JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 2, April 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i2.8517
Hal 84-94
http://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

Sistem Klasifikasi Tingkat Kerusakan Kunci Motor Menggunakan Random Forest dengan Hyperparameter Tuning

Jalu Wira Yuda*, Hastie Audytra, Nur Mahmudah

Fakultas Sains dan Teknologi, Program Studi Teknik Informatika, Universitas Nahdlatul Ulama Sunan Giri, Bojonegoro, Indonesia
Email: [1,*]ghoszaki20@gmail.com, [2]hastie.audytra@gmail.com, [3,] Mudah15@gmail.com
Email Penulis Korespondensi: ghoszaki20@gmail.com
Submitted **26-03-2025;** Accepted **26-04-2025;** Published **30-04-2025**

Abstrak

Kerusakan kunci motor sering menjadi kendala bagi pengguna, sementara proses identifikasinya masih bergantung pada teknisi yang dapat memakan waktu lama dan bersifat subjektif. Penelitian ini mengembangkan sistem klasifikasi tingkat kerusakan kunci motor menggunakan metode Random Forest dengan optimasi hyperparameter. Dataset terdiri dari 1000 sampel yang dikumpulkan melalui observasi dan wawancara teknisi, dengan preprocessing data menggunakan teknik SMOTE untuk mengatasi ketidakseimbangan kelas. Model dilatih dan dioptimasi dengan Random Forest menggunakan GridSearchCV serta dievaluasi dengan akurasi, precision, recall, dan F1-score. Hasil penelitian menunjukkan bahwa model Random Forest yang dioptimasi mencapai akurasi 85,5%, meningkat dari 81,5% sebelum tuning, sehingga mampu mengidentifikasi tingkat kerusakan kunci motor dengan lebih cepat dan akurat. Implementasi sistem ini diharapkan dapat meningkatkan efisiensi layanan perbaikan dan membantu pengguna mengambil tindakan sebelum kerusakan semakin parah.

Kata Kunci: Kerusakan Kunci Motor; Machine Learning; Hyperparameter; Klasifikasi; Random Forest.

Abstract

Motorcycle key damage is often a problem for users, while the identification process still relies on technicians, which can be time-consuming and subjective. This study develops a classification system for motorcycle key damage levels using the Random Forest method with hyperparameter optimization. The dataset consists of 1,000 samples collected through observation and technician interviews, with data preprocessing using the SMOTE technique to address class imbalance. The model is trained and optimized with Random Forest using GridSearchCV and evaluated based on accuracy, precision, recall, and F1-score. The results show that the optimized Random Forest model achieves an accuracy of 85.5%, an improvement from 82% before tuning, enabling faster and more accurate identification of motorcycle key damage levels. The implementation of this system is expected to improve repair service efficiency and help users take action before the damage worsens.

Keywords: Motorcycle Key Damage; Machine Learning; Hyperparameter; Classification Random Forest.

1. PENDAHULUAN

Kerusakan kunci motor merupakan permasalahan yang sering terjadi dan dapat menghambat mobilitas pengguna. Keausan mekanis, faktor lingkungan, serta kesalahan penggunaan sering kali menjadi penyebab utama permasalahan ini. Saat ini, proses identifikasi dan perbaikan kunci motor masih sangat bergantung pada ahli kunci yang mengandalkan pengalaman, sehingga berpotensi menghasilkan diagnosis yang subjektif dan memerlukan waktu lama. Ketidakefisienan ini dapat menyebabkan pengguna mengalami keterlambatan dalam mengakses kendaraan mereka, serta meningkatkan risiko kerusakan lebih lanjut akibat kesalahan dalam diagnosis [1]. Oleh karena itu, diperlukan suatu sistem otomatis yang mampu mengklasifikasikan tingkat kerusakan kunci motor secara cepat dan akurat [2].

Perkembangan teknologi kecerdasan buatan (Artificial Intelligence/AI) dan pembelajaran mesin (Machine Learning/ML) telah memberikan solusi potensial dalam mengotomatisasi berbagai tugas klasifikasi [3]. Salah satu metode yang populer dalam klasifikasi data adalah Random Forest, sebuah algoritma berbasis ensemble learning yang mampu memberikan hasil prediksi dengan akurasi tinggi dibandingkan metode manual [4]. Dengan menggunakan Random Forest, sistem dapat mengidentifikasi tingkat kerusakan kunci motor berdasarkan parameter seperti kondisi fisik kunci, tingkat kesulitan penggunaan, dan riwayat pemakaian. Sistem ini diharapkan dapat membantu teknisi dan pengguna dalam mengevaluasi kondisi kunci dengan lebih cepat, akurat, serta mengurangi subjektivitas dalam diagnosis.

Beberapa penelitian sebelumnya telah mengembangkan sistem berbasis kecerdasan buatan untuk klasifikasi dan prediksi dalam bidang otomotif maupun keamanan kendaraan. Penelitian oleh Qomariyati et al. [5] menggunakan Random Forest Classifier untuk memprediksi kelayakan kendaraan, dengan hasil menunjukkan bahwa akurasi meningkat hingga 94.57% ketika menggunakan 90-100 decision tree, serta peningkatan signifikan dari 69.53% menjadi 94.57% dengan penggunaan semua fitur. Penelitian lain oleh Nugroho [6] membandingkan teknik splitting criteria pada Decision Tree dan Random Forest untuk klasifikasi evaluasi kendaraan, dengan hasil menunjukkan bahwa model Random Forest mencapai akurasi tertinggi sebesar 96,51% menggunakan kriteria gain ratio, sedangkan Decision Tree memperoleh akurasi tertinggi 94,19% dengan kriteria information gain. Selain itu, penelitian oleh Mandenni et al. [7] mengimplementasikan metode Random Forest dan Logistic Regression untuk meningkatkan kualitas operasional layanan kendaraan, dengan hasil menunjukkan bahwa model Random Forest mencapai akurasi 91%, sedangkan Logistic Regression hanya mencapai 72%. Sistem rekomendasi yang dikembangkan dapat memberikan rekomendasi paket promosi dan tanggal penawaran yang lebih efektif untuk meningkatkan kualitas panggilan pelanggan di bengkel otomotif. Penelitian oleh Luqman Hakim et al.[8] memfokuskan pada pengoptimalan klasifikasi malware pada program Android menggunakan algoritma Random Forest yang dioptimalkan melalui GridSearchCV. Hasil penelitian menunjukkan bahwa

JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 2, April 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i2.8517
Hal 84-94
http://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

metode yang diusulkan berhasil mencapai akurasi deteksi malware sebesar 99%. Penelitian lain oleh Ahmad Hasan Mubarok et al. [9] menguji algoritma Random Forest untuk prediksi stunting, mencapai akurasi 91,65% dengan n_estimators 200 dan max_depth 30. Menggunakan data dari Survei Kesehatan Indonesia 2023.

