BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Tabel 2.1 menyajikan ringkasan hasil penelitian terdahulu yang relevan dengan topik penelitian ini, yang digunakan sebagai acuan dan pedoman dalam penyusunan penelitian:

Tabel 2.1. Tabel Penelitian Terdahulu

Jurnal 1 [10]	
Penulis	Bimantyoso Hamdikatama
Tahun	2025
Jurnal	JITK (Jurnal Ilmu Pengetahuan Dan Teknologi Komputer)
Volume	10
Nomor	3
Judul	Beyond Algorithms: An Integrated Approach To Fake News Detection Using Machine Learning Techniques
Metode	XGBoost, SVC, Stacking Ensemble
Dataset	Fake News Dataset
Hasil Penelitian	Model *stacking ensemble* menunjukkan kinerja terbaik dalam deteksi berita hoaks dengan akurasi 82%, melampaui XGBoost (79%) dan SVC (78%), yang menegaskan efektivitas pendekatan kombinasi model dengan dukungan representasi teks berbasis IndoBERT.
Kelemahan/Research Gap	Belum mengevaluasi kemampuan generalisasi model pada berbagai sumber data dan skenario deteksi hoaks yang lebih beragam

Jurnal 2 [11]	
Penulis	Azura Fardhina, Ramadhan Mustaqim Siregar, Mika Ria Waty Br Sibarani, Irsya Chlara Br Ginting, dan Andre Pratama
Tahun	2025
Jurnal	Jumistik (Jurnal Manajemen Informatika, Sistem Informasi dan Teknologi Komputer)
Volume	4
Nomor	1
Judul	SISTEM DETEKSI BERITA HOAKS BERBASIS ALGORITMA NATURAL LANGUAGE PROCESSING (NLP) MENGGUNAKAN BERT
Metode	BERT
Dataset	Dataset Berita Indonesia
Hasil Penelitian	Model *BERT* menunjukkan kinerja unggul dalam deteksi disinformasi dengan akurasi 98%, melampaui *KNN* yang hanya mencapai 93,33%, sehingga menegaskan efektivitas pendekatan berbasis *BERT* dalam analisis teks digital.
Kelemahan/Research Gap	Belum mengeksplorasi pemanfaatan konteks judul dan isi berita secara bersamaan menggunakan model berbasis *transformer*
Jurnal 3 [12]	
Penulis	Alif Gumelar Syah Moeslim, Esa Firmansyah, dan Beben Sutara
Tahun	2025
Jurnal	Jurnal Data Science Indonesia

Volume	5
Nomor	2
Judul	Perbandingan Kinerja XGBoost dan IndoBERT untuk Klasifikasi Teks Kesehatan Bahasa Indonesia
Metode	IndoBERT, XGBoost
Dataset	Dataset pertanyaan (tanya-jawab) kesehatan dalam bentuk teks berbahasa Indonesia
Hasil Penelitian	Hasil menunjukkan bahwa IndoBERT memberikan performa lebih baik dengan akurasi rata-rata 75,44% dibandingkan XGBoost sebesar 69,59%, serta unggul pada metrik *precision*, *recall*, dan F1-*score*. Meskipun demikian, XGBoost memiliki keunggulan dalam efisiensi komputasi dengan waktu pelatihan dan inferensi yang jauh lebih cepat, sehingga menunjukkan adanya *trade-off* antara akurasi dan efisiensi dalam pemilihan model.
Kelemahan/Research Gap	Belum mengevaluasi konsistensi performa IndoBERT pada berbagai dataset, domain, dan konfigurasi pemrosesan data yang berbeda.
Jurnal 4 [13]	
Penulis	Musthofa Ilmi, Ardi Sanjaya, Risky Aswi Ramadhani
Tahun	2025
Jurnal	SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi)
Volume	9
Nomor	-
Judul	Klasifikasi Berita Hoaks Bencana Alam Menggunakan Representasi

	IndoBERT dan Algoritma XGBoost
Metode	IndoBERT, XGBoost
Dataset	Dataset berita bencana alam
Hasil Penelitian	Pendekatan yang mengombinasikan IndoBERT sebagai ekstraksi fitur dan XGBoost sebagai algoritma klasifikasi menunjukkan kinerja sangat tinggi dalam deteksi berita hoaks, dengan akurasi mencapai 99,11% dan f1-*score* sebesar 0,99 pada kedua kelas, yang menegaskan efektivitas kombinasi kedua metode dalam menghasilkan klasifikasi yang akurat dan konsisten.
Kelemahan/Research Gap	Belum menguji kemampuan generalisasi model pada berbagai sumber berita Indonesia dengan karakteristik data yang lebih beragam
Jurnal 5 [14]	
Penulis	Muhammad Yusuf Ridho, Evi Yulianti
Tahun	2024
Jurnal	Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)
Volume	10
Nomor	3
Judul	From Text to Truth: Leveraging IndoBERT and Machine Learning Models for Hoax Detection in Indonesian News
Metode	IndoBERT Base, IndoBERT Large, CNN-LSTM, Logistic Regression, Naïve Bayes, Decision Tree, Random Forest, Gradient Boosting Classifier
Dataset	Dataset Berita Indonesia

Hasil Penelitian	Hasil menunjukkan bahwa IndoBERT Large memberikan performa terbaik dengan akurasi mencapai 98% setelah penerapan teknik *oversampling* SMOTE, diikuti IndoBERT Base sebesar 97%, sementara model lain seperti Naïve Bayes (78%), *Logistic Regression* (76%), *Random Forest* (72%), *Decision Tree* (67%), dan *Gradient Boosting* (70%) menunjukkan performa yang lebih rendah, serta CNN-LSTM mengalami penurunan performa hingga 44%.
Kelemahan/Research Gap	Belum menguji kemampuan generalisasi model pada data berita Indonesia yang lebih beragam dan distribusi kelas yang berbeda
Jurnal 6 [15]	
Penulis	Najwa Aliyah Pramono Putri, Citra Monepta Novaliza, dan Sugiarto Hartono
Tahun	2025
Jurnal	2025 International Conference on Computer Engineering, Network and Intelligent Multimedia (CENIM)
Volume	-
Nomor	-
Judul	Integrated RoBERTa and Machine Learning for Sentiment and Hoax Detection in Indonesian News
Metode	RoBERTa, Random Forest, Logistic Regression
Dataset	Dataset Berita Indonesia
Hasil Penelitian	Hasil menunjukkan bahwa RoBERTa mencapai akurasi 94% dalam analisis sentimen, sementara pada tugas deteksi hoaks, Random Forest memberikan performa terbaik dengan akurasi 96,92%, mengungguli

