BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Proses pencarian penelitian terdahulu yang relevan dilakukan dengan tujuan untuk mendukung pernyataan pada penelitian ini dan berfungsi sebagai sumber teoritis dan referensi primer.

Tabel 2.1 Penelitian Terdahulu

Penelitian Terdahulu		
1.	Nama Jurnal	Brilliance Journal, Vol. 4, No. 2, 500–508
	Judul	Text Classification Algorithms: A Survey
	Penulis	G. Airlangga [19]
	Hasil dan Temuan	Linear SVM dan Gradient Boosting dengan TF-IDF memberi akurasi 95.33%. Ringan dan cocok untuk real-time classification.
2.	Nama Jurnal	*IEEE Access*, vol. 9, 144121–144128
	Judul	A YouTube Spam Comments Detection Scheme Using Cascaded Ensemble Machine Learning Model
	Penulis	Hayoung Oh [20]
	Hasil dan Temuan	Menggunakan enam algoritma ML dan dua model ensemble (hard & soft voting) untuk klasifikasi komentar spam dan ham. Hasil terbaik dicapai oleh metode ensemble dengan soft voting (akurasi 95,06%). Kelebihan: kombinasi model meningkatkan akurasi. Kekurangan: potensi overfitting jika data tidak beragam.
3.	Nama Jurnal	Procedia Computer Science (ScienceDirect), 2020, Vol 177, 593–598
	Judul	Analysis and Classification of User Comments on YouTube Videos
	Penulis	K.M. Kavitha, Asha Shetty, Bryan Abreo, Adline D'Souza, Akarsha Kondana [21]
	Hasil dan Temuan	Meneliti klasifikasi komentar YouTube ke dalam empat kategori: relevant, irrelevant, positive, dan negative. Hasil eksperimen awal (dengan 46 komentar) menunjukkan bahwa metode Association List memiliki performa lebih baik dibanding Bag of Words, khususnya dalam mendeteksi komentar tidak relevan. Akurasi rata-rata >90%. Kekurangan: cakupan data masih terbatas dan eksperimen berskala kecil.
4.	Nama Jurnal	Machine Learning with Applications, 2024,vol. 16, no. 22
	Judul	Spam Detection for YouTube Video Comments Using Machine Learning Approaches
	Penulis	Andrew S. Xiao, Qilian Liang [22]
	Hasil dan Temuan	Dari delapan model yang diuji, Random Forest merupakan model dengan performa terbaik dengan akurasi hingga 98%. Target label: ham dan spam. Model ini unggul dalam AUC, presisi, dan F1-score. Voting classifier juga mendekati hasil terbaik. Kelebihan: akurat dan robust. Kekurangan: beberapa

		model seperti KNN sensitif terhadap dimensi tinggi dan komputasi berat.
5.	Nama Jurnal	Int. J. of Sci. Research in Sci. & Tech,Vol. 11, No. 11, 311–317
	Judul	Random Tree Classifier for Spam Comment Detection
	Penulis	N. Venkatramana et al [23]
	Hasil dan Temuan	Random Tree menunjukkan akurasi 94% dan latency rendah untuk klasifikasi komentar YouTube.
6.	Nama Jurnal	arXiv.org (Preprint), 2020, arXiv:2007.04249
	Judul	Cooking is All About People: Comment Classification on Cookery Channels Using BERT and Classification Models
	Penulis	Subramaniam Kazhuparambil, Abhishek Kaushik [24]
	Hasil dan Temuan	Menggunakan 4.291 komentar dari dua channel YouTube, penelitian ini membandingkan pendekatan ML tradisional dan transformer-based. XLM menjadi model terbaik di sisi Language Models (akurasi 67.31%), sedangkan Random Forest dengan Term Frequency Vectorizer unggul di sisi model tradisional (akurasi 63.59%). Target label: Gratitude, About the Recipe, About the Video, Praising, Hybrid, Undefined, Suggestions & Queries. Kelebihan: XLM efektif untuk teks campuran multibahasa. Kekurangan: model transformer memerlukan komputasi tinggi dan kurang optimal untuk data informal.
7.	Nama Jurnal	Jurnal RESTI, Vol. 7, No. 6, 1371–1377
	Judul	Feature Extraction for Spam YouTube Comments
	Penulis	Jasmir, W. Riyadi [89]
	Hasil dan Temuan	Kombinasi TF-IDF dan CNN meningkatkan akurasi. Model hasilkan F1-score 96% pada komentar obfuscated seperti “s/l/o/t”.
8	Nama Jurnal	MALCOM Journal, Vol. 4, No. 4, 1533–1538
	Judul	Spam Detection in YouTube Comments Using Deep Learning Models
	Penulis	Gregorius Airlangga [90]
	Hasil dan Temuan	LSTM dan GRU mengungguli CNN & MLP. LSTM capai akurasi 95.65% dan F1-score 95.33%. Cocok untuk teks manipulatif.
9.	Nama Jurnal	J. of Information Technologies, Vol. 12, 612–620
	Judul	ALBERT4Spam: Spam Detection on Social Networks
	Penulis	Aqliima Aziz, Cik Feresa Mohd Foozy, Palaniappan Shamala, Zurinah Suradi [91]
	Hasil dan Temuan	Menggunakan ALBERT (lightweight transformer). Akurasi dan explainability mengalami peningkattan dibanding SVM klasik .ALBERT bisa menghasilkan akurasi klasifikasi >96%, diatas model konvensional seperti SVM
10.	Nama Jurnal	PeerJ Computer Science, Vol. 9, e645
	Judul	An improved deep convolutional neural network-based YouTube video classification using textual features
	Penulis	Ali Raza , Faizan Younas , Hafeez Ur Rehman Siddiqui c, Furqan Rustam , Monica Gracia Villar, Eduardo Silva Alvarado, Imran Ashraf j [92]
	Hasil dan Temuan	Mengembangkan CNN yang ditingkatkan untuk klasifikasi video YouTube berdasarkan fitur tekstual, meningkatkan akurasi deteksi konten.

