

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Dalam beberapa tahun terakhir penelitian terkait penggunaan chatbot berbasis NLP untuk mendukung kehidupan sehari-hari telah banyak dikaji terutama dalam bidang pendukung layanan informasi baik dalam komunikasi pendidikan hingga dalam industri. Tabel 2.1 berikut merangkum beberapa penelitian terkait penggunaan chatbot berbasis NLP yang menjadi dasar konseptual dalam penelitian ini. Berikut adalah penelitian terdahulu yang menjadi kajian untuk penelitian ini, antara lain:

Tabel 2. 1 Penelitian Terdahulu

No	Judul dan Peneliti	Nama Jurnal	Metode	Hasil Penelitian
1	<i>Chatbot helpdesk design for digital customer service;</i> Larasati, P; Irawan, A; Anwar, S; Mulya, M; Dewi, M; Nurfatima, I; [3]	<i>Applied Engineering and Technology</i>	<i>H3I</i>	Meningkatkan tanggapan customer service dalam perusahaan dengan menggunakan teknologi chatbot AI berjenis H3I yang diimplementasikan dalam platform pesan instant WhatsApps, Line, Telegram. Hasil yang didapatkan adalah chatbot ini bisa diimplementasikan dalam WhatsApp untuk layanan pelanggan.
2	<i>Analyzing and Tracking Student Educational Program Interests on Social Media with</i>	<i>International Journal of Interactive</i>	<i>CRISP-DM, Naïve Bayes,</i>	Memaparkan aplikasi chatbot untuk menyediakan

No	Judul dan Peneliti	Nama Jurnal	Metode	Hasil Penelitian
	<i>Chatbots Platform and Text Analytics;</i> Nasa-Ngium, P; Nuankaew, W; Nuankaew, P; [10]	<i>Mobile Technologies</i>	<i>KNN, Neural Network</i>	informasi Pendidikan bagi mahasiswa perguruan tinggi menggunakan Teknik data mining dan machine learning yaitu Naïve bayes, KNN, dan Neural Network dan didapatkan hasil akurasi sebesar 88.73%, dan rata-rata 3,97 untuk kepuasan aplikasi.
3	<i>A comparative study of retrieval-based and generative-based chatbots using Deep Learning and Machine Learning;</i> Pandey, S; Sharma, S; [13]	<i>Healthcare Analytics ScienceDirect</i>	<i>RNN, LSTM, Bi-LSTM, GRU, CNN</i>	Membandingkan metode Teknik Deep Learning dan Machine Learning dalam pengembangan chatbot berbasis retrieval-based dan generative-based. Pada model retrieval-based didapatkan Bi-LSTM dengan nilai akurasi tertinggi yaitu hingga 91,57% dan model generative-based dengan desain encoder-decoder mendapatkan akurasi sebesar 94,45%.
4	<i>A comparison of chatbot platforms with the state-of-the-art sentence BERT for answering online student FAQs;</i>	<i>Results in Engineering ScienceDirect</i>	<i>SBERT, Google Dialogflow, Microsoft QnA Maker</i>	Chatbot layanan bantuan mahasiswa baru berdasarkan FAQ yang dapat

No	Judul dan Peneliti	Nama Jurnal	Metode	Hasil Penelitian
	Peyton, K; Unnikrishnan, S [14]			memberikan informasi secara cepat menggunakan metode SBERT dengan hasil F1-score sebesar 0.99.
5	<i>Semantic FAQ Chatbot Using SBERT (Sentence-BERT) and Cosine Similarity for Academic Services;</i> Holis, R; Utomo, P; Hutabarat, B; [15]	<i>Brilliance: Research of Artificial Intelligence</i>	<i>SBERT, Cosine Similarity</i>	Mengembangkan sistem chatbot FAQ berbasis semantic dengan menggunakan SBERT dan Cosine Similarity dan mendapatkan hasil akurasi sebesar 79,2%, presisi 81,7%, recall 96,3%, dan skor F1 sebesar 88,4%.
6	<i>Dice Similarity and TF-IDF for New Student Admissions Chatbot;</i> Prasetya, M; Priyatno, A; [18]	<i>Journal of Artificial Intelligence and Digital Business (RIGGS)</i>	<i>TFIDF</i>	Mengembangkan sistem chatbot berdasarkan NLP untuk membantu customer service penerimaan mahasiswa baru di Universitas dengan menggunakan metode TFIDF dengan hasil nilai recall sebesar 100% dan presisi mencapai 76,92%
7	<i>Enhancing Chatbot User Satisfaction: A Machine Learning Approach Integrating Decision Tree, TF-IDF, and BERTopic;</i> Jianxin Lu [19]	<i>IEEE 6th International Conference on Power, Intelligent Computing and</i>	<i>Integrating Decision Tree, TFIDF, BERTopic</i>	Menggabungkan model ensemble dari Decision Tree, TF-IDF, dan BERTopic untuk meningkatkan akurasi prediksi dan