	Logistic Regression sebesar 93,97%.
Kelemahan/Research Gap	Belum menguji kemampuan generalisasi model pada berbagai sumber berita Indonesia dengan karakteristik data yang lebih beragam
Jurnal 7 [16]	
Penulis	Arnetta Rahmawati, Andry Alamsyah, dan Ade Romadhony
Tahun	2022
Jurnal	International Conference on Information and Communication Technology (ICoICT)
Volume	-
Nomor	-
Judul	Hoax News Detection Analysis using IndoBERT Deep Learning Methodology
Metode	IndoBERT, Support Vector Machine (SVM), Naïve Bayes
Dataset	Dataset Berita Indonesia
Hasil Penelitian	Hasil menunjukkan bahwa IndoBERT memberikan performa terbaik dengan akurasi sebesar 90%, serta didukung oleh nilai *precision*, *recall*, dan F1-*score* yang lebih tinggi dibandingkan model lainnya, sementara Naïve Bayes memperoleh akurasi 80,25% dan SVM sebesar 73,42% dengan performa metrik evaluasi yang relatif lebih rendah.
Kelemahan/Research Gap	Belum menguji performa model pada sumber berita Indonesia dengan karakteristik data yang lebih beragam
Jurnal 8 [17]	
Penulis	Tohpatti Crippa Praha, Widodo, dan Murien Nugraheni

Tahun	2024
Jurnal	International Journal on Informatics Visualization
Volume	8
Nomor	3
Judul	Indonesian Fake News Classification Using Transfer Learning in CNN and LSTM
Metode	CNN, LSTM
Dataset	Dataset Berita Indonesia
Hasil Penelitian	Hasil menunjukkan bahwa model CNN tanpa IndoBERT mencapai akurasi tertinggi sebesar 99,63%, sedangkan LSTM sebesar 92,91%. Setelah integrasi IndoBERT, performa model LSTM meningkat menjadi 97,76% dan mengungguli kombinasi IndoBERT+CNN yang memperoleh akurasi 96,64%, sehingga menunjukkan bahwa penggunaan *transfer learning* dengan IndoBERT mampu meningkatkan kinerja model, khususnya pada arsitektur LSTM.
Kelemahan/Research Gap	Belum dilakukan perbandingan efektivitas transfer learning pada berbagai representasi teks dan arsitektur model untuk klasifikasi hoaks berita Indonesia
Jurnal 9 [18]	
Penulis	Qiuping Li, Fen Fu, Yinjuan Li, Bhunnisa Wisassinthu, Wirapong Chansanam, dan Tossapon Boongoen
Tahun	2025
Jurnal	Journal of Computational and Cognitive Engineering
Volume	4

Nomor	4
Judul	Fake News Detection with Deep Learning: Insights from Multi-dimensional Model Analysis
Metode	BERT Large, RoBERTa Base, TextCNN, BERT-CLS XGBoost, Logistic Regression
Dataset	Dataset Berita Indonesia
Hasil Penelitian	Hasil menunjukkan bahwa BERT Large memberikan performa terbaik dengan akurasi sebesar 99,33%, diikuti oleh TextCNN sebesar 98,77%, RoBERTa sebesar 98,00%, model hybrid BERT+XGBoost sebesar 92,31%, dan *Logistic Regression* sebesar 86,46%, yang menegaskan keunggulan model berbasis *transformer* dalam memahami konteks semantik teks secara lebih mendalam.
Kelemahan/Research Gap	Perbandingan komprehensif antar model deteksi hoaks berdasarkan aspek akurasi, efisiensi, dan interpretabilitas masih terbatas
Jurnal 10 [19]	
Penulis	Lim Bodhi Wijaya, Yosef Nuraga Wicaksana, Sri Saraswati Widhiasari, dan Ari Saptawijaya
Tahun	2024
Jurnal	Buletin Pagelaran Mahasiswa Nasional Bidang Teknologi Informasi dan Komunikasi
Volume	2
Nomor	1
Judul	Pengembangan Model Deteksi Hoaks Berbahasa Indonesia Menggunakan Kombinasi IndoBERT dan BiLSTM
Metode	IndoBERT-BC (fine-tuning), IndoBERT-BiC (feature-based dengan

	BiLSTM), dan IndoBERT-CC (gabungan IndoBERT-BC dan IndoBERT-BiC)
Dataset	Dataset Berita Indonesia
Hasil Penelitian	Hasil menunjukkan bahwa model IndoBERT-CC memberikan performa terbaik dengan akurasi sebesar 98,79%, mengungguli IndoBERT-BC (98,56%) dan IndoBERT-BiC (97,50%), sehingga membuktikan bahwa kombinasi pendekatan *fine-tuning* dan *feature-based* mampu meningkatkan kinerja deteksi hoaks secara signifikan.
Kelemahan/Research Gap	Belum membandingkan efektivitas pendekatan berbasis transformer dan BiLSTM dalam mendeteksi hoaks berbahasa Indonesia.

Berdasarkan hasil sintesis terhadap sepuluh penelitian terdahulu, dapat disimpulkan bahwa pendekatan *machine learning* maupun *deep learning* mampu menghasilkan performa yang tinggi dalam deteksi berita hoaks berbahasa Indonesia. Berbagai algoritma seperti XGBoost, *Random Forest*, SVM, LSTM, CNN, BERT, dan IndoBERT menunjukkan tingkat akurasi yang baik, bahkan beberapa penelitian melaporkan akurasi di atas 95%. Model berbasis *transformer*, khususnya IndoBERT, menunjukkan keunggulan dalam memahami konteks semantik Bahasa Indonesia melalui representasi *contextual embedding* dan mekanisme *self-attention*. Selain itu, beberapa penelitian juga menunjukkan bahwa pendekatan model *hybrid* berpotensi meningkatkan performa klasifikasi dibandingkan penggunaan model tunggal.

Meskipun demikian, hasil analisis kritis terhadap penelitian terdahulu menunjukkan beberapa keterbatasan yang masih perlu diteliti lebih lanjut. Sebagian besar penelitian menggunakan dataset yang berasal dari sumber atau *domain* tertentu sehingga kemampuan generalisasi model terhadap sumber berita yang lebih beragam belum banyak dievaluasi. Beberapa penelitian juga hanya berfokus pada satu pendekatan pemodelan tanpa melakukan perbandingan secara komprehensif

antara metode *machine learning*, *deep learning*, dan model *hybrid*. Selain itu, pengujian menggunakan data eksternal masih relatif terbatas sehingga ketahanan model dalam menghadapi data baru di luar data pelatihan belum dapat diketahui secara menyeluruh. Meskipun berbagai penelitian telah melaporkan tingkat akurasi yang tinggi, efektivitas peningkatan performa yang diperoleh dari penggunaan model yang lebih kompleks, khususnya model *hybrid*, masih belum banyak dianalisis secara komprehensif. Pada penelitian yang menerapkan model *hybrid*, kombinasi IndoBERT dan XGBoost untuk deteksi hoaks berbahasa Indonesia juga masih relatif sedikit dieksplorasi.

Berdasarkan celah penelitian tersebut, penelitian ini mengembangkan dan membandingkan tiga model klasifikasi, yaitu XGBoost, IndoBERT, dan model *hybrid* yang memanfaatkan *embedding* IndoBERT sebagai fitur masukan bagi XGBoost. Penelitian ini menggunakan data yang berasal dari beberapa sumber berita, yaitu AntaraNews, Detik, Kompas, TurnBackHoax, dan Komdigi, serta melakukan *external testing* menggunakan data hasil *web scraping*. Dengan pendekatan tersebut, penelitian ini diharapkan dapat memberikan evaluasi yang lebih komprehensif mengenai pengaruh keberagaman sumber data terhadap performa model, kemampuan generalisasi model terhadap data baru, serta efektivitas pendekatan model *hybrid* dibandingkan model tunggal dalam mendeteksi berita hoaks berbahasa Indonesia.

