Journal of Applied Informatics and Computing (JAIC)
Vol.10, No.4, August 2026, pp. 4109~4123
e-ISSN: 2548-6861

Comparative Analysis of Machine Learning Algorithms for Lung Cancer Classification: A Progressive Evaluation from Preprocessing to Hyperparameter Tuning

Laurentius Joandanu [1]*, Usman Sudibyo [2]**
* Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro, Semarang
111202214179@mhs.dinus.ac.id [1], usman.sudibyo@dsn.dinus.ac.id [2]

Article Info

Article history:

Received 2026-07-02
Revised 2026-08-02
Accepted 2026-08-13

Keyword:

Lung Cancer,
Machine Learning,
Random Forest,
Preprocessing,
Hyperparameter Tuning.

ABSTRACT

Lung cancer is one of the leading causes of cancer-related deaths worldwide, with the main challenge being the difficulty of diagnosis at an early stage. Machine learning-based approaches have been proven to provide efficient solutions in supporting the classification process of this disease. This study proposes a comparative study of five machine learning algorithms, namely K-Nearest Neighbors (KNN), Logistic Regression, Random Forest, CatBoost and LightBGM, for lung cancer classification using the survey_lung_cancer.csv dataset consisting of 309 instances and 16 clinical features. All models were trained using a comprehensive preprocessing pipeline including duplicate data removal, missing values handling, outlier handling using the Interquartile Range (IQR) method, and categorical feature encoding using One-Hot Encoding. Hyperparameter optimization was performed uniformly using RandomizedSearchCV with Stratified K-Fold (k=10) and 50 iterations to ensure a fair comparison between algorithms. Each algorithm was evaluated under three progressive modelling conditions: baseline, after preprocessing, and after hyperparameter tuning, to quantify the individual contribution of each stage to model performance. The results show that Random Forest consistently recorded the best performance across all modelling conditions with an accuracy of 0.9464 and F1-Score of 0.9684 after applying preprocessing and hyperparameter tuning, demonstrating the effectiveness of the proposed progressive preprocessing and hyperparameter tuning pipeline. Learning curve analysis proves that none of the models experienced significant overfitting on the dataset used, indicating stable performance that has yet to be validated on independent clinical data.

I. PENDAHULUAN

Kanker Paru adalah salah satu penyakit yang paling mematikan di dunia, disebabkan oleh pertumbuhan sel-sel abnormal dalam jaringan paru secara tidak terkendali. Menurut WHO, pada tahun 2020 tercatat sekitar 2,21 juta kasus dan 1,80 juta kematian, di mana tingginya angka kematian ini umumnya disebabkan oleh keterlambatan dalam diagnosis, karena gejala awal yang tidak jelas sehingga sering kali penyakit ini baru terdeteksi pada tahap lanjut [1][2]. Risiko terjadinya kanker paru dipengaruhi oleh berbagai faktor, di antaranya adalah riwayat merokok lebih dari 15 tahun yang terdapat pada 62,1% pasien, usia di atas 40 tahun yang mencakup 95,4% kasus, serta laki-laki yang membentuk 71,3% dari total penderita. Selain itu, ada faktor lain yang berkontribusi seperti riwayat keluarga, penyakit paru kronis, paparan debu di tempat kerja, dan konsumsi alkohol [3]. Sebagai informasi klinis, adenokarsinoma adalah jenis histologis yang paling umum, dengan proporsi mencapai 58,6% dari total kasus, dan sebagian besar pasien baru terdiagnosis ketika penyakit berada pada Stadium IV-A [4]. Hal ini selaras dengan pengertian bahwa kanker paru mencakup semua jenis keganasan yang terjadi di paru-paru, baik yang berasal dari jaringan paru itu sendiri maupun yang berasal dari penyebaran dari organ lain [5].

Mengidentifikasi kanker paru secara dini menjadi tantangan terbesar dalam penanganan penyakit ini. Metode diagnosis yang biasa digunakan masih memiliki keterbatasan dalam hal waktu, biaya, serta kesediaan tenaga ahli yang kompeten [6]. Maka dari itu, pendekatan berbasis machine learning semakin banyak digunakan sebagai solusi alternatif yang lebih efisien. Machine learning bisa mengenali pola dari data, mengolah data yang memiliki banyak variabel, dan berjalan sendiri tanpa perlu campur tangan manusia [7]. Dibandingkan dengan cara yang biasa digunakan, pendekatan ini dianggap lebih cepat, lebih murah, dan bisa mendeteksi tanda-tanda awal penyakit yang biasanya tidak terlihat oleh pemeriksaan secara manual.

Beberapa penelitian sudah menunjukkan bahwa berbagai algoritma machine learning memiliki kelebihan dalam mengklasifikasikan penyakit medis. KNN berhasil mengklasifikasikan penyakit kardiovaskular dengan akurasi hingga 91% [8], Logistic Regression mencapai 89% dalam diagnosis penyakit jantung [9], LightGBM mencapai 84% dalam prediksi penyakit diabetes [10], dan CatBoost terbukti efektif dalam klasifikasi penyakit diabetes dengan akurasi 98% [11]. Dalam konteks kanker paru, Random Forest menjadi algoritma yang paling banyak diteliti sebelumnya, sehingga menjadi pembanding utama dalam penelitian ini.

Random Forest adalah algoritma machine learning yang sering digunakan untuk mengklasifikasikan penyakit. Cara kerjanya adalah dengan membuat banyak pohon keputusan menggunakan teknik bagging, lalu menentukan hasil akhir berdasarkan pilihan yang paling banyak didukung.Dengan metode ini, Random Forest bisa mengurangi risiko overfitting, mampu mengolah data yang memiliki banyak variabel, serta memberikan prediksi yang lebih akurat dan konsisten dibandingkan hanya menggunakan satu pohon keputusan saja [12][13]. Penelitian sebelumnya yang menggunakan Random Forest pada dataset survey_lung_cancer.csv berhasil mencapai akurasi sebesar 90,36%, tetapi hanya melakukan preprocessing sederhana seperti menghapus data duplikat dan mengubah label ke bentuk numerik, tanpa memperhatikan data yang tidak normal (outlier) atau menyesuaikan parameter secara tepat [14]. Kualitas data sangat berpengaruh terhadap hasil model karena data yang tidak bersih dapat merusak kinerja model [15]. Penanganan outlier yang tepat terbukti meningkatkan akurasi secara langsung, dan mengubah data kategorik ke bentuk numerik adalah langkah penting yang tidak boleh diabaikan [16]. Selain itu, penyesuaian hyperparameter juga sangat penting untuk meningkatkan performa model, dan RandomizedSearchCV terbukti mampu mencari konfigurasi terbaik dengan waktu komputasi yang lebih efisien dibandingkan GridSearchCV [17].

Meskipun begitu, penelitian sebelumnya, termasuk karya Maulana et al., biasanya hanya menyampaikan hasil akhir dari model tanpa memisahkan dan mengevaluasi kontribusi masing-masing tahap preprocessing serta hyperparameter tuning [14]. Akibatnya, penyebab peningkatan performa yang dilaporkan masih belum jelas. Kesenjangan ini menjadi alasan mengapa diperlukan evaluasi bertahap (progressive evaluation) yang memeriksa kontribusi dari setiap tahap dalam pipeline secara terpisah, mulai dari baseline, setelah preprocessing, hingga setelah hyperparameter tuning.

Berdasarkan perbedaan tersebut, penelitian ini membandingkan lima metode machine learning pada dataset bernama survey_lung_cancer.csv yang berisi 309 data dengan 16 fitur klinis, yaitu KNN, Regresi Logistik, Random Forest, CatBoost, dan LightGBM. Kelima algoritma ini menunjukkan beragam cara dalam mengklasifikasikan data berbentuk tabular. KNN merupakan model berbasis instance yang mudah, Logistic Regression adalah metode linear dasar, Random Forest merupakan model ensemble berbasis bagging yang memungkinkan perbandingan langsung dengan penelitian Maulana et al. [14], sedangkan CatBoost dan LightGBM merupakan contoh model gradient boosting modern.

Pendekatan machine learning klasik dan berbasis pohon keputusan dipilih daripada deep learning karena jumlah data yang digunakan relatif tidak besar, yaitu hanya sebanyak 309 data. Model machine learning klasik umumnya lebih cocok untuk dataset ini karena membutuhkan jumlah data yang lebih sedikit dan bisa belajar dengan efektif tanpa mengalami risiko overfitting yang tinggi, berbeda dengan model deep learning yang biasanya membutuhkan data dalam jumlah yang jauh lebih besar. Membandingkan dengan pendekatan deep learning menggunakan dataset yang lebih besar menjadi arah pengembangan yang penting untuk penelitian selanjutnya.

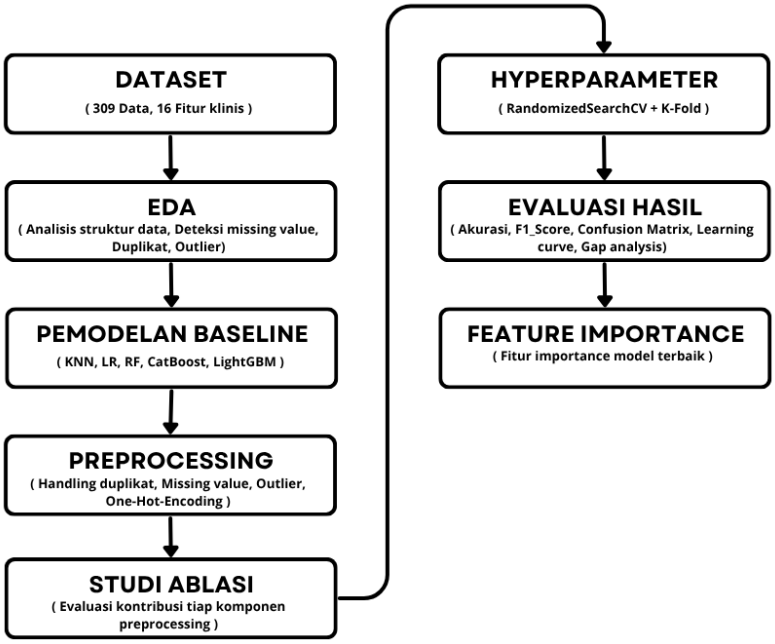
Inovasi utama dalam penelitian ini adalah membongkar kontribusi setiap komponen preprocessing secara terpisah melalui studi ablasi bertahap pada lima algoritma, bukan hanya membandingkan hasil akhirnya saja. Seluruh model dilatih melalui suatu pipeline yang mencakup proses pengelolaan data, seperti menangani data yang duplikat, nilai yang hilang, nilai ekstrem (menggunakan metode IQR), dan melakukan One-Hot Encoding. Untuk menyesuaikan parameter model, digunakan metode RandomizedSearchCV dengan teknik Stratified K-Fold, menggunakan 10 fold dan 50 kali iterasi. Evaluasi model dilakukan berdasarkan akurasi dan skor F1. Penelitian ini menunjukkan bahwa setiap langkah preprocessing memiliki dampak yang berbeda tergantung pada algoritma yang digunakan. Pengolahan outlier dengan metode IQR tidak meningkatkan performa model secara signifikan di semua jenis model, sedangkan One-Hot Encoding hanya memberikan pengaruh yang signifikan pada model Random Forest. Hasil ini bertentangan dengan anggapan bahwa semua tahap preprocessing secara otomatis meningkatkan kinerja model.