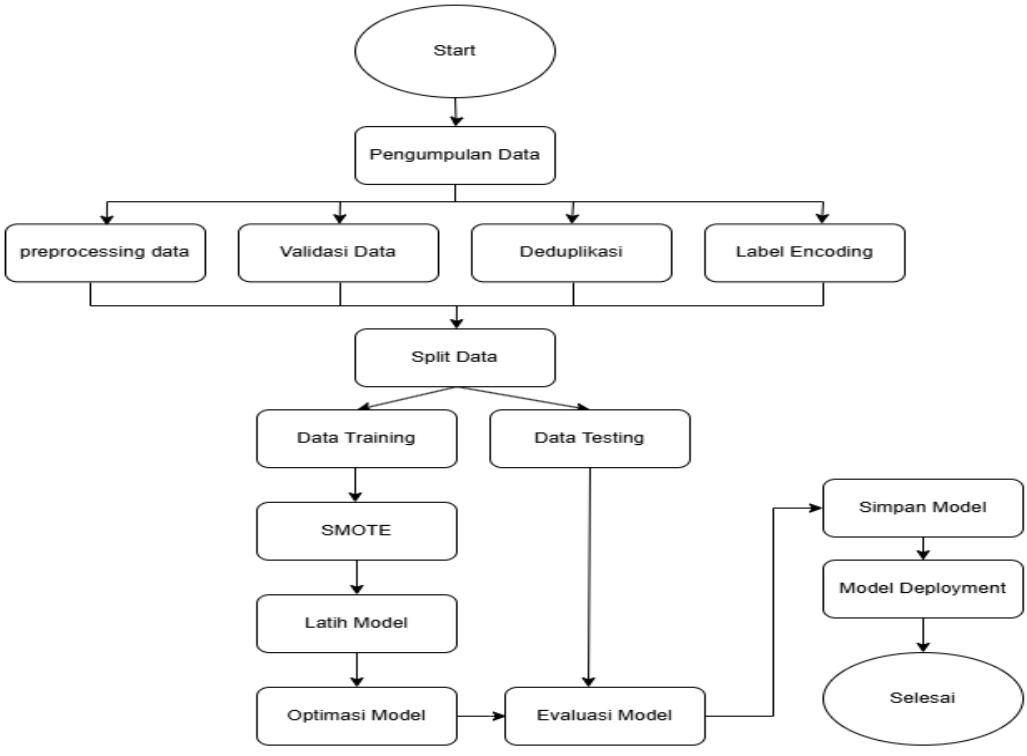
Berdasarkan penelitian terkait tersebut, terdapat kesenjangan penelitian (GAP Analysis) yang dapat diidentifikasi. Sebagian besar penelitian hanya berfokus pada analisis kerusakan sistem kendaraan secara umum, namun belum ada penelitian yang secara spesifik mengembangkan sistem klasifikasi otomatis untuk tingkat kerusakan kunci motor. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sistem klasifikasi tingkat kerusakan kunci motor menggunakan metode Random Forest guna meningkatkan efisiensi dan akurasi dalam proses identifikasi kerusakan.

Penelitian ini bertujuan untuk mengembangkan sistem klasifikasi otomatis untuk mendeteksi tingkat kerusakan kunci motor menggunakan algoritma Random Forest dengan Hyperparameter Tuning. Selain itu, penelitian ini menganalisis performa algoritma Random Forest dalam mengklasifikasikan tingkat kerusakan kunci motor dibandingkan dengan metode manual serta memberikan solusi berbasis AI yang dapat meningkatkan efisiensi waktu dan akurasi dalam proses identifikasi serta perbaikan kunci motor. Dengan adanya sistem ini, diharapkan dapat membantu teknisi dalam mendiagnosis tingkat kerusakan secara lebih objektif, mengurangi kesalahan dalam identifikasi akibat subjektivitas dalam metode manual, serta meningkatkan efektivitas perbaikan kunci motor.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Tahapan penelitian ini dimulai dengan pengumpulan data melalui observasi langsung terhadap kunci motor yang mengalami berbagai tingkat kerusakan. Selain itu, dilakukan wawancara dengan teknisi untuk memahami jenis-jenis kerusakan, penyebabnya, serta metode perbaikannya. Data yang dikumpulkan kemudian diproses dengan teknik preprocessing, seperti encoding data kategorikal, pembagian data menjadi training dan testing, serta penanganan ketidakseimbangan data menggunakan SMOTE. Setelah itu, model Random Forest dilatih untuk melakukan klasifikasi tingkat kerusakan kunci motor, diikuti dengan optimasi menggunakan GridSearchCV agar mendapatkan performa terbaik. Proses ini dapat dilihat pada Gambar 1, yang menunjukkan langkah-langkah utama dalam tahapan penelitian.



Gambar 1. Tahapan Penelitian

Setelah model selesai dilatih, evaluasi dilakukan dengan menggunakan metrik seperti akurasi, precision, recall, dan F1-score untuk mengukur kinerjanya. Model yang sudah optimal kemudian diterapkan dalam sebuah website, di mana backend menggunakan PHP dan MySQL, sedangkan frontend dikembangkan dengan HTML, CSS, dan JavaScript. Pengujian dilakukan untuk memastikan sistem berjalan dengan baik melalui functional testing, usability testing, dan security testing. Terakhir, tahap maintenance dilakukan dengan monitoring sistem, pembaruan fitur, serta mencadangkan data secara berkala agar sistem tetap optimal dan dapat digunakan dalam jangka panjang.

2.2 Pengumpulan Data

JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 2, April 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i2.8517
Hal 84-94
http://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

Penelitian ini menggunakan data yang diperoleh dari observasi langsung terhadap kunci motor yang mengalami berbagai tingkat. Subjek penelitian adalah teknisi atau tukang kunci motor yang memiliki pengalaman kurang lebih 20 tahun dalam menganalisis dan memperbaiki kunci kendaraan bermotor. Keahliannya digunakan sebagai dasar dalam validasi sistem klasifikasi yang dikembangkan.

2.3 Preprocessing Data

Preprocessing data adalah proses yang bertujuan untuk membersihkan, mengorganisir, dan menyiapkan data sebelum digunakan dalam pemodelan machine learning [10]. Langkah ini penting karena data mentah sering kali mengandung nilai yang hilang, data tidak relevan, atau distribusi kelas yang tidak seimbang, yang dapat mempengaruhi akurasi model. Berikut penjelasan mengenai preprocessing data dalam penelitian ini.

