BAB 2
TINJAUAN PUSTAKA

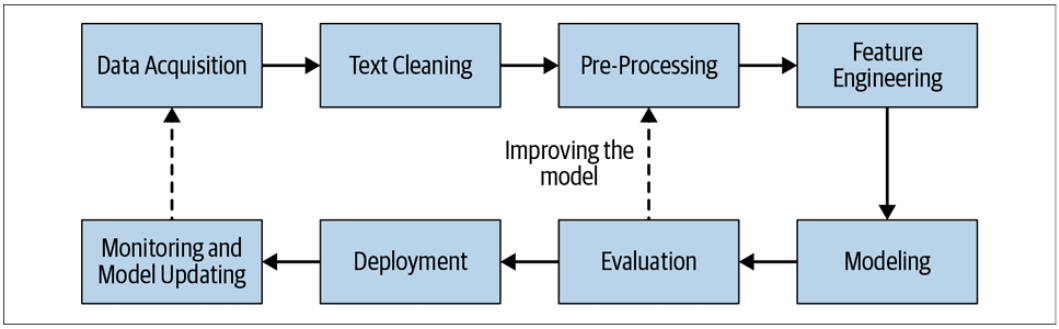
2.1 Landasan Teori

2.1.1 Obat

Obat adalah bahan yang digunakan untuk mencegah, mengurangi rasa sakit, atau menyembuhkan penyakit. Menurut Limbong dkk., (2023), berdasarkan Peraturan Menteri Kesehatan RI Nomor 949/Menkes/Per/IV/2000, penggolongan obat dibagi menjadi beberapa kategori berdasarkan tingkat keamanan, cara pengawasan, dan aturan distribusinya. Kategori pertama adalah obat bebas, yaitu obat yang bisa dibeli tanpa resep dokter di apotek atau toko obat yang resmi, ditandai dengan lingkaran hijau dengan tepi hitam, biasanya digunakan untuk mengatasi masalah kecil yang tidak memerlukan diagnosis dari dokter. Kategori kedua adalah obat bebas terbatas, yang termasuk obat keras yang dapat dibeli tanpa resep dokter, tetapi penggunaannya harus mengikuti petunjuk yang ada pada kemasan, ditandai dengan lingkaran biru dengan tepi hitam. Kategori ketiga adalah obat keras, yaitu obat yang hanya bisa didapatkan dengan resep dokter karena bisa berbahaya jika digunakan tanpa pengawasan medis, ditandai dengan lingkaran merah dengan tepi hitam dan huruf K di dalamnya. Kategori keempat adalah obat narkotika, yaitu obat yang berasal dari tanaman atau bukan dari tanaman, baik alami maupun buatan, yang bisa menyebabkan penurunan atau perubahan kesadaran, menghilangkan rasa sakit, dan menimbulkan ketergantungan fisik atau mental, sehingga distribusi dan penggunaannya diawasi dengan ketat. Kategori kelima adalah obat psikotropika, yaitu zat atau obat yang dapat memengaruhi sistem saraf pusat dan menyebabkan perubahan dalam aktivitas mental dan perilaku penggunanya (Limbong et al., 2023). Dalam penelitian ini, pengelompokan obat menjadi hal yang sangat penting dalam memberi rekomendasi obat alternatif, khususnya untuk memastikan bahwa obat yang disarankan termasuk dalam kategori yang sama dengan obat yang dicari oleh pengguna, agar penggantian bisa dilakukan dengan aman sesuai dengan aturan yang ada.

2.1.2 Natural Language Processing (NLP)

Natural Langauge Processing (NLP) merupakan suatu bidang dalam kecerdasan buatan yang berfokus pada hubungan antara komputer dan bahasa yang digunakan manusia. Tujuannya adalah agar komputer bisa secara otomatis memahami, menganalisis, dan memproses bahasa manusia, sehingga dapat dimanfaatkan dalam berbagai aplikasi, seperti penerjemahan, pengenalan suara, analisis emosi, dan pengambilan informasi dari teks (Turchin et al., 2023). Dalam perkembangannya, metode NLP telah mengalami perubahan dari pendekatan yang menggunakan aturan (*rule-based*) menjadi metode statistik dan *Machine Learning*, yang mana saat ini lebih banyak menggunakan model Deep Learning dan model berbasis Transformers seperti BERT, GPT dan BioBert yang menunjukkan kinerja yang sangat baik dalam berbagai tugas pemrosesan bahasa (Qiu dkk., 2020). Gambar 2.1 berikut merupakan *pipeline* dari NLP:



Gambar 2.1 NLP *Pipeline*

2.1.3 Semantic Matching

Semantic Matching adalah suatu metode untuk membandingkan dan menyamakan teks berdasarkan arti atau makna yang terkandung, bukan hanya pada kesamaan kata secara harfiah. Berbeda dengan pencocokan string yang lebih fokus pada kesesuaian karakter atau kata secara tepat, *Semantic Matching* menekankan pada hubungan makna sehingga dua teks yang menggunakan istilah yang berbeda tetapi memiliki arti yang sama tetap dapat diakui sebagai relevan. Menurut Gao dkk., 2019), *Semantic Matching* memiliki peranan vital dalam sistem berbasis pemrosesan bahasa alami karena bahasa sehari-hari sering kali mengandung sinonim, parafrasa, atau variasi dalam ungkapan. Dengan adanya *Semantic Matching*, sistem dapat memahami maksud dari sebuah teks meskipun bentuk kata-katanya berbeda. Oleh karena itu, *Semantic Matching* menjadi dasar dalam berbagai aplikasi seperti pencarian informasi, sistem tanya jawab, rekomendasi, serta analisis

teks dalam bidang medis.

