

Peningkatan Akurasi Deteksi Liver Disease melalui Hyperparameter Tuning pada Algoritma Random Forest

Cinta Azzaria^{1,a)}, Erna Daniati^{2,a)}, dan Aidina Ristyawan^{3,a)}

^{a)}Program Studi Sistem Informasi,
Fakultas Teknik dan Ilmu Komputer
Universitas Nusantara PGRI Kediri

Author Emails

Corresponding author: ²⁾ernadaniati@unpkediri.ac.id

¹⁾cintaazzaria72@gmail.com

³⁾aidinaristi@unpkediri.ac.id

Abstract. Liver disease is one of the leading causes of global mortality, making early detection crucial to support timely and effective medical intervention. This study aims to improve the accuracy of liver disease detection using the Random Forest algorithm. The dataset used is the Indian Liver Patient Dataset (ILPD), consisting of 583 entries. The research follows the CRISP-DM approach and applies SMOTE to address class imbalance, as well as Grid Search CV for hyperparameter optimization. Unlike previous studies, this research systematically combines SMOTE and Grid Search to enhance model performance on the ILPD dataset. The results show that model accuracy improved from 74% to 75%, with better precision and recall, particularly for the class of patients diagnosed with liver disease. These findings indicate that an optimized Random Forest approach can serve as a reliable method to support early diagnosis of liver disease.

Keywords: Liver Disease Detection, Random Forest, ILPD, SMOTE, Grid Search

Abstraksi. Penyakit liver merupakan salah satu penyebab utama kematian global, sehingga deteksi dini sangat penting untuk mendukung penanganan medis yang cepat dan tepat. Penelitian ini bertujuan meningkatkan akurasi deteksi penyakit liver menggunakan algoritma *Random Forest*. Dataset yang digunakan adalah *Indian Liver Patient Dataset* (ILPD) yang terdiri dari 583 entri. Penelitian mengikuti pendekatan CRISP-DM dan menerapkan teknik SMOTE untuk mengatasi ketidakseimbangan kelas serta *Grid Search CV* untuk optimasi *hyperparameter*. Berbeda dengan penelitian sebelumnya, studi ini menggabungkan SMOTE dan *Grid Search* secara sistematis untuk meningkatkan performa model pada dataset ILPD. Hasil menunjukkan bahwa akurasi model meningkat dari 74% menjadi 75%, dengan perbaikan pada *precision* dan *recall*, khususnya untuk kelas pasien yang terdiagnosis *liver disease*. Temuan ini menunjukkan bahwa *Random Forest* yang dioptimalkan melalui pendekatan ini dapat menjadi metode andal dalam mendukung diagnosis dini penyakit liver.

Kata Kunci : Liver Disease, Random Forest, Hyperparameter Tuning, SMOTE, ILPD, CRISP-DM

PENDAHULUAN

Kematian akibat penyakit hati adalah masalah global yang serius seiring dengan bertambahnya jumlah kasus setiap tahunnya[1]. Faktor penyebabnya sangat beragam, antara lain gangguan autoimun yang menyerang sel-sel hati, efek samping penggunaan obat-obatan tertentu dalam jangka panjang, serta paparan zat toksik atau bahan kimia berbahaya yang dapat merusak jaringan hati secara perlahan. Kompleksitas gejala dan minimnya keluhan pada tahap awal membuat diagnosis dini menjadi tantangan tersendiri bagi tenaga medis. Oleh karena itu, dibutuhkan pendekatan

komputasional berbasis data seperti machine learning untuk membangun sistem klasifikasi prediktif yang mampu mengidentifikasi risiko penyakit liver secara lebih dini, cepat, dan akurat, guna mendukung pengambilan keputusan medis yang lebih efektif. Kemajuan teknologi informasi di bidang kesehatan telah menghasilkan volume data medis yang sangat besar, seperti catatan rekam medis, hasil laboratorium, dan data kesehatan masyarakat. Data ini memiliki potensi besar untuk dimanfaatkan dalam menganalisis perkembangan penyakit liver, mengidentifikasi faktor risiko, serta mendukung pengambilan keputusan klinis. Namun, Keragaman dan kompleksitas data tersebut membutuhkan pendekatan analisis yang tepat [2].