2.2 Teori yang berkaitan

2.2.1 Literasi *Digital*

Literasi *digital* dapat didefinisikan sebagai kemampuan individu untuk mencari, memahami, mengevaluasi, dan memanfaatkan informasi yang diperoleh melalui media digital secara tepat dan bertanggung jawab [20]. Dalam menghadapi penyebaran hoaks di Indonesia, literasi digital menjadi sangat penting karena membantu masyarakat lebih teliti dalam menilai kebenaran informasi yang beredar di internet [21]. Kompetensi ini tidak hanya mencakup keterampilan teknis menggunakan teknologi, tetapi juga melibatkan berpikir kritis dalam menilai isi dan kredibilitas sumber informasi. Menurut Lemhannas RI (2019), rendahnya

kemampuan masyarakat dalam membedakan informasi sahih dan hoaks dipengaruhi oleh keterbatasan pemahaman literasi *digital*, terutama pada aspek evaluasi informasi, berpikir kritis, dan penggunaan media *digital* yang bijak [22]. Hal serupa ditemukan oleh Ismail Zaky Al-Fatih (2024), yang menunjukkan hubungan kuat antara literasi *digital* dan kemampuan seseorang dalam menanggapi informasi secara kritis dan objektif [23]. Individu dengan tingkat literasi digital yang baik cenderung lebih mampu memeriksa keandalan sumber berita, memahami konteks informasi, dan mengurangi kerentanan terhadap informasi yang menyesatkan. Oleh karena itu, peningkatan literasi *digital* menjadi langkah penting untuk membantu masyarakat menyaring dan menggunakan informasi *digital* secara lebih bijaksana, sehingga dampak negatif penyebaran hoaks dapat diminimalkan.

2.2.2 Berita Hoaks

Berita hoaks merupakan informasi atau pernyataan yang berisi data tidak benar, menyesatkan, atau bahkan bersifat fiktif yang sengaja disebarkan untuk mempengaruhi masyarakat, menimbulkan kepanikan, maupun memicu rasa takut [24]. Penyebaran hoaks sering kali terjadi karena masih banyak masyarakat yang langsung mempercayai informasi tanpa melakukan pengecekan terlebih dahulu, terutama jika informasi tersebut berasal dari sumber yang dianggap terpercaya. Pada umumnya, hoaks dibuat untuk membentuk persepsi tertentu dan mempengaruhi opini publik terhadap suatu isu [25]. Perkembangan internet dan media sosial yang sangat pesat membuat penyebaran informasi menjadi jauh lebih cepat dan mudah menjangkau banyak orang dalam waktu singkat. Akibatnya, berita yang tidak benar juga dapat tersebar secara luas tanpa adanya proses verifikasi yang memadai. Kondisi ini meningkatkan risiko penyebaran hoaks di ruang *digital* dan menunjukkan pentingnya kemampuan berpikir kritis dalam menyikapi informasi. Oleh karena itu, masyarakat perlu lebih bijak dalam memeriksa keabsahan informasi dan menilai kredibilitas sumber sebelum mempercayai atau menyebarkannya kepada orang lain [26].

2.2.3 Web Application

Penggunaan aplikasi *web* dalam bidang deteksi berita hoaks dinilai mampu menjadi sarana yang efektif untuk membantu masyarakat memperoleh hasil

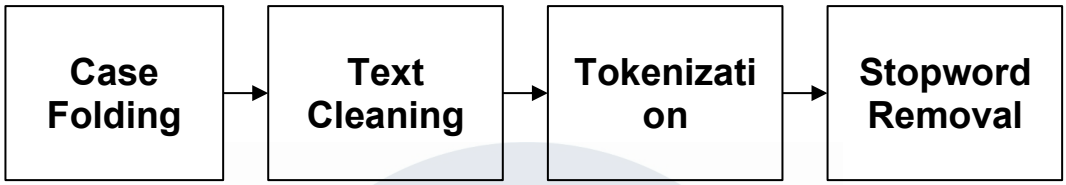
verifikasi informasi secara cepat dan mudah diakses [27]. Dengan memanfaatkan *framework* Streamlit, aplikasi dapat dikembangkan secara interaktif sehingga proses identifikasi berita hoaks dapat dilakukan secara otomatis dan *real-time* melalui dukungan model *machine learning* maupun *deep learning* [28]. Dalam penelitian ini dilakukan implementasi model deteksi hoaks ke dalam sebuah aplikasi *web*. Pengguna dapat memasukkan teks berita ke sistem, lalu aplikasi akan menjalankan tahap *text preprocessing* dan klasifikasi menggunakan model yang telah dilatih sebelumnya. Setelah proses analisis selesai, aplikasi menampilkan prediksi untuk membantu pengguna mengetahui apakah berita tersebut dikategorikan sebagai hoaks atau non-hoaks. Tampilan hasil yang sederhana dan mudah dipahami diharapkan dapat membantu pengguna melakukan pemeriksaan awal terhadap validitas suatu informasi dengan lebih praktis. Selain berfungsi sebagai alat bantu deteksi hoaks, aplikasi ini juga diharapkan dapat mendukung peningkatan literasi digital masyarakat dengan mendorong pengguna untuk lebih kritis dalam menyikapi informasi yang beredar di media *digital*.

2.2.4 Natural Language Processing dan Text Preprocessing

Natural Language Processing (NLP) adalah cabang ilmu komputer yang fokus pada pengembangan sistem agar mampu memahami dan memproses bahasa manusia secara alami. Dengan NLP, komputer dapat mengolah, menafsirkan, dan menghasilkan teks dengan cara yang mendekati kemampuan komunikasi manusia [29]. Salah satu penerapan populer NLP adalah analisis sentimen, yaitu teknik untuk mengidentifikasi dan mengelompokkan opini dalam teks menjadi kategori seperti positif, negatif, atau netral. Dalam penelitian ini, NLP dipakai untuk mendukung identifikasi hoaks melalui analisis pola bahasa pada teks berita [30].

Sebelum data teks dipakai untuk analisis dan pemodelan, dilakukan serangkaian tahap pra-pemrosesan teks (*text preprocessing*). Tahapan ini krusial karena kualitas hasil *text preprocessing* sangat mempengaruhi performa model [31]. Secara umum, langkah-langkah *text preprocessing* meliputi *case folding*, *tokenization*, dan *stopword removal* [32]. Alur *text preprocessing* yang dipakai dalam penelitian ini

ditampilkan pada Gambar 2.1 sebagai dasar pengolahan data teks sebelum memasuki tahap pemodelan.