No	Judul dan Peneliti	Nama Jurnal	Metode	Hasil Penelitian
		<i>Systems (ICPICS)</i>		menunjukkan peningkatan signifikan dalam akurasi dan metrik kepuasan pengguna dibandingkan model tradisional.
8	<i>HANA: An AI Chatbot for Islamic Jurisprudence on Menstruation using SBERT and TF-IDF;</i> Rafah Masuzzahra, T; Umam, K; Mustofa, H; Handayani, M; [20]	<i>Journal of Applied Informatics and Computing (JAIC)</i>	<i>TF-IDF, SBERT</i>	Mengusulkan dan mengimplementasikan HANA sebuah chatbot AI untuk Telegram dengan menggunakan gabungan model TF-IDF dan SBERT yang mendapatkan skor cosine similarity hingga 87% dan menunjukkan kepuasan pelanggan sebesar 86,7%.
9	<i>A Hybrid Ai-Based University Students Queries Chatbot Using NLP And SBERT Technologies;</i> Muhammad Hassan Zaidi, S; Ahmed, S; Husain, I; Hasan Qureshi, B; Ahmad Naz, S; Razaque, A; [21]	<i>Spectrum of Engineering Sciences</i>	<i>NLP, SBERT, TF-IDF</i>	Merancang chatbot yang digunakan untuk menangani pertanyaan mahasiswa terkait penerimaan, akademik, dan administrasi dengan menggunakan Teknik NLP yang menggabungkan SBERT, TF-IDF, dan Cosine Similarity yang menghasilkan model SBERT memiliki akurasi tertinggi dari model

No	Judul dan Peneliti	Nama Jurnal	Metode	Hasil Penelitian
				lainnya sehingga bisa untuk menangani pertanyaan mahasiswa secara efisien.
10	<i>Refilling Knowledge and Intention to Use Travel Chatbot: Indonesian Context;</i> Setiawan, j; Princes, E; [22]	<i>Advances in Economics, Business and Management Research</i>	<i>AI, Blockchain</i>	Pemanfaatan chatbot untuk travelling dengan mengintegrasikan artificial intelligent (AII) dan Blockchain yang didapatkan bahwa chatbot traveling memiliki respon positif dan akan sangat membantu untuk perjalanan yang optimal

Berdasarkan penelitian sebelumnya yang dirangkum pada tabel 2.1, dapat disimpulkan bahwa penerapan teknologi Artificial Intelligence (AI) dan *Natural Language Processing (NLP)* pada pengembangan chatbot telah berkembang pesat dan diaplikasikan di berbagai bidang seperti layanan pelanggan, pendidikan, administrasi akademik, travelling, hingga sistem keagamaan yang dibantu oleh chatbot.

Beragam pendekatan digunakan untuk selalu meningkatkan kinerja chatbot agar lebih akurat dan bisa membantu permasalahan yang ada baik dari sisi pencocokan kata (*lexical matching*) maupun pemahaman konteks semantik (*semantic understanding*). Metode *Term Frequency-Inverse Document Frequency (TF-IDF)* yang berbasis lexical terbukti sangat efektif untuk mengidentifikasi kesamaan antar pertanyaan berdasarkan kata kunci yang muncul. Pada penelitian oleh Prasetya dan Priyanto [18] menunjukkan bahwa dengan menggunakan TF-IDF chatbot mampu menghasilkan nilai recall sempurna dalam sistem chatbot penerimaan mahasiswa baru, yang mana menandakan kemampuan model ini dalam

menangkap kesamaan kata yang eksplisit antar kalimat. Tetapi dalam metode ini masih memiliki keterbatasan dalam memahami makna kalimat yang berbeda tapi masih memiliki arti yang sama secara semantik.

Untuk mengatasi hal ini, pada penelitian berikutnya banyak beralih ke model berbasis semantik seperti Sentence-BERT (SBERT). Model SBERT mampu merepresentasikan kalimat dalam bentuk vektor berdimensi tinggi sehingga dapat mengukur kesamaan makna antar kalimat. Dalam penelitian Peyton dan Unnikrishnan [14] membuktikan bahwa SBERT memiliki efektivitas dengan nilai *F1-score* sebesar 0.99 dalam menjawab FAQ mahasiswa, sedangkan penelitian oleh Holis dkk. [15] juga menunjukkan performa SBERT yang baik yaitu dengan mencapai nilai *recall* hingga 96,3%.

Pada penelitian terbaru dalam pengembangan chatbot adalah dengan menggabungkan pendekatan lexical dan semantic kedalam satu sistem yang disebut *hybrid retrieval*. Pada penelitian oleh Rafah Masuzzahra dkk. [20] dan penelitian oleh Muhammad Hassan dkk. [21] telah mengkombinasikan TF-IDF dan SBERT untuk meningkatkan akurasi pencarian jawaban chatbot. Dalam hasil penelitian ini menunjukkan bahwa model hybrid ini menghasilkan tingkat similarity dan kepuasan pengguna yang lebih tinggi dibandingkan dengan penggunaan model tunggal.

Meskipun demikian, penelitian terdahulu terdahulu pada tabel 2.1 didapatkan bahwa sebagian besar masih difokuskan pada konteks eksternal seperti layanan pelanggan dan sistem akademik. Belum banyak penelitian yang menerapkan pendekatan *hybrid* ini untuk kebutuhan internal perusahaan, khususnya dalam mendukung operasional sistem SAP yang memiliki kompleksitas tinggi dan memerlukan bantuan digital untuk membantu pengguna non-teknis. Maka karena itu, penelitian ini berupaya mengisi celah dalam *research gap* dengan mengembangkan chatbot helpdesk FAQ berbasis hybrid TF-IDF dan SBERT yang diimplementasikan di lingkungan PT Sampoerna Agro Tbk. Chatbot ini dirancang untuk memahami pertanyaan bebas (*free text*) dari semua pengguna SAP di

perusahaan dan memberikan jawaban yang paling relevan berdasarkan model kombinasi kemiripan dari leksikal dan semantic. Dengan menggunakan pendekatan ini, diharapkan sistem bisa memberikan solusi yang lebih efisien, akurat, dan adaptif terhadap kebutuhan operasional perusahaan, sekaligus menjadi kontribusi dalam penerapan *Big Data Analytics* dan NLP di sektor industri.