Penerapan *machine learning* dalam proses *data mining* telah memberikan kontribusi yang signifikan dalam meningkatkan kemampuan analisis data di bidang kesehatan, terutama dalam mengidentifikasi penyakit yang kompleks seperti penyakit liver. Pentingnya peningkatan akurasi dan efisiensi algoritma klasifikasi telah mendorong banyak penelitian dan implementasi secara berkelanjutan. Dalam hal ini, studi [3] menunjukkan bahwa algoritma *Decision Tree* mencapai akurasi tertinggi sebesar 72,74%, mengungguli *Neural Network*, K-NN, dan *Naïve Bayes*. Hasil ini sejalan dengan penelitian [4] yang mencatat akurasi 70,29% untuk *Decision Tree* dan 67,05% untuk *Naïve Bayes*. Penelitian lainnya oleh [5] melaporkan bahwa *Random Forest* mampu menghasilkan akurasi sebesar 70,60%, mengungguli *Naïve Bayes*. Sementara itu, beberapa studi terkini juga memberikan perspektif tambahan, seperti yang disampaikan oleh [6] yang mencatat akurasi 70% untuk K-NN, serta [7] yang memperoleh akurasi tertinggi sebesar 71,79% menggunakan algoritma ANN. Meskipun menunjukkan performa yang menjanjikan, setiap algoritma memiliki keterbatasan masing-masing. Beberapa di antaranya sensitif terhadap nilai pencilan (*outlier*), sulit diinterpretasikan secara langsung, dan rentan mengalami *overfitting*.

Untuk mengatasi berbagai kelemahan tersebut, algoritma *Random Forest* dianggap lebih unggul karena kemampuannya dalam menangani data dengan berbagai jenis fitur, ketahanannya terhadap data ekstrem, serta kemampuannya membentuk model yang stabil dan lebih mudah dipahami [3]. Selain keunggulan yang dimilikinya, kinerja algoritma *Random Forest* masih dapat ditingkatkan melalui penyesuaian parameter atau yang dikenal dengan *hyperparameter tuning*. Proses tuning melibatkan pengaturan beberapa aspek penting seperti jumlah pohon yang digunakan, kedalaman maksimum tiap pohon, dan jumlah minimum data pada setiap cabang keputusan [4]. Tujuannya adalah agar model dapat lebih sesuai dengan pola dan karakteristik data yang dianalisis, sehingga hasil prediksi menjadi lebih akurat.

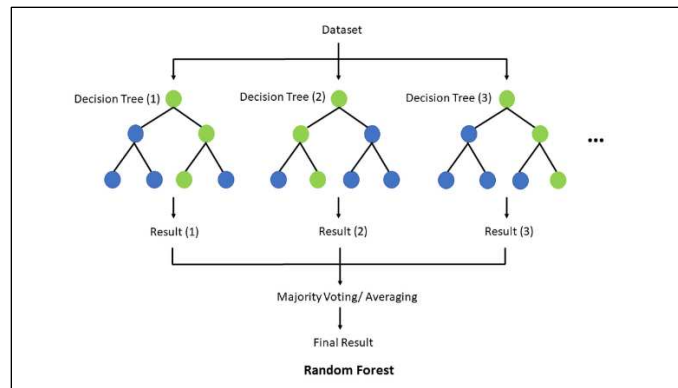
Salah satu pendekatan yang sering diterapkan untuk menemukan susunan parameter terbaik adalah *Grid Search*, yaitu teknik pencarian sistematis untuk menemukan nilai parameter yang menghasilkan performa model optimal [5]. Berdasarkan hasil penelitian [5], penerapan *hyperparameter tuning* menggunakan *Grid Search* pada algoritma *Random Forest* terbukti mampu meningkatkan akurasi model secara signifikan, yaitu sebesar 18,48% setelah dilakukan tuning dengan *Grid Search*. Hal ini menunjukkan bahwa penggunaan *Grid Search* berkontribusi langsung dalam meningkatkan kinerja model klasifikasi pada kasus deteksi penyakit liver.