Gambar 2.1 Tahapan Text Preprocessing

Case folding adalah penyeragaman teks dengan mengubah semua karakter menjadi huruf kecil agar format data lebih konsisten, umumnya dilakukan di Python dengan fungsi *lower()* [33]. Tahap ini bertujuan untuk mengurangi variasi penulisan yang disebabkan oleh perbedaan penggunaan huruf kapital dan huruf kecil sehingga kata yang memiliki makna sama dapat direpresentasikan secara konsisten oleh model. Langkah berikutnya adalah *text cleaning* atau normalisasi teks, yaitu pembersihan dari elemen tak diinginkan seperti tanda baca, angka, simbol, URL, karakter khusus, dan spasi berlebih yang dapat mengganggu proses analisis teks. Tahap ini mencakup berbagai teknik untuk memperbaiki ketidaksempurnaan data mentah sehingga siap diproses lebih lanjut [34]. Selain itu, *text cleaning* dilakukan untuk mengurangi *noise* pada data sehingga informasi yang relevan dapat diekstraksi dengan lebih baik oleh model klasifikasi. Setelah itu dilakukan *tokenization*, yaitu pemecahan teks menjadi unit-unit kecil (*token*) yang umumnya berupa kata-kata [35]. Tahap ini diperlukan agar teks dapat direpresentasikan dalam bentuk yang dapat diproses oleh algoritma *machine learning* maupun *deep learning* untuk melakukan analisis lebih lanjut. Tahap terakhir adalah *stopword removal*, yaitu penghilangan kata-kata umum yang sering muncul namun memberikan kontribusi kecil terhadap makna, seperti kata penghubung. Dalam penelitian ini, penghapusan *stopword* menggunakan daftar dari pustaka NLP Python, NLTK, agar fitur teks yang dihasilkan lebih relevan untuk klasifikasi [35]. Tahap ini bertujuan untuk mengurangi kata-kata yang kurang informatif sehingga model dapat lebih fokus pada kata-kata yang memiliki kontribusi penting dalam membedakan berita hoaks dan non-hoaks. Keseluruhan rangkaian *text preprocessing* ini berperan penting dalam meningkatkan kualitas data teks, mengurangi *noise*, menyeragamkan

representasi kata, serta membantu model mengekstraksi informasi yang lebih relevan sehingga dapat menghasilkan performa klasifikasi yang lebih baik.

2.2.5 Machine Learning

Machine learning merupakan cabang dari *Artificial Intelligence* (AI) yang memungkinkan komputer mempelajari pola dari data dan membuat keputusan otomatis tanpa pemrograman eksplisit untuk setiap tugas tertentu [37]. Teknologi ini memungkinkan sistem belajar dari pengalaman dan memperbaiki performanya berdasarkan data yang diproses, sehingga banyak dimanfaatkan untuk menyelesaikan masalah secara cepat, efisien, dan adaptif [38].

Berbagai studi telah mengevaluasi algoritma *machine learning* seperti *Logistic Regression*, *Decision Tree*, *Gradient Boosting*, dan *Random Forest* untuk tugas-tugas klasifikasi. Hasil penelitian menunjukkan bahwa pendekatan *machine learning* mampu memberikan performa yang baik, termasuk dalam mendeteksi hoaks dengan akurasi tinggi [39]. Secara umum, *machine learning* membangun model dari pola dan hubungan dalam dataset melalui dua tahapan utama: training untuk melatih model dengan data berlabel, dan testing untuk mengukur kemampuan model memprediksi data baru [40].

Dalam praktik, *machine learning* dibagi menjadi tiga pendekatan utama [41]:

1. *Supervised Learning*

   Model dilatih menggunakan data berlabel untuk mempelajari hubungan antara fitur dan label, umum dipakai pada klasifikasi dan regresi.

2. *Unsupervised Learning*

   Model bekerja pada data tanpa label untuk menemukan pola atau struktur tersembunyi, sering digunakan pada *clustering* dan reduksi dimensi.

3. *Reinforcement Learning*

   Model belajar melalui interaksi dengan lingkungan dengan menerima umpan balik (*reward*/*punishment*), sehingga dapat memperbaiki strategi

dan meningkatkan performanya secara bertahap melalui proses *trial* dan *error*.

Penelitian ini menerapkan *supervised learning*, model dikembangkan menggunakan dataset berlabel yang terbagi menjadi dua kelas, yakni hoaks dan non-hoaks. Melalui proses pelatihan tersebut, model dapat mempelajari pola bahasa dan karakteristik setiap kelas sehingga mampu dalam mengklasifikasikan berita baru dengan akurasi yang memadai.

2.2.6 Ensemble Learning

Ensemble Learning adalah pendekatan yang menggabungkan beberapa model dalam meningkatkan kualitas prediksi. Model-model dasar (*weak learners*) yang memiliki kemampuan prediksi yang masih terbatas apabila digunakan secara individual, dikombinasikan sehingga menghasilkan performa yang lebih baik, stabil, dan akurat dibandingkan satu model tunggal. Secara umum, metode *ensemble* dibagi menjadi tiga kategori, yakni *bagging*, *boosting*, dan *stacking* [42].

1. *Bagging*

Metode Bagging (*Bootstrap Aggregating*) membangun beberapa model paralel dengan setiap model dilatih pada *subset* data yang diambil secara acak melalui *bootstrap sampling*. Prediksi akhir muncul dari penggabungan hasil model-model tersebut guna meningkatkan stabilitas dan akurasi [43].

2. *Stacking*

Metode *Stacking* menggabungkan keluaran beberapa *base learner* (*level-0*) terlebih dahulu menghasilkan prediksi terhadap data, yang selanjutnya digunakan sebagai masukan bagi *meta learner* (*level-1*) dalam menghasilkan prediksi akhir. Mekanisme bertingkat inilah menjadi ciri khas *stacking* [44].

3. *Boosting*

Metode *Boosting* membangun model secara berurutan, dimana setiap model baru difokuskan dalam memperbaiki kesalahan prediksi dari model sebelumnya. Dalam proses pelatihannya, data diberikan bobot tertentu yang akan diperbarui berdasarkan tingkat kesalahan pada iterasi sebelumnya. Dengan mekanisme

tersebut, model akan lebih fokus pada data yang sulit diklasifikasikan sehingga performa prediksi dapat meningkat secara bertahap dan *weak learner* dapat berkembang menjadi *strong learner* [43],[45].

Pada penelitian ini, algoritma XGBoost termasuk ke dalam kategori metode *boosting*. XGBoost bekerja dengan membangun serangkaian *decision tree* secara bertahap untuk menghasilkan model klasifikasi yang kuat. Pada setiap iterasi, algoritma memberi perhatian lebih pada data yang sebelumnya salah prediksi untuk memperbaiki kesalahan pada langkah berikutnya. Pendekatan ini membantu algoritma XGBoost dalam mencapai performa klasifikasi tinggi dengan efisiensi komputasi, dengan adanya keefektifikan dalam proses penanganan data yang berdimensi besar serta pendistribusian data yang tidak seimbang. Oleh sebab itu, XGBoost dinilai sesuai untuk diterapkan dalam proses deteksi berita hoaks pada media *digital* berbahasa Indonesia. Tabel 2.2 menyajikan perbandingan karakteristik antara algoritma XGBoost dan algoritma *ensemble learning* lainnya, yaitu *Random Forest*.