2.1.1 Posisi Penelitian Ini

Berdasarkan hasil kajian terhadap penelitian-penelitian terdahulu, penelitian ini menempati posisi yang berbeda dan mengisi celah penelitian (*research gap*) yang belum banyak dibahas pada studi sebelumnya. Sebagian besar penelitian terdahulu berfokus pada penerapan chatbot berbasis NLP dalam konteks eksternal seperti layanan pelanggan, pendidikan, dan administrasi akademik. Namun, belum ditemukan penelitian yang secara khusus mengembangkan chatbot berbasis *hybrid retrieval* untuk mendukung operasional sistem SAP ERP di lingkungan perusahaan Indonesia. Sistem SAP memiliki karakteristik domain yang kompleks, mencakup modul yang beragam serta penggunaan *transaction code (tcode)* dan terminologi teknis yang sulit dipahami oleh sebagian besar pengguna non-teknis. Kondisi ini menempatkan penelitian ini pada domain yang lebih khusus dibandingkan penelitian serupa sebelumnya.

Selain itu, penelitian ini juga berbeda dari sisi penggunaan bahasa. Sebagian besar penelitian terdahulu memanfaatkan data dan percakapan berbahasa Inggris, sedangkan konteks penggunaan SAP di perusahaan Indonesia umumnya melibatkan pertanyaan dalam Bahasa Indonesia, campuran Inggris–Indonesia, serta variasi penulisan yang tidak baku. Hal ini menjadikan tantangan tersendiri karena model harus mampu menangani teks bebas (*free text*) yang tidak selalu sesuai dengan format pertanyaan pada dokumen FAQ. Penelitian ini mengatasi tantangan tersebut dengan menggunakan SBERT versi multilingual dan melakukan normalisasi teks internal.

Perbedaan penting lainnya terletak pada arsitektur sistem yang digunakan. Penelitian ini mengimplementasikan model *hybrid retrieval* yang terdiri dari dua tahap utama, yaitu penyaringan awal menggunakan TF-IDF untuk memilih kandidat dokumen paling dekat secara leksikal, kemudian melakukan penyusunan ulang (*reranking*) menggunakan SBERT berdasarkan kesamaan semantik. Pendekatan *lexical filter* dan *semantic rerank* seperti ini belum banyak diterapkan secara eksplisit pada konteks SAP, terutama dengan penyesuaian bobot kombinasi skor sebesar $\alpha = 0.6$ untuk SBERT dan 0.4 untuk TF-IDF.

Dari sisi evaluasi, penelitian ini juga menawarkan kontribusi yang lebih komprehensif dibandingkan penelitian terdahulu. Evaluasi tidak hanya menggunakan satu metrik sederhana, melainkan melibatkan *Top-K Accuracy*, *Recall@5*, *Mean Reciprocal Rank (MRR)*, serta pengukuran *latency* atau waktu respon model, yang sangat relevan untuk konteks operasional sistem enterprise. Selain itu, dilakukan juga uji coba pengguna internal untuk menilai pengalaman penggunaan dan kesesuaian jawaban dalam skenario penggunaan nyata.

Dengan demikian, penelitian ini menempati posisi strategis sebagai penelitian yang mengimplementasikan *hybrid retrieval* NLP pada sistem SAP ERP internal Indonesia, dengan pipeline yang lebih terstruktur serta evaluasi yang lebih menyeluruh dibanding penelitian terdahulu.

2.2 Teori yang digunakan

2.2.1 Chatbot

Chatbot merupakan program komputer yang dirancang untuk berinteraksi dengan manusia melalui pesan teks maupun suara [23]. Teknologi ini memungkinkan pengguna memperoleh informasi atau layanan secara otomatis tanpa harus berhubungan langsung dengan manusia [24]. Pada awalnya, chatbot hanya berbasis aturan sederhana (*rule-based*) dengan pencocokan kata kunci tertentu, namun seiring perkembangan teknologi kini

banyak yang sudah memanfaatkan Natural Language Processing (NLP) [11], [21]. Dengan adanya NLP, chatbot dapat memahami maksud percakapan, menafsirkan konteks, dan memberikan jawaban yang lebih relevan [15].

Integrasi dengan kecerdasan buatan (*Artificial Intelligence/AI*) juga membuat chatbot semakin cerdas karena mampu belajar dari interaksi sebelumnya dan menyesuaikan respon berdasarkan kebutuhan pengguna [25]. Di Indonesia, chatbot banyak digunakan dalam dunia bisnis, terutama untuk mendukung layanan pelanggan melalui aplikasi *internal*, *website*, maupun platform komunikasi seperti *WhatsApp Business* [2], [26]. Tujuan utamanya adalah memberikan layanan yang cepat, efisien, hemat biaya, dan tersedia sepanjang waktu. Selain itu, chatbot juga kerap difungsikan sebagai helpdesk digital yang membantu menjawab pertanyaan berulang, memberikan informasi produk, hingga mendukung proses transaksi [27].