II. METODE

enelitian ini memiliki alur yang terstruktur dan sistematis, dirancang untuk memastikan setiap tahapan dapat dievaluasi kontribusinya secara terpisah terhadap peningkatan performa model klasifikasi. Alur penelitian ini terdiri dari delapan tahapan utama yang saling berkaitan, mulai dari persiapan

data hingga interpretasi hasil model, sebagaimana diilustrasikan secara ringkas pada

Gambar *1*.



Gambar 1. Alur Penelitian

Penelitian ini memiliki alur delapan tahapan berurutan sebagaimana diilustrasikan pada

Gambar *1*, dimulai dari inisialisasi dataset, Exploratory Data Analysis (EDA), pemodelan baseline, preprocessing data, studi ablasi komponen preprocessing, hyperparameter tuning, evaluasi hasil, hingga analisis feature importance, yang keseluruhannya dirancang secara sistematis untuk menghasilkan model dengan performa yang optimal.

A. Dataset

Penelitian ini menggunakan data sekunder yang bersifat data tabular dan bersumber dari platform Kaggle melalui tautan https://www.kaggle.com/datasets/ajisofyan/survey-lung-cancer dengan nama file survey_lung_cancer.csv. Dataset tersebut terdiri dari 309 entri dan 16 fitur yang berisi data survei klinis pasien.

TABEL 1.
FITUR DAN PENJELASANNYA

No	Fitur	Tipe Data	Deskripsi
1	GENDER	object	Jenis kelamin pasien
2	AGE	int64	Usia pasien dalam tahun
3	SMOKING	int64	Kebiasaan merokok (0 = Tidak, 1 = Ya)
4	YELLOW FINGERS	int64	Kondisi jari menguning (0 = Tidak, 1 = Ya)
5	ANXIETY	int64	Riwayat kecemasan (0 = Tidak, 1 = Ya)
6	PEER PRESSURE	int64	Pengaruh tekanan sosial (0 = Tidak, 1 = Ya)
7	CHRONIC DISEASE	int64	Riwayat penyakit kronis (0 = Tidak, 1 = Ya)
8	FATIGUE	int64	Gejala kelelahan (0 = Tidak, 1 = Ya)
9	ALLERGY	int64	Riwayat alergi (0 = Tidak, 1 = Ya)
10	WHEEZING	int64	Gejala mengi atau napas berbunyi (0 = Tidak, 1 = Ya)
11	ALCOHOL CONSUMING	int64	Kebiasaan konsumsi alkohol (0 = Tidak, 1 = Ya)
12	COUGHING	int64	Gejala batuk (0 = Tidak, 1 = Ya)
13	SHORTNESS OF BREATH	int64	Gejala sesak napas (0 = Tidak, 1 = Ya)
14	SWALLOWING DIFFICULTY	int64	Kesulitan menelan (0 = Tidak, 1 = Ya)

15	CHEST PAIN	int64	Gejala nyeri dada (0 = Tidak, 1 = Ya)
16	LUNG_ CANCER	int64	Label target (YES = Kanker paru, NO = Tidak kanker paru)

Berdasarkan Tabel 1, sebagian besar fitur dalam dataset merupakan data survei berbasis gejala dengan nilai biner 0 dan 1, di mana 0 menandakan tidak adanya gejala atau kondisi tersebut dan 1 menandakan keberadaannya. Fitur Age merupakan satu-satunya fitur numerik kontinu yang merepresentasikan usia pasien, sedangkan fitur Gender dan LUNG_CANCER merupakan fitur kategori yang perlu diubah ke format numerik pada tahap preprocessing sebelum digunakan dalam pelatihan model

B. Exploratory Data Analysis (EDA)

Sebelum dilakukan preprocessing, tahap Exploratory Data Analysis (EDA) dilakukan untuk memahami karakteristik dan distribusi data secara menyeluruh. EDA merupakan langkah awal yang penting dalam proses analisis data untuk mengidentifikasi pola, kualitas, serta sebaran nilai pada setiap fitur sebelum model dibangun.

Langkah pertama adalah memeriksa struktur data untuk mengetahui berapa banyak baris dan kolom yang ada, serta jenis data setiap fitur. Pemeriksaan ini bertujuan untuk memahami kondisi dan ciri-ciri dataset secara awal sebelum melakukan analisis lebih lanjut.

Selanjutnya dilakukan analisis tentang bagaimana distribusi kelas target LUNG_CANCER, yaitu antara kelas "YES" dan "NO", terbagi dalam dataset. Analisis ini penting karena ketidakseimbangan kelas bisa memengaruhi hasil kerja model dan perlu dipertimbangkan dalam memilih metrik evaluasi serta cara penanganannya di tahap berikutnya.

Pada tahap ini, dilakukan identifikasi awal mengenai adanya missing values dengan menggunakan perintah df.isnull().sum() serta pengecekan data duplikat dengan df.duplicated(). Hasil identifikasi ini digunakan sebagai dasar untuk menentukan langkah-langkah preprocessing yang harus dilakukan.

Setiap fitur yang ada kemudian dianalisis dengan menggunakan visualisasi berupa histogram dan boxplot agar bisa mendeteksi adanya outlier serta memahami bagaimana nilai-nilai tersebut tersebar pada setiap atribut. Visualisasi ini membantu mengenali fitur-fitur yang memiliki outlier dan membutuhkan penanganan khusus saat preprocessing data.

C. Pemodelan Baseline

Sebelum proses preprocessing dilakukan, model dibuat terlebih dahulu langsung dari data mentah sebagai referensi awal untuk mengukur kemampuan dasar setiap algoritma. Di tahap ini, baris yang memiliki kolom target kosong akan dihapus terlebih dahulu, lalu fitur berupa numerik dan kategorik dipisahkan berdasarkan tipe datanya.

Setiap fitur kategorik selanjutnya diubah dengan metode Label Encoding agar bisa diolah oleh algoritma machine learning. Data selanjutnya dibagi secara proporsi (80:20) menjadi data latih dan data uji.

Lima algoritma yang dibandingkan, yaitu K-Nearest Neighbors (KNN), Logistic Regression, Random Forest, CatBoost, dan LightGBM kemudian dilatih dengan menggunakan parameter default tanpa ada penyesuaian tambahan. Hasil penilaian pada tahap ini digunakan sebagai patokan awal untuk mengetahui seberapa besar dampak dari proses preprocessing data dan hyperparameter tuning yang dilakukan pada tahap berikutnya.

D. Preprocessing Data

Preprocessing data dilakukan secara rinci agar kualitas data tetap terjaga sebelum model dimulai pelatihannya. Penelitian terdahulu menyatakan bahwa proses preprocessing adalah langkah pertama yang bertujuan menjaga kualitas data tetap baik dan memungkinkan pengambilan informasi penting dari sekumpulan data [15].

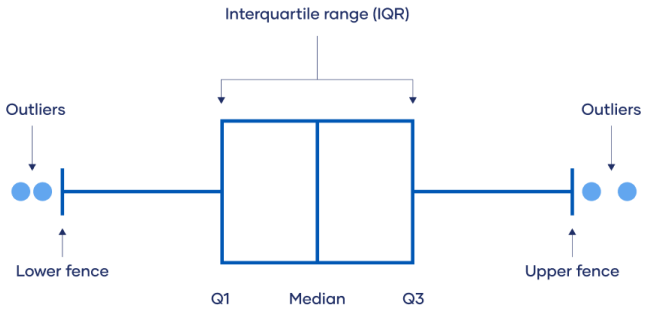
Tahap pertama adalah penanganan data duplikat. Deteksi duplikat dilakukan dengan menggunakan fungsi df.duplicated() dan penghapusan dilakukan dengan drop_duplicates() agar tidak ada bias dalam proses pelatihan, karena data yang sama bisa muncul di data latih dan data uji sekaligus, sehingga hasil evaluasi model tidak akurat.

Selanjutnya dilakukan penanganan nilai yang hilang dengan menggunakan perintah df.isnull().sum() untuk mendeteksi adanya data yang tidak lengkap. Nilai yang hilang dalam fitur numerik diisi dengan nilai median dari setiap fitur, karena median lebih kuat terhadap pengaruh outlier dibandingkan rata-rata, sehingga nilai imputasi yang dihasilkan lebih representatif.

Penanganan outlier kemudian dilakukan menggunakan metode Interquartile Range (IQR). IQR didefinisikan sebagai selisih antara kuartil ketiga dan kuartil pertama, sebagaimana ditunjukkan pada persamaan (1).

$$IQR = Q3 - Q1 \quad \textbf{(1)}$$

Nilai yang berada di luar batas lower bound (Q1 − 1,5×IQR) dan upper bound (Q3 + 1,5×IQR) di-capping ke nilai batas tersebut. Ilustrasi struktur IQR beserta komponen-komponennya, termasuk posisi Q1, median, Q3, lower fence, upper fence, serta outlier, ditampilkan pada Gambar 2.



Gambar 2. Visualisasi IQR dalam boxplot

Visualisasi boxplot dilakukan sebelum dan sesudah penanganan untuk memverifikasi hasilnya. Pengelolaan outlier yang efektif terbukti berkontribusi langsung pada peningkatan akurasi dan kinerja model.