2.3.1 Validasi Data

Validasi Data adalah proses untuk memastikan bahwa data yang digunakan dalam penelitian atau sistem memiliki kualitas yang baik, akurat, dan sesuai dengan standar yang ditetapkan [11]. Data yang memiliki nilai kosong atau tidak relevan akan dihapus untuk mencegah gangguan dalam analisis dan pelatihan model.

2.3.2 Deduplikasi

Deduplikasi Data adalah proses identifikasi dan penghapusan entri data yang duplikat dalam suatu dataset untuk meningkatkan efisiensi penyimpanan dan kualitas data. Data yang memiliki kemiripan atau berulang dihapus untuk menghindari bias dalam model.

2.3.3 Label Encoding

Label Encoding adalah teknik dalam preprocessing data yang digunakan untuk mengonversi data kategorikal atau data text menjadi bentuk numerik [12]. Data akan dilakukan label encoding dikarenakan dalam proses analisis data, model hanya dapat lebih mudah memahami dan mengolah data dalam bentuk angka. Contoh label encoding dalam penelitian ini.

Tabel.1 Label Encoding

Tingkat Kerusakan	Label Encoding
Ringan	0
Sedang	1
Parah	2

Berdasarkan Tabel 1, setiap kategori dikonversi ke nilai numerik agar dapat digunakan oleh algoritma Random Forest. Hal ini bertujuan untuk memastikan data bersifat terstruktur dan dapat diproses secara efisien.

2.3.4 Split Data

Split data adalah proses membagi dataset menjadi dua atau lebih bagian untuk melatih dan menguji model machine learning. Data akan dibagi menjadi 80% untuk data training dan 20% untuk data testing, dimana data latih akan masuk ke proses SMOTE dikarenakan data yang tidak seimbang. Tujuan utama dari pembagian data (split data) adalah untuk menguji seberapa baik model dapat bekerja pada data yang belum pernah digunakan saat pelatihan [13]. Dengan cara ini, kita dapat mengetahui apakah model mampu mengenali pola baru dan memberikan hasil yang akurat ketika digunakan pada data yang berbeda.

2.3.5 SMOTE

SMOTE (Synthetic Minority Over-sampling Technique) adalah teknik dalam preprocessing data yang digunakan untuk menangani ketidakseimbangan kelas dalam dataset[14]. Data dalam penelitian ini menunjukkan lebih banyak kasus kerusakan ringan dibanding kerusakan sedang dan parah, model akan lebih sering memprediksi kerusakan ringan, meskipun ada beberapa kasus yang lebih sedang dan parah. Dilakukan oversampling untuk mengatasi ketidakseimbangan data dengan menambah jumlah kelas minoritas dengan menggunakan teknik SMOTE, jumlah sampel dalam kelas minoritas dapat ditingkatkan secara sintetis agar distribusi data lebih seimbang [15].

2.4 Random Forest

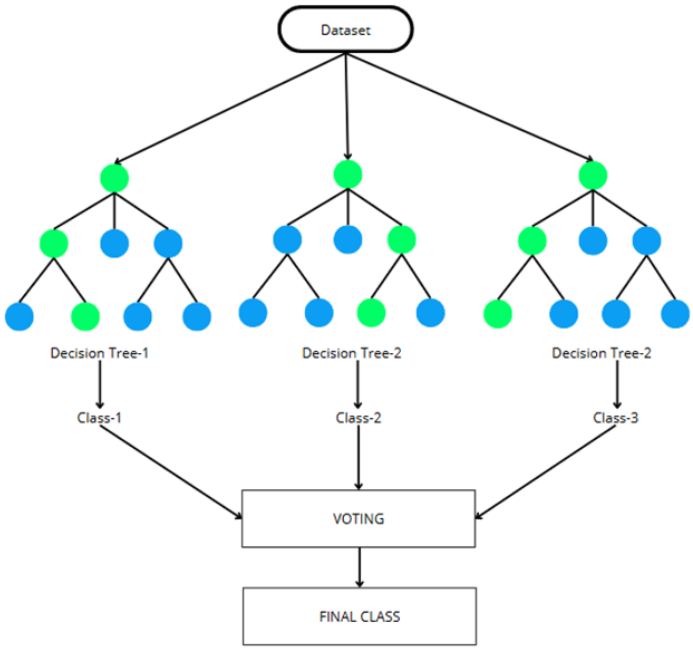
Random Forest adalah algoritma ensemble learning yang terdiri dari kumpulan decision trees yang bekerja secara kolektif untuk meningkatkan akurasi dalam klasifikasi maupun regresi. Dengan menggunakan pendekatan bagging (Bootstrap Aggregating), Random Forest membangun banyak pohon keputusan dan mengombinasikan hasilnya melalui mekanisme voting mayoritas dalam klasifikasi [16].

Proses kerja Random Forest dimulai dengan Bootstrap Sampling, di mana dataset awal diambil secara acak dengan pengembalian (sampling with replacement) untuk membentuk subset data bagi setiap pohon keputusan. Selanjutnya, dilakukan pembangunan pohon keputusan, di mana setiap pohon dibuat menggunakan metode CART (Classification and Regression Trees) dengan memilih subset fitur secara acak untuk membagi data hingga mencapai kedalaman tertentu atau data tidak dapat dibagi lebih lanjut. Setelah semua pohon terbentuk, hasil akhirnya ditentukan dengan voting mayoritas

JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 2, April 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i2.8517
Hal 84-94
http://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

atau Rata-rata, di mana dalam klasifikasi, setiap pohon memberikan prediksi dan hasil akhir ditentukan berdasarkan kelas dengan suara terbanyak [17].

Untuk memberikan pemahaman lebih jelas mengenai cara kerja algoritma Random Forest, Gambar 2 menampilkan arsitektur dasar Random Forest dan alur prediksi yang dilakukan.