Dengan memanfaatkan keunggulan algoritma *Random Forest* dan menerapkan *hyperparameter tuning* secara sistematis menggunakan *Grid Search*, penelitian ini menghadirkan pendekatan baru dalam meningkatkan akurasi deteksi penyakit liver sejak tahap awal yaitu terletak pada optimalisasi performa model klasifikasi melalui penyetelan parameter yang terstruktur. Model yang dikembangkan ini diharapkan dapat digunakan sebagai referensi dalam studi lanjutan yang fokus pada peningkatan akurasi diagnosis dan pengembangan metode deteksi dini yang lebih efektif untuk penyakit liver.

TINJAUAN PUSTAKA

Random Forest

Random Forest adalah algoritma *ensemble learning* yang bekerja dengan membentuk sejumlah pohon keputusan (*decision tree*) dan menggabungkan prediksinya agar menghasilkan output akhir [6]. Algoritma ini unggul dalam mengelola data dengan banyak fitur, memberikan hasil yang konsisten, dan mampu meminimalkan risiko *overfitting* [7]. Karena fleksibilitasnya, *Random Forest* dapat diterapkan baik pada tugas klasifikasi maupun regresi. Untuk memberikan gambaran yang lebih jelas mengenai cara kerja algoritma ini, berikut ditampilkan ilustrasi dari *Random Forest*.



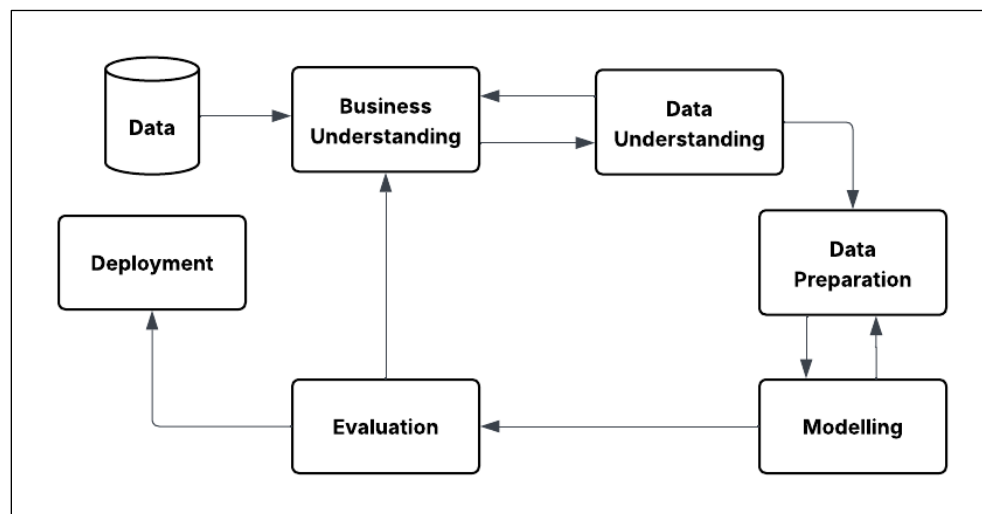
GAMBAR 1. Algoritma *Random Forest*

Grid Search Cross Validation

Salah satu cara untuk memilih *hyperparameter* pada fungsi *kernel* adalah dengan menggunakan metode *Grid Search* yang dikombinasikan dengan *Cross Validation*. *Grid search* merupakan metode pencarian sistematis terhadap kombinasi parameter untuk menemukan konfigurasi terbaik dari suatu model. Teknik ini bekerja dengan membangun dan mengevaluasi model berdasarkan setiap kombinasi *hyperparameter* yang telah ditentukan sebelumnya [8]. Akibatnya, *Grid Search* cenderung memerlukan sumber daya komputasi yang besar dan waktu yang cukup lama untuk menemukan hasil yang optimal. *Cross validation* (CV) sendiri adalah metode yang sering digunakan untuk evaluasi model. Jenis yang paling sederhana dari CV adalah metode holdout yang memisahkan data menjadi dua set, yaitu data training dan data testing [9].

METODE PENELITIAN

Metode CRISP-DM (*Cross Industry Standard Process for Data Mining*) digunakan dalam penelitian ini sebagai kerangka kerja yang terstruktur dan standar untuk membangun sistem data mining secara sistematis. Pendekatan ini dipilih karena dinilai sangat cocok untuk membangun model klasifikasi berbasis *machine learning*, khususnya dalam mendiagnosis penyakit liver. CRISP-DM adalah metodologi yang terdiri dari enam tahapan inti yaitu *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modelling*, *Evaluation*, dan *Deployment* [11].