2.1.4 Text Preprocessing

Text Preprocessing merupakan proses awal yang sangat penting dalam NLP. Proses ini bertujuan untuk mengubah teks yang belum diproses menjadi format yang lebih teratur, bersih, dan siap untuk digunakan oleh algoritma pemrosesan bahasa. Tahapan Preprocessing Text berfungsi untuk mengurangi noise pada saat pemrosesan, menstandarkan teks, serta meningkatkan kualitas representasi data sebelum dilakukan analisis lebih mendalam (Khurana dkk., 2023).

Beberapa tahapan umum yang ada dalam *Preprocessing Text* adalah sebagai berikut:

2.1.4.1 Case Folding

Case Folding adalah metode *preprocessing* yang sering digunakan dalam *pipeline* NLP. *Lowercasing* adalah salah satu dari 13 teknik *preprocessing* yang banyak diteliti pengaruhnya terhadap ketepatan *classifier* dalam analisa teks. Walaupun efektif dalam mengurangi variasi pada token, teknik ini berpotensi meningkatkan kebingungan contohnya, “Apple” (nama perusahaan) dan “apple” (buah) akan dianggap sebagai entitas yang sama sehingga penggunaannya perlu disesuaikan dengan jenis tugas NLP yang dilakukan (Shukla & Dwivedi, 2024).

2.1.4.2 Cleaning/Removing Punctuation

Cleaning/Removing Punctuation yaitu proses menghapus karakter-karakter khusus seperti angka, emoji, atau tanda baca (seperti titik, koma, tanya) yang tidak diperlukan dan tidak memiliki pengaruh signifikan terhadap makna teks. Tujuannya dari proses ini adalah untuk mengurangi *noise* pada data sehingga algoritma pemrosesan dapat lebih fokus pada isi yang isi yang bermakna. Dalam kasus teks medis yang tidak teratur, langakah ini sangat penting karena banyaknya karakter non-alfanumerik yang sering muncul dalam catatan medis atau masukan dari pengguna (Pradana & Hayaty, 2019).

2.1.4.3 Cleaning/Removing Unwanted Characters

Cleaning/removing unwanted characters merupakan salah satu tahapan yang bertujuan untuk menghilangkan elemen yang tidak dibutuhkan, seperti tanda baca, simbol, emoji atau karakter khusus yang tidak berperan dalam proses

pencocokan teks. Langkah ini sangat penting karena tambahan karakter pada teks bisa membuat perbedaan dalam cara penyajian meskipun teks tersebut memiliki arti yang sama. Dalam hal data obat, tahap pembersihan sangat penting karena nama obat sering kali dicampur dengan informasi lain seperti dosis, satuan, bentuk sediaan, atau tanda baca tertentu. Wibawa dkk., (2023), melakukan *text mining* pada data rekam medis elektronik dan menjelaskan bahwa tanda baca dihilangkan pada tahap awal pengolahan agar pola penulisan nama obat dan kadar obat dapat dikenali dengan lebih baik. Ini menunjukkan bahwa menghapus karakter yang tidak diperlukan dapat membantu sistem dalam memahami struktur informasi obat dengan lebih konsisten.

2.1.4.4 Whitespace Normalization

Whitespace normalization/removal adalah salah satu langkah dalam yang bertujuan untuk menghilangkan spasi yang tidak perlu dan menyamakan format spasi pada data teks. Selama pengolahan teks, data asli sering kali memiliki cara penulisan yang tidak seragam, seperti adanya spasi berlebihan di awal, akhir, atau di antara kata-kata. Ketidaksamaan ini bisa membuat sistem menganggap dua teks yang sebenarnya identik sebagai data yang berbeda. Palomino & Aider, (2022), menyatakan bahwa pengolahan teks diperlukan untuk membersihkan data teks mentah sebelum digunakan dalam analisis berikutnya. Langkah-langkah pengolahan berperan dalam mengurangi ketidakkonsistenan dan gangguan dalam teks, sehingga data menjadi lebih siap untuk diproses oleh sistem.