Langkah berikutnya adalah mengubah fitur kategorik menjadi bentuk numerik menggunakan metode One-Hot Encoding dari library Scikit-learn. Metode ini membuat kolom baru berbentuk angka 0 atau 1 untuk setiap kategori yang berbeda, sehingga menghindari asumsi hubungan ordinal antar kategori yang dapat menyesatkan model. Hasil encoding selanjutnya digabungkan dengan fitur numerik menggunakan pd.concat() sehingga membentuk dataframe akhir yang sudah siap digunakan untuk membuat model.

Terakhir, data yang sudah melewati semua tahapan preprocessing dibagi menjadi data latih dan data uji dengan rasio (80:20). Kelima model kemudian dilatih kembali dengan data yang sudah dibersihkan, dan hasilnya dibandingkan dengan pemodelan baseline untuk melihat dampak dari proses preprocessing terhadap performa model.

E. Studi Ablasi Komponen Preprocessing

Untuk mengetahui seberapa besar setiap langkah preprocessing berpengaruh terhadap hasil model, dilakukan studi ablasi (ablation study) dengan cara menghilangkan komponen satu per satu secara bertahap. Dalam skema ini, komponen preprocessing ditambahkan secara bertahap dan berurutan ke dalam pipeline, mulai dari data mentah atau baseline, lalu dilanjutkan dengan menghapus data yang duplikat, menangani nilai yang hilang, menangani outlier menggunakan metode IQR, hingga melakukan One-Hot Encoding pada fitur kategorik. Performa model dicek setiap kali komponen baru ditambahkan, sehingga bisa dilihat bagaimana setiap komponen berkontribusi dari perubahan performa antara setiap tahap. Proses studi ablasi menggunakan lima algoritma yang dibandingkan dalam penelitian, yaitu KNN, Logistic Regression, Random Forest, CatBoost, dan LightGBM, untuk melihat apakah kontribusi setiap tahap preprocessing sama di semua algoritma atau berbeda antara satu dan lainnya.

Lima tahap yang diuji secara bertahap adalah: tahap dasar menggunakan data mentah tanpa dilakukan pengolahan awal, tahap pertama ditambah dengan penghapusan data yang duplikat, tahap kedua ditambah dengan penanganan data yang tidak lengkap, tahap ketiga ditambah dengan penanganan data ekstrem menggunakan metode IQR, dan tahap keempat ditambah dengan One-Hot Encoding pada fitur kategorik, yang setara dengan pipeline pengolahan data secara lengkap. Setiap tahap dilatih dan dicek dengan cara membagi data yang sama untuk kelima algoritma agar hasilnya tetap seimbang, menggunakan akurasi dan F1-Score sebagai acuan penilaian.

F. Hyperparameter Tuning

Hyperparameter tuning adalah proses menemukan kombinasi hyperparameter terbaik untuk memaksimalkan kinerja dan kemampuan generalisasi model. Proses ini dilakukan menggunakan RandomizedSearchCV yang memilih kombinasi hyperparameter secara acak sejumlah iterasi yang telah ditetapkan, sehingga lebih efisien dibandingkan GridSearchCV yang harus mengevaluasi seluruh kombinasi secara menyeluruh.

Ruang pencarian hyperparameter untuk masing-masing algoritma disesuaikan dengan karakteristik setiap model dan disajikan pada Tabel 2.

TABEL 2.
RUANG PENCARIAN HYPERPARAMETER

Model	Hyperparameter	Rentang
KNN	n_neighbors	[7 - 31]
	weights	{uniform, distance}
	metric	{euclidean, manhattan}
	p	[1 - 3]
Logistic Regression	solver	{saga}
	penalty	{l1, l2, elasticnet}
	C	[0.001 – 1000]
	l1_ratio	[0.1 – 0.9]
Random Forest	n_estimators	[300 – 2000]
	max_depth	[10 – 100]
	min_samples_split	[2 – 20]
	min_samples_leaf	[1 – 8]
	max_features	{sqrt, log2}
	bootstrap	{True, False}
	criterion	{gini, entropy}
CatBoost	iterations	[100 – 500]
	learning_rate	[0.01 – 0.1]
	depth	[4 – 8]
	l2_leaf_reg	[1 – 9]
	border_count	[32, 64, 128]
LightGBM	n_estimators	[100 – 500]
	learning_rate	[0.01 – 0.1]
	max_depth	[3 – 7]
	num_leaves	[15, 31, 63]
	min_child_samples	[5 – 30]
	subsample	[0.8 – 1.0]
	colsample_bytree	[0.8 – 1.0]

Berdasarkan Tabel 2, setiap algoritma memiliki ruang pencarian yang berbeda karena mempunyai karakteristik dan parameter masing-masing. Random Forest memiliki ruang pencarian yang lebih luas dengan tujuh hyperparameter yang perlu dioptimalkan, sedangkan KNN memiliki ruang

pencarian yang lebih sederhana dengan hanya empat hyperparameter. Perbedaan dalam ruang pencarian ini menunjukkan tingkat kekompleksan dari setiap algoritma dalam proses optimasi.

Konfigurasi yang digunakan untuk semua algoritma adalah Stratified K-Fold Cross Validation dengan 10 fold (k=10) dan 50 iterasi pencarian (n_iter=50). Stratified K-Fold dipakai agar rasio kelas di setiap bagian tetap sama seperti pada data asli, sehingga proses evaluasi selama mencari parameter terbaik menjadi lebih akurat dan representatif. Metrik yang digunakan sebagai dasar penyesuaian adalah F1-Score dengan parameter class_weight='balanced' untuk mengatasi ketidakseimbangan kelas yang teridentifikasi pada tahap EDA.

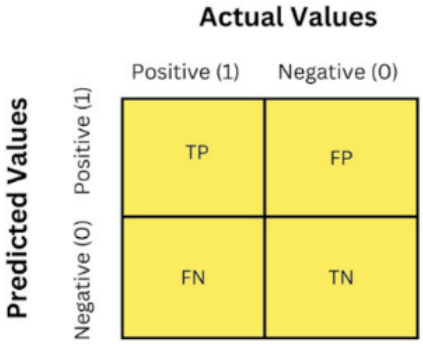
Pemilihan 10 fold (k=10) dilakukan karena nilai ini biasanya digunakan dalam validasi silang karena mampu memberikan keseimbangan yang baik antara bias dan varians dalam estimasi kinerja model dibandingkan dengan nilai k yang lebih kecil seperti 5 fold (k=5). Selain itu, nilai 10 fold (k=10) tetap memastikan jumlah data yang cukup pada setiap bagian (fold) mengingat ukuran dataset dalam penelitian ini relatif kecil, yaitu 276 data setelah proses preprocessing selesai. Nilai n_iter=50 pada RandomizedSearchCV dipilih sebagai kesepakatan antara kemampuan untuk mengeksplorasi ruang pencarian hyperparameter dan penghematan waktu komputasi, karena sumber daya komputasi yang tersedia terbatas dalam penelitian ini.

Setelah menemukan kombinasi hyperparameter terbaik melalui proses pencarian, kombinasi tersebut digunakan untuk melatih kembali setiap model menggunakan data latih dan kemudian diuji dengan data uji. Hasil evaluasi tersebut selanjutnya dibandingkan dengan pemodelan baseline dan pemodelan setelah dilakukan preprocessing agar mengetahui seberapa besar peran penyesuaian hyperparameter dalam meningkatkan performa masing-masing algoritma.

G. Evaluasi Hasil

Setiap model dinilai berdasarkan dua ukuran utama, yaitu tingkat akurasi dan F1-Score. Akurasi mengukur berapa persen dari prediksi yang benar dibandingkan semua prediksi yang dilakukan model, sedangkan F1-Score adalah rata-rata harmonis dari presisi dan recall, yang sangat berguna untuk dataset dengan kelas yang tidak seimbang.

Hasil prediksi dari setiap model juga ditampilkan dalam bentuk heatmap melalui confusion matrix. Confusion matrix adalah tabel yang menunjukkan bagaimana kinerja model klasifikasi dalam membuat prediksi dengan membandingkan hasil prediksi model terhadap label kelas yang sebenarnya. Confusion matrix terdiri dari empat bagian utama, yaitu True Positive (TP) yang menunjukkan berapa banyak sampel positif yang berhasil diprediksi dengan benar, True Negative (TN) yang menunjukkan berapa banyak sampel negatif yang diprediksi dengan benar, False Positive (FP) yang menunjukkan berapa banyak sampel negatif yang salah diprediksi sebagai positif, dan False Negative (FN) yang menunjukkan berapa banyak sampel positif yang salah diprediksi sebagai negatif. Contoh struktur confusion matrix yang digunakan dalam penelitian ini ditunjukkan pada Gambar 3.



Gambar 3. Tabel Confusion Matrix

Dalam klasifikasi kanker paru, nilai False Negative (FN) mendapat perhatian besar karena kesalahan dalam mengenali pasien yang benar-benar menderita kanker bisa menyebabkan dampak serius terhadap perawatan medis.

Selain mengevaluasi performa, juga dilakukan analisis overfitting terhadap semua model dengan menggunakan learning curve. Overfitting terjadi ketika model terlalu menyesuaikan diri dengan data pelatihan sehingga tidak bisa bekerja baik pada data yang baru. Learning curve digunakan untuk menunjukkan pola kinerja F1-Score pada data latih dan data uji seiring bertambahnya jumlah data latih. Model yang tidak mengalami overfitting akan memiliki kurva hasil pelatihan dan pengujian yang hampir mirip dan mendekat, yang menunjukkan bahwa model mampu memahami pola secara umum dan bukan hanya menghafal data pelatihan. Sebagai pelengkap kuantitatif terhadap learning curve, dihitung pula selisih (gap) antara F1-Score data latih dan F1-Score data uji pada model final setiap algoritma, di mana nilai gap yang kecil mengindikasikan kemampuan generalisasi yang baik. Perbandingan performa ketiga kondisi pemodelan, yaitu model baseline, model dengan preprocessing, dan model dengan preprocessing dan hyperparameter tuning, dilakukan dalam bentuk tabel dan grafik batang agar perbedaan dapat dilihat dengan jelas.