GAMBAR 2. Metode CRISP-DM

Berikut penjelasan tahapan CRISP-DM yang digambarkan pada Gambar 1:

1. *Bussines Understanding*

Tahap ini berfokus pada pemahaman mendalam terhadap tujuan bisnis serta permasalahan yang ingin dipecahkan. Tahap ini meliputi penentuan sasaran proyek dan kebutuhan stakeholder kepentingan agar hasil analisis yang diperoleh dapat memberikan manfaat nyata dan terukur.

2. *Data Understanding*

Setelah memahami tujuan bisnis, langkah berikutnya adalah mengumpulkan data awal dan melakukan eksplorasi terhadap data tersebut. Proses ini bertujuan untuk mengenali karakteristik data, kualitasnya, serta potensi permasalahan seperti data yang hilang, nilai ekstrem (outlier), dan ketidakseimbangan kelas yang dapat memengaruhi hasil analisis.

3. *Data Preparation*

Proses pembuatan model membutuhkan data yang sudah diolah. Tahapan ini mencakup pembersihan data, pengelolaan nilai yang hilang, transformasi data, encoding variabel kategorik, dan penerapan teknik SMOTE untuk menyeimbangkan distribusi kelas dengan tujuan untuk menghasilkan data berkualitas yang sesuai untuk proses pemodelan. Metode yang dikenal sebagai SMOTE (*Synthetic Minority Over-sampling Technique*) bertujuan untuk menyeimbangkan jumlah data antar kelas dengan cara membuat data baru dari kelas minoritas melalui interpolasi antar sampel yang ada [10].

4. *Modelling*

Pada tahap ini dilakukan pemilihan algoritma yang sesuai yaitu *Random Forest*. kemudian dilakukan pelatihan dan pengujian model menggunakan data yang telah disiapkan. Selain itu, dilakukan penyetelan parameter dengan *Grid Search CV* untuk memperoleh model dengan performa terbaik.

5. *Evaluation*

Setelah model dikembangkan, langkah selanjutnya adalah menilai kinerjanya dengan menggunakan metrik yang relevan seperti akurasi, presisi, recall, atau F1-score. Evaluasi juga mencakup peninjauan terhadap sejauh mana model mampu menjawab permasalahan bisnis yang telah ditetapkan di awal. Jika performa model belum memuaskan, maka dapat dilakukan iterasi kembali ke tahapan sebelumnya untuk dilakukan perbaikan.

6. *Deployment*

Tahapan terakhir adalah mengimplementasikan model ke dalam sistem yang digunakan secara operasional agar hasil prediksi dapat dimanfaatkan secara nyata. Selain itu, penting untuk memantau performa model secara berkala serta pemeliharaan agar model selalu relevan dan efektif menghadapi perubahan kondisi atau data terbaru.

HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil implementasi penelitian serta pembahasannya secara sistematis berdasarkan tahapan dalam kerangka kerja CRISP-DM (*Cross Industry Standard Process for Data Mining*).

1. *Bussines Understanding*

Langkah pertama adalah memahami permasalahan utama yang dihadapi, yaitu perlunya deteksi dini penyakit liver untuk membantu proses diagnosis. Dalam tahap ini dilakukan identifikasi kebutuhan, tujuan proyek, dan manfaat potensial dari solusi berbasis data, agar sistem yang dikembangkan dapat memberikan dampak nyata pada proses pengambilan keputusan.