Tabel 2.1 merangkum berbagai penelitian yang relevan dalam pengembangan model klasifikasi berbasis machine learning untuk mendeteksi komentar bermasalah, khususnya spam di platform YouTube. Studi-studi ini memberikan pijakan penting dalam merancang sistem moderasi otomatis, sekaligus mengidentifikasi pendekatan yang paling efektif berdasarkan dataset, metode, dan skenario spesifik. Penelitian yang dilakukan oleh Xiao dan Liang menunjukkan performa luar biasa dari model Random Forest untuk mendeteksi spam pada komentar YouTube, dengan rata-rata akurasi mencapai 98% dan precision 100%. Selain itu, pendekatan ensemble seperti Voting Classifier juga menunjukkan kinerja yang konsisten tinggi. Penelitian ini menyoroti efektivitas klasifikasi berbasis TF-IDF dan optimasi hyperparameter sebagai strategi penting untuk meningkatkan ketepatan deteksi [22]. Sementara itu, Aiyara dan Shetty mengeksplorasi pemanfaatan character-based N-Gram sebagai metode ekstraksi fitur. Dengan dataset berisi 13.000 komentar, mereka menemukan bahwa kombinasi N-Gram (n=6) dan algoritma Random Forest memberikan F1-score yang sangat tinggi (0.9813), menandakan bahwa detail karakteristik dari komentar teks sangat berpengaruh dalam klasifikasi. Pendekatan ini terbukti mampu mendeteksi komentar manipulatif yang sering kali tidak dapat dikenali oleh metode konvensional [23]. Kazhuparambil dan Kaushik membandingkan efektivitas model machine learning tradisional dengan transformer-based language model seperti XLM. Dengan menggunakan 4.291 komentar berbahasa Inggris-Malayalam dari kanal memasak YouTube, model XLM memberikan performa terbaik di antara model lainnya (akurasi 67,31%), sementara Random Forest dengan Term Frequency Vectorizer tetap unggul di sisi model klasik. Penelitian ini juga menegaskan pentingnya mempertimbangkan multibahasa dan variasi konteks dalam klasifikasi komentar [24]. Di sisi lain, penelitian oleh Kavitha mengevaluasi dua pendekatan utama: Bag of Words dan Association List. Hasil menunjukkan bahwa pendekatan Association List lebih efektif untuk mendeteksi komentar tidak relevan dan positif, meskipun eksperimen dilakukan pada dataset yang kecil (46 komentar). Studi ini memberikan wawasan awal terkait representasi konteks dalam klasifikasi komentar YouTube [21]. Hayoung Oh mengembangkan model klasifikasi berbasis ensemble (Ensemble Soft Voting)

dengan kombinasi enam algoritma (SVM, Logistic Regression, Naive Bayes, Random Forest, CART), dan berhasil mencapai akurasi hingga 95,06%. Temuan ini memperkuat asumsi bahwa kombinasi beberapa algoritma dapat memberikan hasil yang lebih stabil dan andal dibandingkan model tunggal. Sebagai dasar teoritis [20], G.AirlanggaDari penelitian sebelumnya, kita melihat masih belum ada yang mengeksplorasi deteksi komentar yang mengandung unsur promosi perjudian online, terutama di YouTube. Sehingga, penelitian ini di desain untuk mengembangkan sistem klasifikasi dengan menggabungkan spesifik akurasi dan juga consider semantics dari komentar tersebut. Hasil beberapa tes awal menunjukkan bahwa pendekatan dari Linear SVM dan Gradient Boosting di kombinasikan dengan fitur TF-IDF memiliki tingkat akurasi sebesar 95,33%. Pendekatan ini menunjukkan performa pendekatan tersebut dan secara ringan mengcomputate dengan nilai berhasil consi adalah recommended sistem melakukan tes [19].

Berdasarkan keenam penelitian tersebut, terdapat benang merah bahwa metode *deep learning*, terutama model yang dapat menangani konteks dan urutan seperti LSTM atau model hybrid, menunjukkan akurasi yang lebih tinggi dibandingkan model klasik. Namun, beberapa penelitian juga menyoroti bahwa dengan pemilihan fitur dan preprocessing yang tepat, model klasik masih bisa memberikan performa yang kompetitif. Dalam konteks penelitian ini, yang berfokus pada moderasi komentar YouTube terkait konten judi online, pendekatan berbasis *Large Language Model* seperti Mistral 7B memiliki peluang untuk melampaui semua pendekatan sebelumnya. Hal ini disebabkan oleh kemampuan LLM dalam memahami konteks yang kompleks, makna implisit, serta gaya bahasa yang mungkin tidak langsung menunjukkan kata-kata spam secara eksplisit. Selain itu, penggunaan *instruction tuning* dapat lebih mengarahkan model dalam memahami jenis spam tertentu seperti yang berkaitan dengan perjudian. Dengan membandingkan hasil dari penelitian-penelitian terdahulu, terlihat jelas bahwa masih terdapat ruang eksplorasi dalam peningkatan performa model klasifikasi komentar spam, terutama dengan memanfaatkan kekuatan dari LLM modern dan teknik penyetelan yang lebih kontekstual seperti instruction

tuning. Oleh karena itu, penelitian ini mengambil pendekatan yang berbeda dengan menggunakan Mistral 7B dan instruction tuning untuk menyusun model klasifikasi yang lebih adaptif dan presisi dalam mendeteksi komentar judi online di YouTube. Berbagai penelitian telah dilakukan untuk mengembangkan sistem deteksi komentar spam, khususnya di platform seperti YouTube. Studi-studi tersebut banyak memanfaatkan algoritma machine learning (Naïve Bayes, SVM, Random Forest), deep learning (CNN, LSTM, BiLSTM), serta teknik ekstraksi fitur seperti TF-IDF dan Word2Vec. Tabel 2.1 merangkum lebih dari 10 studi terkini yang dipublikasikan antara tahun 2020 hingga 2024, dengan mencantumkan teknik, dataset, dan hasil performa (akurasi atau F1-score). Namun demikian, sebagian besar studi masih bersifat umum dan belum menargetkan konteks lokal, khususnya komentar spam yang mempromosikan judi online di kanal YouTube berbahasa Indonesia. Sebagian besar penelitian juga masih menggunakan dataset bahasa Inggris atau bersifat campuran, dengan konteks promosi spam secara luas (e.g., penipuan, phishing), bukan spesifik perjudian.