Tabel 2.2 Komparasi XGBoost dan Random Forest

Aspek	XGBoost	Random Forest
Metode Ensemble	Boosting	Bagging
Mekanisme Pembentukan Pohon	Pohon dibangun secara berurutan, dimana setiap pohon memperbaiki kesalahan pohon sebelumnya	Pohon dibangun secara independen dan paralel
Penanganan Bias dan Variance	Lebih efektif mengurangi bias dan variance	Lebih fokus mengurangi variance
Regularisasi	Lebih kompleks dan membutuhkan tuning hyperparameter	Tidak memiliki mekanisme regularisasi bawaan
Kecepatan Komputasi	Umumnya menghasilkan akurasi yang lebih tinggi	Performa baik, namun sering kalah dari XGBoost pada banyak kasus

2.2.7 Deep Learning

Konsep *deep learning* awalnya dikemukakan oleh Marton dan Säljö pada tahun 1976 sebagai pendekatan yang menekankan pemahaman yang lebih dalam terhadap materi serta hubungan antar konsepnya. Pendekatan ini melampaui pemrosesan kognitif sederhana dengan menekankan penguasaan makna yang menyeluruh [46]. Seiring kemajuan komputasi dan kecerdasan buatan, istilah dari *deep learning* sendiri berevolusi menjadi teknik di bidang *Artificial Intelligence* yang memanfaatkan jaringan saraf tiruan berlapis (*multi-layer artificial neural networks*) untuk memproses dan mempelajari pola dari data [47]. Dalam penelitian ini, pendekatan *deep learning* diwujudkan melalui pemanfaatan model IndoBERT dalam mendukung proses analisis pada deteksi berita hoaks berbahasa Indonesia.

Dalam penelitian ini, pendekatan *deep learning* diwujudkan melalui penggunaan IndoBERT, yaitu adaptasi BERT yang telah dilatih secara khusus menggunakan korpus teks berbahasa Indonesia. Dengan memanfaatkan *contextual embedding*, IndoBERT mampu menangkap hubungan semantik, konteks kalimat, serta karakteristik linguistik Bahasa Indonesia secara lebih efektif dibandingkan metode berbasis representasi statistik tradisional. Kemampuan tersebut menjadikan IndoBERT relevan untuk tugas klasifikasi teks, termasuk deteksi berita hoaks, karena model dapat mengidentifikasi pola bahasa dan informasi konteks [48],[49].

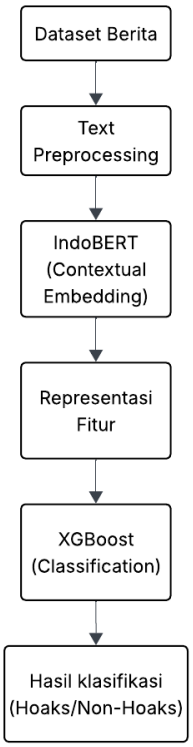
2.2.8 Teori Hybrid

Model *hybrid* adalah pendekatan dalam *machine learning* yang mengkombinasikan lebih dari satu metode atau algoritma untuk mencapai kinerja yang lebih baik dibandingkan model tunggal. Pendekatan ini memanfaatkan keunggulan masing-masing algoritma sehingga sistem lebih mampu menangani masalah yang kompleks [43]. Dengan menggabungkan beberapa teknik, model *hybrid* tidak hanya memperbaiki kemampuan generalisasi tetapi juga mengurangi kelemahan yang muncul bila setiap metode digunakan sendiri-sendiri. Selain itu, model *hybrid* berguna untuk mengatasi kendala seperti keterbatasan jumlah data, ketidakseimbangan kelas, dan kompleksitas karakteristik data [50]. Beberapa studi melaporkan bahwa pendekatan model *hybrid* menghasilkan performa klasifikasi yang lebih stabil dan lebih tahan terhadap masalah *class imbalance* dibandingkan

model tunggal [51]. Dalam bidang analisis sentimen maupun klasifikasi teks, penggunaan model *hybrid* juga sering memberikan hasil evaluasi yang lebih baik dibandingkan pendekatan konvensional lainnya [52]. Bahkan, beberapa studi melaporkan bahwa penggabungan beberapa teknik dalam satu model dapat meningkatkan nilai evaluasi seperti *accuracy*, *precision*, *recall*, dan *F1-score*, dengan peningkatan performa mencapai sekitar 8–16% dibandingkan penggunaan model individual [53]. Oleh karena itu, pendekatan model *hybrid* pada penelitian ini dipandang relevan untuk meningkatkan kemampuan sistem dalam mendeteksi berita hoaks pada media *digital* berbahasa Indonesia.

Secara teoritis, kombinasi IndoBERT dan XGBoost dalam model *hybrid* didasarkan pada karakteristik dan keunggulan yang saling melengkapi. IndoBERT merupakan model berbasis *Transformer* yang mampu menghasilkan contextual embedding sehingga dapat memahami hubungan semantik, konteks kalimat, serta makna kata berdasarkan penggunaannya dalam teks. Namun, proses klasifikasi akhir pada model berbasis *Transformer* umumnya dilakukan menggunakan lapisan klasifikasi sederhana (*fully connected layer*) yang mungkin belum sepenuhnya optimal dalam memanfaatkan representasi fitur yang kompleks. Di sisi lain, XGBoost merupakan algoritma *ensemble* berbasis *gradient boosting* yang memiliki kemampuan kuat dalam mempelajari pola *non-linear*, melakukan seleksi fitur secara efektif, serta menghasilkan klasifikasi yang akurat pada data berdimensi tinggi. Oleh karena itu, penelitian ini memanfaatkan representasi fitur hasil ekstraksi IndoBERT sebagai masukan bagi XGBoost untuk proses klasifikasi. Melalui pendekatan tersebut, model *hybrid* diharapkan mampu menggabungkan kemampuan pemahaman konteks bahasa dari IndoBERT dengan kemampuan klasifikasi berbasis *boosting* dari XGBoost, sehingga menghasilkan performa deteksi berita hoaks yang lebih baik dibandingkan penggunaan masing-masing model secara terpisah.

Gambar 2.2 menampilkan visualisasi *diagram* arsitektur model *hybrid* yang digunakan dalam penelitian ini, yakni sebagai berikut:



Gambar 2.2 Arsitektur model *hybrid*

Pada visualisasi arsitektur model *hybrid* ditunjukkan alur pengolahan data yang menggabungkan kemampuan representasi bahasa dari IndoBERT dengan kemampuan klasifikasi dari XGBoost. Proses diawali dengan dataset berita yang melalui tahap *text preprocessing* untuk membersihkan dan menyiapkan data teks sebelum diproses lebih lanjut. Selanjutnya, teks yang telah di praproses dimasukkan ke dalam model IndoBERT untuk menghasilkan *contextual embedding*, yaitu representasi fitur yang mampu menangkap makna kata berdasarkan konteks kalimat. Hasil representasi fitur tersebut kemudian digunakan sebagai masukan bagi algoritma XGBoost untuk melakukan proses klasifikasi. Melalui pendekatan ini, model *hybrid* memanfaatkan kemampuan IndoBERT dalam memahami konteks semantik bahasa Indonesia serta kemampuan XGBoost dalam membangun model klasifikasi yang efektif. Pada tahap akhir, model menghasilkan keluaran berupa klasifikasi berita ke dalam kategori Hoaks atau Non-Hoaks.