2. *Data Understanding*

Setelah tujuan bisnis ditetapkan, langkah berikutnya adalah melakukan analisis awal terhadap data. Riset ini memanfaatkan *dataset Indian Liver Patient Dataset (ILPD)* dari *UCI Machine Learning Repository* yang terdiri dari 583 entri dengan beberapa atribut dan satu variabel target. Proses eksplorasi awal dilakukan untuk mengidentifikasi tipe data, pola distribusi, serta potensi masalah kualitas data.

```

Age                int64
Gender             object
Total_Bilirubin    float64
Direct_Bilirubin   float64
Alkaline_Phosphotase int64
Alamine_Aminotransferase int64
Aspartate_Aminotransferase int64
Total_Proteins     float64
Albumin            float64
Albumin_and_Globulin_Ratio float64
Label              int64
dtype: object

```

GAMBAR 3. Tipe data ILPD

Berdasarkan analisis awal yang ditampilkan pada Gambar 3, dataset ILPD terdapat lima atribut dengan format numerik desimal (float64), lima atribut numerik integer (int64), serta satu atribut bertipe non-numerik (object). Selain itu, ditemukan adanya nilai yang hilang (*missing value*) pada atribut “*Albumin_and Globulin_Ratio*” sebanyak 4 data yang ditunjukkan pada Gambar 4. Hasil eksplorasi awal juga ditampilkan pada Gambar 5 yang menunjukkan adanya 13 baris data duplikat pada dataset yang memiliki nilai yang identik di semua atribut. Apabila data duplikat dan nilai yang hilang tidak ditangani, maka dapat menimbulkan bias karena model belajar dari informasi yang berulang atau tidak lengkap.

```

Missing Values Before Handling:
Age                0
Gender             0
Total_Bilirubin    0
Direct_Bilirubin   0
Alkaline_Phosphotase 0
Alamine_Aminotransferase 0
Aspartate_Aminotransferase 0
Total_Proteins     0
Albumin            0
Albumin_and_Globulin_Ratio 4
Label              0
dtype: int64

```

GAMBAR 4. Temuan *Missing Value*

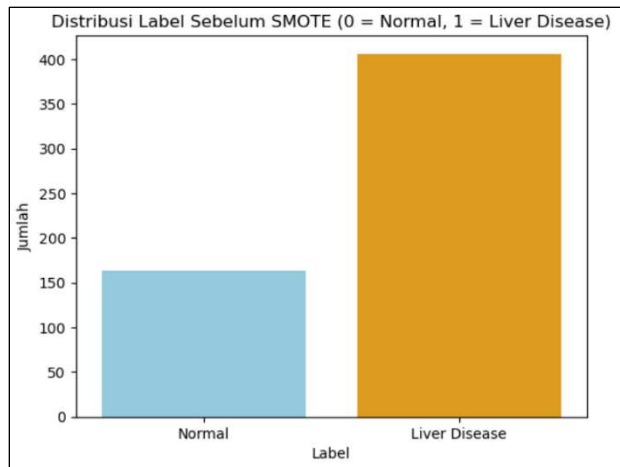
Data duplikat ditemukan: 13

	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Proteins	Albumin	Album
18	40	Female	0.9	0.3	293	232	245	6.8	3.1	
19	40	Female	0.9	0.3	293	232	245	6.8	3.1	
25	34	Male	4.1	2.0	289	875	731	5.0	2.7	
26	34	Male	4.1	2.0	289	875	731	5.0	2.7	
33	38	Female	2.6	1.2	410	59	57	5.6	3.0	