Analisis kritis terhadap literatur menunjukkan bahwa pendekatan berbasis deep learning cenderung unggul dalam konteks data yang tidak terstruktur dan berformat bebas, sementara model seperti SVM dan Gradient Boosting tetap kompetitif untuk kebutuhan real-time karena efisiensi komputasinya. Misalnya, Airlangga (2023) melaporkan bahwa BiLSTM memiliki akurasi tinggi (96,2%), tetapi waktu latih yang jauh lebih tinggi dibandingkan Linear SVM dengan TF-IDF, yang mencapai akurasi 95,33%. Untuk memperjelas perbandingan antar pendekatan, Gambar 2.1 menyajikan diagram bubble yang memetakan hubungan antara teknik yang digunakan dan performa (akurasi/F1), berdasarkan dataset dan bahasa yang dianalisis. Dari pemetaan tersebut tampak jelas bahwa belum ada studi yang secara eksplisit mengembangkan model klasifikasi untuk komentar spam perjudian online dalam bahasa Indonesia, sehingga penelitian ini diarahkan untuk mengisi gap tersebut.

2.2 Tinjauan Teori

2.2.1 Text Classification

Text classification atau klasifikasi teks merupakan proses pengelompokan dokumen teks ke dalam kategori yang telah ditentukan sebelumnya. Proses ini melibatkan representasi teks, pemilihan fitur, dan penerapan algoritma pembelajaran mesin. salah satu tugas utama dalam pemrosesan bahasa alami (NLP) yaitu mengelompokkan dokumen teks secara otomatis ke dalam kategori yang telah ditentukan berdasarkan isi dan konteksnya. Proses ini berperan krusial dalam berbagai aplikasi, seperti deteksi spam, analisis sentimen, dan moderasi konten, sehingga memungkinkan pengelolaan dan pemrosesan data teks dalam skala besar secara lebih cepat dan efisien [25], [26].

2.2.2 Moderasi Komentar Otomatis

Moderasi komentar otomatis adalah proses penggunaan teknologi berbasis NLP dan pembelajaran mesin untuk menyaring, mengklasifikasikan, dan mengelola komentar pengguna secara real-time. Tujuan utamanya adalah untuk mendeteksi dan menangani konten yang berpotensi berbahaya, seperti ujaran kebencian, pelecehan, spam, atau informasi palsu, guna menciptakan lingkungan daring yang aman dan inklusif. Moderasi komentar otomatis merupakan proses penting dalam pengelolaan konten berbasis pengguna di lingkungan daring, khususnya pada platform media sosial dan forum diskusi. Kebutuhan terhadap moderasi otomatis semakin meningkat seiring dengan lonjakan interaksi daring yang membuat moderasi manual menjadi tidak lagi praktis dan kurang efisien [27] [28].

2.2.3 Undersampling

Undersampling merupakan salah satu teknik penting dalam *machine learning* yang digunakan untuk mengatasi permasalahan ketidakseimbangan kelas (*class imbalance*) dalam suatu dataset. Ketidakseimbangan ini terjadi ketika jumlah sampel dari kelas mayoritas jauh lebih banyak dibandingkan kelas minoritas, sehingga menyebabkan model cenderung bias terhadap kelas mayoritas. Tantangan tersebut berdampak pada menurunnya akurasi prediksi,

terutama dalam mengidentifikasi kelas minoritas, yang justru seringkali memiliki signifikansi tinggi dalam aplikasi kritis seperti deteksi penipuan atau diagnosis penyakit langka [29]. Prinsip dasar undersampling adalah dengan secara selektif mengurangi jumlah sampel dari kelas mayoritas untuk mencapai distribusi kelas yang lebih seimbang. Dengan pendekatan ini, model dapat lebih optimal dalam mengenali pola yang terdapat pada kelas minoritas, yang mungkin menyimpan informasi penting dan relevan secara praktis [30]. Beragam metode undersampling telah dikembangkan untuk mempertahankan representasi data yang efektif sambil mengurangi jumlah sampel kelas mayoritas. Salah satu metode paling sederhana adalah Random Undersampling, di mana sampel dari kelas mayoritas dihapus secara acak [31]. Metode yang lebih canggih mencakup Tomek Links, yang mengidentifikasi dan menghapus sampel dari kelas mayoritas yang memiliki jarak terdekat dengan kelas minoritas, sehingga meningkatkan pemisahan antar kelas [32]. Pendekatan lain seperti Cluster Centroids menggantikan sampel mayoritas dengan titik centroid dari hasil klasterisasi, yang memungkinkan pengurangan redundansi sambil mempertahankan informasi inti [33].

2.2.4 Representasi Teks dengan TF-IDF

Representasi teks merupakan tahapan penting dalam pemrosesan data teks sebelum dapat digunakan dalam berbagai algoritma pembelajaran mesin. Salah satu metode representasi teks yang paling umum digunakan adalah *Term Frequency-Inverse Document Frequency* (TF-IDF)[34]. TF-IDF adalah teknik pembobotan kata yang menggabungkan dua konsep utama, yaitu frekuensi kemunculan kata dalam suatu dokumen (*term frequency* - TF) dan frekuensi kebalikan kemunculan kata di seluruh korpus (*inverse document frequency* - IDF). Term Frequency mengukur seberapa sering suatu kata muncul dalam sebuah dokumen [34], sedangkan *Inverse Document*

Frequency bertujuan untuk mengurangi bobot kata-kata yang terlalu sering muncul di banyak dokumen dan kurang memberikan informasi khusus [35].

Rumus dasar:

Perhitungan TF-IDF untuk suatu kata w dalam dokumen d dirumuskan sebagai berikut:

$$TF - IDF(w, d) = TF(w, d) \times IDF(w)$$

2.1 Rumus Perhitungan TF-IDF

dengan komponen:

Term Frequency (TF):

$$TF(w, d) = \frac{Jumlah\ kemunculan\ kata\ w\ dalam\ dokumen\ d}{Total\ Jumlah\ kata\ dalam\ dokumen\ d}$$

2.2 Rumus TF

Inverse Document Frequency (IDF):

$$IDF(w) = log(\frac{N}{n_w})$$

2.3 Rumus IDF

keterangan:

- N adalah jumlah total dokumen dalam korpus,
- n_w adalah jumlah dokumen yang mengandung kata w.