Gambar 2. Random Forest [18]

Gambar 2 menunjukkan proses kerja algoritma Random Forest secara visual. Penjelasan detail dari gambar tersebut disampaikan pada uraian berikut:

a. Dalam proses pemisahan node pada pohon keputusan, digunakan Gini Impurity untuk mengukur seberapa "murni" atau homogen suatu node dalam decision tree. Rumusnya adalah:

$$Gini(D) = 1 - \sum_{i=1}^{n} p_i^2 \quad (1)$$

Di mana D adalah dataset dalam node tertentu, n adalah jumlah kelas dalam dataset, dan p_i^2 adalah probabilitas suatu data termasuk dalam kelas ke-i.

b. Information Gain digunakan untuk memilih fitur terbaik dalam pemisahan data. Semakin besar IG, semakin baik fitur tersebut dalam membagi data. Rumusnya adalah:

$$IG = \text{Entropy}(Parent) - \sum_{j=1}^{m} \frac{|D_j|}{|D|} \text{Entropy}(D_j) \quad (2)$$

Di mana Entropy(*Parent*) adalah entropy sebelum pemisahan, D*j* adalah subset data setelah pemisahan berdasarkan fitur tertentu, $\frac{|D_j|}{|D|}$ adalah proporsi jumlah data dalam subset dibandingkan dengan total data.

c. Setelah semua Decision Tree selesai dibuat dalam Random Forest, setiap pohon memberikan prediksi terhadap data baru. Hasil akhirnya ditentukan menggunakan voting mayoritas. Rumus voting mayoritas:

$$\hat{y} = \arg\max_{k} \sum_{j=1}^{m} 1(h_j(x) = k) \quad (3)$$

Di mana $\hat{y}$ adalah kelas yang dipilih sebagai hasil akhir, $h_j(x)$ adalah prediksi dari pohon ke-j, k adalah kelas yang mungkin ada dalam data, m adalah jumlah total pohon dalam hutan.

2.5 Optimasi Model

Optimasi model Random Forest bertujuan untuk meningkatkan kinerja model dalam melakukan klasifikasi atau regresi dengan cara menyesuaikan hyperparameter yang mempengaruhi proses pembelajaran [19]. Agar model dapat bekerja secara optimal, di penelitian ini dilakukan optimasi hyperparameter menggunakan GridSearchCV. GridSearchCV merupakan metode pencarian grid yang mencoba berbagai kombinasi hyperparameter untuk menemukan konfigurasi terbaik yang memberikan akurasi tertinggi [20]. Pada model Random Forest, penggunaan nilai default berarti model akan bekerja dengan pengaturan bawaan yang telah ditentukan oleh Scikit-Learn. Sementara itu, optimasi hyperparameter bertujuan untuk menemukan kombinasi parameter terbaik yang dapat meningkatkan performa model.

2.6 Evaluasi Model

JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 2, April 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i2.8517
Hal 84-94
http://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

Setelah model Random Forest dilatih dan dioptimasi menggunakan HyperParameter GridSearchCV, langkah berikutnya adalah evaluasi model untuk mengukur performa dan keakuratan dalam mengklasifikasikan tingkat kerusakan kunci motor. Evaluasi dilakukan menggunakan data uji (20%) yang tidak digunakan selama pelatihan untuk memastikan model memiliki generalisasi yang baik dan tidak hanya bekerja dengan baik pada data latih.
Evaluasi untuk mengukur kinerja model, digunakan beberapa metrik evaluasi seperti akurasi, precision, recall, dan F1-Score.

2.7 Simpan Model

Model yang sudah berhasil dilatih dan dievaluasi akan disimpan agar dapat digunakan kembali di masa mendatang tanpa perlu melatih ulang. Penyimpanan model dilakukan untuk memudahkan proses deployment atau integrasi model ke dalam aplikasi yang akan digunakan untuk prediksi otomatis.

2.8 Model Deployment

Setelah model Random Forest berhasil dilatih, dioptimasi, dievaluasi, dan disimpan, langkah selanjutnya adalah deployment. Deployment adalah proses mengintegrasikan model ke dalam aplikasi agar dapat digunakan untuk melakukan prediksi secara langsung oleh pengguna.

3. HASIL DAN PEMBAHASAN

3.1 Hasil Pengumpulan Data

Pengambilan data dalam penelitian ini dilakukan selama periode Januari 2023 hingga Mei 2024. Pada penelitian ini, dataset yang digunakan berasal dari hasil observasi terhadap kunci motor yang mengalami berbagai tingkat kerusakan. Dataset terdiri dari 1000 data sampel dengan 12 variabel utama. Untuk memberikan pemahaman lebih lengkap terhadap struktur data yang digunakan dalam penelitian ini, berikut disajikan daftar lengkap variabel yang dimasukkan dalam dataset klasifikasi kerusakan kunci motor.

Tabel 2. Variabel Penelitian

Variabel	Kode	Keterangan
Tipe Sepeda Motor	VB1	Jenis sepeda motor yang mengalami kerusakan
Bagian Kunci	VB2	Bagian dari sistem kunci yang mengalami kerusakan seperti kunci utama, kunci jok atau kunci tangki.
Masalah Kunci	VB3	Jenis kerusakan yang terjadi pada kunci, apakah mancet, susah diputar, atau kendala lainnya.
Kondisi Anak Kunci	VB4	Apakah anak kunci mengalami aus, bengkok, atau masih bagus.
Kesulitan Memutar	VB5	Seberapa sulit kunci diputar.
Kondisi Lubang Kunci	VB6	Apakah lubang kunci mengalami karat, aus, atau kotor.
Frekuensi Masalah	VB7	Seberapa sering kunci mengalami masalah kerusakan.
Riwayat Servis	VB8	Apakah kunci pernah diperbaiki sebelumnya.
Riwayat Pengunaan	VB9	Seberapa sering sepeda motor digunakan.
Tingkat Kerusakan	-	Variabel yang akan menilai tingkat kerusakan kunci dari ringan, sedang, dan parah.
Solusi	VB10	Jalan keluar yang bisa dilakukan
Estimasi Biaya	VB11	Estimasi biaya yang mungkin akan dikeluarkan

Dari tabel 2, dapat diketahui bahwa dataset terdiri dari beragam variabel yang merepresentasikan kondisi fisik, riwayat, serta tingkat dari kerusakan kunci. Informasi ini digunakan sebagai fitur untuk proses klasifikasi tingkat kerusakan.

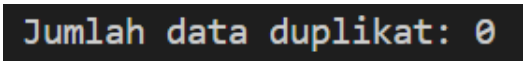
3.2 Preprocessing Data

3.3.1 Validasi Data

Validasi data dilakukan secara manual ketika penyusunan ulang data. Tidak ditemukan data yang hilang ataupun kosong.

3.2.2 Deduplikasi

Dalam tahap preprocessing, dilakukan deduplikasi data untuk memastikan bahwa tidak ada entri ganda yang dapat mempengaruhi hasil pelatihan model. Proses ini dilakukan menggunakan pustaka Python pandas. Hasil dari pengecekan data duplikat ditampilkan pada Gambar 3.



Gambar 3. Hasil Cek Data Duplikat

JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 2, April 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i2.8517
Hal 84-94
http://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

Dari Gambar 3 terlihat bahwa data tidak mengandung entri duplikat.