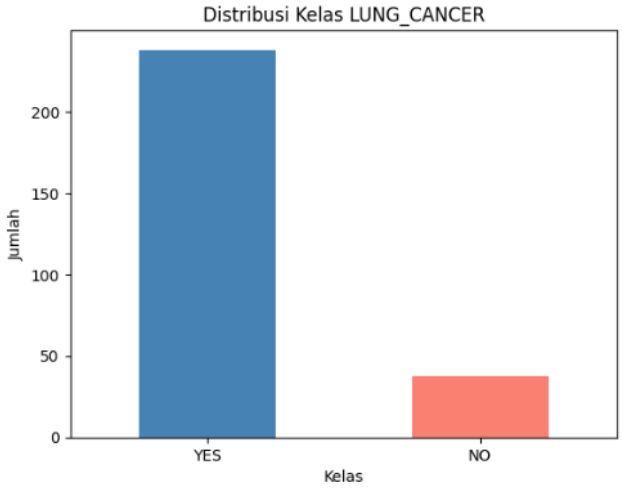
H. Analisis Feature Importance

Untuk mengetahui karakteristik fitur yang paling berpengaruh terhadap keputusan klasifikasi, dilakukan analisis feature importance menggunakan atribut feature_importances_ dari model Random Forest setelah hyperparameter tuning, yaitu model dengan performa terbaik di antara kelima algoritma yang dibandingkan. Nilai feature importance pada Random Forest dihitung berdasarkan rata-rata penurunan impurity (mean decrease in impurity) yang disumbangkan oleh setiap fitur di seluruh pohon keputusan dalam ensemble, kemudian diurutkan dan divisualisasikan dalam bentuk grafik batang untuk mengidentifikasi fitur klinis yang paling dominan dalam memprediksi kanker paru.

III. HASIL DAN PEMBAHASAN

A. *Hasil Exploratory Data Analysis (EDA)*

Pemeriksaan awal terhadap dataset menunjukkan bahwa file survey_lung_cancer.csv memiliki 309 baris data dan 16 kolom, serta berisi berbagai jenis data yang berbeda.



Gambar 4. Distribusi Kelas LUNG_CANCER

Dari hasil analisis distribusi kelas target pada Gambar 4, terlihat bahwa ada ketidakseimbangan dalam jumlah data antar kelas. Kelas "YES" memiliki jumlah data yang lebih banyak, sekitar 235 data, sedangkan kelas "NO" hanya memiliki sekitar 39 data. Kondisi ini harus diperhatikan saat memilih metrik penilaian karena model yang hanya memprediksi kelas mayoritas tetap bisa memiliki akurasi yang tinggi meskipun sebenarnya tidak mampu mengenali kelas minoritas. Oleh karena itu, F1-Score digunakan sebagai metrik utama evaluasi dan parameter class_weight='balanced' diterapkan pada proses hyperparameter tuning untuk menangani ketidakseimbangan ini.

Pemeriksaan missing values dilakukan menggunakan df.isnull().sum() untuk mengetahui apakah terdapat nilai yang hilang pada setiap fitur dataset.

```
# Cek missing value
df.isnull().sum()
```

	0
GENDER	0
AGE	0
SMOKING	0
YELLOW_FINGERS	0
ANXIETY	0
PEER_PRESSURE	0
CHRONIC DISEASE	0
FATIGUE	0
ALLERGY	0
WHEEZING	0
ALCOHOL CONSUMING	0
COUGHING	0
SHORTNESS OF BREATH	0
SWALLOWING DIFFICULTY	0
CHEST PAIN	0
LUNG_CANCER	0

Gambar 5. Hasil Pemeriksaan Missing Value

Hasil pemeriksaan yang ditunjukkan pada Gambar 5 memperlihatkan bahwa seluruh 16 fitur dalam dataset memiliki nilai 0, yang berarti tidak terdapat missing values sama sekali pada dataset ini. Meskipun demikian, langkah penanganan missing values tetap disiapkan dalam pipeline preprocessing menggunakan imputasi nilai median sebagai antisipasi apabila ditemukan nilai yang hilang pada data baru.

Pemeriksaan data duplikat dilakukan menggunakan df.duplicated() untuk mendeteksi baris-baris yang memiliki nilai identik pada seluruh atribut.

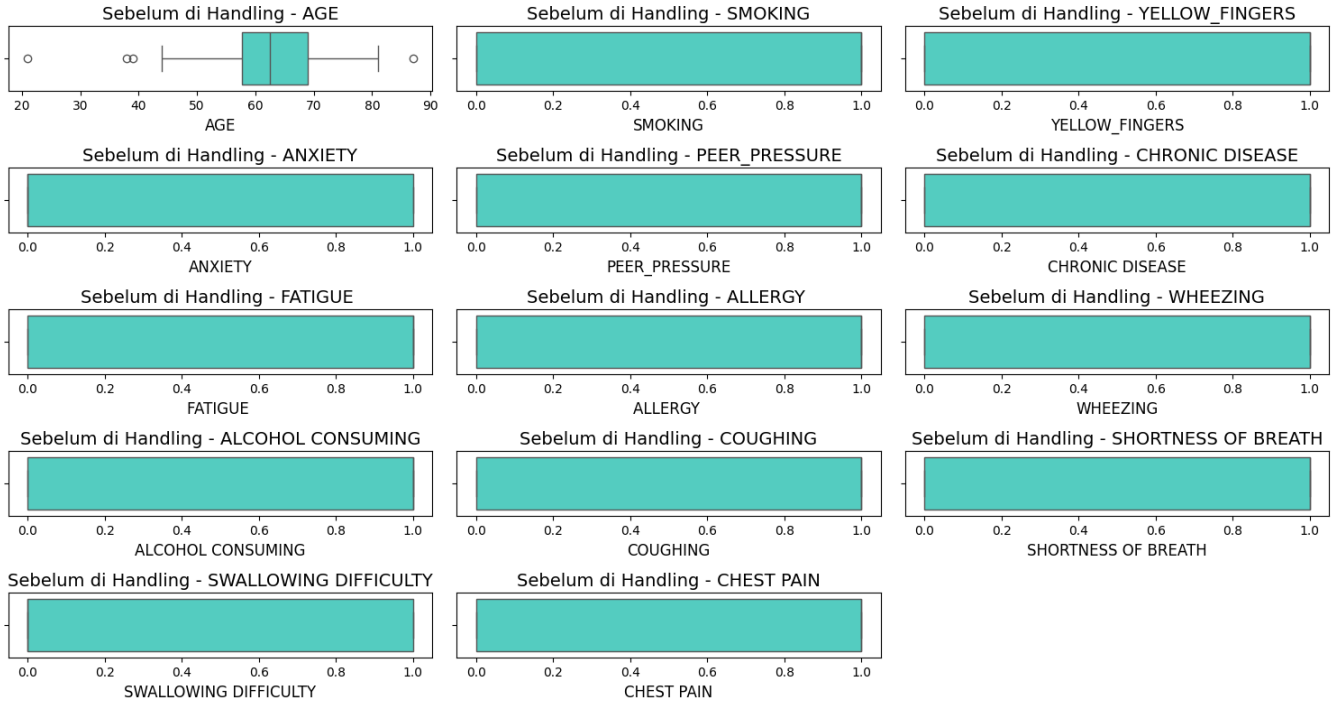
```
# Cek jumlah baris duplikat
print("Jumlah data duplikat:", df.duplicated().sum())

Jumlah data duplikat: 33
```

Gambar 6. Hasil Pemeriksaan Data Duplikat

Hasil pemeriksaan yang ditunjukkan pada Gambar 6 menemukan sebanyak 33 data duplikat dari total 309 entri dataset. Keberadaan data duplikat ini perlu ditangani karena dapat menyebabkan bias dalam proses pelatihan model dan membuat hasil evaluasi menjadi tidak valid.

Analisis outlier menggunakan visualisasi boxplot juga dilakukan pada tahap Exploratory Data Analysis (EDA) untuk mengidentifikasi fitur-fitur yang memiliki outlier sebelum dilakukan penanganan.



Gambar 7. Hasil Pemeriksaan Outlier

Hasil analisis boxplot sebelum penanganan ditunjukkan pada Gambar 7, yang memperlihatkan bahwa hanya fitur AGE yang menunjukkan adanya outlier, sementara fitur-fitur lainnya yang berbasis nilai biner tidak menunjukkan adanya outlier. Identifikasi ini menjadi dasar dalam menentukan langkah penanganan outlier yang perlu dilakukan pada tahap preprocessing.

B. Hasil Pemodelan Baseline

Pemodelan *baseline* dilakukan menggunakan kelima algoritma dengan parameter *default* tanpa melalui tahap *preprocessing* terlebih dahulu. Tujuannya adalah untuk mengetahui performa dasar masing-masing algoritma sebelum dilakukan perbaikan data. Hasil evaluasi pemodelan *baseline* disajikan pada Tabel 3.

TABEL 3.
HASIL PEMODELAN BASELINE

Model	Akurasi	F1_Score
K-Nearest Neighbors (KNN)	0,9032	0,9464
Logistic Regression	0,9032	0,9455
Random Forest	0,9194	0,9533
CatBoost	0,9032	0,9444
LightGBM	0,8871	0,9346

Hasil pada Tabel 3 menunjukkan bahwa Random Forest mencatatkan akurasi dan F1-*Score* tertinggi, yaitu 0,9194 dan 0,9533. LightGBM menghasilkan performa terendah dengan akurasi 0,8871 dan F1-*Score* 0,9346. Secara umum, seluruh model sudah menunjukkan performa yang cukup baik meskipun data belum melalui tahap preprocessing maupun hyperparameter tuning sama sekali.