2.2.9 Confusion Matrix

Confusion matrix adalah metode evaluasi yang digunakan untuk menilai kinerja model klasifikasi dengan membandingkan prediksi model terhadap label sebenarnya pada data uji. Melalui metode ini, jumlah prediksi benar dan salah terlihat jelas, sehingga memudahkan dalam identifikasi terhadap pola kesalahan model [54]. Karena mudah dipahami dan cukup efektif, *confusion matrix* seringkali digunakan sebagai teknik evaluasi dalam penelitian klasifikasi [55]. Secara umum, *confusion matrix* terdiri dari empat komponen utama yang menggambarkan hasil prediksi model [56], yaitu:

1. *True Positive* (TP): Model memprediksi kelas positif dan label sebenarnya juga positif.
2. *True Negative* (TN): Model memprediksi kelas negatif dan label sebenarnya negatif.
3. *False Positive* (FP): Model memprediksi positif padahal label sebenarnya negatif.
4. *False Negative* (FN): Model memprediksi negatif padahal label sebenarnya positif.

Berdasarkan keempat komponen tersebut, performa model dapat dianalisis menggunakan beberapa metrik evaluasi standar, yaitu *Accuracy*, *Precision*, *Recall*, dan *F1-Score*, yang dijelaskan sebagai berikut:

1. *Accuracy* (Akurasi)

   Accuracy mengukur tingkat ketepatan keseluruhan model dalam membuat prediksi [57]. Nilai *accuracy* diperoleh dengan membandingkan jumlah prediksi benar terhadap total data yang diuji. Pada *confusion matrix*, nilai ini dihitung dari jumlah prediksi benar pada diagonal utama dibandingkan dengan total data yang diuji. Rumus 2.1 menyajikan perhitungan *accuracy* sebagai berikut:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Rumus 2.1 Rumus *Accuracy*

2. *Precision* (Presisi)

   Precision menilai ketepatan dalam memprediksi positif yang dibuat model, yang dimana menunjukan seberapa besar prediksi prediksi positif yang dihasilkan model benar-benar sesuai dengan kondisi sebenarnya [57]. *Precision* dihitung dengan membandingkan *true positive* terhadap seluruh data yang diprediksi positif oleh model. Rumus 2.2 menyajikan perhitungan *precision* sebagai berikut:

$$Precision = \frac{TP}{TP + FP}$$

Rumus 2.2 Rumus *Precision*

3. *Recall* (Sensivitas)

   Recall mengukur kemampuan model dalam menemukan seluruh kelas positif yang sebenarnya, yakni proporsi data positif yang berhasil dikenali oleh model [58]. Metrik ini menunjukkan seberapa banyak data positif yang berhasil dikenali dengan benar oleh model. Nilai recall dihitung dari perbandingan antara *true positive* dengan jumlah keseluruhan data positif. Rumus 2.3 menyajikan perhitungan *recall* sebagai berikut:

$$Recall = \frac{TP}{TP + FN}$$

Rumus 2.3 Rumus *Recall*

4. *F1-Score*

   F1-Score adalah rata-rata harmonik antara *precision* dan *recall*, yang dimana sangat berguna terutama pada dataset dengan distribusi kelas tidak seimbang karena memberikan gambaran kinerja yang lebih representatif dibandingkan hanya menggunakan *accuracy* [58]. Nilai F1-*Score* menunjukkan keseimbangan antara ketepatan prediksi positif dan

kemampuan mendeteksi kasus positif. Rumus 2.4 menyajikan perhitungan F1-*Score* sebagai berikut:

$$F1\text{-}Score = 2 * \frac{precision * recall}{precision + recall}$$

Rumus 2.4 Rumus *F1-Score*

2.3 Framework/Algoritma yang digunakan

2.3.1 IndoBERT (*Bidirectional Encoder Representations from Transformers*)

IndoBERT adalah model Natural Language Processing (NLP) yang dirancang secara khusus dalam menangani Bahasa Indonesia. Meskipun penggunaan Bahasa Indonesia di internet terus meningkat, pengembangan teknologi NLP untuk bahasa ini masih menghadapi berbagai tantangan, terutama keterbatasan sumber daya linguistik dan ketersediaan dataset dalam jumlah besar. Untuk memenuhi kebutuhan tersebut, IndoNLU mengembangkan IndoBERT sebagai adaptasi dari arsitektur BERT yang dilatih menggunakan dataset Indo4B. Dataset tersebut berisi sekitar empat miliar kata Bahasa Indonesia yang dikumpulkan dari berbagai sumber teks dan telah melalui tahap *preprocessing* sebelumnya [59].

Dalam pemrosesan teks, IndoBERT menerapkan metode *WordPiece tokenization*, yaitu teknik yang memecah kata menjadi unit-unit sub-kata (*sub-word*). Pendekatan ini memungkinkan model mengatasi permasalahan *out-of-vocabulary* (OOV), seperti penggunaan kata tidak baku, kesalahan penulisan (*typo*), maupun bahasa gaul, karena model masih dapat mengenali bagian kata yang telah dipelajari sebelumnya. Berbeda dengan metode representasi kata konvensional seperti GloVe yang menghasilkan representasi tetap untuk setiap kata, IndoBERT mampu memahami makna kata berdasarkan konteks penggunaannya dalam kalimat. Kemampuan tersebut didukung oleh mekanisme *self-attention* yang menjadi komponen utama pada arsitektur *transformer* yang digunakan oleh IndoBERT [60].

Transformer merupakan arsitektur jaringan saraf yang saat ini banyak digunakan dalam berbagai tugas *Natural Language Processing* (NLP). Popularitasnya

didukung oleh kemampuannya dalam memproses data secara paralel, menangani dependensi jangka panjang antar-kata, serta memanfaatkan data pelatihan dalam skala besar secara efisien [61]. Perkembangan arsitektur dan teknik *pre-training* juga memungkinkan pembangunan model dengan kapasitas yang lebih besar sehingga dapat menghasilkan representasi bahasa yang lebih kaya dan mendukung berbagai tugas NLP secara efektif [62]. Salah satu komponen utama yang mendukung kinerja *transformer* adalah mekanisme *self-attention*, yaitu proses yang memungkinkan model menentukan tingkat perhatian terhadap setiap token dalam suatu kalimat berdasarkan hubungan kontekstual antar-token. Perhitungan mekanisme *self-attention* tersebut ditunjukkan pada Rumus 2.5 sebagai berikut:

$$Attention(\mathrm{Q}, \mathrm{K}, \mathrm{V}) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)\mathrm{V}$$