Pemanfaatan chatbot tidak hanya meningkatkan efisiensi operasional perusahaan, tetapi juga menjaga konsistensi informasi yang diberikan kepada pelanggan serta meningkatkan pengalaman pengguna. Oleh karena itu, chatbot saat ini menjadi salah satu bagian penting dari strategi transformasi digital di berbagai sektor, mulai dari bisnis, pendidikan, kesehatan, perbankan, hingga layanan publik [28], [29], [30].

2.2.2 Natural Language Processing (NLP)

Natural Language Processing (NLP) merupakan cabang dari kecerdasan buatan (*Artificial Intelligence/AI*) yang berfokus pada interaksi antara komputer dan bahasa manusia [13]. Secara umum, NLP bertujuan untuk memungkinkan sistem komputer menganalisis, mengekstraksi, serta memahami informasi dari teks atau ucapan sehingga mampu menafsirkan maksud dan konsep yang terkandung dalam bahasa manusia [31].

Dalam prosesnya, NLP memiliki dua pendekatan utama, yaitu sintaksis dan semantik. Analisis sintaksis berfokus pada struktur bahasa, seperti tata bahasa, susunan kata, dan hubungan antar elemen dalam sebuah kalimat.

Sementara itu, analisis semantik menitikberatkan pada pemahaman makna yang terkandung dalam teks, sehingga sistem tidak hanya membaca kata secara literal tetapi juga mampu menangkap maksud dari sebuah pernyataan [20].

Selain itu, NLP memiliki turunan yang lebih spesifik, yaitu *Natural Language Understanding (NLU)* dan *Natural Language Generation (NLG)* [1]. NLU merupakan bidang yang lebih dalam dari NLP yang berfokus pada interpretasi input pengguna, sehingga sistem dapat mengenali maksud (intent) dan entitas yang relevan dari sebuah teks atau ucapan. NLU banyak digunakan pada agen interaktif, sistem edukasi, maupun perangkat elektronik seperti smartphone yang membutuhkan pemahaman bahasa alami untuk menjalankan perintah pengguna [32].

Sementara itu, NLG berfokus pada proses menghasilkan keluaran dalam bentuk bahasa alami yang dapat dipahami manusia. NLG digunakan untuk mengubah data mentah atau informasi terstruktur menjadi kalimat yang mudah dipahami. Aplikasi NLG dapat ditemukan pada chatbot berbasis voice recognition system, asisten virtual, hingga sistem rekomendasi yang memberikan penjelasan otomatis kepada pengguna [1].

Dengan kombinasi NLP, NLU, dan NLG, sistem berbasis percakapan seperti chatbot dapat berfungsi lebih efektif, interaktif, dan mendekati cara manusia berkomunikasi.

2.2.3 FAQ Helpdesk

Frequently Asked Questions (FAQ) merupakan kumpulan pertanyaan dan jawaban yang disusun berdasarkan masalah yang paling sering dihadapi oleh pengguna suatu sistem atau layanan. FAQ berfungsi sebagai basis pengetahuan (*knowledge base*) yang dapat digunakan untuk menjawab pertanyaan berulang secara cepat dan konsisten tanpa memerlukan intervensi langsung dari tim dukungan teknis.

Dalam konteks chatbot, FAQ digunakan sebagai sumber data utama dalam sistem retrieval-based chatbot. Ketika pengguna mengajukan

pertanyaan, sistem akan mencari kemiripan antara pertanyaan tersebut dengan data FAQ yang telah tersimpan untuk menemukan jawaban yang paling relevan. Dengan cara ini, chatbot mampu memberikan respons otomatis yang efisien dan mengurangi beban kerja petugas helpdesk.

Helpdesk sendiri merupakan layanan yang memberikan bantuan teknis kepada pengguna, baik terkait kendala penggunaan sistem, transaksi, maupun prosedur operasional. Dalam lingkungan *Enterprise Resource Planning* (ERP) seperti SAP, helpdesk berperan penting membantu pengguna memahami proses bisnis dan menyelesaikan masalah yang timbul selama penggunaan sistem. Namun, karena tingginya volume pertanyaan, layanan helpdesk konvensional sering kali tidak efisien dan memerlukan banyak sumber daya manusia.

Oleh karena itu, penerapan FAQ Helpdesk berbasis chatbot menjadi solusi efektif untuk meningkatkan efisiensi layanan dukungan teknis. Dengan mengintegrasikan data FAQ ke dalam sistem chatbot, pengguna dapat memperoleh jawaban secara mandiri, cepat, dan akurat. Selain meningkatkan kepuasan pengguna, penerapan ini juga membantu perusahaan mengoptimalkan proses operasional dan mengurangi beban kerja tim IT maupun helpdesk.

2.2.4 Evaluasi Information Retrieval Model

Karena penelitian ini menggunakan pendekatan retrieval-based, maka diperlukan metrik evaluasi yang dapat mengukur kemampuan sistem dalam menemukan kembali jawaban yang benar dari basis pengetahuan. Metrik-metrik yang digunakan dalam penelitian ini antara lain:

2.2.4.1 Top-K Accuracy

Top-K Accuracy merupakan salah satu metrik evaluasi yang paling umum digunakan dalam sistem *information retrieval* dan sistem rekomendasi untuk menilai kemampuan model dalam menghasilkan item relevan pada posisi teratas hasil pencarian. Dalam konteks chatbot

berbasis FAQ retrieval, *Top-K Accuracy* digunakan untuk mengukur apakah jawaban yang benar muncul dalam daftar K hasil teratas yang dikembalikan oleh sistem.