2.1.4.5 Tokenization

Tokenization adalah proses yang memecah teks menjadi bagian-bagian kecil yang disebut token. Token ini bisa berupa kata, bagian dari kata, karakter, atau unit dalam *byte-level.* Tujuan utama dari tokenisasi adalah untuk mengubah teks yang tidak teratur menjadi format yang teratur agar bisa dipahami secara matematis. Tokenisasi yang dihasilkan menjadi komponen dasar dalam model *machine learning*. Tokenisasi adalah tahapan yang penting dalam menyiapkan data untuk model bahasa, dan cara yang dipilih untuk melakukan tokenisasi bisa mempengaruhi hasil, tergantung pada sifat bahasa yang sedang diproses. Secara umum, ada tiga metode utama: tokenisasi tingkat kata yang memecah teks berdasarkan ruang dan tanda baca, tokenisasi tingkat karakter yang memecah teks

hingga ke tiap karakter, serta tokenisasi bagian kata seperti *Byte Pair Encoding* (BPE) dan *WordPiece*. *Byte Pair Encoding* (BPE) dan *WordPiece* ini menggabungkan kelebihan dari keduanya dengan memecah kata-kata langka menjadi bagian yang lebih umum sehingga mampu menangani variasi bentuk kata dan kata-kata yang tidak ada dalam kosakata (Siino et al., 2024).

Dengan tahapan-tahapan ini, *preprocessing text* bisa dilihat sebagai dasar yang penting dalam NLP. Tahapan ini tidak hanya bertujuan untuk membersihkan data, tetapi juga mempengaruhi tingkat kualitas representasi teks yang akan diterapkan di langkah analisis berikutnya.

2.1.5 Text Representation

Text Representation atau Representasi teks merupakan proses mengkonversi data dalam bentuk bahasa alami menjadi format numerik atau vektor agar dapat diproses oleh komputer dalam berbagai jenis tugas pemrosesan bahasa alami. Proses ini sangat penting karena komputer tidak dapat menangkap makna teks ketika dalam format aslinya, melainkan memerlukan representasi matematis yang dapat dipahami oleh algoritma. Minaee dkk., (2022), menyatakan bahwa tujuan dari representasi teks adalah untuk menangkap informasi terkait leksikal (kata-kata), semantik (arti), serta konteks dari sebuah kalimat, sehingga model dapat mengenali hubungan antara kata-kata dan pemahaman keseluruhan teks. Ini menjadikan representasi teks sebagai dasar utama dalam hampir semua aplikasi pemrosesan bahasa alami, seperti pengelompokan teks, analisis emosi, ekstraksi entitas, dan sistem pencarian informasi.

2.1.5.1 TF-IDF (Term Frequency-Inverse Document Frequency)

Term Frequency-Inverse Document Frequency (TF-IDF) merupakan metode pembobotan kata secara statistik yang digunakan dalam *information retrieval* dan NLP untuk menentukan seberapa penting sebuah kata dalam suatu dokumen diantara banyak dokumen atau korpus. TF-IDF adalah rumus matematis yang paling sering digunakan dalam *information retrieval*, yang dipahami sebagai cara sederhana untuk mengukur seberapa sering sebuah kata muncul dalam satu dokumen tertentu dibandingkan dengan banyak dokumen lainnya. Berbagai variasi dari TF-IDF sering digunakan sebagai metode penilaian kata dalam berbagai

aplikasi analisis teks. Secara teknis, nilai TF-IDF dihitung dari hasil kali dua bagian: *Term Frequency* (TF) yang mengukur seberapa sering sebuah kata muncul dalam sebuah dokumen, dan *Inverse Document Frequency* (IDF) yang menilai seberapa langka kata tersebut di seluruh kumpulan dokumen, sehingga kata-kata yang muncul banyak di berbagai dokumen memiliki nilai rendah, sementara kata-kata yang lebih unik pada dokumen tertentu mendapatkan nilai tinggi (Heryawan et al., 2024). Menurut Septiani & Isabela, (2022) rumus nilai TF-IDF adalah sebagai berikut:

$$tf = 0{,}5 + 0{,}5 \times \frac{tf}{\max(tf)}$$

$$idf_t = \log(\frac{D}{df_t})$$

$$W_{d.t} = tf_{d.t} \times idf_{d.t}$$

2-1

Dimana:

- D = Dokumen ke-d
- t = Term ke-t dari dokumen
- W = Bobot ke-d terhadap term ke-t
- tf = Jumlah kemunculan term I dalam dokumen
- idf = *Inversed Document Frequency*
- df = Banyak dokumen yang mengandung term i