C. Hasil Preprocessing Data

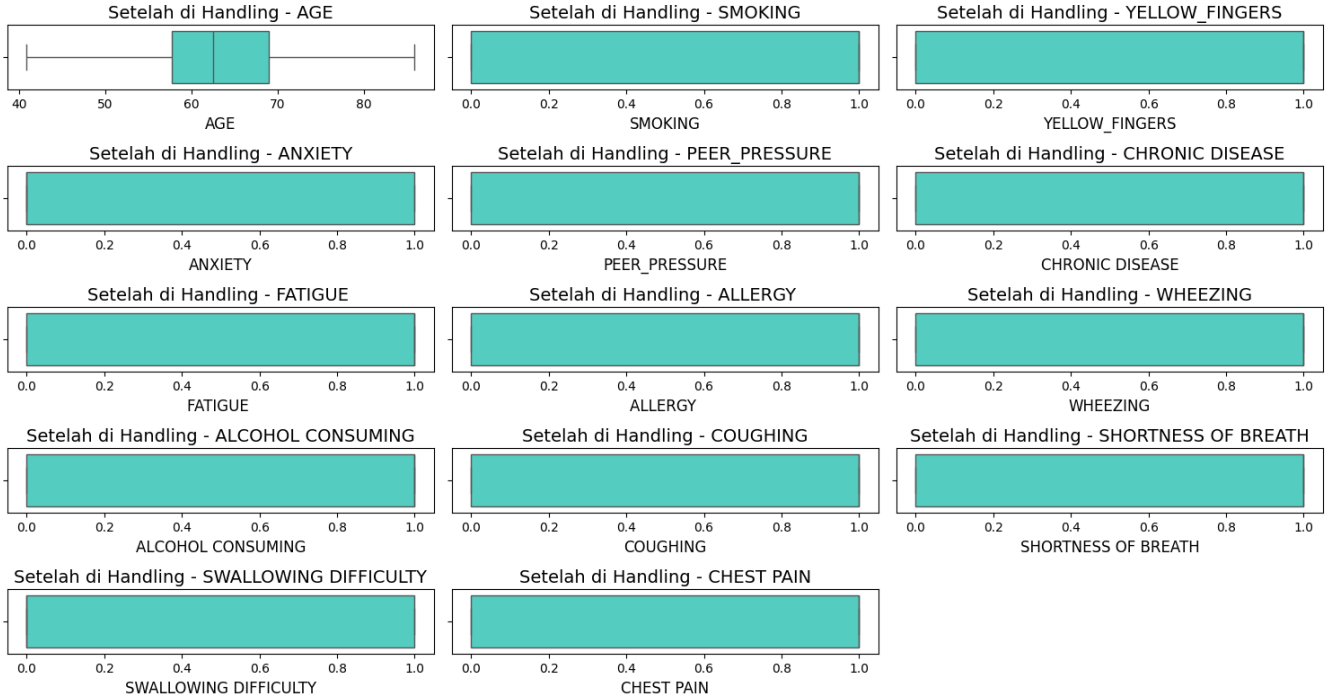
Penanganan data duplikat dilakukan dengan menghapus 33 baris duplikat yang ditemukan pada tahap EDA menggunakan fungsi drop_duplicates().

```
df = df.drop_duplicates().reset_index(drop=True)
print("Setelah hapus:", df.duplicated().sum())

Setelah hapus: 0
```

Gambar 8. Hasil Data Duplikat Setelah Dihapus

Hasil penghapusan dikonfirmasi menggunakan df.duplicated().sum() yang menunjukkan nilai 0 sebagaimana ditunjukkan pada Gambar 8, yang berarti seluruh data duplikat berhasil dihapus sehingga jumlah data berkurang dari 309 menjadi 276 entri.



Gambar 9. Boxplot Sesudah Penanganan Outlier

Penanganan outlier dilakukan menggunakan metode Interquartile Range (IQR). Berdasarkan hasil analisis boxplot yang ditunjukkan pada Gambar 7, ditemukan bahwa hanya fitur AGE yang memiliki outlier sebelum dilakukan penanganan, sementara fitur-fitur lainnya yang berbasis nilai biner tidak menunjukkan adanya outlier. Nilai yang berada di luar batas lower bound (Q1 − 1,5×IQR) dan upper bound (Q3 + 1,5×IQR) di-capping ke nilai batas tersebut. Setelah penanganan, visualisasi boxplot pada Gambar 9 memperlihatkan bahwa outlier pada fitur AGE berhasil ditangani dengan baik.

Fitur kategorik kemudian diubah menjadi representasi numerik menggunakan One-Hot Encoding dengan OneHotEncoder dari Scikit-learn. Metode ini menghasilkan kolom biner baru untuk setiap kategori unik sehingga menghindari asumsi hubungan ordinal antar kategori yang dapat menyesatkan model. Hasil encoding kemudian digabungkan dengan fitur numerik menggunakan pd.concat() untuk membentuk dataframe final. Daftar fitur yang dihasilkan setelah proses encoding disajikan pada Tabel 4.

TABEL 4.
FITUR DATASET SETELAH ENCODING

No	Fitur	Tipe Data
1	AGE	Fitur numerik asli
2	SMOKING	Fitur numerik asli
3	YELLOW_FINGERS	Fitur numerik asli
4	ANXIETY	Fitur numerik asli
5	PEER_PRESSURE	Fitur numerik asli
6	CHRONIC DISEASE	Fitur numerik asli
7	FATIGUE	Fitur numerik asli
8	ALLERGY	Fitur numerik asli
9	WHEEZING	Fitur numerik asli
10	ALCOHOL CONSUMING	Fitur numerik asli
11	COUGHING	Fitur numerik asli
12	SHORTNESS OF BREATH	Fitur numerik asli
13	SWALLOWING DIFFICULTY	Fitur numerik asli
14	CHEST PAIN	Fitur numerik asli
15	GENDER_FEMALE	Hasil One-Hot Encoding dari GENDER
16	GENDER_MALE	Hasil One-Hot Encoding dari GENDER

Berdasarkan Tabel 4, proses One-Hot Encoding pada fitur GENDER menghasilkan dua kolom baru yaitu GENDER_FEMALE dan GENDER_MALE. Fitur target LUNG_CANCER dipisahkan dari dataset sebelum proses encoding dilakukan karena fitur ini digunakan sebagai label prediksi dan tidak diikutsertakan sebagai fitur masukan model. Dengan demikian, total fitur yang digunakan sebagai masukan model setelah encoding menjadi 16 fitur yang seluruhnya bertipe numerik dan siap digunakan dalam proses pemodelan.

Setelah seluruh tahapan preprocessing selesai, data dibagi dengan perbandingan 80:20 dan kelima model dilatih ulang menggunakan data yang telah bersih. Hasil evaluasi disajikan pada Tabel 5.

TABEL 5.
HASIL PEMODELAN DENGAN PREPROCESSING

Model	Akurasi	F1_Score
K-Nearest Neighbors (KNN)	0,8929	0,9412
Logistic Regression	0,9107	0,9495
Random Forest	0,9286	0,9583
CatBoost	0,9107	0,9485
LightGBM	0,8750	0,9278

Hasil pada Tabel 5 menunjukkan adanya perubahan performa setelah diterapkan *preprocessing*. Random Forest mencatatkan akurasi tertinggi sebesar 0,9286 dengan F1-Score sebesar 0,9583. Peningkatan performa yang paling terlihat terjadi pada Logistic Regression dan CatBoost, yang akurasinya naik dari 0,9032 menjadi 0,9107. Sebaliknya, KNN dan LightGBM mengalami sedikit penurunan akurasi setelah preprocessing, yang menunjukkan bahwa kedua algoritma ini lebih sensitif terhadap perubahan distribusi data.

TABEL 6.
PERBANDINGAN PERFORMA BASELINE DAN SETELAH PREPROCESSING

Model	Baseline		Setelah Preprocessing		Gap	
	Accuracy	F1_Score	Accuracy	F1_Score	Accuracy	F1_Score
K-Nearest Neighbors (KNN)	0,9032	0,9464	0,8929	0,9412	-0,0103	-0,0052
Logistic Regression	0,9032	0,9455	0,9107	0,9495	+0,0075	+0,0040
Random Forest	0,9194	0,9533	0,9286	0,9583	+0,0092	+0,0050
CatBoost	0,9032	0,9444	0,9107	0,9485	+0,0075	+0,0041
LightGBM	0,8871	0,9346	0,8750	0,9278	-0,0121	-0,0068

Berdasarkan Tabel 6, terlihat bahwa Random Forest mengalami peningkatan akurasi dari 0,9194 menjadi 0,9286 setelah diterapkan preprocessing, begitu pula Logistic Regression dan CatBoost yang meningkat dari 0,9032 menjadi 0,9107. Sementara KNN dan LightGBM mengalami sedikit penurunan akurasi.

D. Hyperparameter Tuning

Hyperparameter tuning dilakukan menggunakan RandomizedSearchCV dengan Stratified K-Fold (k=10) dan 50 iterasi pencarian. Pendekatan ini dipilih karena lebih efisien dibandingkan GridSearchCV dan tetap mampu menemukan kombinasi hyperparameter yang optimal. Hasil evaluasi seluruh model setelah hyperparameter tuning disajikan pada Tabel 7.

TABEL 7.
HASIL PEMODELAN DENGAN PREPROCESSING DAN HYPERPARAMETER

Model	Akurasi	F1_Score
K-Nearest Neighbors (KNN)	0,8929	0,9412
Logistic Regression	0,8750	0,9293
Random Forest	0,9464	0,9684
CatBoost	0,9107	0,9495
LightGBM	0,8750	0,9278

Hasil pada Tabel 7 menunjukkan bahwa Random Forest mencatatkan peningkatan performa yang paling signifikan setelah hyperparameter tuning, dengan akurasi meningkat dari 0,9286 menjadi 0,9464 dan F1-Score dari 0,9583 menjadi 0,9684. KNN dan LightGBM mempertahankan performanya masing-masing pada akurasi 0,8929 dan 0,8750 tanpa perubahan berarti setelah tuning. CatBoost juga relatif stabil dengan akurasi tetap 0,9107, meskipun F1-Score-nya sedikit meningkat menjadi 0,9495. Sementara itu, Logistic Regression justru mengalami sedikit penurunan akurasi menjadi 0,8750, yang mengindikasikan bahwa model linear ini kurang cocok untuk karakteristik dataset ini meskipun sudah dilakukan optimasi. Perbandingan performa ketiga kondisi per model disajikan pada Tabel 8.