Secara konsep, *Top-K Accuracy* bernilai 1 apabila jawaban yang benar berada pada peringkat ke-1 hingga ke-K, dan bernilai 0 apabila jawaban benar berada di luar daftar tersebut. Misalnya, *Top-1 Accuracy* mengukur seberapa sering model menempatkan jawaban yang tepat pada peringkat pertama, sedangkan *Top-3 Accuracy* menilai seberapa sering jawaban yang benar muncul di posisi peringkat 1–3. Karena pengguna chatbot biasanya hanya memperhatikan beberapa respon teratas, metrik ini sangat relevan dalam menilai apakah chatbot dapat memberikan jawaban yang tepat tanpa harus ditelusuri terlalu jauh.

Keunggulan *Top-K Accuracy* adalah kesederhanaannya dalam memberikan gambaran performa praktis: semakin tinggi nilai *Top-K*, semakin baik kemampuan sistem dalam menyajikan jawaban akurat pada hasil awal pencarian. Dalam penelitian ini, *Top-K Accuracy* digunakan untuk membandingkan kinerja TF-IDF, SBERT, dan model *hybrid* dalam menampilkan jawaban yang benar pada posisi teratas sehingga dapat mencerminkan efektivitas model dalam skenario penggunaan nyata.

2.2.4.2 Recall@K

Recall@K merupakan metrik yang mengukur seberapa sering sistem berhasil "menangkap" atau menyertakan jawaban benar di antara K dokumen teratas yang diretrieval. Berbeda dengan *Top-K Accuracy* yang menilai posisi jawaban benar, *Recall@K* hanya peduli apakah jawaban itu muncul dalam K kandidat teratas, tanpa mempertimbangkan peringkat pastinya.

Pada tugas FAQ retrieval yang hanya memiliki satu jawaban benar untuk setiap pertanyaan, *Recall@K* bernilai 1 apabila jawaban yang

relevan muncul dalam daftar K dokumen hasil retrieval, dan bernilai 0 apabila tidak muncul sama sekali. Dengan demikian, metrik ini menggambarkan seberapa komprehensif model dalam menjangkau jawaban relevan pada ruang pencarian terbatas.

Recall@K sangat sesuai digunakan untuk mengukur kemampuan model TF-IDF dan SBERT dalam menghasilkan kandidat yang benar sebelum proses penyusunan ulang pada model *hybrid*. Secara praktis, nilai *Recall@K* yang tinggi menunjukkan bahwa sistem tidak "kehilangan" jawaban yang benar, bahkan ketika variasi bahasa pengguna berbeda dari format FAQ yang ada. Pada lingkungan enterprise seperti SAP ERP, hal ini penting agar chatbot dapat tetap memberikan rekomendasi relevan meskipun pertanyaan tidak identik secara kata.

2.2.4.3 Mean Reciprocal Rank (MRR)

Mean Reciprocal Rank (MRR) adalah metrik evaluasi yang digunakan untuk mengukur kualitas peringkat dalam sistem pencarian, yaitu seberapa tinggi posisi jawaban benar dalam daftar hasil retrieval. MRR menggunakan *reciprocal rank*, yaitu nilai kebalikan dari posisi ranking jawaban yang benar. Jika jawaban benar berada pada peringkat pertama, nilai reciprocal rank adalah 1; bila berada pada peringkat kedua bernilai 1/2, peringkat ketiga bernilai 1/3, dan seterusnya.

$$MRR = \frac{1}{N} \sum_{i=1}^N \frac{1}{r_i}$$

Rumus 1 Mean Reciprocal Rank

Dengan menghitung nilai rata-rata *reciprocal* rank untuk seluruh pertanyaan, MRR memberikan gambaran yang lebih detail mengenai kemampuan model dalam memprioritaskan jawaban yang benar di bagian atas hasil pencarian. Metrik ini jauh lebih informatif dibandingkan metrik biner seperti *Top-K Accuracy*, karena mampu membedakan antara

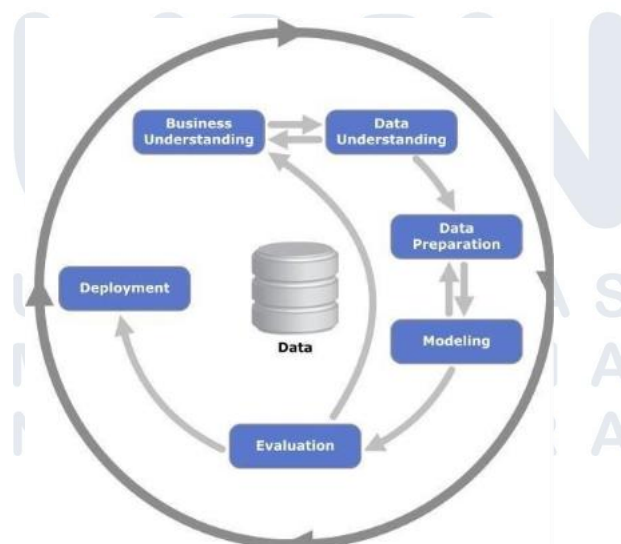
sistem yang selalu menempatkan jawaban benar di peringkat pertama dengan sistem yang sering menempatkan jawaban benar di urutan lebih rendah.