2.1.5.2 Word2Vec

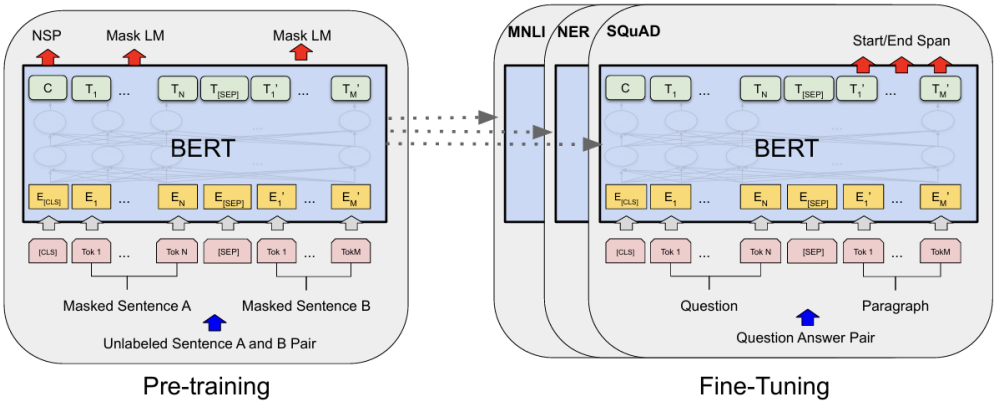
Word2Vec adalah metode representasi yang berbasis word embedding, yang mengonversi kata-kata menjadi vektor dengan dimensi tetap. Terdapat dua arsitektur utama dalam Word2Vec, yaitu *Continuous Bag of Words* (CBOW) dan Skip-Gram. Keunggulan dari Word2Vec adalah kemampuannya untuk merepresentasikan kata dengan memperhatikan konteks di sekitarnya, sehingga kata-kata yang memiliki makna yang mirip akan memiliki representasi vektor yang saling mendekat (Anjani & Irmanda, 2024). Untuk implementasi, Word2vec dapat digunakan dengan menggunakan berbagai *library* Python, salah satunya adalah Gensim yang menawarkan fungsi yang memudahkan dalam melatih model CBOW maupun Skip-Gram serta mendapatkan representasi vektor dari kata-kata. Di

samping itu, *framework* seperti TensorFlow dan PyTorch juga mendukung pengembangan Word2Vec yang disesuaikan melalui *layer embedding*. Untuk bahasa Indonesia, ada beberapa model *pre-trained* yang telah dibuat dengan menggunakan *library* IndoWord2Vec untuk bahasa ini, yang telah dilatih pada kumpulan data besar Bahasa Indonesia untuk menghasilkan representasi makna yang lebih tepat sesuai konteks lokal (Putri et al., 2021).

2.1.5.3 FastText

FastText dikembangkan oleh Facebook AI Research dan memiliki pendekatan serupa dengan Word2Vec, namun memperhatikan struktur sub-kata. Dengan demikian, FastText memiliki kelebihan dalam menangani kata-kata baru (*out of vocabulary*/OOV), karena bisa membangun representasi dari potongan n-gram karakter (Siti Khomsah et al., 2022). Untuk implementasi, FastText bisa dioperasikan dengan menggunakan *library* resmi dari Facebook AI yang mendukung pelatihan model secara *unsupervised* maupun *supervised*. Di samping itu, *library* Gensim juga menawarkan modul FastText yang lebih *user-friendly*, terutama untuk peneliti yang sudah biasa menggunakan Word2Vec, karena strukturnya mirip. Dalam konteks Bahasa Indonesia, terdapat model *pre-trained* FastText yang dirilis oleh Facebook AI Research. Model ini dilatih dengan menggunakan korpus besar dari Common Crawl dan Wikipedia dalam Bahasa Indonesia, sehingga dapat menciptakan representasi vektor yang lebih baik untuk morfologi serta variasi kata dalam Bahasa Indonesia (Young & Rusli, 2019).

2.1.5.4 IndoBERT

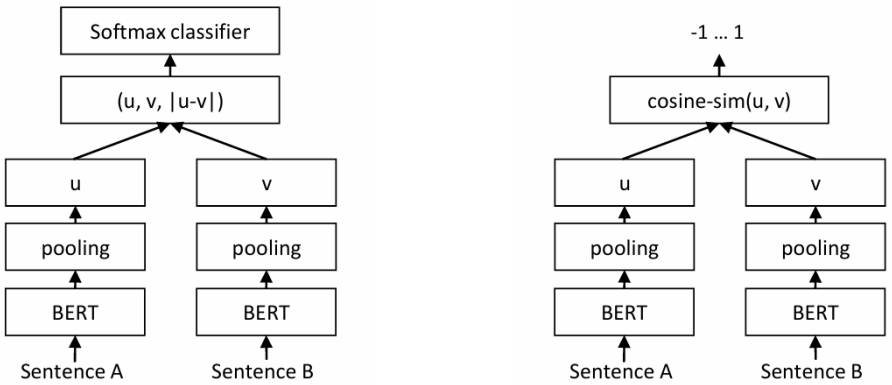


Gambar 2.2 Prosedur Metode IndoBERT Secara Umum

IndoBERT adalah model *Bidirectional Encoder Representations from Transformers* (BERT) yang dirancang khusus untuk bahasa Indonesia. Model ini mampu memahami konteks kata dengan cara dua arah (dari kiri dan kanan) sehingga lebih tepat dalam menafsirkan makna semantik dibandingkan dengan metode konvensional. IndoBERT terbukti lebih unggul dalam berbagai tugas NLP untuk bahasa Indonesia, seperti klasifikasi teks, ekstraksi entitas, dan kesamaan semantik (Kartika et al., 2023). Menurut Devlin et al., (2019), pada Gambar 2.2 terdapat prosedur metode IndoBERT secara umum.