3.2.3 Label Encoding

Data kategorikal dalam dataset diubah ke bentuk numerik agar dapat diproses oleh model pembelajaran mesin. Proses ini dilakukan menggunakan Label Encoding. Gambar 4 menunjukkan cuplikan kode program untuk melakukan encoding menggunakan pustaka sklearn.preprocessing.

```
label_encoders = {}
for column in df.select_dtypes(include=['object']).columns:
    le = LabelEncoder()
    df[column] = le.fit_transform(df[column])
    label_encoders[column] = le
```

Gambar 4. Kode Label Encoding

Dari hasil kode di Gambar 4, data kategorikal telah berhasil dikonversi ke dalam format numerik, sehingga lebih mudah diolah oleh model machine learning. Seperti yang ditunjukkan dalam Tabel 4, setiap nilai unik dalam data kategorikal mendapatkan representasi angka tertentu. Tabel 3 dan Tabel 4 berikut memperlihatkan perbandingan data sebelum dan sesudah dilakukan proses *Label Encoding*.

Tabel 3. Sebelum Label Encoding

No	VB1	VB2	VB3	VB4	…	Tingkat kerusakan	VB10	VB11
1	Mio	Kunci Setang	Kunci setang tidak berfungsi	Aus(parah)	…	Ringan	Pembersihan dan pelumasan	25000
2	Supra X125	Kunci Setang	Kunci setang tidak berfungsi	Bengkok	…	Parah	Pembersihan dan pelumasan	30000
3	Vario	Kunci Magnet	Pengaman magnet tidak bisa dibuka	Bengkok	…	Ringan	Bawa ke ahli kunci	25000
…	…	…	….	…	…	…	…	…
1000	Supra X125	Kunci Setang	Kunci seret saat diputar	Bagus	…	Ringan	Pembersihan dan pelumasan	25000

Tabel 4. Sesudah Label Encoding

No	VB1	VB2	VB3	VB4	…	Tingkat kerusakan	VB10	VB11
1	16	3	7	0	…	0	0	25000
2	29	3	7	3	…	2	0	10000
3	32	2	10	3	…	0	1	25000
…	…	…	….	…	…	…	…	…
1000	29	3	3	2	…	0	0	25000

Perbandingan Tabel 3 dan Tabel 4 memperlihatkan bahwa data telah dikodekan dengan Label Encoding, yang memungkinkan sistem untuk memprosesnya dengan model Random Forest. Misalnya, nama motor seperti "Mio" dan "Supra X125" telah dikonversi menjadi angka 16 dan 29 secara berurutan.