2.2.5 Algoritma Machine Learning untuk Klasifikasi

Dalam klasifikasi teks, algoritma *machine learning* berfungsi untuk mempelajari pola dari data berlabel dan mengkategorikan data baru secara otomatis. Beberapa algoritma yang umum digunakan dalam tugas klasifikasi teks meliputi:

2.2.5.1 Logistic Regression

Logistic Regression adalah algoritma klasifikasi linear yang digunakan untuk memprediksi probabilitas dari suatu kelas. Meskipun namanya mengandung kata "regression", algoritma ini digunakan untuk klasifikasi biner maupun multikelas. Logistic Regression bekerja dengan menggunakan fungsi sigmoid untuk memetakan input ke rentang probabilitas antara 0 dan 1. Logistic Regression adalah metode statistik yang digunakan untuk menganalisis hubungan antara satu atau lebih variabel independen (baik kategorikal maupun kontinu) dengan variabel dependen biner. Model ini memprediksi probabilitas kejadian suatu peristiwa dengan menggunakan fungsi logit, yang merupakan transformasi logaritma dari odds suatu kejadian.

Sebagai contoh, dalam konteks penelitian kesehatan, regresi logistik sering digunakan untuk menentukan faktor-faktor risiko yang berkontribusi terhadap hasil kesehatan tertentu, seperti keberhasilan atau kegagalan pengobatan [36].

2.2.5.2 Multinomial Naive Bayes

Multinomial Naïve Bayes adalah salah satu varian dari algoritma Naïve Bayes yang secara khusus digunakan untuk data diskrit, seperti jumlah kemunculan kata dalam dokumen. Algoritma ini mengasumsikan bahwa fitur-fitur (misalnya, kata-kata dalam teks) bersifat independen satu sama lain, dan mengikuti distribusi multinomial. Model ini sederhana, cepat, dan efektif untuk tugas klasifikasi teks skala besar seperti deteksi spam dan analisis sentimen [37].

2.2.5.3 Support Vector Machine (LinearSVC)

Linear Support Vector Classifier (LinearSVC) merupakan implementasi khusus dari metode *Support Vector Machines* (SVM)

yang secara spesifik dirancang untuk menangani tugas klasifikasi teks. Algoritma ini bekerja dengan mengidentifikasi *hyperplane* optimal yang mampu memisahkan kelas-kelas data secara efisien dalam ruang fitur berdimensi tinggi [38]. LinearSVC sangat sesuai untuk data teks berdimensi tinggi yang direpresentasikan menggunakan teknik *Term Frequency-Inverse Document Frequency* (TF-IDF), yang banyak digunakan dalam berbagai aplikasi pemrosesan bahasa alami [39]. Selain itu, desain LinearSVC juga mengintegrasikan teknik regularisasi, yang berperan penting dalam mengurangi risiko *overfitting*, terutama dalam kasus jumlah fitur yang sangat besar akibat ekstraksi informasi dari data teks [40].

2.2.5.4 Decision Tree

Decision Tree merupakan algoritma *machine learning* yang serbaguna dan beroperasi berdasarkan model berbentuk pohon untuk pengambilan keputusan. Algoritma ini dapat digunakan secara efektif untuk tugas klasifikasi maupun regresi dengan cara membagi dataset menjadi subset-subset yang lebih kecil berdasarkan atribut tertentu, membentuk struktur hierarkis yang terdiri dari node (merepresentasikan fitur), cabang (merepresentasikan aturan keputusan), dan daun (merepresentasikan hasil) [41]. Pada setiap node internal, algoritma memilih fitur yang paling optimal untuk memisahkan kelas data berdasarkan kriteria tertentu, seperti Gini Index atau Information Gain [42]. Tujuan dari proses pemilihan ini adalah untuk meminimalkan tingkat impuritas dalam subset yang dihasilkan, sehingga data yang mengalir ke cabang-cabang berikutnya menjadi semakin homogen.

Dalam konteks klasifikasi teks, Decision Tree memanfaatkan fitur-fitur representasi teks untuk membangun aturan keputusan

yang mampu mengklasifikasikan teks ke dalam kelas yang telah ditentukan [43]. Salah satu keunggulan utama dari penggunaan Decision Tree adalah tingkat interpretabilitasnya yang tinggi, di mana setiap keputusan yang diambil dapat ditelusuri secara transparan dari node akar hingga ke daun, sehingga memudahkan pengguna dalam memahami logika klasifikasi yang dilakukan [44].

2.2.5.5 Random Forest

Random Forest merupakan algoritma *ensemble learning* yang menonjol dengan memanfaatkan kekuatan dari banyak pohon keputusan (*decision trees*) untuk menyelesaikan tugas klasifikasi maupun regresi. Algoritma ini membangun sekumpulan pohon keputusan melalui proses yang disebut *bootstrap sampling*, di mana setiap pohon dilatih menggunakan subset acak dari data pelatihan. Pendekatan ini tidak hanya meningkatkan stabilitas dan akurasi model, tetapi juga mengurangi risiko *overfitting* dengan meningkatkan keragaman antar pohon individual, sehingga memungkinkan penggabungan prediksi dari seluruh pohon menjadi sebuah hasil akhir yang lebih andal [45], [46].

Mekanisme kerja Random Forest melibatkan dua strategi utama, yaitu *bootstrap aggregating* (bagging) dan *random feature selection*. Pada fase pelatihan, setiap pohon dibangun dari sampel bootstrap, yakni pemilihan subset data secara acak dengan pengembalian (*with replacement*). Proses ini memungkinkan algoritma untuk memperkirakan kesalahan prediksi secara efektif menggunakan data yang tidak terpilih dalam sampel (*out-of-bag samples*). Selain itu, pada setiap percabangan pohon, hanya sebagian acak dari fitur yang digunakan untuk menentukan pemisahan, yang berfungsi untuk mengurangi korelasi antar pohon [47]

Dalam proses prediksi, Random Forest menerapkan strategi yang berbeda tergantung pada jenis tugas yang dihadapi. Untuk tugas klasifikasi, Random Forest menggunakan mekanisme *majority voting*, yaitu memilih kelas yang paling sering diprediksi oleh seluruh pohon sebagai keluaran akhir. Sedangkan untuk tugas regresi, hasil prediksi dihitung dengan mengambil rata-rata dari seluruh output pohon, sehingga menghaluskan anomali individual dan meningkatkan akurasi keseluruhan [47], [48]. Pendekatan ini menghasilkan prediksi yang umumnya lebih andal dibandingkan dengan prediksi dari satu pohon keputusan tunggal, yang sangat menguntungkan pada dataset yang kompleks dan rentan terhadap *overfitting* [49].