2.1.5.5 SBERT

Sentence-BERT (SBERT) merupakan metode representasi teks yang menghasilkan *sentence embedding*, yaitu representasi sebuah kalimat dalam bentuk vektor numerik yang memuat informasi semantik. SBERT menggunakan arsitektur siamese atau *bi-encoder* sehingga setiap teks dapat diproses secara terpisah, kemudian tingkat kemiripan antarteks dihitung berdasarkan kedekatan vektornya, salah satunya menggunakan *Cosine Similarity*. Pendekatan ini memungkinkan proses pencarian dan pemeringkatan dokumen dilakukan secara lebih efisien (Ding et al., 2023). Menurut Reimers & Gurevych, (2019), pada Gambar 2.3 berikut terlampir alur dari metode SBERT secara umum:



Gambar 2.3 Alur Metode SBERT Secara Umum

Dalam bidang biomedis, SBERT dapat digunakan untuk mencocokkan *query* pengguna dengan teks yang memiliki makna relevan meskipun menggunakan susunan kata yang berbeda. Pada penelitian (Weber & Toddenroth, 2024) menunjukkan bahwa SBERT mampu memberikan kinerja yang baik dalam proses *biomedical information retrieval*. Selain itu, pada penelitian (Pandey et al., 2022),

penggunaan *embedding* kontekstual berbasis SBERT juga dapat membantu mengidentifikasi kesamaan semantik yang tidak dapat ditangkap hanya berdasarkan kemunculan kata yang sama.

Dengan membandingkan kelima metode representasi teks ini, sistem dapat menentukan pendekatan terbaik dalam menghasilkan representasi yang sesuai untuk analisis *semantic matching*.

2.1.6 Cosine Similarity

Cosine Similarity digunakan untuk mengukur seberapa mirip dua kata atau objek dengan melihat arah vektornya. Kemiripan *cosinus* digunakan untuk mengevaluasi seberapa mirip suatu produk dengan alternatif obat berdasarkan fitur-fitur tertentu seperti komposisi atau indikasi. Tingkat kemiripan dinyatakan dalam angka 0 dan 1, dimana nilai yang semakin mendekati 1 menunjukkan bahwa kesamaan antar kata semakin besar (Indriyani et al., 2025). Menurut Selvaraj dkk., (2024), untuk rumus *Cosine Similarity* adalah sebagai berikut:

$$\cos(\theta) = \frac{A \cdot B}{||A|| \cdot ||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \cdot \sqrt{\sum_{i=1}^{n} B_i^2}}$$

2-2

Dimana:

- A, B = Dua vektor yang dibandingkan kemiripannya
- $A \cdot B$ = Perkalian antara vektor *A* dan *B,* dihitung sebagai $\sum_{i=1}^{n} A_i B_i$
- $||A||$ = Norma L2 dari vektor *A,* dihitung sebagai $\sqrt{\sum_{i=1}^{n} A_i^2}$
- $||B||$ = Norma L2 dari vektor *B,* dihitung sebagai $\sqrt{\sum_{i=1}^{n} B_i^2}$

2.1.7 Facebook AI Similarity Search (FAISS)

Facebook AI *Similarity Search* (FAISS) adalah *library open-source* yang yang dibuat oleh Meta AI (yang sebelumnya dikenal sebagai Facebook AI Research) untuk mencari dan mengelompokkan vektor dengan banyak dimensi dengan cara yang cepat pada kumpulan data yang besar. Menurut Douze dkk., (2025), FAISS adalah alat pengindeksan yang berguna untuk mencari, mengelompokkan, mengompres, dan mengubah vektor dalam jumlah besar, mulai dari jutaan hingga miliaran data, sambil menyeimbangkan antara cepatnya

pencarian, penggunaan memori, dan ketepatan hasil. Tidak seperti mesin pencari yang menggunakan kata kunci biasa, FAISS berfungsi dengan menggunakan representasi vektor dari informasi, kemudian mengukur jarak antara vektor dalam ruang berdimensi tinggi untuk menemukan item yang paling mirip berdasarkan isi kontennya.

FAISS menawarkan berbagai jenis indeks seperti IndexFlatL2 yang digunakan untuk pencarian menyeluruh menggunakan jarak Euclidean dan IndexIVFFlat yang menerapkan teknik pemisahan *Inverted File Index* (IVF) untuk mempercepat pencarian dengan membagi vektor menjadi bagian-bagian kecil, sehingga hanya kelompok terdekat yang diperiksa saat pencarian tanpa perlu menelusuri seluruh kumpulan data. Dalam bidang *Natural Language Processing* (NLP), FAISS biasanya digabungkan dengan model *embedding* seperti BERT, Word2Vec, atau FastText untuk menciptakan sistem pencarian yang dapat menemukan dokumen atau item yang paling sesuai berdasarkan kesamaan makna, bukan hanya cocok dengan kata kunci (Douze et al., 2025).

Dalam penelitian ini, FAISS digunakan sebagai bagian dari sistem pengindeksan vektor untuk mempercepat pencarian kesamaan antara berbagai obat dalam basis data, sehingga sistem rekomendasi obat alternatif dapat berjalan dengan lancar meskipun data yang digunakan sangat besar.

2.1.8 Pengujian Sistem

Pengujian sistem dilakukan untuk mengetahui kinerja model rekomendasi dan memastikan fungsi sistem berjalan sesuai dengan kebutuhan. Pengujian dibagi menjadi dua bagian, yaitu evaluasi Top-K dan *Blackbox Testing*.