3.2.4 Split Data

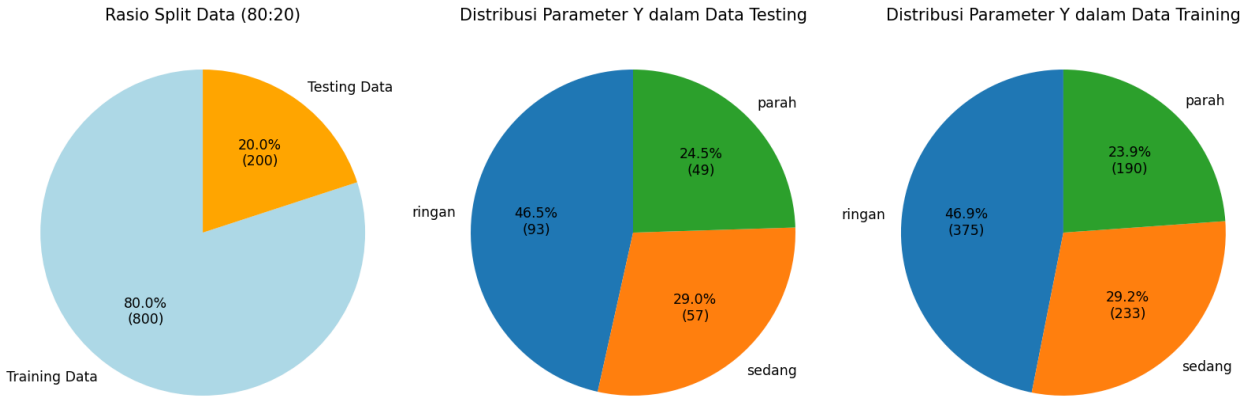
Dataset dibagi menjadi data latih dan data uji menggunakan fungsi train_test_split dari Scikit-Learn. Pembagian dilakukan dengan proporsi 80% data latih dan 20% data uji. Penggunaan parameter random_state=42 memastikan pembagian data yang konsisten pada setiap kali program dijalankan. Gambar 5 memperlihatkan kode Python yang digunakan untuk melakukan proses split data.

```
# Split data (80% training, 20% testing)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Gambar 5. Kode Split Data

Tujuan dari code pada gambar 5 adalah untuk memastikan bahwa data yang digunakan untuk melatih model tidak sama dengan data yang digunakan untuk mengujinya, sehingga kinerja model dapat dinilai secara objektif. Hasil distribusi kelas setelah pembagian tersebut dapat dilihat lebih lanjut pada Gambar 6.

JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 2, April 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i2.8517
Hal 84-94
http://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

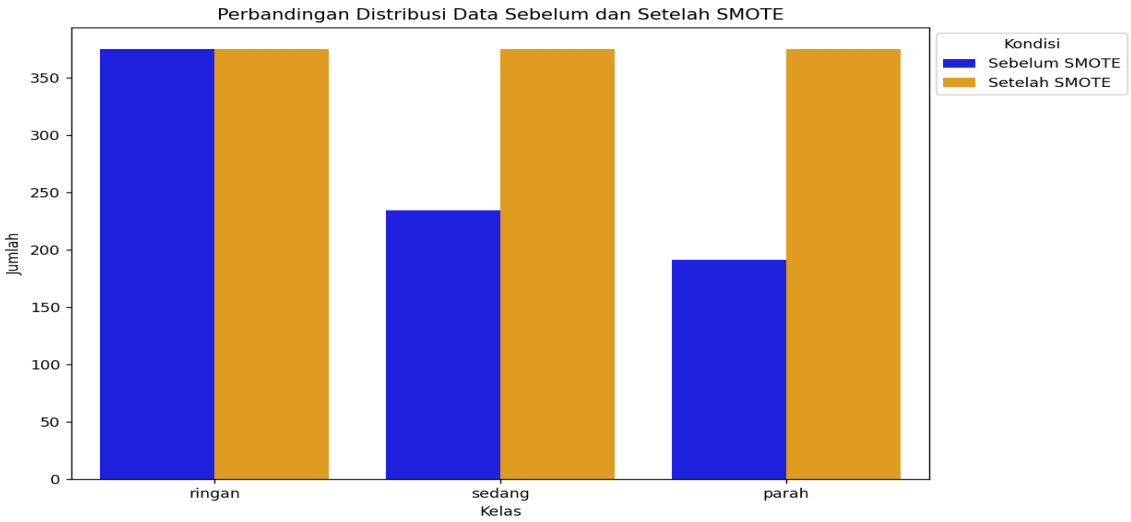


Gambar 6. Split Data

Gambar 6 memperlihatkan visualisasi dari hasil pembagian data tersebut, termasuk jumlah total data yang masuk ke dalam set pelatihan (training set) dan set pengujian (testing set). Selain itu, gambar ini juga menunjukkan distribusi masing-masing parameter atau fitur dalam kedua set tersebut, untuk memastikan bahwa tidak terjadi ketidakseimbangan atau bias dalam representasi data antar set. Hal ini penting agar model tidak hanya belajar dari pola yang dominan di satu set saja, tetapi memiliki pemahaman yang seimbang terhadap keseluruhan data.

3.2.5 SMOTE

Bisa dilihat pada Gambar 6 bahwa setelah dilakukan split data, variabel Y tidak seimbang. Sebanyak lebih dari 46% atau sekitar 375 data menunjukkan kategori kerusakan ringan. Oleh karena itu, dilakukan proses SMOTE untuk menyeimbangkan distribusi kelas sebelum model dilatih. Gambar 7 menunjukkan distribusi data sebelum dan sesudah dilakukan SMOTE.



Gambar 7. Hasil SMOTE

Gambar 7 memperlihatkan bahwa jumlah sampel untuk kelas minoritas berhasil ditambah secara sintetis sehingga distribusi kelas menjadi seimbang, yang penting untuk menghindari bias dalam pelatihan model.

3.3 Random Forest

Pada tahap awal, model Random Forest diterapkan dengan hyperparameter default untuk melihat baseline performa sebelum dilakukan optimasi. Untuk mengetahui kinerja awal dari model sebelum dilakukan tuning, model Random Forest diuji dengan pengaturan default seperti ditunjukkan dalam Tabel 5.

Tabel 5. Default Random Forest

Hyperparameter	Nilai Default
n_estimators (Jumlah pohon)	100
max_depth (Kedalaman maksimum pohon)	None
max_features (Jumlah fitur yang digunakan setiap split)	'sqrt'
min_samples_split (Minimal sampel untuk membagi node)	2

JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 2, April 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i2.8517
Hal 84-94
http://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

min_samples_leaf (Minimal sampel dalam leaf node)	1
criterion (Metode pemisahan node)	'gini'

Berdasarkan Tabel 5, model awal menggunakan default. Ini berguna sebagai baseline untuk mengevaluasi seberapa banyak peningkatan performa yang diberikan oleh optimasi hyperparameter.

3.4 Optimasi Model

Untuk meningkatkan performa klasifikasi, dilakukan tuning hyperparameter menggunakan GridSearchCV. Rentang nilai yang diuji disajikan dalam Tabel 6.

Tabel 6. GridSearchCV

Hyperparameter	Nilai yang dicoba
n_estimators (Jumlah pohon)	100, 200, 300, 400, 500
max_depth (Kedalaman maksimum pohon)	5, 10, 15, 20, 25, 30
max_features (Jumlah fitur yang digunakan setiap split)	‘auto’,'sqrt', 'log2'
min_samples_split (Minimal sampel untuk membagi node)	2, 5, 10
min_samples_leaf (Minimal sampel dalam leaf node)	1, 2, 4
criterion (Metode pemisahan node)	'gini', 'entropy'

Seperti terlihat pada Tabel 6, GridSearchCV mengevaluasi berbagai kombinasi parameter yang memungkinkan untuk mendapatkan konfigurasi model Random Forest yang optimal. Proses tuning ini bertujuan untuk meningkatkan performa model, yang kemudian akan dievaluasi secara kuantitatif pada tahap berikutnya menggunakan metrik seperti akurasi, precision, recall, dan F1-score.

3.5 Evaluasi Model

Sebelum dilakukan tuning, model Random Forest diuji dengan pengaturan default. Evaluasi awal ini memberikan baseline performa yang digunakan sebagai pembanding setelah optimasi. Hasil evaluasi model default disajikan pada Gambar 8.

```
Akurasi Model Random Forest: 0.82

Laporan Klasifikasi:
              precision    recall  f1-score   support

       parah       0.83      0.82      0.82        49
      ringan       0.83      0.77      0.80        93
      sedang       0.80      0.90      0.85        58

    accuracy                           0.82       200
   macro avg       0.82      0.83      0.82       200
weighted avg       0.82      0.82      0.82       200
```

Gambar 8. Hasil Model Default

Dari hasil model default pada gambar 8, akurasi model tercatat sebesar 82%. Namun, nilai recall untuk kelas parah masih rendah, menunjukkan bahwa model kurang andal dalam mengidentifikasi kerusakan berat. Setelah dilakukan tuning hyperparameter menggunakan GridSearchCV, performa model meningkat secara signifikan. Hasil evaluasi terhadap model yang telah dioptimasi disajikan pada Gambar 9.

```
Akurasi Model Random Forest: 0.855

Laporan Klasifikasi:
              precision    recall  f1-score   support

       parah       0.84      0.94      0.88        49
      ringan       0.90      0.77      0.83        93
      sedang       0.82      0.91      0.86        58

