dengan ambang maksimal 3 halaman per dokumen dan 5 dokumen teratas untuk memastikan keragaman sumber referensi.

Tabel 1. Simulasi Skor Retrieval dengan Keyword Boost

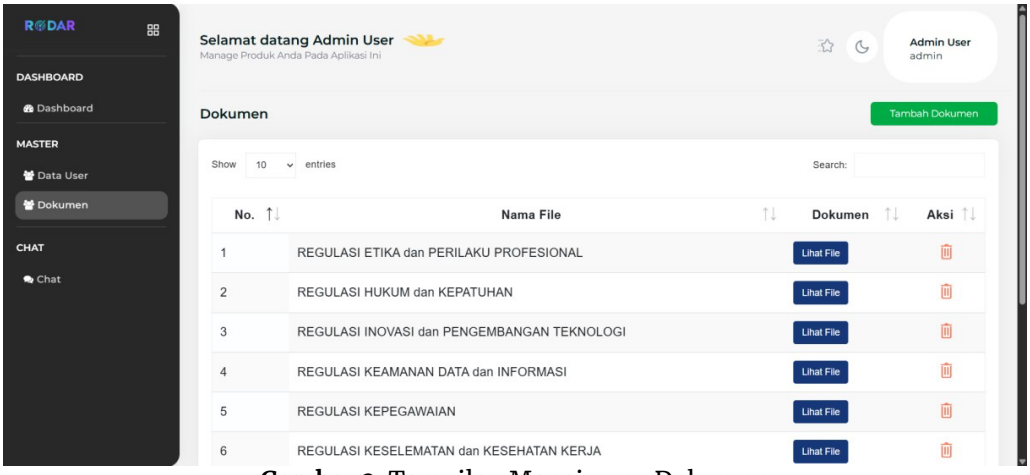
Halaman	Cosine Similarity	Boost	Skor Akhir	Urutan
A (Halaman 5, SK PNS)	0,72	0,25	0,97	1
B (Halaman 8, SK PNS)	0,68	0,00	0,68	3
C (Halaman 2, Pedoman CUTI)	0,55	0,25	0,80	2

d. Prompt Engineering dan Generasi Jawaban

Prompt terdiri atas peran asisten, blok konteks yang memuat nama file dan halaman, serta instruksi untuk menjawab hanya berdasarkan konteks dalam bahasa Indonesia formal. Generator menggunakan gemini-3.6-flash dengan timeout 120 detik, kemudian keluaran diformat menjadi HTML dan disertai rujukan dokumen. Parameter

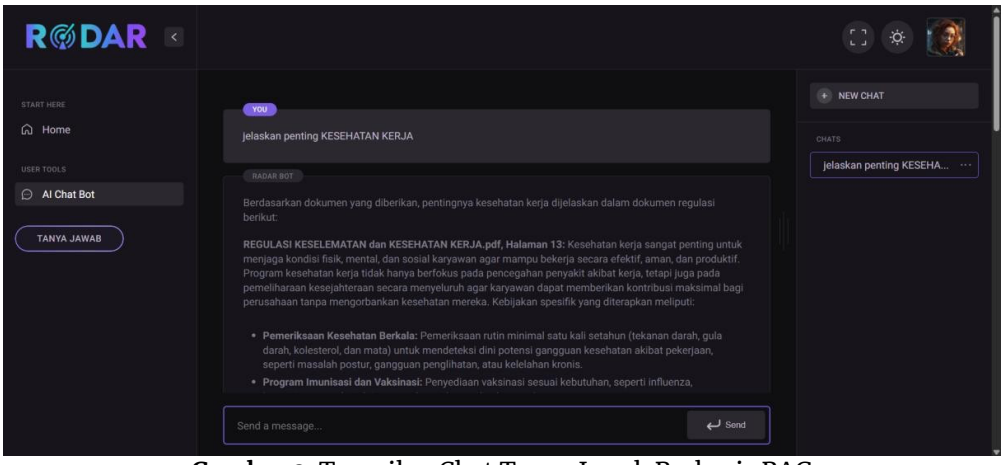
generasi lain perlu dicatat pada versi akhir untuk memastikan reprodusibilitas.

3. Tampilan dan Pengujian Sistem



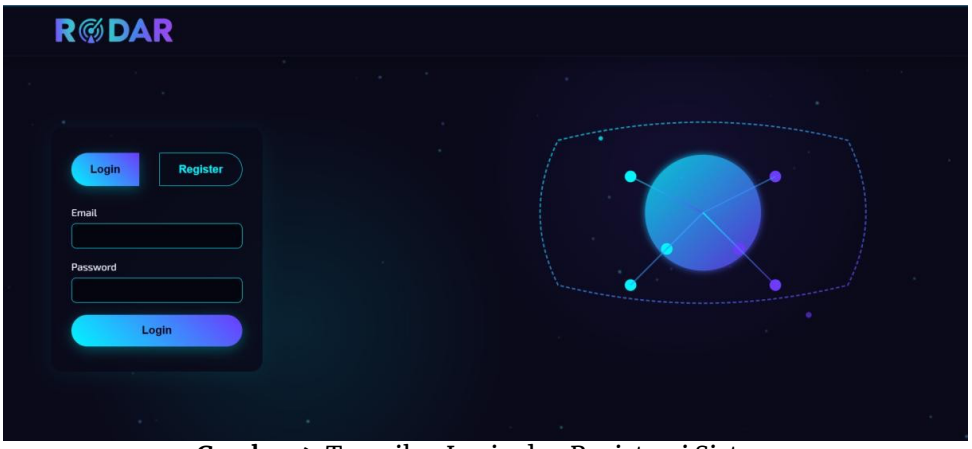
Gambar 2. Tampilan Manajemen Dokumen

Gambar 2 menunjukkan modul manajemen dokumen yang digunakan untuk mengunggah dan mengelola dokumen PDF sebagai basis pengetahuan. Dokumen yang telah diunggah menjadi sumber ekstraksi teks dan pembentukan embedding.