Rumus 2.5 Rumus *Self-Attention* IndoBERT

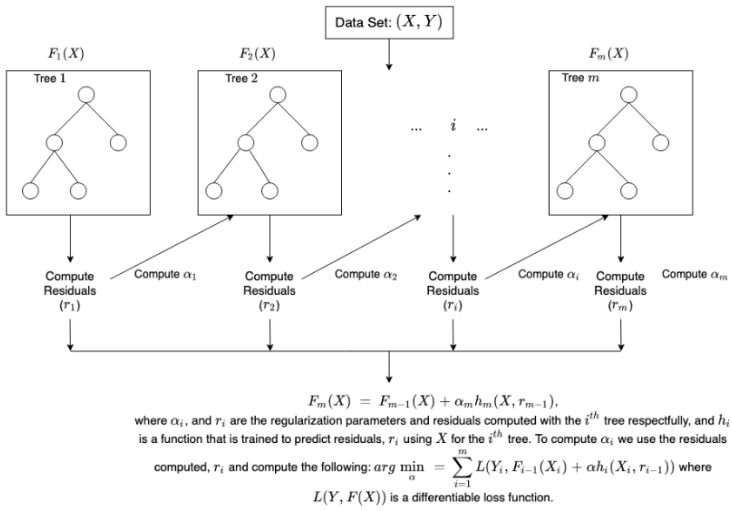
Pada persamaan tersebut, *Query* (Q), *Key* (K), dan *Value* (V) merupakan representasi vektor dari token masukan yang digunakan dalam mekanisme *self-attention*. Ketiga komponen tersebut berperan dalam menentukan tingkat keterkaitan antar-token di dalam suatu kalimat. Proses perhitungan diawali dengan membandingkan *Query* (Q) terhadap *Key* (K) untuk menghasilkan skor perhatian (*attention score*) yang menunjukkan tingkat relevansi antara satu token dengan token lainnya. Selanjutnya, skor tersebut dinormalisasi dengan membaginya menggunakan akar kuadrat dari dimensi *Key* ($\sqrt{d_k}$) guna menjaga stabilitas nilai perhitungan. Hasil normalisasi kemudian diproses menggunakan fungsi *softmax* untuk menghasilkan bobot perhatian (*attention weight*) yang merepresentasikan tingkat kepentingan setiap token. Bobot perhatian tersebut selanjutnya dikalikan dengan *Value* (V) sehingga menghasilkan representasi kontekstual yang telah mempertimbangkan informasi dari seluruh token dalam urutan masukan [63]. Melalui mekanisme ini, IndoBERT mampu menangkap hubungan semantik dan konteks antar-kata secara lebih efektif, sehingga dapat menghasilkan representasi teks yang lebih kaya dan mendukung berbagai tugas klasifikasi teks, termasuk deteksi berita hoaks berbahasa Indonesia.

2.3.2. XGBoost (*eXtreme Gradient Boosting*)

XGBoost (*eXtreme Gradient Boosting*) adalah algoritma *ensemble learning* berbasis *boosting* yang diperkenalkan oleh Tianqi Chen pada tahun 2014. Algoritma ini dikembangkan dari konsep *gradient boosting*, yaitu pendekatan pembelajaran yang berfokus pada perbaikan kesalahan prediksi dari model sebelumnya [64]. Dalam implementasinya, XGBoost banyak digunakan untuk menyelesaikan permasalahan klasifikasi maupun regresi dengan memanfaatkan kerangka *Gradient Boosting Decision Tree* (GBDT) [65].

Sebagai algoritma berbasis *boosting*, XGBoost membangun sejumlah *decision tree* secara bertahap dan berurutan. Setiap pohon keputusan yang baru dibentuk bertujuan untuk mengoreksi kesalahan yang masih ada dalam prediksi pada pohon sebelumnya sehingga kinerja model meningkat seiring iterasi [66]. Selama proses pelatihan, bobot model akan terus diperbarui agar hasil prediksi menjadi lebih *optimal* dan akurat [6366

Selain memiliki kemampuan klasifikasi yang baik, XGBoost juga dikenal karena efisiensi serta kestabilannya dalam proses komputasi. Algoritma ini dilengkapi dengan mekanisme regularisasi yang berfungsi untuk mengendalikan kompleksitas model dan mengurangi dari adanya risiko *overfitting*. Walaupun menerapkan regularisasi, XGBoost tetap dapat bekerja cepat dan hemat memori, sehingga banyak dipilih dalam penelitian dan aplikasi berbasis *machine learning* [67].

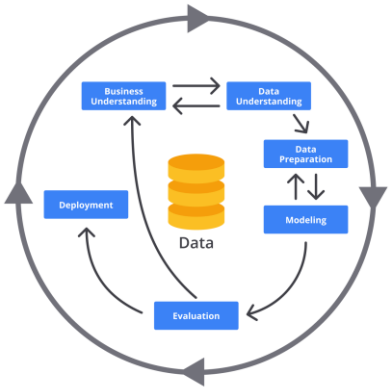


Gambar 2.3 *XGBoost Classifier*

Sumber: [68]

Cara kerja XGBoost dapat dipahami sebagai proses pembentukan *decision tree* yang menggunakan fitur-fitur data sebagai titik percabangan (*conditional node*). Struktur pohon berkembang hingga mencapai *leaf node* yang mewakili keputusan klasifikasi [69]. Pada setiap iterasi pelatihan, XGBoost menghitung nilai residual untuk mengukur tingkat kesalahan prediksi model sebelumnya. Nilai residual tersebut kemudian dimanfaatkan untuk membangun pohon berikutnya agar kesalahan prediksi dapat diperbaiki secara bertahap [70]. Proses ini berlangsung secara berulang hingga residual mencapai batas tertentu (*threshold*) yang dianggap optimal dalam menentukan kelas data. Selain menghasilkan performa klasifikasi yang tinggi, XGBoost juga memiliki berbagai keunggulan praktis, seperti waktu komputasi yang relatif cepat dan penggunaan memori yang efisien. Kombinasi kemampuan tersebut membuat XGBoost banyak diterapkan di berbagai *domain*, mulai dari kesehatan, analisis risiko kredit, hingga riset metagenomik [71].

2.3.3. CRISP-DM (*Cross-Industry Standard Process for Data Mining*)



Gambar 2.4 Metode CRISP-DM

Sumber: [72]

CRISP-DM (*Cross-Industry Standard Process for Data Mining*) adalah salah satu metodologi yang umum dipakai dalam kegiatan proses *data mining*. Metode ini dikembangkan oleh konsorsium perusahaan dengan dukungan dari Komisi Eropa pada tahun 1996 dan hingga sekarang masih sering dijadikan acuan dalam berbagai penelitian maupun proyek pengolahan data [73]. CRISP-DM dikenal memiliki alur kerja yang terstruktur, sistematis, serta fleksibel sehingga sesuai diterapkan pada proyek yang memiliki tujuan analisis yang jelas. Melalui pendekatan tersebut, proses pengolahan data dapat dilakukan secara lebih terarah sekaligus membantu memahami kebutuhan bisnis dan eksplorasi data secara mendalam [74],[75]. Metodologi CRISP-DM tersusun dari enam tahapan yang saling terhubung dan membentuk sebuah siklus proses, yakni sebagai berikut:

1. *Business Understanding*

   Tahap *business understanding* merupakan fase awal dalam metodologi CRISP-DM yang berfokus pada pemahaman tujuan, kebutuhan, serta permasalahan bisnis secara menyeluruh. Pada tahap ini, permasalahan yang ada diterjemahkan menjadi permasalahan *data mining* yang lebih terstruktur sehingga dapat dijadikan dasar dalam menentukan strategi dan arah penelitian.

2. *Data Understanding*

   Tahap *data understanding* dilakukan setelah kebutuhan bisnis dan tujuan penelitian dipahami dengan baik. Proses ini dimulai dengan pengumpulan data awal dan identifikasi informasi yang relevan untuk analisis. Selanjutnya, data dipelajari lebih lanjut melalui pemeriksaan kualitas data serta pencarian pola-pola tertentu yang dapat mendukung proses penelitian.