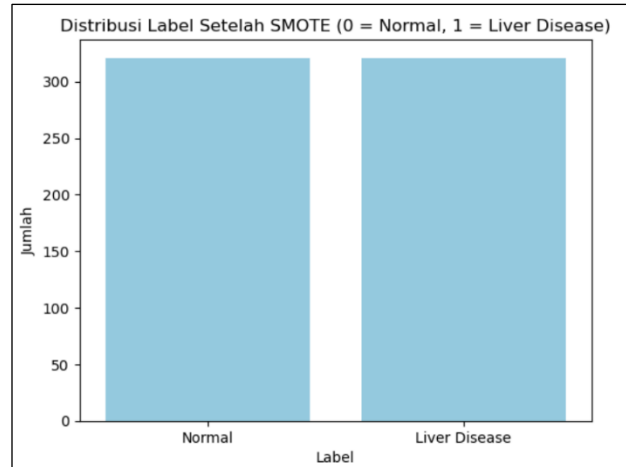
Gambar 5. Data Duplikasi

3. Data Preparation

Setelah tahap pemahaman data selesai, data dipersiapkan agar siap digunakan dalam proses pelatihan model. Langkah pertama dalam tahap ini adalah menangani nilai yang hilang (*missing value*) yang sebelumnya telah diidentifikasi pada tahap *Data Understanding*. Nilai yang hilang ditangani menggunakan teknik imputasi median pada fitur numerik untuk menjaga kestabilan distribusi data. Selanjutnya, dilakukan proses penghapusan data duplikat untuk menghindari bias dalam proses pelatihan model. Kemudian dilakukan encoding terhadap atribut kategorikal yakni pada atribut “Gender” agar dapat diubah menjadi nilai numerik menggunakan teknik *Label Encoding* sehingga bisa diproses oleh algoritma. Setelah itu, data dipisahkan menjadi fitur (X) dan label (y) sebagai persiapan untuk tahap pelatihan model. Mengacu pada hasil eksplorasi awal, distribusi kelas target pada dataset ILPD tergolong tidak seimbang (*imbalanced*), di mana jumlah pasien dengan liver bermasalah jauh lebih banyak dibandingkan dengan pasien yang sehat. Untuk mengatasi hal ini, maka diterapkan teknik SMOTE. Proses ini dilakukan hanya pada *data training* setelah proses pemisahan data menjadi 80% *data training* dan 20% *data testing* agar evaluasi model tetap objektif dan tidak terpengaruh oleh data hasil *oversampling*. Distribusi data ditampilkan pada Gambar 6a dan Gambar 6b untuk menunjukkan perbedaan sebelum dan sesudah diterapkannya SMOTE.



GAMBAR 6a. Distribusi Label sebelum SMOTE



GAMBAR 6b. Distribusi Label setelah SMOTE

4. Modelling

Pada tahap ini, data yang telah dipersiapkan sebelumnya akan digunakan untuk melatih model dan kemudian diuji kemampuannya dalam mengklasifikasikan data baru secara akurat. Dalam proyek ini, algoritma yang digunakan untuk pengujian adalah *Random Forest*. Pengujian dilakukan dalam dua tahap, yaitu:

a. Tanpa Tuning (*Default Parameters*)

Pada tahap awal, model *Random Forest* dibangun menggunakan parameter default dari pustaka *scikit-learn*. Ini dilakukan untuk melihat performa dasar model tanpa penyesuaian apapun terhadap konfigurasi parameternya. Model dilatih menggunakan data pelatihan yang telah diseimbangkan dengan teknik SMOTE dan dievaluasi pada data pengujian. Hasil dari evaluasi model *Random Forest* tanpa tuning ditampilkan pada Tabel 1.

Tabel 1. Hasil *Classification Report* model *Random Forest* tanpa Tuning

Class	Precision	Recall	F1-Score	Support
0	0.48	0.38	0.42	29
1	0.80	0.86	0.83	85
Accuracy			0.74	114

Berdasarkan hasil evaluasi awal, model *Random Forest* tanpa tuning parameter mampu mencapai tingkat akurasi sebesar 74%, yang menunjukkan performa klasifikasi yang cukup baik secara umum. Namun, performa model menunjukkan perbedaan antara masing-masing kelas. Pada kelas 0 (Normal), precision tercatat sebesar 0.48 dan recall sebesar 0.38, menandakan bahwa kemampuan model dalam mengidentifikasi pasien yang tidak mengalami gangguan liver masih terbatas. Sebaliknya, untuk kelas 1 (*Liver Disease*) model menunjukkan hasil yang lebih unggul dengan precision sebesar 0.80 dan recall 0.86. Hal ini mengindikasikan bahwa model cukup efektif dalam mendeteksi pasien yang mengalami gangguan liver.

b. Dengan Tuning menggunakan *Grid Search CV*

Sebagai bagian dari upaya peningkatan kinerja model, dilakukan *hyperparameter tuning* menggunakan pendekatan *Grid Search Cross-Validation* (*GridSearchCV*). Metode *GridSearchCV* digunakan untuk secara sistematis mengevaluasi berbagai kombinasi parameter guna menemukan konfigurasi yang memberikan performa terbaik. Parameter yang disesuaikan meliputi jumlah pohon (*n_estimators*) dengan nilai 100 dan 200, kedalaman maksimum pohon (*max_depth*) dengan opsi *None*, 10, dan 20, serta jumlah minimum sampel untuk pemisahan node (*min_samples_split*) yang diuji dengan nilai 2 dan 5.