    accuracy                           0.85       200
   macro avg       0.85      0.88      0.86       200
weighted avg       0.86      0.85      0.85       200
```

Gambar 9. Hasil Model Dengan Hyperparameter GridSearchCV

JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 2, April 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i2.8517
Hal 84-94
http://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

Dapat disimpulkan bahwa perbandingan evaluasi sebelum dan sesudah menggunakan Hyperparameter GridSearchCV adalah sebagai berikut:

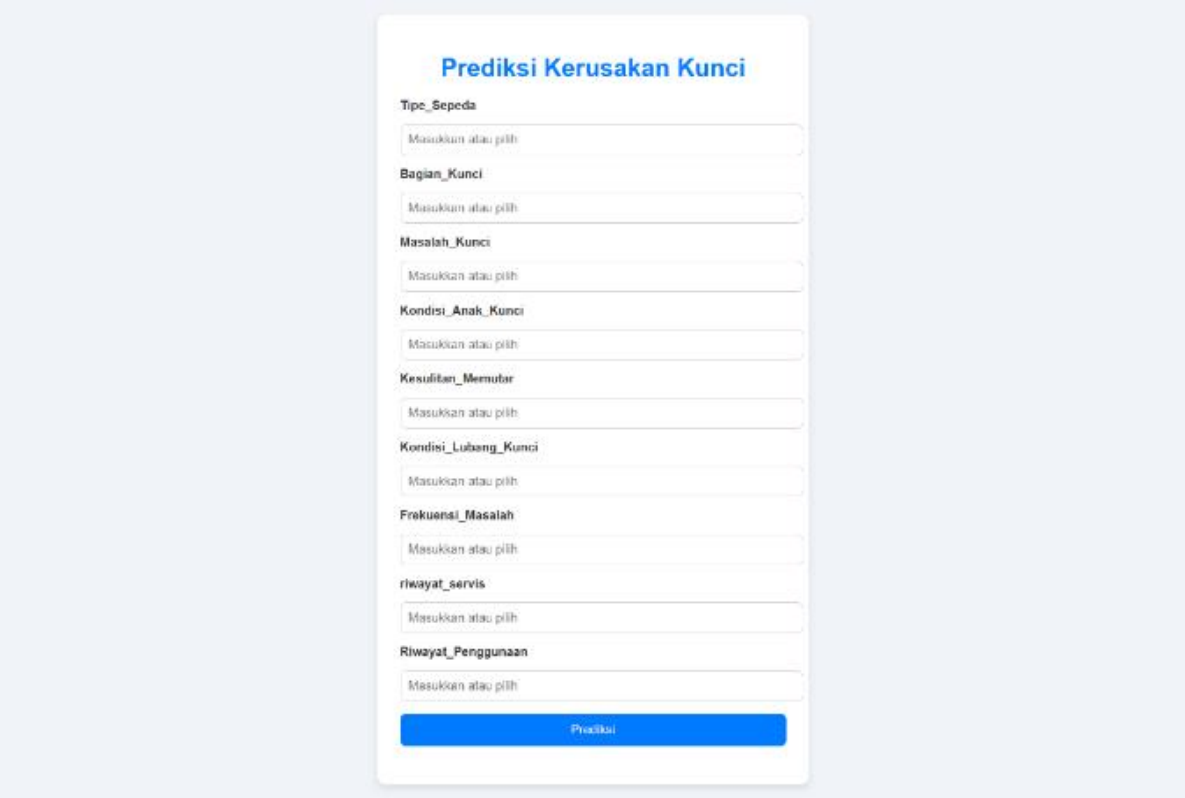
a. Akurasi : dapat dilihat bahwa terjadi peningkatan akurasi dari 82% menjadi 85.5%. Hal ini menunjukkan bahwa penyesuaian hyperparameter melalui GridSearchCV berhasil meningkatkan kinerja model dalam melakukan klasifikasi.
b. Recall : model yang telah dioptimasi jauh lebih baik dalam menangkap kasus kelas parah, meningkat dari 0.82 menjadi 0.94. Recall untuk kelas ringan tetap di angka 0.77, sedangkan kelas sedang meningkat dari 0.90 menjadi 0.91. Dari hasil ini, dapat disimpulkan bahwa model setelah tuning hyperparameter lebih baik dalam mengenali kasus parah, tetapi masih memiliki performa yang sama pada kelas ringan.
c. F1-Score : kelas parah mengalami peningkatan dari 0.82 menjadi 0.88, kelas ringan tetap di angka 0.83, dan kelas sedang meningkat dari 0.85 menjadi 0.86. Ini menunjukkan bahwa keseimbangan antara precision dan recall pada kelas parah dan sedang mengalami peningkatan, sementara kelas ringan masih tetap stabil. Secara keseluruhan, baik macro average maupun weighted average mengalami peningkatan dari 0.82 menjadi 0.85, yang menandakan bahwa model bekerja lebih baik dalam mengklasifikasikan semua kelas setelah dilakukan optimasi hyperparameter
d. Precision : model yang telah dioptimasi mengalami peningkatan terutama pada kelas ringan, yang sebelumnya memiliki precision sebesar 0.83 dan meningkat menjadi 0.90. Precision pada kelas parah juga mengalami sedikit peningkatan dari 0.83 menjadi 0.84, sementara kelas sedang meningkat dari 0.80 menjadi 0.82. Hal ini menunjukkan bahwa model yang telah dioptimasi lebih baik dalam mengidentifikasi kelas ringan, meskipun kelas lainnya juga mengalami perbaikan.

3.6 Simpan Model

Hasil akan disimpan dalam format .pkl untuk selanjutnya diterapkan dalam website.

3.7 Model Deployment

Setelah melalui serangkaian tahapan mulai dari pengolahan data, pelatihan model, evaluasi, hingga penyimpanan model, langkah selanjutnya adalah melakukan deployment agar model dapat digunakan secara praktis oleh pengguna. Dalam tahap ini, model yang telah dikembangkan diintegrasikan ke dalam sebuah website berbasis web yang menyediakan antarmuka pengguna (UI) untuk melakukan prediksi dengan mudah. Antarmuka pengguna (UI) dari sistem ini ditampilkan pada Gambar 10.



Gambar 10. UI Website

Gambar 10 UI sistem menampilkan formulir input yang memudahkan pengguna mengisi data karakteristik kerusakan kunci motor, yang kemudian akan diproses oleh model untuk menghasilkan klasifikasi tingkat kerusakan. Setelah pengguna mengisi data pada UI input, sistem akan menampilkan hasil klasifikasi kerusakan berdasarkan input tersebut. Gambar 11 memperlihatkan hasil prediksi yang diberikan oleh model.

JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 2, April 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i2.8517
Hal 84-94
http://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

Gambar 11. UI Hasil Prediksi

Gambar 11 UI hasil prediksi disajikan dengan jelas, lengkap dengan informasi kategori kerusakan, saran solusi yang dapat diambil oleh pengguna, serta estimasi biaya.

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, dapat disimpulkan bahwa model Random Forest yang dikembangkan mampu mengklasifikasikan tingkat kerusakan kunci motor dengan tingkat akurasi yang cukup tinggi. Model awal dengan parameter default memiliki akurasi sebesar 81.5%, sedangkan setelah dilakukan optimasi hyperparameter menggunakan GridSearchCV, akurasi meningkat menjadi 85.5%. Peningkatan ini menunjukkan bahwa optimasi hyperparameter berkontribusi dalam meningkatkan performa model dalam melakukan klasifikasi. Evaluasi model menggunakan precision, recall, dan F1-score menunjukkan bahwa model yang telah dioptimasi memiliki kemampuan yang lebih baik dalam menangkap pola data, khususnya pada kategori kerusakan parah yang sebelumnya kurang terdeteksi dengan baik. Hasil dari confusion matrix juga menunjukkan bahwa kesalahan klasifikasi berkurang setelah optimasi dilakukan. Selain itu, penerapan teknik SMOTE dalam proses pelatihan model terbukti efektif dalam menangani ketidakseimbangan kelas pada dataset. Dengan adanya teknik ini, model dapat memberikan prediksi yang lebih akurat untuk kelas minoritas tanpa mengorbankan performa pada kelas mayoritas.