2.2.5.6 XGBoost (Extreme Gradient Boosting)

XGBoost (*Extreme Gradient Boosting*) adalah algoritma *ensemble* berbasis pohon keputusan yang sangat efisien dan memiliki kinerja tinggi, dikembangkan untuk meningkatkan akurasi serta efisiensi penggunaan sumber daya komputasi pada tugas klasifikasi dan regresi. Dirancang sebagai penyempurnaan dari metode *Gradient Boosting* tradisional, XGBoost mengintegrasikan berbagai optimasi yang meningkatkan kecepatan dan efektivitasnya, terutama ketika digunakan pada dataset berukuran besar [50]. Prinsip dasar dari XGBoost adalah penambahan pohon keputusan secara bertahap (*sequential addition*), di mana setiap pohon baru dibangun untuk memperbaiki kesalahan prediksi yang dilakukan oleh pohon sebelumnya. Proses ini memanfaatkan teknik *gradient descent* untuk meminimalkan fungsi loss yang dipilih, sehingga secara bertahap menghasilkan model prediktif yang semakin akurat [51].

Salah satu inovasi utama dari XGBoost adalah penerapan teknik regularisasi, yang berfungsi untuk mengendalikan overfitting. Hal

ini menjadi aspek penting mengingat metode gradient boosting tradisional cenderung rentan terhadap overfitting, terutama ketika berhadapan dengan model yang kompleks dan data yang mengandung noise [51].

Selain itu, XGBoost juga mengintegrasikan metode seleksi fitur yang lebih maju serta mendukung pemrosesan paralel selama tahap pembangunan pohon. Dukungan terhadap pemrosesan paralel ini memungkinkan percepatan waktu pelatihan secara signifikan dibandingkan pendekatan gradient boosting konvensional [52].

2.2.5.7 LightGBM

Light Gradient Boosting Machine (LightGBM) merupakan algoritma *ensemble* berbasis pohon keputusan yang dikembangkan oleh Microsoft Research, dengan tujuan untuk meningkatkan kecepatan pelatihan serta efisiensi penggunaan memori pada tugas klasifikasi dan regresi. Algoritma ini secara efektif mengatasi tantangan yang terkait dengan dataset berskala besar dan berdimensi tinggi dengan mengoptimalkan pendekatan *Gradient Boosting Decision Tree* (GBDT) tradisional [53]. Berbeda dengan banyak algoritma boosting konvensional yang membangun pohon secara **level-wise**—di mana seluruh node pada tingkat kedalaman tertentu dibagi secara simultan—LightGBM menerapkan strategi pertumbuhan **leaf-wise**. Pada setiap iterasi, algoritma ini memilih daun (*leaf*) dengan potensi pengurangan loss terbesar untuk dilakukan pemisahan, sehingga menghasilkan model yang lebih akurat dibandingkan metode level-wise pada jumlah pohon yang sama [54].

Pendekatan inovatif dalam membangun pohon secara leaf-wise memungkinkan LightGBM untuk mengalokasikan sumber daya komputasi secara lebih efisien dengan berfokus pada bagian data yang paling informatif. Strategi ini secara signifikan meningkatkan

kemampuan algoritma dalam mencapai presisi prediksi yang lebih tinggi, sekaligus mengurangi kebutuhan sumber daya, terutama dalam situasi yang melibatkan representasi fitur berdimensi tinggi seperti TF-IDF atau word embeddings. Karakteristik tersebut menjadikan LightGBM sebagai pilihan utama dalam aplikasi seperti klasifikasi teks dan berbagai bidang lainnya yang menuntut skalabilitas tinggi serta kecepatan pelatihan [55], [56].

2.2.5.8 AdaBoost

AdaBoost (Adaptive Boosting) merupakan salah satu algoritma ensemble learning yang berperan penting dalam meningkatkan performa model klasifikasi dengan cara menggabungkan beberapa weak learner menjadi satu strong learner. Algoritma ini pertama kali diperkenalkan oleh Freund dan Schapire, dan dikenal luas karena kemampuannya dalam meningkatkan akurasi model tanpa memerlukan perubahan signifikan terhadap algoritma dasar yang digunakan [57]. Mekanisme utama dari AdaBoost terletak pada penyesuaian bobot terhadap sampel yang diklasifikasikan secara salah oleh model sebelumnya. Proses ini dilakukan secara bertahap, di mana setiap *weak learner* dilatih secara berurutan, dengan fokus yang lebih besar pada sampel yang sulit diklasifikasikan. Hasil akhir dari proses iteratif ini merupakan agregasi prediksi dari seluruh model melalui metode *weighted majority voting* untuk klasifikasi atau *weighted sum* untuk regresi [58]. Keunggulan utama AdaBoost adalah kemampuannya dalam menangani dataset yang tidak seimbang, suatu tantangan yang umum dalam pengembangan model *machine learning*. Strategi reweighting terhadap sampel yang sulit memungkinkan AdaBoost untuk memberikan perhatian lebih pada kasus-kasus penting yang cenderung diabaikan oleh algoritma konvensional [59].