Dalam penelitian ini, penggunaan MRR sangat penting mengingat model hybrid melakukan dua tahap: *lexical filtering* menggunakan TF-IDF dan *semantic reranking* menggunakan SBERT. MRR memberikan indikasi langsung apakah proses reranking benar-benar berhasil mengangkat jawaban relevan ke peringkat atas. Nilai MRR yang tinggi menandakan bahwa model tidak hanya menemukan jawaban yang benar, tetapi juga secara konsisten menempatkannya pada posisi yang mudah dilihat pengguna, sehingga meningkatkan efektivitas interaksi chatbot.

2.3 Framework dan Algoritma yang digunakan

2.3.1 CRISP-DM

Cross-Industry Standard Process for Data Mining (CRISP-DM) merupakan sebuah standar yang ditunjukkan untuk melakukan proses analisis strategi pemecahan masalah dari bisnis atau unit penelitian di suatu industri [33]. CRISP-DM memiliki 6 tahapan implementasi diantaranya adalah sebagai berikut:



Gambar 2. 1 Diagram Flow CRISP-DM

Berdasarkan diagram flowchart pada gambar 2.1, terdapat 6 proses yang diimplementasikan dalam penelitian ini, berikut penjelasan masing-masing tahapannya:

1. Business Understanding

Dalam proses ini dilakukan penentuan tujuan, kebutuhan, batasan, serta strategi awal untuk mencapai tujuan yang ingin dicapai. Pada langkah ini di klasifikasikan tujuan bisnis, analisis permasalahan yang ada, dan mengidentifikasi batasan-batasan penelitian. Penelitian ini menentukan masalah yang ada di perusahaan yaitu membutuhkan program chatbot yang bisa membantu operasional penggunaan SAP di perusahaan

2. Data Understanding

Setelah mengetahui kebutuhan bisnis, tahapan selanjutnya adalah mengumpulkan dan mengenali data yang akan digunakan untuk penelitian. Dalam penelitian ini digunakan data FAQ dari tim helpdesk SAP dan user guide penggunaan SAP yang terdiri dari 407 pasang pertanyaan dan jawaban dari 11 modul SAP yang digunakan di perusahaan. Proses ini dilakukan dengan melihat eksplorasi awal data dengan melihat dokumen FAQ dan user guide perusahaan.

3. Data Preparation

Pada tahapan ini dilakukan persiapan data dengan mengekstrak data FAQ SAP yang berupa dokumen word kedalam dataset csv untuk persiapan modelling dan memastikan data berkualitas. Proses ini melibatkan ekstrak dokumen word, pembersihan data, transformasi, pemetaan kode transaksi, dan pemilihan data yang sesuai.

4. Modeling

Dalam tahapan ini, dilakukan pengaplikasian data yang telah dipersiapkan sebelumnya kedalam model utama yaitu TF-IDF dan SBERT yang kemudian disimpan kedalam model pickle data berupa file npy, npz, dan.pkl yang kemudian di gabungkan melalui hybrid retrieval berdasarkan nilai alpha lexical dan semantic tertentu untuk pengaturan modelnya.

5. Evaluation

Pada tahapan evaluation dilakukan pengujian model agar sejalan dengan kebutuhan bisnis dan teknis. Proses evaluasi ini menguji model terhadap data pertanyaan yang sesuai dan menilai kinerja model dengan menggunakan metrik evaluasi berupa akurasi top answer, MRR, presision, recall, dan avg latency. Dengan melihat metrik-metrik tersebut dapat membantu menilai berapa baik model dalam menghasilkan prediksi yang sesuai dengan tujuan penelitian dan kebutuhan bisnis sehingga dapat di terapkan secara langsung di perusahaan secara nyata.

6. Deployment

Pada proses implementasi yang merupakan tahapan akhir dengan menerapkan model yang sudah di evaluasi sebelumnya ke dalam konteks penggunaan nyata. Proses ini dilakukan dengan menerapkan model ke dalam framework atau aplikasi Flask app sebagai website yang dapat diakses oleh user internal berdasarkan server testing perusahaan dimana hal ini dapat membantu pengguna dalam mengatasi permasalahan tentang SAP sebelumnya tanpa perlu menunggu tim helpdesk SAP.

2.3.2 TF-IDF

TF-IDF merupakan salah satu *metode feature extraction* klasik yang sangat berpengaruh dalam bidang Information Retrieval (IR) dan Natural Language Processing (NLP). Teknik ini digunakan untuk merepresentasikan teks ke dalam bentuk numerik sehingga dapat diproses oleh algoritma

komputasi. Konsep dasarnya adalah mengukur seberapa penting suatu kata dalam dokumen dibandingkan dengan seluruh kumpulan dokumen(corpus)[18].

Komponen pertama, *Term Frequency (TF)*, menghitung frekuensi kemunculan kata dalam suatu dokumen. Semakin sering sebuah kata muncul, maka semakin besar pula bobot TF-nya. Namun, kata-kata umum seperti “the”, “and”, atau “di” yang sering muncul di banyak dokumen tidak memiliki nilai informasi yang tinggi. Karena itu, diperkenalkan komponen kedua yaitu *Inverse Document Frequency (IDF)*, yang berfungsi mengurangi bobot kata-kata yang terlalu umum dan memperkuat kata-kata yang lebih jarang dan bersifat unik terhadap konteks dokumen tertentu.