Dengan kombinasi tersebut, terdapat 12 konfigurasi parameter yang diuji dalam proses ini. Evaluasi dilakukan menggunakan *5-fold cross-validation*, yaitu teknik pembagian data pelatihan menjadi lima bagian yang secara bergantian digunakan untuk pelatihan dan validasi(). Strategi ini bertujuan untuk memperoleh estimasi performa model yang lebih stabil dan menghindari *overfitting*. Hasil evaluasi dari seluruh konfigurasi disusun dengan menampilkan skor rata-rata validasi silang (*mean_test_score*) untuk masing-masing kombinasi yang ditampilkan pada Gambar 7 yang berbentuk dataframe.

	param_n_estimators	param_max_depth	param_min_samples_split	\
0	100	None	2	
1	200	None	2	
2	100	None	5	
3	200	None	5	
4	100	10	2	
5	200	10	2	
6	100	10	5	
7	200	10	5	
8	100	20	2	
9	200	20	2	
10	100	20	5	
11	200	20	5	
mean_test_score				
0	0.684281			
1	0.666770			
2	0.660201			
3	0.660177			
4	0.664548			
5	0.660153			
6	0.675585			
7	0.671142			
8	0.688677			
9	0.668968			
10	0.660201			
11	0.660177			

Best Parameters:
{ 'max_depth': 20, 'min_samples_split': 2, 'n_estimators': 100 }

Best Mean Cross-Validation Score: 0.6887

GAMBAR 7. Hasil *Grid Search CV*

Konfigurasi parameter terbaik berdasarkan hasil *Grid Search CV* yaitu *n_estimators* = 100, *max_depth* = 20 dan *min_samples_split* = 2. Kombinasi tersebut menghasilkan nilai *mean_test_score* tertinggi sebesar 0.688677 atau 68,87% yang merupakan performa paling optimal di antara seluruh konfigurasi yang diuji. Performa model yang telah dituning ditampilkan pada Tabel 2.

TABEL 2. Hasil *Classification Report* model *Random Forest* dengan tuning

Class	Precision	Recall	F1-Score	Support
0	0.48	0.38	0.42	29
1	0.80	0.86	0.83	85
Accuracy			0.74	114

Setelah dilakukan penyetelan parameter menggunakan metode *Grid Search CV*, performa model *Random Forest* mengalami peningkatan dengan akurasi keseluruhan mencapai 75%. Hasil ini menunjukkan adanya peningkatan stabilitas dibandingkan dengan model awal. Untuk kelas 0 (Normal), *precision* tercatat sebesar 0.50 dan *recall* sebesar 0.38, yang menggambarkan adanya sedikit peningkatan dalam mendeteksi pasien yang sehat. Sementara itu, pada kelas 1 (*Liver Disease*), model menunjukkan performa yang solid dengan *precision* sebesar 0.80 dan *recall* sebesar 0.87. Nilai ini mencerminkan kemampuan model yang semakin baik dalam mengidentifikasi pasien yang mengidap penyakit liver setelah dilakukan tuning parameter.

5. Evaluation

Evaluasi hasil klasifikasi dilakukan untuk menilai sejauh mana model mampu mendeteksi penyakit liver secara akurat dan konsisten. Setelah penerapan hyperparameter tuning menggunakan *Grid Search CV*, akurasi model meningkat menjadi 75%, menunjukkan peningkatan yang marginal namun konsisten terhadap kinerja awal. Selain itu, nilai *precision* dan *recall* pada kelas 1 juga mengalami peningkatan menjadi masing-masing 0.80 dan 0.87, yang mengindikasikan bahwa model semakin andal dalam mengidentifikasi pasien dengan penyakit liver. Penerapan teknik *Synthetic Minority Over-sampling Technique* (SMOTE) turut memberikan kontribusi positif dalam menyeimbangkan distribusi kelas, sehingga mengurangi bias terhadap kelas mayoritas. Namun demikian, kemampuan model dalam mengenali kelas minoritas (pasien non-liver) masih menunjukkan keterbatasan, sebagaimana tercermin dari nilai *recall* yang relatif lebih rendah pada kelas 0.