2.2.5.9 CatBoost

CatBoost (Categorical Boosting) merupakan pustaka *machine learning* berbasis *gradient boosting* yang dikembangkan oleh Yandex, dan dirancang secara khusus untuk menangani fitur kategorikal secara efisien tanpa memerlukan proses encoding manual seperti one-hot encoding maupun label encoding [60]. Kemampuan khas ini menyederhanakan tahapan *preprocessing* data dan sekaligus mengurangi risiko bias yang sering muncul akibat transformasi fitur kategorikal yang tidak tepat [61]. *CatBoost* mengintegrasikan metode statistik lanjutan untuk menangani fitur kategorikal secara langsung dalam kerangka kerjanya, sehingga mampu mempertahankan karakteristik penting dari data dan meminimalkan kemungkinan terjadinya kebocoran informasi (*information leakage*) selama proses pelatihan. Salah satu inovasi utama dari CatBoost adalah penerapan teknik **ordered boosting**, yang secara khusus dikembangkan untuk mengatasi masalah *prediction shift* yang umum terjadi pada algoritma boosting konvensional. Teknik ini menjaga distribusi data selama proses pembentukan pohon secara iteratif, dan secara efektif mencegah kebocoran informasi target ke dalam model selama pelatihan.Pendekatan ini berkontribusi besar terhadap peningkatan akurasi dan stabilitas model, terutama pada kondisi dataset yang tidak seimbang [62]. Fleksibilitas dan efisiensinya menjadikan CatBoost pilihan ideal untuk tugas-tugas *machine learning* yang melibatkan fitur kategorikal dalam jumlah besar, dengan peningkatan performa signifikan dibandingkan pendekatan konvensional.

2.2.5.10 MLP Classifier

Multilayer Perceptron (MLP) Classifier merupakan salah satu arsitektur penting dalam jaringan saraf tiruan (artificial neural network) yang banyak digunakan untuk menyelesaikan berbagai tugas klasifikasi. Sebagai bagian dari jaringan saraf feedforward, MLP terdiri atas tiga komponen utama, yaitu lapisan input, satu atau lebih lapisan tersembunyi, dan lapisan output. Setiap lapisan dibentuk oleh sekumpulan neuron yang saling terhubung secara penuh (fully connected) dan menggunakan fungsi aktivasi untuk melakukan transformasi nonlinier terhadap data masukan.Operasi dasar MLP melibatkan pemetaan data input ke label output melalui proses pembelajaran berbasis *backpropagation*. Dalam proses ini, bobot antar neuron disesuaikan secara iteratif untuk meminimalkan kesalahan prediksi berdasarkan fungsi loss yang telah ditentukan, yang berperan penting dalam mengoptimalkan kinerja model [63]. Fungsi aktivasi memegang peran sentral dalam arsitektur MLP karena kemampuannya memperkenalkan sifat nonlinier, yang memungkinkan model mempelajari hubungan kompleks dalam data. Fungsi aktivasi yang umum digunakan pada lapisan tersembunyi adalah ReLU *(Rectified Linear Unit)*, sedangkan untuk lapisan output biasanya digunakan softmax atau sigmoid, bergantung pada jenis klasifikasi—apakah multiclass atau biner [64].

Kemampuan ini menjadikan MLP efektif dalam berbagai aplikasi seperti klasifikasi teks, analisis citra, pengenalan pola, dan prediksi deret waktu. Keunggulan utama MLP terletak pada kemampuannya dalam menangkap hubungan nonlinier yang kompleks, yang tidak dapat ditangani oleh model klasifikasi linier sederhana [65].

2.3 Framework dan Algoritma

2.3.1 RandomizedSearchCV

RandomizedSearchCV merupakan metode optimasi hyperparameter yang kuat dan banyak digunakan dalam pengembangan model *machine learning*, dikenal karena kemampuannya dalam meningkatkan performa model melalui identifikasi kombinasi hyperparameter yang optimal. Pemilihan hyperparameter yang tepat secara signifikan dapat meningkatkan kemampuan model untuk melakukan generalisasi terhadap data baru serta secara bersamaan mengurangi risiko *overfitting* [66]. Secara operasional, RandomizedSearchCV bekerja dengan melakukan *sampling* secara acak dari distribusi nilai hyperparameter yang telah ditentukan sebelumnya, kemudian mengevaluasi kombinasi-kombinasi tersebut menggunakan teknik validasi silang (*cross-validation*) [67]. Pendekatan ini memungkinkan penemuan kombinasi hyperparameter yang menguntungkan tanpa harus mengeksplorasi seluruh ruang parameter secara ekshaustif.

Selain efisiensi, RandomizedSearchCV juga menawarkan fleksibilitas tinggi, karena pengguna dapat menentukan distribusi probabilistik untuk masing-masing hyperparameter, tidak terbatas pada nilai-nilai diskrit. Hal ini memungkinkan eksplorasi ruang parameter yang lebih luas dan beragam dengan beban komputasi yang tetap terkendali [67].

2.3.2 Cross-Validation

Cross-validation merupakan teknik evaluasi model yang penting dalam *machine learning*, yang berfungsi untuk menilai kemampuan model dalam melakukan generalisasi terhadap data yang belum pernah dilihat sebelumnya. Metodologi ini bertujuan untuk mengatasi potensi bias dalam estimasi performa model dengan cara membagi dataset ke dalam beberapa subset atau lipatan (*folds*), sehingga penggunaan keseluruhan dataset dapat dioptimalkan [68]. Mekanisme utama dalam cross-validation melibatkan pembagian dataset menjadi dua bagian, yaitu data latih (*training set*) dan data uji (*testing set*). Dalam setiap iterasi,

sebagian data digunakan untuk melatih model, sementara bagian lainnya disisihkan untuk evaluasi. Proses ini diulang berkali-kali dengan berbagai pembagian data, sehingga setiap subset secara bergantian berfungsi sebagai data latih maupun data uji dalam siklus iteratif [69].

Di antara berbagai strategi cross-validation, k-Fold Cross-Validation merupakan salah satu metode yang paling umum digunakan. Pada teknik ini, dataset dibagi secara merata menjadi kkk bagian, di mana pada setiap iterasi, satu bagian digunakan sebagai data uji, sementara k−1k-1k−1 bagian lainnya digunakan untuk pelatihan model. Prosedur ini dilakukan sebanyak K kali, dan hasil evaluasi model diambil dari rata-rata kinerja pada seluruh iterasi tersebut [68]. Penelitian menunjukkan bahwa k-Fold Cross-Validation sangat efektif di berbagai konteks, termasuk dalam penggunaan jaringan saraf tiruan (*artificial neural networks*) dan dataset yang tidak seimbang, di mana varian teknik seperti **stratified k-Fold Cross-Validation** disarankan untuk mempertahankan distribusi kelas yang representatif di seluruh lipatan [70]. Aspek kritis dari cross-validation tidak hanya terletak pada implementasinya, tetapi juga pada dampaknya terhadap evaluasi model. Apabila diterapkan dengan benar, cross-validation dapat secara substansial mengurangi risiko *overfitting*, karena kinerja model divalidasi pada berbagai pembagian data. Berdasarkan studi empiris, teknik seperti *10-Fold Cross-Validation* sangat populer karena mampu menghasilkan estimasi akurasi yang andal [71].