Hasil dari perhitungan TF-IDF adalah sebuah vektor numerik untuk setiap dokumen, di mana setiap elemen vektor mewakili bobot kepentingan kata-kata. Algoritma ini sering digunakan untuk mengubah teks menjadi nilai yang lebih bermakna dengan menggunakan persamaan (1) berikut.

$$w_{i,j} = tf_{i,j} \times \log \left(\frac{N}{df_i} \right)$$

Rumus 2 Rumus TF-IDF

Dengan penjelasan rumus sebagai berikut:

- $w_{i,j}$ = bobot kata ke-i pada dokumen ke-j
- $tf_{i,j}$ = frekuensi kemunculan kata ke-i dalam dokumen ke-j
- df_i = jumlah dokumen yang mengandung kata ke-i
- N = jumlah total dokumen dalam koleksi

TF-IDF mewakili jumlah total kemunculan istilah ke-i dalam sebuah dokumen yang mana df itu mewakili keseluruhan dokumen yang mengandung istilah ke-1, dan N mewakili jumlah dari total keseluruhan dokumen [18].

Representasi perhitungan ini merefleksikan kecocokan leksikal atau kata kunci murni antara query pengguna dan dokumen FAQ. TF-IDF digunakan

untuk mengekstrak kata kunci dari pertanyaan yang diajukan agar chatbot dapat mencari jawaban yang paling sesuai dengan basis data yang sudah ada [20]. Meskipun cepat dan efisien, kelemahan utama TF-IDF adalah ketidakmampuannya memahami makna kontekstual atau sinonim. Sebagai contoh, TF-IDF tidak akan menganggap "PR" dan "Purchase Requisition" memiliki makna yang sama jika keduanya tidak pernah muncul secara bersamaan, sehingga diperlukan metode yang lebih canggih untuk mengatasi bahasa bebas (free text).

2.3.3 SBERT

TF-IDF Sentence-BERT (SBERT) merupakan pengembangan dari model BERT (Bidirectional Encoder Representations from Transformers) yang dirancang oleh Devlin et al. pada tahun 2018 [15]. BERT pada dasarnya adalah model transformer-based yang mampu memahami konteks kata berdasarkan arah dua sisi (kiri dan kanan) dalam sebuah kalimat. Namun, BERT standar tidak secara langsung dioptimalkan untuk menghitung kesamaan antar kalimat (*sentence similarity*), karena menghasilkan representasi kata yang terpisah-pisah.

SBERT memperluas kemampuan ini dengan menambahkan Siamese dan Triplet Network di atas arsitektur BERT untuk menghasilkan sentence embeddings, yaitu representasi vektor berdimensi tinggi yang mewakili makna keseluruhan kalimat. Hasilnya, dua kalimat dengan makna yang sama akan memiliki jarak vektor yang dekat dalam ruang embedding. Contohnya, kalimat “bagaimana cara membuat purchase order” dan “apa tcode untuk membuat PO” akan menghasilkan embedding yang berdekatan karena memiliki maksud (*intent*) yang serupa.

SBERT bekerja melalui dua tahap utama:

1. *Encoding* yang melihat setiap kalimat diubah menjadi vektor tetap misal pada penelitian ini diubah menjadi sebanyak

384 dimensi, menggunakan model transformer multibahasa dengan nama "*paraphrase-multilingual-MiniLM-L12-v2*"

2. *SimilarityComputation* dimana proses untuk melihat kesamaan antar kalimat yang dihitung menggunakan metrik seperti *cosine similarity*, yang mengukur jarak sudut antara dua vektor.

Keunggulan utama SBERT dibandingkan model berbasis TF-IDF adalah kemampuannya memahami sinonim, parafrase, dan variasi bahasa alami. SBERT tidak hanya membandingkan kata demi kata, tetapi juga memperhatikan konteks kalimat secara keseluruhan. Namun, tantangan dari model ini adalah kebutuhan komputasi yang lebih tinggi, terutama ketika diaplikasikan pada dataset berukuran besar [16].

Dalam sistem chatbot FAQ SAP ini, SBERT digunakan pada tahap re-ranking untuk menilai seberapa relevan makna pertanyaan pengguna terhadap kandidat jawaban hasil pencarian awal oleh TF-IDF. Kombinasi kedua pendekatan ini memberikan keseimbangan antara kecepatan dan akurasi semantik.

2.3.4 Hybrid Retrieval

Hybrid Retrieval merupakan pendekatan pencarian informasi yang mengombinasikan metode berbasis kecocokan kata (TF-IDF) dan metode berbasis pemahaman makna (Sentence-BERT/SBERT) untuk meningkatkan relevansi hasil pencarian. Pendekatan ini digunakan untuk mengatasi keterbatasan masing-masing metode tunggal, khususnya pada sistem chatbot FAQ SAP yang harus menangani pertanyaan teknis sekaligus pertanyaan dengan bahasa bebas (*free text*).

Dalam penelitian ini, TF-IDF digunakan untuk menangkap kesesuaian istilah teknis dan kata kunci eksplisit yang umum digunakan dalam sistem SAP, seperti nama modul dan *transaction code* (*t-code*). Sementara itu, SBERT digunakan untuk memahami kesamaan makna antar kalimat, sehingga sistem

tetap dapat mengenali pertanyaan yang disampaikan dengan variasi bahasa atau parafrase yang berbeda dari data FAQ.