TABEL 8.
PERBANDINGAN KESELURUHAN KETIGA KONDISI PEMODELAN

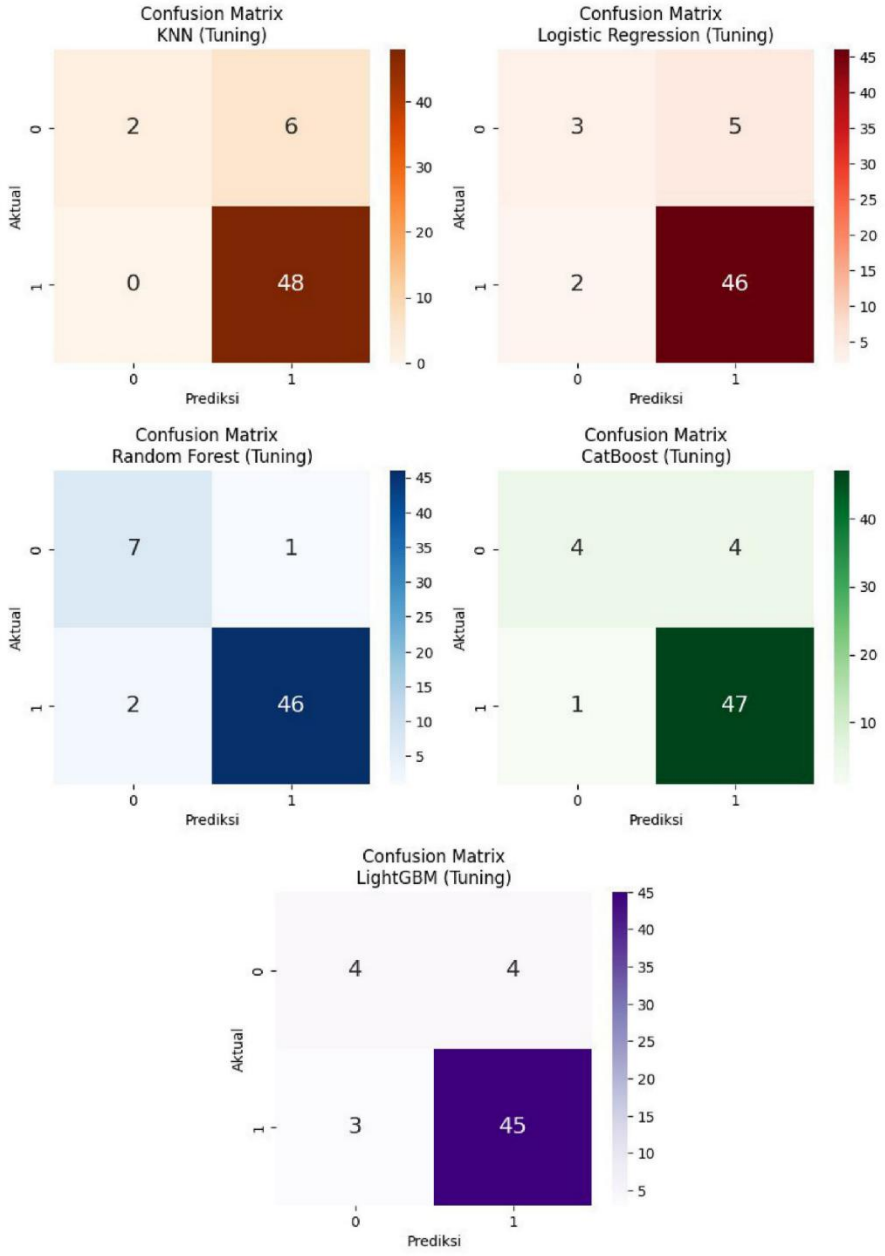
Model	Baseline		Setelah Preprocessing		Setelah Hyperparameter Tuning		Gap	
	Accuracy	F1_Score	Accuracy	F1_Score	Accuracy	F1_Score	Accuracy	F1_Score
K-Nearest Neighbors (KNN)	0,9032	0,9464	0,8929	0,9412	0,8929	0,9412	−0,0103	−0,0052
Logistic Regression	0,9032	0,9455	0,9107	0,9495	0,8750	0,9293	−0,0282	−0,0162
Random Forest	0.9194	0.9533	0.9286	0.9583	0.9464	0.9684	+0,0270	+0,0151
CatBoost	0,9032	0,9444	0,9107	0,9485	0,9107	0,9495	+0,0075	+0,0051
LightGBM	0,8871	0,9346	0,8750	0,9278	0,8750	0,9278	−0,0121	−0,0068

Berdasarkan Tabel 8, Random Forest secara konsisten unggul di seluruh kondisi pemodelan dan mencatatkan performa terbaik setelah hyperparameter tuning dengan akurasi 0,9464 dan F1-Score 0,9684, sedikit lebih tinggi

dibandingkan penelitian Maulana et al. yang melaporkan akurasi sebesar 0,9036 menggunakan Random Forest pada dataset yang sama [14]. Meskipun demikian, perbandingan ini perlu dibaca secara kritis dan tidak dapat langsung diklaim sebagai bukti keunggulan pipeline yang diusulkan. Maulana et al. hanya melakukan penghapusan data duplikat dan encoding fitur kategorik menggunakan Label Encoder, tanpa penanganan outlier maupun One-Hot Encoding, serta mengevaluasi model menggunakan satu kali pembagian data latih dan data uji dengan rasio 70:30 tanpa cross-validation maupun proses hyperparameter tuning, sementara penelitian ini menerapkan preprocessing yang lebih menyeluruh termasuk One-Hot Encoding, Stratified K-Fold Cross Validation (k=10), dan RandomizedSearchCV dengan rasio data latih dan data uji sebesar 80:20. Perbedaan cakupan preprocessing, strategi validasi, komposisi data latih dan data uji, serta ada tidaknya hyperparameter tuning antara kedua penelitian membuat selisih akurasi sebesar 0,0428 tersebut tidak dapat sepenuhnya diatribusikan pada pipeline yang diusulkan dalam penelitian ini. Perbandingan yang lebih objektif memerlukan replikasi protokol Maulana et al. pada pembagian data dan skema validasi yang identik dengan penelitian ini, yang berada di luar cakupan penelitian saat ini.

E. Evaluasi Hasil

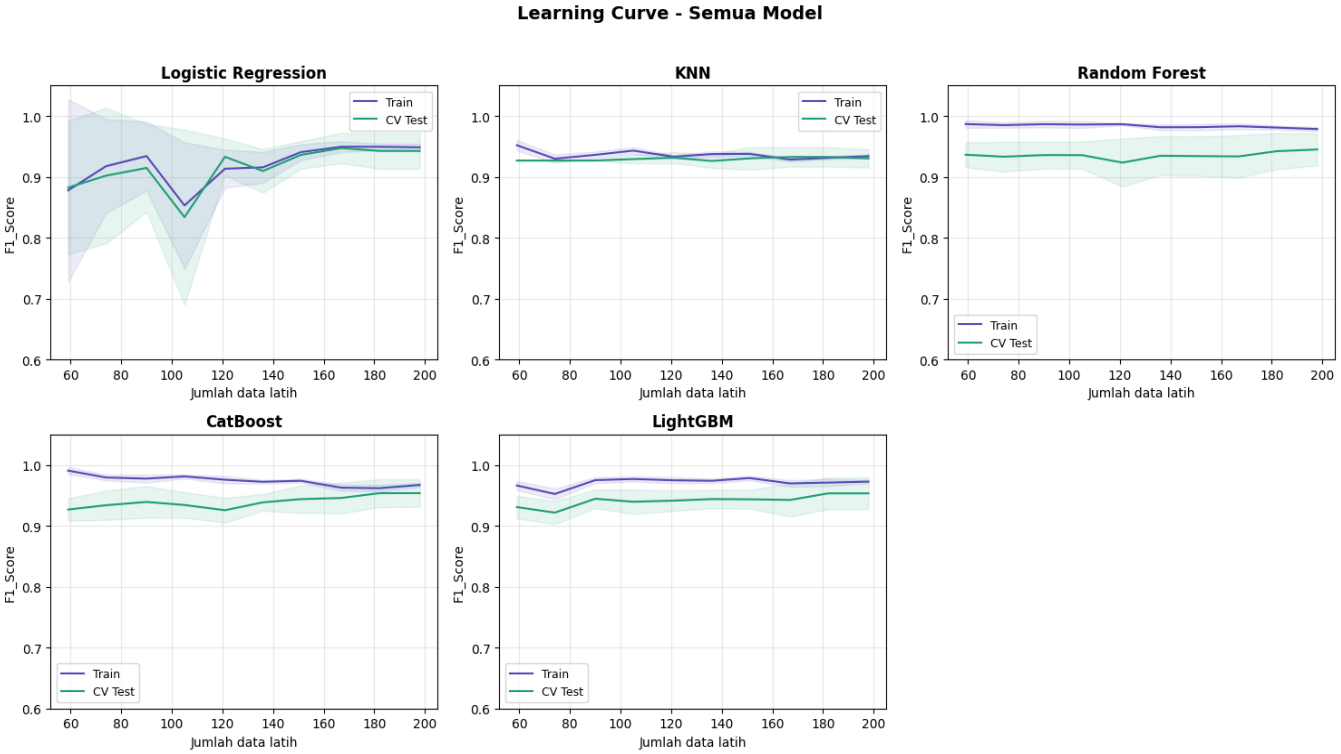
Hasil evaluasi confusion matrix dari seluruh model setelah hyperparameter tuning disajikan pada Gambar 10.



Gambar 10. Confusion Matrix Tiap Model

Berdasarkan Gambar 10, Random Forest menghasilkan prediksi terbaik dengan 7 True Negative, 46 True Positive, 1 False Positive, dan hanya 2 False Negative. Nilai False Negative yang rendah pada Random Forest sangat penting dalam konteks klasifikasi kanker paru, karena kesalahan dalam mengidentifikasi pasien yang sebenarnya menderita kanker dapat berakibat serius pada penanganan medis.

Analisis overfitting dilakukan menggunakan learning curve yang memperlihatkan tren performa F1-Score pada data latih dan data uji seiring bertambahnya jumlah data latih. Hasil learning curve seluruh model disajikan pada Gambar 11.