2.4 Tools dan software yang digunakan

2.4.1 Scikit-learn

Scikit-learn merupakan salah satu pustaka Python paling populer untuk pengembangan aplikasi *machine learning*, terutama berkat kemudahan penggunaannya serta dokumentasinya yang sangat komprehensif. Pustaka ini dirancang untuk menawarkan berbagai algoritma *machine learning*

yang efisien, mencakup berbagai tugas seperti klasifikasi, regresi, clustering, reduksi dimensi, pemilihan model, dan preprocessing data [72], [73]. Scikit-learn dibangun di atas pustaka numerik yang kuat, seperti *NumPy* dan *SciPy*, yang memungkinkan pelaksanaan operasi matematis kompleks secara efisien [74]. Berkat fondasi ini, Scikit-learn telah menjadi alat penting baik dalam penelitian akademis maupun dalam pengembangan industri, sekaligus memberikan kemudahan bagi pemula untuk memulai pembelajaran dalam bidang *machine learning*. Arsitektur Scikit-learn memanfaatkan fleksibilitas bahasa Python, memungkinkan pengguna untuk mengimplementasikan berbagai strategi machine learning tanpa memerlukan banyak kode dasar (boilerplate code). Filosofi desain ini mendorong fleksibilitas dan eksperimen, sehingga praktisi dapat dengan cepat membuat prototipe dan memvalidasi model mereka [72]

Kemampuan *Scikit-learn* diperkuat lebih lanjut oleh keterkaitannya dengan pustaka lain yang mapan. Sebagai contoh, *NumPy* berfungsi sebagai tulang punggung untuk manipulasi array, sedangkan *Matplotlib* menyediakan alat visualisasi data—fitur yang penting untuk interpretasi hasil model [74]. Integrasi ini memungkinkan peneliti dan pengembang untuk mengeksekusi alur kerja yang kompleks, mulai dari preprocessing data hingga *deployment model*, secara lebih mulus [75]. Dalam tahap *preprocessing*, misalnya, Scikit-learn menyediakan berbagai metode seperti normalisasi, encoding, dan imputasi data yang hilang. Untuk tahap pemodelan, library ini mencakup berbagai algoritma seperti decision tree, random forest, support vector machine, dan naive bayes, yang siap digunakan hanya dengan beberapa baris kode. Tak hanya itu, Scikit-learn juga mendukung teknik evaluasi model melalui metrik seperti akurasi, precision, recall, F1-score, dan confusion matrix, serta proses tuning hyperparameter menggunakan teknik seperti grid search dan random search.

2.4.2 Pandas dan NumPy

Pandas merupakan pustaka Python open-source yang telah memperoleh popularitas luas untuk keperluan manipulasi dan analisis data. Kontribusi utamanya terletak pada penyediaan struktur data yang fleksibel dan efisien, seperti DataFrame dan Series, yang memudahkan penanganan, pemrosesan, dan analisis dataset berukuran besar secara intuitif. Keunggulan utama Pandas adalah kemampuannya dalam mengelola data tabular dalam format dua dimensi yang menyerupai struktur spreadsheet, sehingga sangat sesuai untuk berbagai tugas *data preprocessing* yang penting dalam menyiapkan dataset sebelum diterapkan pada model *machine learning* [76].

Penggunaan *Pandas* dalam konteks manipulasi data telah didukung oleh berbagai studi yang menyoroti peran pentingnya dalam proses pra-analitik. Misalnya, sejumlah peneliti melaporkan penggunaan Pandas untuk membaca, membersihkan, dan mengorganisasi dataset sebelum dilakukan pelatihan dan evaluasi algoritma machine learning. Langkah-langkah preprocessing ini sangat penting untuk memastikan bahwa data berada dalam kondisi yang dapat digunakan, termasuk penanganan nilai hilang (*missing values*), transformasi data kategorikal, serta penghapusan duplikasi, di mana semua tahapan ini dapat berpengaruh signifikan terhadap kinerja model [77]. Selain fungsinya dalam pembersihan dan manipulasi data, *Pandas* juga terintegrasi secara mulus dengan pustaka *Python* lainnya, memungkinkan pengguna untuk menggabungkan berbagai alat dan teknik analitik dalam satu lingkungan kerja.

Sementara itu, *NumPy*, atau dikenal sebagai *Numerical Python*, merupakan pustaka fundamental dalam ekosistem Python yang dirancang khusus untuk mendukung operasi komputasi numerik berkinerja tinggi. Salah satu struktur data utamanya, yaitu *n-dimensional array* (*ndarray*), memungkinkan penyimpanan dan manipulasi data numerik skala besar

secara efisien. Struktur ini memudahkan pengguna dalam menangani dataset kompleks serta melakukan operasi yang akan menjadi sulit atau tidak efisien apabila menggunakan tipe data standar *Python* [74], [78]. Keunggulan utama dari *NumPy* terletak pada kemampuannya untuk melakukan operasi vektorisasi. Operasi ini memungkinkan berbagai perhitungan matematis dan aljabar linear dilakukan secara serentak pada seluruh elemen *array* tanpa memerlukan penggunaan loop eksplisit, sehingga menghasilkan efisiensi komputasi yang jauh lebih baik. Dengan demikian, kode program yang menggunakan *NumPy* menjadi lebih ringkas, mudah dibaca, serta lebih mudah dipelihara. Kemampuan ini sangat bermanfaat ketika menangani dataset berukuran besar yang membutuhkan kalkulasi matematis kompleks secara berulang, menjadikan *NumPy* alat yang esensial bagi ilmuwan data dan analis data [79].

Selain itu, *NumPy* memainkan peran penting dalam mempercepat pemrosesan data. Implementasi operasi vektorisasi dalam *NumPy* secara signifikan mengurangi waktu komputasi dibandingkan metode iteratif tradisional. Peningkatan kinerja ini sangat krusial ketika bekerja dengan dataset berskala besar, karena memungkinkan analisis yang lebih cepat dan mendalam. Efisiensi yang ditawarkan *NumPy* juga memungkinkan praktisi untuk mengembangkan algoritma yang lebih kompleks tanpa mengalami hambatan performa, sehingga semakin memperkuat posisinya sebagai pustaka dasar dalam komputasi ilmiah [80].