Proses *hybrid retrieval* dilakukan dengan menggabungkan skor kemiripan dari kedua metode tersebut menggunakan fungsi pembobotan (*weighted combination*). Skor akhir *hybrid retrieval* dihitung menggunakan persamaan berikut:

$$Score_{Hybrid}(q, d) = \alpha \cdot Score_{SBERT}(q, d) + (1 - \alpha) \cdot Score_{TF-IDF}(q, d)$$

Rumus 3 Hybrid Retrieval

Dengan penjelasan rumus sebagai berikut:

- q = merupakan *query* yang diajukan oleh pengguna,
- d = merupakan dokumen FAQ kandidat,
- α = merupakan parameter pembobotan dengan nilai antara 0 dan 1.

Pada penelitian ini, nilai α ditetapkan sebesar 0,6, sehingga komponen semantik memiliki bobot 60% dan komponen leksikal memiliki bobot 40%. Pemilihan bobot ini bertujuan untuk memberikan penekanan lebih pada pemahaman konteks pertanyaan, tanpa mengabaikan kecocokan istilah teknis yang menjadi karakteristik utama pertanyaan SAP.

Dengan menggunakan pendekatan hybrid retrieval ini, sistem chatbot mampu menyeimbangkan kecepatan pencarian berbasis kata kunci dan kemampuan pemahaman bahasa alami. Pendekatan ini memungkinkan chatbot memberikan jawaban yang lebih relevan baik untuk pertanyaan yang bersifat eksplisit maupun pertanyaan yang disampaikan dalam bentuk bahasa bebas, sehingga lebih sesuai dengan kebutuhan pengguna SAP di lingkungan perusahaan.

2.3.5 Jupyter Notebook

Jupyter Notebook adalah alat interaktif berbasis web yang digunakan untuk menulis dan menjalankan kode *Python* secara langsung [35]. *Platform* ini

mendukung kombinasi antara teks, kode, dan visualisasi, sehingga memudahkan proses dokumentasi dan eksplorasi data. Dalam penelitian ini, *Jupyter Notebook* digunakan untuk pengolahan dan ekstraksi data, pelatihan dan penyusunan model, serta pengevaluasi model yang dibuat.

Keunggulan *Jupyter Notebook* adalah kemampuannya untuk menjalankan kode secara berurutan, memungkinkan pengguna untuk menguji setiap langkah pengolahan data atau model secara terpisah, sehingga mempercepat proses debugging dan pengembangan. Kombinasi Python dan *Jupyter Notebook* menjadikannya tools yang ideal untuk penelitian berbasis data dan pembelajaran mesin.

2.3.6 Visual Studio Code

Visual studio code adalah sebuah alat editor kode dengan jenis sumber terbuka (*open source*) yang dikembangkan oleh *Microsoft* untuk sistem kerja *Windows*, *Linux*, dan *mac OS* [36]. *Visual studio code* digunakan untuk memudahkan penulisan code yang mendukung beberapa jenis bahasa pemrograman yang dimanfaatkan untuk menulis, mengedit, dan mengelola kode berdasarkan berbagai bahasa pemrograman. *Visual studio code* menyediakan berbagai macam fitur penyorotan sintaks, *auto-completion*, dan *IntelliSense* yang memudahkan dan membantu pengguna untuk menulis code tanpa banyak kesalahan. Berdasarkan fitur-fitur yang ada, *Visual Studio Code* menjadi salah satu alat penting dalam pengembangan perangkat lunak di seluruh dunia mulai dari proyek sederhana hingga proyek profesional [37].

2.3.7 Flask Framework

Flask framework adalah *micro web framework* berbasis *Python* yang digunakan untuk membangun aplikasi web secara ringan dan fleksibel [34]. *Flask* dikembangkan oleh Armin Ronacher sebagai bagian dari proyek Werkzeug dan Jinja2, serta termasuk dalam kategori *micro-framework* karena tidak menyertakan komponen kompleks seperti sistem *ORM (Object Relational Mapping)* atau validasi form secara default.

Keunggulan utama *Flask* terletak pada kesederhanaan dan kemudahan modifikasinya. *Framework* ini menyediakan struktur minimal yang dapat diperluas melalui ekstensi sesuai kebutuhan pengembang. Misalnya, *Flask* dapat dengan mudah diintegrasikan dengan pustaka machine learning seperti *scikit-learn*, *TensorFlow*, atau *PyTorch*, serta dapat digunakan untuk membangun API layanan model NLP seperti chatbot berbasis TF-IDF dan SBERT.

Dalam penelitian ini, *Flask* digunakan untuk membangun antarmuka pengguna (*User Interface*) berbasis web yang menghubungkan model NLP dengan pengguna akhir. Melalui *Flask*, sistem chatbot dapat dijalankan secara lokal maupun di-deploy ke server internal perusahaan. *Framework* ini menangani rute percakapan, input pertanyaan dari pengguna, pemrosesan data menggunakan model *hybrid retrieval*, serta menampilkan jawaban hasil pemrosesan secara real-time melalui browser.

Selain itu, *Flask* juga mendukung pengelolaan session pengguna, pencatatan log chat ke dalam file CSV, serta integrasi dengan API eksternal seperti pengiriman email otomatis ke PIC SAP ketika chatbot tidak dapat menemukan jawaban. Dengan arsitektur yang sederhana namun kuat, *Flask* menjadi pilihan ideal untuk penelitian ini karena kemampuannya menggabungkan fleksibilitas, kecepatan pengembangan, dan kemudahan integrasi antar komponen sistem.