Gambar 3. Tampilan Chat Tanya Jawab Berbasis RAG

Gambar 3 menunjukkan interaksi pengguna dengan sistem tanya jawab. Respons sistem menampilkan jawaban berbasis konteks dokumen dan referensi nama dokumen serta halaman sumber.



Gambar 4. Tampilan Login dan Registrasi Sistem

Konteks halaman terpilih dimasukkan ke dalam prompt bersama kueri pengguna. Gambar 3 menunjukkan bahwa sistem menampilkan jawaban, nama dokumen, dan nomor halaman sehingga pengguna dapat menelusuri sumber respons.

Hasil pengujian fungsional dirangkum pada Tabel 2. Status "Sesuai" menunjukkan bahwa keluaran yang diamati memenuhi perilaku yang dirancang pada skenario pengujian, tetapi tidak menunjukkan akurasi retrieval atau factual correctness jawaban.

Secara fungsional, cosine similarity menyediakan pemeringkatan semantik, sedangkan Levenshtein distance menormalkan variasi karakter sebelum retrieval. Keyword boost dapat mengubah urutan kandidat pada simulasi, dan fallback LIKE menyediakan jalur pencarian ketika skor semantik rendah. Tanpa baseline dan ablation, kontribusi individual setiap komponen terhadap kinerja belum dapat ditentukan.

Keterbatasan utama penelitian adalah tidak adanya dataset pertanyaan berlabel, baseline, ablation, dan metrik retrieval seperti Recall@k, MRR, atau nDCG. Kualitas jawaban juga belum dievaluasi dengan faithfulness, answer relevance, atau penilaian manusia. Selain itu, unit retrieval per halaman dan pemotongan 3.000 karakter dapat menghilangkan konteks, koreksi typo berbasis kamus korpus berpotensi salah mengganti istilah, serta bobot boost 0,25 dan ambang fallback 0,2 masih bersifat heuristik.

Tabel 2. Ringkasan Pengujian Fungsional

Skenario	Masukan/Kondisi	Keluaran yang Diamati	Status
Autentikasi	Akun aktif dan tidak aktif	Akun tidak aktif ditolak oleh sistem	Sesuai
Dokumen	Unggah dokumen PDF	Teks per halaman dan embedding tersimpan	Sesuai
Tanya jawab	Kueri relevan, typo, dan tidak relevan	Retrieval, fallback, jawaban, dan rujukan diproses	Sesuai
Compliance	Permintaan analisis ISO	Keluaran terstruktur dengan rujukan halaman	Sesuai

Kesimpulan

Penelitian ini menghasilkan implementasi sistem tanya jawab dokumen regulasi berbasis RAG yang menjalankan ekstraksi per halaman, embedding, pemeringkatan cosine similarity, normalisasi typo Levenshtein, keyword boost, fallback LIKE, generasi jawaban melalui Gemini API, dan penyajian rujukan dokumen. Pengujian fungsional menunjukkan bahwa komponen utama beroperasi sesuai rancangan, sedangkan simulasi perhitungan menjelaskan perilaku skor dan koreksi typo. Hasil tersebut mendukung kelayakan implementasi, tetapi belum membuktikan peningkatan akurasi retrieval atau kualitas jawaban. Validasi berikutnya harus menggunakan korpus dan kueri berlabel, baseline semantic retrieval, ablation setiap komponen, metrik Recall@k, MRR, atau nDCG, serta evaluasi faithfulness dan answer relevance agar manfaat metode dapat ditentukan secara empiris.

Daftar Pustaka

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proc. NAACL-HLT, pp. 4171–4186, 2019.

[2] A. Vaswani, N. Shazeer, N. Parmar, et al., "Attention Is All You Need," in Advances in Neural Information Processing Systems 30, 2017.

[3] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, and H. Wang, "Retrieval-Augmented Generation for Large Language Models: A Survey," arXiv preprint arXiv:2312.10997, 2023.

[4] B. Ni, Z. Liu, L. Wang, Y. Lei, Y. Zhao, X. Cheng, Q. Zeng, L. Dong, Y. Xia, Y. Fan, et al., "Towards Trustworthy Retrieval Augmented Generation for Large Language Models: A Survey," arXiv preprint arXiv:2502.06872, 2025.

[5] I. K. W. Adnyana, R. T. Tayane, F. Fahmi, F. Wicaksono, B. Nugraha, and E. Y. Ali, "Mitigating Hallucinations in Large Language Models via Retrieval Augmented Generation: A Systematic Review of n8n-Based Implementations," EDUKASIA, vol. 7, no. 1, pp. 605–618, 2026.

[6] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in Advances in Neural Information Processing Systems 33, 2020.

[7] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in Proc. 2019 Conf. Empirical Methods in Natural Language Processing, 2019.

[8] J. Pardede and A. Yudistira, "Comparison of Cosine Similarity, Rabin-Karp, and Levenshtein Distance Algorithms for Plagiarism Detection in Document," Ultimatics: Jurnal Teknik Informatika, vol. 17, no. 1, pp. 63–71, 2025.

[9] J. Juvekar and A. Purwar, "COS-Mix: Cosine Similarity and Distance Fusion for Improved Information Retrieval," arXiv preprint arXiv:2406.00638, 2024.

[10] S. Es, J. James, L. Espinosa-Anke, and S. Schockaert, "RAGAS: Automated Evaluation of Retrieval Augmented Generation," arXiv preprint arXiv:2309.15217, 2023.