Gambar 11. Visualisasi Learning Curve

Berdasarkan Gambar 11, seluruh model menunjukkan pola learning curve yang sehat di mana kurva data latih dan data uji cenderung konvergen seiring bertambahnya jumlah data latih. Logistic Regression menunjukkan kurva yang stabil dengan tren yang terus meningkat dan konvergen pada nilai F1-Score sekitar 0,94, menandakan model ini belajar dengan baik tanpa overfitting. KNN menunjukkan kurva yang paling stabil di antara semua model, dengan nilai train dan CV test yang sangat berdekatan sepanjang proses pelatihan, mengindikasikan kemampuan generalisasi yang sangat baik. Random Forest menunjukkan gap yang konsisten antara kurva train dan CV test, namun kedua kurva tetap berada pada rentang nilai yang tinggi yaitu di atas 0,93, yang menunjukkan bahwa model ini memiliki kapasitas belajar yang tinggi namun tetap mampu melakukan generalisasi dengan baik. CatBoost dan LightGBM menunjukkan pola yang serupa satu sama lain, dengan kurva train yang berada pada rentang 0,95 hingga hampir 1,0 dan kurva CV test yang secara bertahap mendekat seiring bertambahnya jumlah data latih, mengindikasikan bahwa kedua model berbasis gradient boosting ini memiliki kapasitas belajar yang tinggi namun tetap mampu melakukan generalisasi dengan baik. Secara keseluruhan, tidak ada model yang menunjukkan tanda-tanda overfitting yang signifikan, sehingga seluruh model yang dihasilkan dalam penelitian ini terbukti stabil pada dataset yang digunakan. Perlu dicatat bahwa stabilitas ini merepresentasikan performa model pada dataset survey_lung_cancer.csv dan belum tentu tergeneralisasi pada data klinis independen di luar dataset penelitian ini.

Sebagai pelengkap kuantitatif terhadap analisis learning curve, dihitung pula selisih (gap) antara F1-Score data latih dan F1-Score data uji pada model final setiap algoritma setelah preprocessing dan hyperparameter tuning. Hasil perhitungan gap disajikan pada

Tabel 9.

TABEL 9.
ANALISIS GAP F1_SCORE DATA LATIH DAN DATA UJI

Model	F1-Score Train	F1-Score Test	Gap
K-Nearest Neighbors (KNN)	0,9347	0,9412	−0,0065
CatBoost	0,9757	0,9495	0,0262
Logistic Regression	0,9476	0,9293	0,0183
LightGBM	0,9757	0,9278	0,0479
Random Forest	0,9757	0,9684	0,0073

Hasil pada

Tabel 9 menunjukkan bahwa empat dari lima model, yaitu Logistic Regression, Random Forest, CatBoost, dan LightGBM, memiliki gap positif antara F1-Score data latih dan data uji, dengan nilai tertinggi pada LightGBM sebesar 0,0479 dan terendah pada Random Forest sebesar 0,0073. Nilai gap yang relatif lebih besar pada LightGBM dan CatBoost dibandingkan model lainnya mengindikasikan potensi overfitting ringan pada kedua algoritma berbasis gradient boosting ini, meskipun performa pada data uji tetap berada pada rentang yang baik dengan F1-Score di atas 0,92. Sebaliknya, KNN menunjukkan gap negatif sebesar −0,0065, di mana F1-Score pada data uji justru sedikit lebih tinggi dibandingkan data latih; pola ini bukan indikasi overfitting, melainkan variasi wajar yang dapat terjadi pada algoritma non-parametrik seperti KNN akibat pembagian data dengan ukuran dataset yang terbatas. Secara keseluruhan, gap yang relatif kecil pada Random Forest memperkuat hasil analisis learning curve yang menunjukkan model ini memiliki kemampuan generalisasi paling baik di antara kelima algoritma yang diuji.

F. Hasil Studi Ablasi

Hasil studi ablasi kumulatif pada kelima algoritma disajikan pada Tabel 10.

TABEL 10.
HASIL STUDI ABLASI TIAP PROSES PREPROCESSING (F1_SCORE)

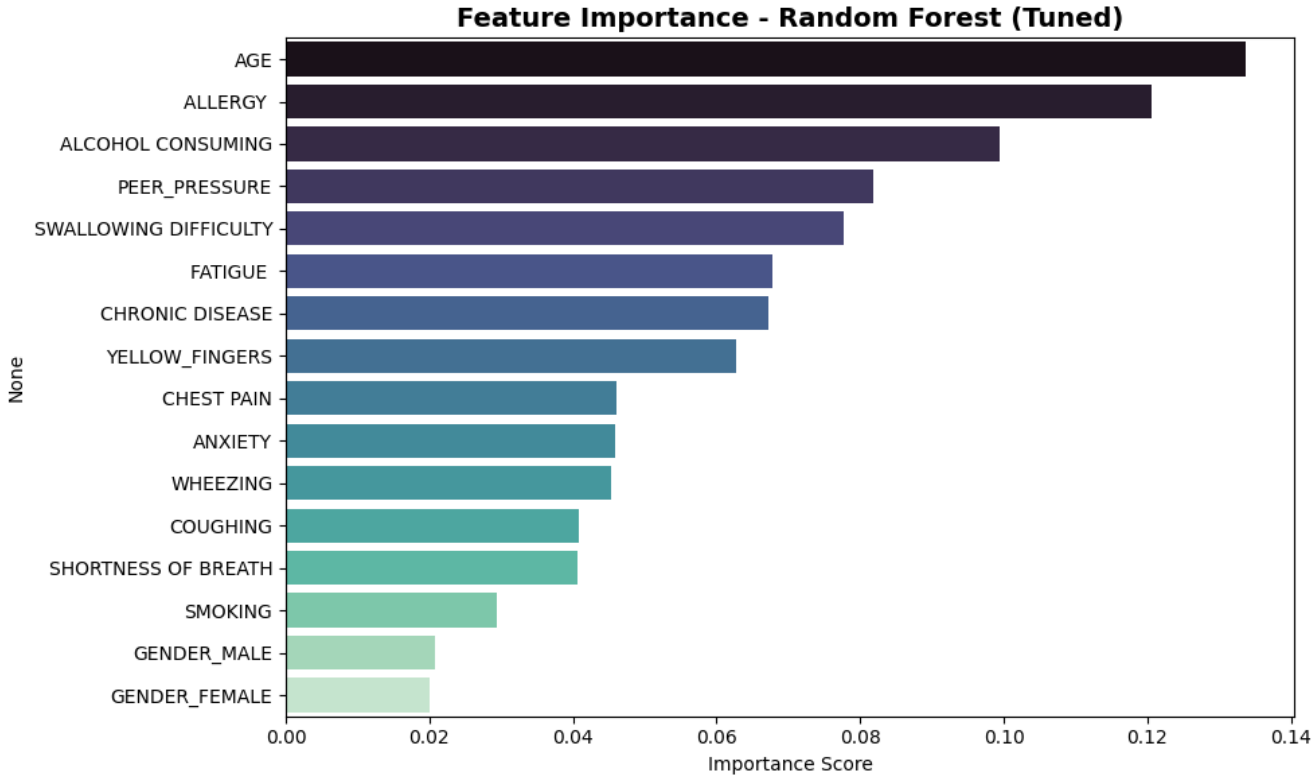
Tahap	KNN	LR	RF	CatBoost	LightGBM
Baseline	0,9464	0,9455	0,9533	0,9444	0,9346
Hapus Duplikat	0,9412	0,9495	0,9485	0,9485	0,9278
Missing Value Handling	0,9412	0,9495	0,9485	0,9485	0,9278
Outlier Handling (IQR)	0,9412	0,9495	0,9485	0,9485	0,9278
One-Hot Encoding (Full)	0,9412	0,9495	0,9583	0,9485	0,9278

Hasil pada Tabel 10 menunjukkan bahwa kontribusi tiap komponen preprocessing bervariasi antar tahap maupun antar algoritma. Penghapusan data duplikat memberikan efek yang beragam: performa Logistic Regression dan CatBoost meningkat, sementara performa KNN, Random Forest, dan LightGBM justru sedikit menurun pada tahap ini. Penanganan missing value tidak memberikan perubahan performa pada seluruh model, sejalan dengan hasil pemeriksaan awal yang menunjukkan bahwa dataset ini tidak memiliki nilai yang hilang. Penanganan outlier menggunakan metode IQR juga tidak menunjukkan perubahan F1-Score pada seluruh model dibandingkan tahap sebelumnya, mengonfirmasi dan memperkuat temuan bahwa distribusi outlier pada fitur AGE tidak cukup memengaruhi kinerja kelima algoritma yang diuji, tidak hanya pada Random Forest. One-Hot Encoding hanya memberikan peningkatan performa yang terlihat pada Random Forest, dari F1-Score 0,9485 menjadi 0,9583, sementara keempat model lainnya tidak menunjukkan perubahan pada tahap ini. Temuan ini menegaskan bahwa penanganan outlier IQR tidak memberikan kontribusi terukur terhadap performa model pada dataset ini di seluruh algoritma yang diuji, sekaligus menunjukkan bahwa kontribusi tiap komponen preprocessing bersifat spesifik terhadap algoritma yang digunakan dan tidak dapat digeneralisasi begitu saja antar model.

G. Analisis Feature Importance

Untuk menganalisis karakteristik fitur yang paling berpengaruh terhadap hasil klasifikasi, dilakukan analisis feature importance menggunakan atribut feature_importances_ dari model Random Forest setelah hyperparameter tuning, yaitu model dengan performa terbaik di antara kelima algoritma yang dibandingkan. Feature importance pada Random Forest dihitung berdasarkan rata-rata penurunan impurity (mean decrease in impurity) yang disumbangkan oleh setiap fitur di seluruh pohon keputusan

dalam ensemble. Hasil feature importance untuk seluruh 16 kolom fitur (15 fitur klinis, dengan GENDER direpresentasikan sebagai dua kolom biner hasil One-Hot Encoding) disajikan pada Gambar 12.



Gambar 12. Visualisasi Feature Importance

Berdasarkan Gambar 12, AGE tercatat sebagai fitur dengan kontribusi tertinggi terhadap keputusan klasifikasi model Random Forest sebesar 0,1337, diikuti oleh ALLERGY sebesar 0,1205 dan ALCOHOL_CONSUMING sebesar 0,0994. Dominasi AGE ini sejalan dengan karakteristiknya sebagai satu-satunya fitur numerik kontinu dalam dataset, yang secara umum menyediakan lebih banyak kemungkinan titik pemisahan (split point) bagi Random Forest dibandingkan fitur biner lainnya, sehingga cenderung berkontribusi lebih besar terhadap penurunan impurity secara kumulatif di seluruh pohon keputusan. Sebaliknya, GENDER_FEMALE sebesar 0,0200, GENDER_MALE sebesar 0,0208, dan SMOKING sebesar 0,0295 tercatat sebagai tiga fitur dengan kontribusi terendah. Temuan bahwa SMOKING memiliki kontribusi yang relatif kecil menarik untuk dicermati, karena riwayat merokok secara klinis dan dalam literatur umumnya dianggap sebagai faktor risiko utama kanker paru. Salah satu kemungkinan penjelasannya adalah tingginya prevalensi riwayat merokok pada dataset ini (62,1% dari seluruh pasien sebagaimana disebutkan pada bagian Pendahuluan), yang menyebabkan fitur tersebut kurang variatif dan kurang diskriminatif secara statistik dibandingkan fitur dengan distribusi yang lebih seimbang seperti ALLERGY dan ALCOHOL_CONSUMING. Fitur GENDER, yang direpresentasikan sebagai dua kolom biner (GENDER_MALE dan GENDER_FEMALE) hasil One-Hot Encoding, secara konsisten menunjukkan kontribusi terendah di antara seluruh fitur, yang mengindikasikan bahwa jenis kelamin bukan merupakan faktor pembeda yang signifikan pada dataset ini. Temuan feature importance ini memberikan konteks tambahan terhadap performa klasifikasi yang telah dibahas sebelumnya dan dapat menjadi dasar pertimbangan klinis serta arah penelitian lanjutan.