3. *Data Preparation*

   Tahap *data preparation* berfokus pada proses persiapan dataset agar siap digunakan pada tahap pemodelan. Pada fase ini dilakukan berbagai proses seperti seleksi atribut yang relevan, pembersihan data, transformasi data, hingga pembentukan dataset akhir. Tahapan ini bersifat iteratif sehingga dapat dilakukan berulang kali sesuai kebutuhan model yang akan dikembangkan.

4. *Modeling*

   Tahap *modeling* merupakan proses pembangunan model setelah data dinyatakan siap digunakan. Pada fase ini dipilih metode atau algoritma yang sesuai, kemudian dilakukan proses pelatihan model dan optimasi parameter guna memperoleh performa terbaik. Dalam implementasinya, beberapa model dapat dibangun secara bersamaan sehingga memungkinkan adanya penyesuaian kembali pada tahap persiapan data apabila diperlukan.

5. *Evaluation*

   Setelah model selesai dibangun, tahap *evaluation* dilakukan untuk menilai apakah model telah memenuhi tujuan penelitian dan kebutuhan bisnis yang ditetapkan sebelumnya. Tahap ini penting dilakukan sebelum model diterapkan secara nyata agar kualitas, keandalan, dan performa prediksi model dapat dipastikan terlebih dahulu.

6. *Deployment*

Tahap *deployment* adalah fase penutup dalam kerangka CRISP-DM. Pada tahap ini, model dan temuan analisis yang telah dibangun diimplementasikan atau disajikan dalam bentuk yang bisa langsung dimanfaatkan oleh pengguna dan *stakeholder*. Dengan demikian, keluaran penelitian tidak sekedar berhenti pada evaluasi, melainkan juga diwujudkan dalam aplikasi praktis yang dapat diterapkan pada kondisi nyata.

2.4 Tools/software yang digunakan

2.4.1 Python

Python merupakan bahasa pemrograman tingkat tinggi yang diperkenalkan oleh Guido van Rossum pada tahun 1991. Seiring waktu, Python menjadi salah satu bahasa pemrograman yang seringkali digunakan karena dengan adanya ketersediaan sintaksnya yang ringkas, mudah dibaca, dan fleksibel, sehingga cocok untuk beragam kebutuhan teknis seperti pengembangan aplikasi, kecerdasan buatan (AI), *machine learning*, *deep learning*, pengembangan *web*, serta analisis data [76].

Salah satu faktor utama yang membuat Python banyak digunakan adalah tersedianya berbagai pustaka (*library*) yang mendukung proses komputasi dan pengolahan data secara lebih efisien. Sebagai contoh, NumPy digunakan untuk kebutuhan komputasi numerik dan pengolahan *array*, sedangkan Pandas dimanfaatkan untuk manipulasi, pengelolaan, serta analisis data secara lebih praktis [77]. Keberadaan *library* tersebut yang kaya mempercepat dan mempermudah proses pengolahan data serta pengembangan solusi. Python juga menyediakan pustaka visualisasi seperti Matplotlib yang dapat digunakan untuk membuat berbagai jenis grafik, yakni seperti *scatter plot*, *line chart*, *bar chart*, dan *pie chart*. Pustaka ini juga mendukung pengaturan tampilan grafik, seperti penambahan judul, label sumbu, dan penyesuaian visual lainnya. Di sisi lain, NumPy memiliki peran penting dalam proses komputasi matematis karena mampu menangani operasi numerik berbasis *array* secara efisien [78].

Dalam penelitian ini, Python digunakan sebagai perangkat utama karena memiliki ekosistem pustaka yang komprehensif dan kemampuannya dalam mendukung seluruh tahapan penelitian, mulai dari pengolahan data, pembangunan model klasifikasi, hingga penerapan *Natural Language Prerocessing* (NLP). Beberapa *library* yang dimanfaatkan meliputi Pandas dan NumPy untuk manipulasi data, Scikit-Learn dan XGBoost untuk pembuatan serta evaluasi model klasifikasi, serta *Transformers* untuk penerapan IndoBERT dalam memahami konteks teks berbahasa Indonesia. Selain itu, sifat Python yang bersifat *open-source*, fleksibel, dan didukung komunitas pengguna yang luas memudahkan dalam proses pengembangan, pengujian, dan reproduksi penelitian.

2.4.2 Streamlit

Streamlit adalah *framework* sumber terbuka (*open-source*) berbasis Python yang dirancang dalam melakukannya proses penyederhanaan dalam pembuatan aplikasi *web* interaktif, khususnya dalam bidang *data science* dan *machine learning*. *Framework* ini banyak digunakan karena proses pengembangannya relatif sederhana dan tidak mengharuskan pengembang menguasai teknologi *web* seperti HTML, CSS, maupun JavaScript secara mendalam. Dengan hanya memanfaatkan kemampuan dasar pemrograman Python, pengguna sudah dapat membangun aplikasi *web* yang interaktif dan fungsional dengan lebih cepat [79]. Selain memiliki proses pengembangan yang praktis, Streamlit juga menyediakan berbagai komponen antarmuka interaktif yang dapat langsung digunakan dalam aplikasi, seperti *slider*, *text input*, *checkbox*, *selectbox*, dan berbagai elemen lainnya. Kehadiran fitur-fitur tersebut membuat proses pembuatan aplikasi menjadi lebih efisien sekaligus mempermudahan pengembang dalam membangun sistem berbasis *web* tanpa perlu melakukan pengkodean antarmuka yang rumit [80].

2.4.3 Jupyter Notebook

Jupyter Notebook adalah perangkat lunak yang sering digunakan untuk analisis data dan pengembangan program menggunakan bahasa Python. Platform ini memungkinkan pengguna untuk mengintegrasikan kode program,

hasil eksekusi, dokumentasi, serta visualisasi data ke dalam satu dokumen interaktif [81]. Dengan fasilitas tersebut, pengguna dapat menjalankan kode, menuliskan penjelasan teori, menampilkan hasil simulasi, dan membuat visualisasi data secara langsung dalam satu lingkungan kerja yang terhubung. Kondisi ini membuat proses pembelajaran maupun analisis data menjadi lebih praktis dan mudah dipahami, termasuk bagi pengguna yang masih belum memiliki pengalaman pemrograman yang mendalam [82].

Selain mendukung pengolahan data, Jupyter Notebook juga menyediakan struktur kerja yang fleksibel dan modular sehingga kode, teks *markdown*, dan visualisasi interaktif dapat disusun secara terorganisir dalam satu dokumen. Meskipun demikian, penggunaan Jupyter Notebook tidak terlepas dari beberapa kekurangan. Sejumlah penelitian menyoroti potensi munculnya penulisan kode yang kurang api, perilaku program yang tidak selalu konsisten, hingga kendala dalam mereproduksi hasil analisis secara berulang [83] Meskipun memiliki beberapa keterbatasan, Jupyter Notebook tetap menjadi platform yang luas pemanfaatannya dalam konteks pendidikan dan penelitian, karena mampu membantu pengguna menghubungkan konsep teoritis dengan implementasi praktis secara langsung [84],[82].