KESIMPULAN

Temuan utama dari penelitian ini menunjukkan bahwa penerapan algoritma *Random Forest* yang dikombinasikan dengan teknik penyeimbangan data SMOTE serta penyetelan hyperparameter melalui *Grid Search CV* berkontribusi dalam meningkatkan performa model dalam mendeteksi penyakit liver pada dataset ILPD. Hasil evaluasi memperlihatkan bahwa model awal memiliki akurasi sebesar 74%, yang kemudian meningkat menjadi 75% setelah dilakukan tuning. Meskipun peningkatan tersebut tergolong marginal, hal ini tetap mencerminkan adanya perbaikan yang konsisten dalam stabilitas dan kemampuan generalisasi model, khususnya dalam mengidentifikasi pasien dengan penyakit liver. Dengan demikian, kombinasi antara optimasi parameter dan penyeimbangan data dapat dianggap efektif dalam memperbaiki kinerja klasifikasi, meskipun peningkatan akurasi secara absolut tidak signifikan.

DAFTAR PUSTAKA

- [1] W. Izaaz Hernando and P. Yuhhy Hernanda, “Studi Literatur Sistematis Konsumsi Alkohol terhadap Penyakit Liver Kronik,” 2024.
- [2] A. F. Abadi, N. Alamsyah, F. G. Retnanto, E. Daniati, and A. Ristyawan, “Penerapan Data Mining dalam Mengklasifikasi Penyakit Stroke Menggunakan Algoritma,” Online, 2024.
- [3] H. T. Santoso, F. A. Felmidi, A. N. Fadhila, A. Ristyawan, and E. Daniati, “Analisis Kinerja Algoritma Data Mining pada Klasifikasi Tingkat Obesitas dengan K-Fold Cross Validation dan AUC,” Online, 2024.
- [4] A. Priyono, M. Shodiq, D. P. Alvinsyah, and S. A. Hidayah, “METODE RANDOM FOREST UNTUK MEMUDAHKAN KLASIFIKASI DIAGNOSIS PENYAKIT

- MENTAL,” *Jurnal Informatika Medis (J-INFORMED)*, vol. 2, no. 1, pp. 1–4, Jun. 2024, doi: 10.52060/im.v2i1.2119.
- [5] N. Faoziatun Khusna *et al.*, “Implementasi Random Forest dalam Klasifikasi Kasus Stunting pada Balita dengan Hyperparameter Tuning Grid Search,” *Seminar Nasional Sains Data*, vol. 2024, 2024.
- [6] P. T. S. Ningsih, M. Gusvarizon, and R. Hermawan, “Analisis Sistem Pendeteksi Penipuan Transaksi Kartu Kredit dengan Algoritma Machine Learning,” *Jurnal Teknologi Informatika dan Komputer*, vol. 8, no. 2, pp. 386–401, Sep. 2022, doi: 10.37012/jtik.v8i2.1306.
- [7] F. Ardyansyah, E. Daniati, and A. Ristyawan, “Pemanfaatan Data Mining untuk Analisis Keputusan,” Online, 2024.
- [8] D. Statistika Bisnis, F. Vokasi, and I. Teknologi Sepuluh Nopember, “KLASIFIKASI AKSES INTERNET OLEH ANAK-ANAK DAN REMAJA DEWASA DI JAWA TIMUR MENGGUNAKAN SUPPORT VECTOR MACHINE MUKTI RATNA DEWI,” *J. Ris. & Ap. Mat*, vol. 4, no. 1, pp. 17–27, 2020.
- [9] R. Tuntun, K. Kusriani, and K. Kusnawi, “Analisis Perbandingan Kinerja Algoritma Klasifikasi dengan Menggunakan Metode K-Fold Cross Validation,” *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 6, no. 4, p. 2111, Oct. 2022, doi: 10.30865/mib.v6i4.4681.
- [10] A. Nugroho and D. Harini, “Teknik Random Forest untuk Meningkatkan Akurasi Data Tidak Seimbang,” *JSITIK*, vol. 2, no. 2, 2024, doi: 10.53624/jsitik.v2i2.XX.