2.1.8.1 Evaluasi top-k

Evaluasi top-k dilakukan untuk menilai seberapa baik rekomendasi yang diberikan oleh sistem. Dalam sistem rekomendasi, evaluasi top-k adalah mekanisme untuk mengambil keputusan dimana sistem memberikan sejumlah K item yang paling sesuai berdasarkan skor kesamaan tertinggi dari semua pilihan yang ada. Menurut Jadon & Patil, (2024), evaluasi top-k bekerja dengan menghitung skor kesamaan antara item *query* dengan semua item dalam *database,* lalu mengurutkan hasilnya dari yang tertinggi hingga terendah. Setelah

mengurutkan hasilnya, evaluasi top-k memilih K item teratas untuk diberikan sebagai rekomendasi kepada pengguna. Metode ini menjadi dasar utama dalam sistem rekomendasi yang menggunakan *content-based filtering* karena dapat membantu sistem untuk mencari item terkait dari kumpulan data yang besar dengan cara yang efisien.

Berdasarkan penelitian Jadon & Patil, (2024), penilaian sistem rekomendasi meliputi kumpulan serangkaian metrik yang dirancang untuk mengukur berbagai fitur kerja sistem, dimulai dari akurasi prediksi hingga kualitas urutan rekomendasi yang diberikan kepada pengguna. Untuk metrik yang digunakan dalam penelitian ini antara lain:

1) *Precision,* yaitu mengukur jumlah item yang sesuai diantara rekomendasi terbaik yang diberikan kepada pengguna.
2) *Recall,* yaitu menghitung seberapa banyak item yang berhasil ditangkap dalam rekomendasi teratas dibandingkan dengan seluruh item relevan yang tersedia.
3) *Mean Average Precision* (MAP), yaitu mengevaluasi urutan rekomendasi dengan menghitung rata-rata ketepatan pada setiap posisi dimana item yang relevan ditemukan, sehingga memberikan gambaran lengkap tentang kualitas *ranking* rekomendasi.
4) *Mean Reciprocal Rank* (MRR), bekerja dengan mengukur posisi dari item relevan pertama dalam daftar rekomendasi dengan menghitung rata-rata *reciprocal rank* dari seluruh *query,* dimana angka yang lebih tinggi menunjukkan bahwa sistem selalu menempatkan item relevan di urutan teratas.
5) *Normalized Discounted Cumulative Gain* (NDCG), digunakan untuk mengukur kualitas peringkat atau rekomendasi suatu sistem seperti sistem rekomendasi atau mesin pencari. Metrik ini menilai seberapa tepat sistem dalam menyusun suatu item dari yang paling relevan hingga yang tidak relevan.

2.1.8.2 Blackbox Testing

Blackbox testing merupakan pengujian yang menilai kinerja sistem berdasarkan input dan output serta respon sistem tanpa memperhatikan struktur internal dari kode (Pirdaus & Hidayana, 2024). Fokus utama dari pengujian ini adalah untuk mengkonfirmasi fungsionalitas sistem dari perspektif pengguna dengan memastikan setiap fitur dapat memberikan hasil yang sesuai dengan

harapan saat menerima input tertentu. Dalam *blackbox testing*, penguji tidak perlu mengetahui cara implementasi kode di dalamnya tetapi cukup memastikan bahwa sistem bekerja sesuai dengan spesifikasi yang ditentukan. Pendekatan ini dianggap tepat untuk menguji sistem rekomendasi obat berbasis web karena pengujian dapat langsung dilakukan pada antarmuka dan fungsi yang akan digunakan oleh pengguna akhir, seperti penginputan gejala, tampilan hasil rekomendasi, serta respons sistem terhadap input yang tidak valid.

2.2 Penelitian Terdahulu

Penelitian oleh R dkk., (2025) melakukan penelitian dengan menggunakan *machine learning* dan NLP di bidang farmasi. Penelitian ini berfokus untuk meningkatkan sistem rekomendasi resep obat dan pengelolaan perawatan pasien. Dataset yang digunakan pada penelitian ini bersumber dari platform apotek *online* yang berjumlah lebih dari 11.000 data obat. Data ini mencakup informasi seperti nama obat, kegunaan, komposisi, dan efek samping. Sistem yang dibuat menggabungkan TF-IDF untuk mengambil fitur dari teks, BERT *embeddings* untuk memahami teks medis yang kompleks, dan *Cosine Similarity* untuk mengevaluasi kesamaan antara *query* pengguna dengan seluruh entri data obat yang ada. Hal ini bertujuan untuk menghasilkan top-5 rekomendasi obat yang paling sesuai. Hasil penilaian menunjukkan bahwa model yang diajukan mencapai tingkat akurasi sebesar 97%, *precission* 98,8%, *recall* 96,3%, dan F1-*score* 96,5%. Hasil ini lebih baik dibandingkan dengan metode lain seperti *machine learning* konvensional yaitu 79% dan *K-Nearest Neighbors* sebesar 87,5%.