IV. KESIMPULAN

Penelitian ini dilakukan menggunakkan perbandingan lima algoritma machine learning, yaitu K-Nearest Neighbors (KNN), CatBoost, Logistic Regression, LightGBM, dan Random Forest, dalam mengklasifikasikan kanker paru menggunakan dataset survey_lung_cancer.csv dengan pendekatan evaluasi bertahap melalui tiga kondisi pemodelan: baseline, setelah preprocessing, dan setelah hyperparameter tuning.

Temuan utama penelitian ini menunjukkan bahwa setiap tahapan pipeline memberikan dampak yang berbeda terhadap masing-masing algoritma. Preprocessing meningkatkan performa Logistic Regression, Random Forest, dan CatBoost, sementara KNN dan LightGBM lebih sensitif terhadap perubahan distribusi data sehingga mengalami sedikit penurunan akurasi. Hyperparameter tuning menggunakan RandomizedSearchCV dengan Stratified K-Fold (k=10) dan

50 iterasi berhasil meningkatkan performa Random Forest secara signifikan, menjadikannya algoritma terbaik dengan akurasi 0,9464 dan F1-Score 0,9684.

Kontribusi utama penelitian ini adalah pembuktian bahwa pipeline yang terstruktur dan seragam antar algoritma merupakan pendekatan yang lebih adil dalam studi komparatif machine learning. Hasil ini juga menunjukkan bahwa preprocessing dan hyperparameter tuning bukan sekadar langkah teknis, melainkan faktor penentu yang signifikan dalam memilih algoritma yang tepat untuk dataset tertentu. Analisis learning curve serta gap F1-Score data latih dan data uji membuktikan bahwa seluruh model tidak mengalami overfitting pada dataset penelitian ini. Meski demikian, klaim keandalan model sebagai alat bantu klasifikasi kanker paru masih terbatas pada karakteristik dataset survey_lung_cancer.csv yang digunakan, karena penelitian ini belum melibatkan validasi pada data klinis independen dari populasi atau institusi lain.

Penelitian ini juga memiliki beberapa batasan. Jumlah iterasi RandomizedSearchCV (n_iter=50) belum dijustifikasi dengan pemeriksaan konvergensi. Evaluasi hanya dilakukan pada satu dataset publik yang relatif kecil (309 data) tanpa validasi eksternal. Ketidakseimbangan kelas pada dataset telah diidentifikasi pada tahap EDA dan ditangani melalui penggunaan F1-Score sebagai metrik utama serta parameter class_weight='balanced' pada proses tuning, namun evaluasi belum didukung oleh metrik tambahan seperti Matthews Correlation Coefficient (MCC) atau balanced accuracy yang dapat memberikan gambaran lebih komprehensif pada kondisi kelas yang tidak seimbang. Analisis feature importance juga masih terbatas pada mean decrease in impurity dari Random Forest tanpa dukungan metode Explainable AI seperti SHAP. Keunggulan performa Random Forest dibandingkan algoritma lain juga belum diuji signifikansinya secara statistik, misalnya menggunakan uji Wilcoxon signed-rank atau paired t-test terhadap hasil cross-validation, sehingga belum dapat dipastikan apakah perbedaan performa antar algoritma benar-benar bermakna secara statistik atau hanya disebabkan oleh variasi acak.

Untuk penelitian selanjutnya, disarankan menggunakan dataset yang lebih besar dan beragam agar kemampuan generalisasi model lebih baik. Selain itu, eksplorasi metode seleksi fitur dan pengujian metode hyperparameter tuning alternatif seperti Bayesian Optimization dapat menjadi arah pengembangan yang menarik. Validasi eksternal menggunakan data klinis independen dari populasi atau institusi yang berbeda juga sangat diperlukan untuk mengonfirmasi kemampuan generalisasi model sebelum dapat dipertimbangkan untuk aplikasi klinis yang sesungguhnya.

DAFTAR PUSTAKA

[1] I. Buana and D. Agustian Harahap, "Asbestos, Radon Dan Polusi Udara Sebagai Faktor Resiko Kanker Paru Pada Perempuan Bukan Perokok," 2022. doi: 10.29103/averrous.v8i1.7088.

[2] S. Mustofa *et al.*, "Laporan Kasus Kanker Paru Kiri Jenis Adenokarsinoma Dengan Hemoptisis Non Masive [2023]," *Jurnal Ilmu Kedokteran dan Kesehatan*, May 2023, doi: 10.33024/jikk.v10i5.10085.

[3] S. Alfarisa, S. Wahyuni, and Efriza, "Karakteristik Pasien Kanker Paru di RSUP Dr. M. Djamil Padang Tahun 2021," *Scientific Journal*, Nov. 2023, doi: 10.56260/sciena.v2i6.116.

[4] A. F. Hamdani, W. Purbaningsih, and W. Y. Nalapraya, "Karakteristik Demografi dan Klinikopatologi Pasien Kanker Paru di RSUD Al−Ihsan," *Jurnal Riset Kedokteran*, pp. 97–102, Dec. 2023, doi: 10.29313/jrk.v3i2.2959.

[5] S. Salsabila, S. A. Intan, and D. W. Fitrina, "Gambaran Tipe Sel Kanker Paru Berdasarkan Usia, Jenis Kelamin, dan Paparan Rokok di RSUP Dr. M. Djamil Padang Tahun 2018-2020," *Jurnal Ilmu Kesehatan Indonesia*, vol. 4, no. 4, pp. 281–288, Dec. 2023, doi: 10.25077/jikesi.v4i4.1118.

[6] S. Jiwandana Pinasthika, "Aplikasi Pembelajaran Mesin dalam Pengolahan Data Citra untuk Bidang Medis: Sebuah Kajian Pustaka," *Journal of Information Engineering and Technology (JIETY*, vol. 2, no. 1, pp. 3026–6459, Mar. 2024, doi: 10.21831/jiety.v2i1.249.

[7] R. S. Nurhalizah, R. Ardianto, and P. Purwono, "Analisis Supervised dan Unsupervised Learning pada Machine Learning: Systematic Literature Review," *Jurnal Ilmu Komputer dan Informatika*, vol. 4, no. 1, pp. 61–72, Aug. 2024, doi: 10.54082/jiki.168.

[8] V. Artanti, M. Faisal, and F. Kurniawan, "Klasifikasi Cardiovascular Diseases Menggunakan Algoritma K-Nearest Neighbors (KNN) Classification of Cardiovascular Diseases using K-Nearest Neighbors (KNN) Algorithm," vol. 23, no. 2, pp. 467–479, May 2024, doi: 10.62411/tc.v23i2.10061.

[9] D. Fabiyanto and Z. Pratama Putra, "Validasi Efektivitas Logistic Regression untuk Diagnosa Penyakit Jantung melalui Pendekatan Machine Learning," *Jurnal Ilmiah FIFO*, vol. 16, no. 2, p. 158, Nov. 2024, doi: 10.22441/fifo.2024.v16i2.006.

[10] E. Ramadanti, D. A. Dinathi, C. Sri, K. Aditya, and R. Chandranegara, "Diabetes Disease Detection Classification Using Light Gradient Boosting (LightGBM) With Hyperparameter Tuning," *Jurnal dan Penelitian Teknik Informatika*, vol. 8, no. 2, 2024, doi: 10.33395/v8i2.13530.

[11] N. L. Sabili, R. Fajri, and M. Umbara, "Klasifikasi Penyakit Diabetes Menggunakan Algoritma Categorical Boosting Dengan Faktor Risiko Diabetes," 2024.

[12] D. A. Hadi and D. A. N. Sirodj, "Metode Random Forest untuk Klasifikasi Penyakit Diabetes," *Bandung Conference Series: Statistics*, vol. 3, no. 2, pp. 428–435, Aug. 2023, doi: 10.29313/bcss.v3i2.8354.

[13] B. Swarnadwip and S. Bose, "Random Forests: The Wisdom of Crowds in Action," *Journal of Emerging Trends in Computer Science and Applications (JETCSA)*, vol. 1, no. 1, pp. 67–91, Apr. 2025, doi: 10.65525/jetcsa.v1i1.5.

[14] A. Maulana, A. Pratama, D. Primanda, and N. Hariyanto, "Model Prediksi Kanker Paru-Paru dengan Random Forest Lung Cancer Prediction Model with Random Forest," *Jurnal Sisfotenika*, vol. 15, no. 2, 2025, doi: 10.30700/sisfotenika.v15i1.569.

[15] Y. Chithra, P. Kiran, and M. P B, "The Novel Method for Data Preprocessing CLI," *Advances in Intelligent Systems and Technologies*, pp. 117–120, Dec. 2022, doi: 10.53759/aist/978-9914-9946-1-2_21.

[16] T. Gori, A. Sunyoto, and H. Al Fatta, "Preprocessing Data dan Klasifikasi untuk Prediksi Kinerja Akademik Siswa," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 11, no. 1, pp. 215–224, Feb. 2024, doi: 10.25126/jtiik.20241118074.

[17] T. A. E. Putri, T. Widiharih, and R. Santoso, "Penerapan Tuning Hyperparameter Randomsearchcv Pada Adaptive Boosting Untuk Prediksi Kelangsungan Hidup Pasien Gagal Jantung," *Jurnal Gaussian*, vol. 11, no. 3, pp. 397–406, Jan. 2023, doi: 10.14710/j.gauss.11.3.397-406.