2.4.3 SciPy

SciPy merupakan pustaka open-source yang tangguh dan dirancang khusus untuk kebutuhan komputasi ilmiah dan rekayasa. Pustaka ini dibangun di atas kemampuan dasar **NumPy** dalam manipulasi array multidimensi, dan memperluas fungsionalitasnya dengan menyediakan berbagai fungsi numerik dan algoritma tingkat lanjut yang memungkinkan pelaksanaan operasi ilmiah kompleks. Kemampuan tersebut menjadikan SciPy relevan dalam berbagai bidang seperti fisika,

matematika, dan biologi komputasional [81]. Filosofi desain SciPy mengutamakan skalabilitas dan stabilitas, memungkinkan pustaka ini menangani perhitungan komputasional yang kompleks secara efisien—yang merupakan elemen kunci dalam penelitian ilmiah [82]. Salah satu keunggulan utama SciPy adalah kemudahannya dalam berintegrasi dengan pustaka lain di dalam ekosistem Python, seperti Pandas dan Matplotlib, sehingga mendukung alur kerja yang komprehensif dalam machine learning dan analisis data [83]

Interoperabilitas ini memperkuat fungsionalitas SciPy; pengguna dapat berpindah dengan mulus dari manipulasi data menggunakan Pandas ke operasi numerik menggunakan SciPy, dengan memanfaatkan kekuatan masing-masing pustaka untuk memperoleh wawasan yang bermakna dari dataset yang kompleks [84]. Selain itu, SciPy menyediakan berbagai algoritma penting seperti metode optimasi dan integrasi, yang menjadikannya pilihan utama bagi peneliti yang bekerja di bidang yang memerlukan pemodelan komputasional secara rinci [85]. Seiring berkembangnya kebutuhan dalam komputasi ilmiah, SciPy tetap menjadi salah satu alat utama yang digunakan peneliti, didukung oleh kemampuannya untuk terus beradaptasi dan mengintegrasikan algoritma mutakhir sesuai dengan kebutuhan disiplin ilmu yang beragam.

2.4.4 XGBoost dan LightGBM Library

XGBoost (*Extreme Gradient Boosting*) merupakan pustaka machine learning yang tangguh, mengimplementasikan algoritma boosting berbasis pohon keputusan yang dirancang dengan fokus pada efisiensi, kecepatan, dan performa model yang tinggi. Inti dari arsitektur XGBoost adalah kerangka kerja gradient boosting yang dilengkapi dengan berbagai optimasi penting, yang memungkinkan pustaka ini mengungguli metode gradient boosting konvensional dalam hal waktu pelatihan dan penggunaan memori, sekaligus meningkatkan kemampuan generalisasi model. Salah satu keunggulan XGBoost adalah kemampuannya dalam

melakukan analisis pentingnya fitur (feature importance analysis), yang memberikan wawasan mengenai kontribusi masing-masing variabel terhadap hasil prediksi. Penelitian telah menunjukkan efektivitas XGBoost dalam konteks tersebut, misalnya dalam memprediksi risiko cedera pada olahraga kompetitif [57]. XGBoost mendukung berbagai jenis tugas pembelajaran, termasuk klasifikasi, regresi, dan *ranking*, menjadikannya pustaka yang fleksibel dalam berbagai aplikasi. Kemampuan ini diperkuat dengan integrasi mekanisme regularisasi L1 dan L2, yang berperan penting dalam mengendalikan kompleksitas model dan mengurangi risiko *overfitting*, suatu tantangan umum dalam pengembangan model *machine learning*. Studi lebih lanjut menunjukkan keberhasilan penerapan XGBoost di lingkungan klinis, khususnya untuk mengevaluasi biomarker kritis yang mempengaruhi prediksi mortalitas, yang menunjukkan kemampuannya dalam menangani data kompleks dengan strategi regularisasi yang efektif [58]. Selain itu, XGBoost juga mendukung komputasi paralel, yang memungkinkan pemrosesan data secara simultan, serta menangani nilai yang hilang (*missing values*) secara otomatis. Fitur-fitur ini meningkatkan fleksibilitas dan efisiensi XGBoost, menjadikannya sangat sesuai untuk pengolahan dataset besar yang umum dijumpai dalam proyek *machine learning* modern. Performa XGBoost telah divalidasi di berbagai lingkungan kompetitif, termasuk dalam kompetisi data science berskala besar seperti Kaggle. Kemampuannya dalam menangani data berdimensi tinggi dan kompleks telah menjadi salah satu kunci kesuksesannya [50]

Sementara itu, LightGBM (*Light Gradient Boosting Machine*) merupakan pustaka *machine learning* open-source tingkat lanjut yang dikembangkan oleh Microsoft, dirancang untuk menyelesaikan tugas-tugas klasifikasi dan regresi dengan fokus pada efisiensi dan kecepatan. Pustaka ini sangat efektif dalam menangani dataset berukuran besar dan berdimensi tinggi, terutama berkat algoritma konstruksi pohon yang dioptimalkan, yang memungkinkan waktu pelatihan lebih cepat dan konsumsi memori yang

lebih rendah dibandingkan metode tradisional seperti XGBoost [86]. Salah satu inovasi utama dalam LightGBM adalah penerapan strategi pertumbuhan pohon leaf-wise, yaitu memilih percabangan berdasarkan pengurangan loss terbesar, bukan berdasarkan tingkat kedalaman pohon. Pendekatan ini sering kali menghasilkan model dengan akurasi yang lebih tinggi pada berbagai aplikasi, sebagaimana ditunjukkan dalam studi tentang analisis sentimen pada teks pendek, di mana kemampuan LightGBM dalam mengelola ruang fitur berdimensi tinggi menjadi faktor penting dalam meningkatkan akurasi klasifikasi [87]. LightGBM menunjukkan performa yang kompetitif dibandingkan dengan model ensemble tradisional seperti *Random Forest* dan *Gradient Boosting*, memperlihatkan ketangguhan dalam menghasilkan model dengan performa yang baik [88].