Penelitian yang dilakukan oleh Zomorodi dkk., (2022) adalah merancang sistem rekomendasi obat yang lengkap dengan menggunakan berbagai fitur pasien dan obat secara bersamaan. Fitur yang digunakan adalah kondisi kesehatan pasien, usia, jenis kelamin, efek samping dari obat, serta kategori obat. Sistem ini dibangun dengan menggunakan pendekatan NLP untuk *preprocessing sentiment analysis* yang kemudian dikombinasikan dengan metode *neural network* dan algoritma *recommender system* berbasis *matrix factorization* untuk menghasilkan rekomendasi. Hasil dari evaluasi menunjukkan bahwa model yang dikembangkan dapat meningkatkan akurasi, sensitivitas, dan tingkat keberhasilan masing-masing sebesar 26%, 34%, dan 40% jika dibandingkan dengan metode *matrix factorization*

konvensional, serta dapat meningkatkama rata-rata dari ketiga metrik tersebut sebesar 31%, 29%, dan 28% dibandingkan dengan metode *machine learning* lainnya.

Dalam permasalahan lain, Garg, (2021) melakukan penelitian dengan membuat sistem yang merekomendasikan obat berdasarkan analisis sentimen dari ulasan yang diberikan pasien. Penelitian ini menggunakan beberapa metode pengolahan teks, yaitu *Bag-of-Words* (BoW), TF-IDF, Word2Vec, dan *Manual Feature Analysis.* Metode-metode ini kemudian digabungkan dengan berbagai algoritma klasifikasi untuk memberikan rekomendasi obat terbaik sesuai dengan penyakit yang dialami. Penilaian dilakukan dengan menggunakan metrik seperti *precision, recall,* F1-*score,* akurasi, dan AUC *score.* Hasil penelitian mengungkapkan bahwa kombinasi TF-IDF dengan *classifier* LinearSVC yang memberikan hasil terbaik dibandingkan dengan metode lainnya dengan tingkat akurasi mencapai 93%. Ini menunjukkan bahwa pendekatan yang menggunakan frekuensi kata seperti TF-IDF masih dapat bersaing dengan metode berbasis *embedding* seperti Word2Voc dalam hal klasifikasi teks di bidang farmasi.

Sementara itu, Sutriawan dkk., (2025) melakukan penelitian dengan membandingkan lima model *text embedding* yaitu TF-IDF, Word2Vec, FastText, BERT dan GPT. Penilaian ini dilakukan untuk mengklasifikasikan kalimat ambigu dalam korpus berita berbahasa Indonesia. Dengan menggunakan dataset XL-Sum dari BBC *News* yang terdiri dari 90.00 lebih kalimat. Hasil dari penelitian ini menunjukkan bahwa mengkombinasikan GPT dengan *Gaussian Naive Bayes* memberikan hasil terbaik dalam menemukan kalimat ambigu, dengan *recall* sebesar 71% dan F1-*Score* 60%. Di sisi lain, TF-IDF yang dipasangkan dengan *Bagging Classifier* berhasil mencapai akurasi tertinggi yaitu 83% untuk kalimat yang jelas. Sementara itu, Word2Vec dan *FastText* kurang efektif dalam memahami hubungan makna secara keseluruhan, sedangkan model berbasis transformer seperti BERT dan GPT lebih unggul dalam memahami konteks yang rumit dalam teks.

Terakhir, studi yang dilakukan oleh Aydoğan Kılıç dkk., (2025) adalah membandingkan lima metode untuk mengekstrak fitur, yaitu TF-IDF, Word2Vec, GloVe, FastText, dan BERT. Mereka menggunakan dataset rekam medis MIMIC-III untuk mengklasifikasikan kode ICD dengan model *deep learning* seperti NN,

CNN, dan BiLSTM. Hasil dari penelitian ini menunjukkan bahwa FastText dengan arsitektur skip-gram mendapatkan *micro* ROC-AUC tertinggi sebesar 91,74% dan *micro recall* terbaik yaitu sebesar 68,16% saat menggunakan BiLSTM. Di sisi lain, Word2Vec dengan metode CBOW menghasilkan *micro* F1-*score* tertinggi sebesar 68,84% dan *macro* F1-*score* sebesar 59,67%. Kesimpulan dari penelitian ini adalah FastText adalah metode ekstraksi fitur yang paling konsisten saat digunakan dalam model *deep learning*, meskipun perbedaan kinerja antara metode cenderung menurun ketika kualitas *preprocessing* dan kompleksitas model meningkat. Pada Tabel 2.1 berikut merupakan perbandingan dari variabel penelitian:

Tabel 2.1 Perbandingan Variabel Penelitian

Nama	**R dkk., (2025)**	**Zomorodi dkk., (2022)**	**Garg, (2021)**	**Sutriawan dkk., (2025)**	**Aydoğan Kılıç dkk., (2025)**
Judul	Enhancing Drug Discovery and Patient Care Through Advanced Analytics with The Power of NLP and Machine Learning in Pharmaceutical Data Interpretation	RECOMED: A Comprehensive Pharmaceutical Recommendation System	Drug Recommendation System Based on Sentiment Analysis of Drug Reviews Using Machine Learning	Performance Evaluation of Text Embedding Models for Ambiguity Classification in Indonesian News Corpus: A Comparative Study of TF-IDF, Word2Vec, FastText, BERT, and GPT	Comparative Study of Feature Extraction Methods for Automated ICD Code Classification Using MIMIC-III Medical Notes and Deep Learning Models
Tujuan	Mengoptimalkan sistem rekomendasi resep obat berbasis teks medis menggunakan NLP dan ML	Membangun sistem rekomendasi obat komprehensif berbasis fitur pasien dan obat	Membangun sistem rekomendasi obat berbasis analisis sentimen ulasan pasien dengan membandingkan beberapa metode	Membandingkan performa lima model text embedding untuk klasifikasi kalimat ambigu pada korpus berita berbahasa Indonesia	Membandingkan metode ekstraksi fitur secara sistematis untuk klasifikasi kode ICD otomatis pada catatan medis

Nama	R dkk., (2025)	Zomorodi dkk., (2022)	Garg, (2021)	Sutriawan dkk., (2025)	Aydoğan Kılıç dkk., (2025)
			representasi teks		
Metode	BERT embeddings + TF-IDF + *Cosine Similarity*	NLP preprocessing + neural network + matrix factorization	BoW, TF-IDF, Word2Vec + algoritma klasifikasi (LinearSVC, Naive Bayes, dll.)	TF-IDF, Word2Vec, FastText, BERT, GPT + Logistic Regression, Random Forest, Bagging, Multinomial NB, Gaussian NB	TF-IDF, Word2Vec (CBOW & skip-gram), GloVe, FastText (CBOW & skip-gram), BERT + NN, CNN, BiLSTM
Dataset/ Objek Uji	>11.000 data obat dari platform apotek online (Kaggle)	Data dari Drugs.com dan Druglib.com	UCI Drug Review Dataset (Drugs.com)	XL-Sum BBC News Indonesia (>90.000 kalimat)	MIMIC-III-50 (11.368 discharge summaries ICU, 50 kode ICD)
Hasil/ Akurasi	Akurasi 97%, Precision 96,8%, Recall 96,3%, F1-score 96,5%	Akurasi, sensitivitas, dan hit rate meningkat masing-masing 26%, 34%, dan 40% dibanding baseline	TF-IDF + LinearSVC unggul dengan akurasi 93% dibanding metode lainnya	GPT + Gaussian NB: Recall 71%, F1-score 60% (kelas ambigu); TF-IDF + Bagging: Akurasi tertinggi 83% (kelas tidak ambigu)	BiLSTM + FastText (skip-gram): micro ROC-AUC 91,74%, micro Recall 68,16%; BiLSTM + Word2Vec (CBOW): micro F1 64,84%, micro Precision 62,34%
Relevansi dengan Penelitian Ini	Menggunakan gabungan BERT dan TF-IDF ditambah *Cosine Similarity* untuk saran obat yang berfokus pada teks yang dapat	Sistem rekomendasi obat komprehensif yang memperhatikan karakteristik medis obat yang dapat digunakan sebagai acua	Perbandingan metode representasi teks (TF-IDF, Word2Vec, BoW) untuk rekomendasi obat, yang relevan sebagai baseline	Perbandingan TF-IDF, Word2Vec, FastText, dan BERT pada Bahasa Indonesia yang relevan sebagai justifikasi	Perbandingan sistematis TF-IDF, Word2Vec, FastText, dan BERT pada data medis yang relevan sebagai landasan pemilihan

Nama	R dkk., (2025)	Zomorodi dkk., (2022)	Garg, (2021)	Sutriawan dkk., (2025)	Aydoğan Kılıç dkk., (2025)
	digunakan sebagai alasan untuk pendekatan pencocokan semantik di bidang farmasi.	n untuk struktur sistem rekomendasi di bidang farmasi.	perbandingan metode	pemilihan dan perbandingan metode embedding dalam konteks bahasa Indonesia	metode ekstraksi fitur untuk data teks klinis/farmasi

Berdasarkan analisis terhadap beberapa penelitian sebelumnya yang terdapat pada Tabel 2.1, dapat diketahui bahwa penelitian-penelitian tersebut telah membahas isu yang berkaitan dengan pennelitian ini. Namun, mereka memiliki keterbatasan dalam hal metode representasi teks yang dibandingkan, konteks bahasa dan jenis data yang digunakan, serta integrasi sistem rekomendasi obat alternatif secara menyeluruh yang didasarkan pada atribut klinis seperti indikasi, komposisi, dan usia pasien.

Penelitian ini berbeda dari penelitian sebelumnya karena berfokus pada masalah rekomendasi obat alternatif dalam konteks farmasi di Indonesia dengan membandingkan empat metode representasi teks, yaitu TF-IDF, Word2Vec, FastText, dan IndoBERT. Pendekatan yang digunakan adalah *semantic mathing* yang berbasis pada *Cosine Similarity* dan rekomendasi top-k, serta membandingkan kesamaan dalam komposisi, indikasi obat, serta kecocokan usia sebagai faktor dalam rekomendasi. Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi bagi pengembangan sistem rekomendasi obat berbasis web yang lebih sesuai secara klinis dan linguistik dalam menjawab permasalahan yang ada serta melengkapi hasi-hasil dari penelitian sebelumnya.