UCAPAN TERIMAKASIH

Penulis mengucapkan terima kasih kepada Universitas Nahdlatul Ulama Sunan Giri, Bojonegoro, atas dukungan fasilitas dan sumber daya dalam pelaksanaan penelitian ini. Ucapan terima kasih juga disampaikan kepada teknisi kunci motor yang telah memberikan wawasan dan data yang berharga dalam pengumpulan dataset. Penulis menghargai bimbingan dari dosen pembimbing serta rekan-rekan peneliti yang telah memberikan masukan konstruktif dalam penyempurnaan penelitian ini.

REFERENCES

[1] A. Kurniawan, G. Atmaja, M. Nawawi, O. Hidayat, A. Latif, and R. E. Syahputra, “Sistem Pakar Diagnosa Kerusakan Mesin Sepeda Motor Dengan Menggunakan Metode Forward Chaining,” Teknik dan Multimedia, vol. 1, no. 2, 2023, [Online]. Available: https://scholar.google.com.

[2] R. Mega Yulianto and R. Gunawan, “Dirgamaya Jurnal Manajemen dan Sistem Informasi Sistem Pakar Mendeteksi Kerusakan Komponen Kelistrikan Sepeda Motor Matic Injeksi Menggunakan Fuzzy Sugeno.”

[3] R. Rakhmat Sani, Y. Ayu Pratiwi, S. Winarno, E. Devi Udayanti, and dan Farrikh Al Zami, “Analisis Perbandingan Algoritma Naive Bayes Classifier dan Support Vector Machine untuk Klasifikasi Hoax pada Berita Online Indonesia,” 2022.

[4] A. Nugroho, “Analisa Splitting Criteria Pada Decision Tree dan Random Forest untuk Klasifikasi Evaluasi Kendaraan,” JSITIK: Jurnal Sistem Informasi dan Teknologi Informasi Komputer, vol. 1, no. 1, pp. 41–49, Dec. 2022, doi: 10.53624/jsitik.v1i1.154.

[5] L. N. Qomariyati, S. Nurpadillah, N. F. Rosyidin, F. Sofyan, and L. R. Mubarak, “Jurnal FUSE-Teknik Elektro |Vol. 4 | No. 1 | Halaman 21-30,” 2024.

[6] A. Nugroho, “Analisa Splitting Criteria Pada Decision Tree dan Random Forest untuk Klasifikasi Evaluasi Kendaraan,” JSITIK: Jurnal Sistem Informasi dan Teknologi Informasi Komputer, vol. 1, no. 1, pp. 41–49, Dec. 2022, doi: 10.53624/jsitik.v1i1.154.

[7] N. Made Ika Marini Mandenni et al., “IMPLEMENTASI MACHINE LEARNING UNTUK MENINGKATKAN KUALITAS OPERASIONAL SERVICE KENDARAAN DENGAN METODE RANDOM FOREST DAN LOGISTIC REGRESSION IMPLEMENTATION OF MACHINE LEARNING TO IMPROVE THE QUALITY OF VEHICLE SERVICE OPERATIONS WITH RANDOM FOREST AND LOGISTIC REGRESSION METHODS,” Jurnal Praksis dan Dedikasi (JPDS) Oktober, vol. 7, no. 2, pp. 206–213, 2024, doi: 10.17977/um022v7i2p206-213.

[8] L. Hakim, Z. Sari, A. Rizaldy Aristyo, and S. Pangestu, “Optimzing Android Program Malware Classification Using GridSearchCV Optimized Random Forest,” Computer Network, Computing, Electronics, and Control Journal, vol. 9, no. 2, pp. 173–180, 2024.

[9] A. H. Mubarok, P. Pujiono, D. Setiawan, D. F. Wicaksono, and E. Rimawati, “Parameter Testing on Random Forest Algorithm for Stunting Prediction,” sinkron, vol. 9, no. 1, pp. 107–116, Jan. 2025, doi: 10.33395/sinkron.v9i1.14264.

[10] A. P. Joshi and B. V. Patel, "Data Preprocessing: The Techniques for Preparing Clean and Quality Data for Data Analytics Process," Oriental journal of computer science and technology, vol. 13, no. 0203, pp. 78–81, Jan. 2021, doi: 10.13005/ojcst13.0203.03.

[11] M. P. J. van der Loo and E. de Jonge, "Data Validation," in Wiley StatsRef: Statistics Reference Online, Wiley, 2020, pp. 1–7. doi: 10.1002/9781118445112.stat08255.

[12] C. Herdian, A. Kamila, and I. G. Agung Musa Budidarma, "Studi Kasus Feature Engineering Untuk Data Teks: Perbandingan Label Encoding dan One-Hot Encoding Pada Metode Linear Regresi," Technologia : Jurnal Ilmiah, vol. 15, no. 1, p. 93, Jan. 2024, doi: 10.31602/tji.v15i1.13457.

[13] Y. A. Prasetyo, E. Utami, and A. Yaqin, "Pengaruh Komposisi Split Data Terhadap Performa Akurasi Analisis Sentimen Algoritma Naïve Bayes dan SVM," Journal homepage: Journal of Electrical Engineering and Computer (JEECOM), vol. 6, no. 2, 2024, doi: 10.33650/jeecom.v4i2.

[14] A. Syukron, E. Saputro, and P. Widodo, "Penerapan Metode Smote Untuk Mengatasi Ketidakseimbangan Kelas Pada Prediksi Gagal Jantung," 2023. [Online]. Available: https://doi.org/10/25047/jtit.v10i1.312

[15] I. Dataset, P. Kebangkrutan, P. Wilda, I. Sabilla, and C. B. Vista, "Jurnal Politeknik Caltex Riau," 2021. [Online]. Available: https://jurnal.pcr.ac.id/index.php/jkt/

[16] D. P. Sinambela, H. Naparin, M. Zulfadhilah, and N. Hidayah, "Implementasi Algoritma Decision Tree dan Random Forest dalam Prediksi Perdarahan Pascasalin," Jurnal Informasi dan Teknologi, vol. 5, no. 3, pp. 58–64, Sep. 2023, doi: 10.60083/jidt.v5i3.393.

[17] H. Tantyoko, D. Kartika Sari, and A. R. Wijaya, "PREDIKSI POTENSIAL GEMPA BUMI INDONESIA MENGGUNAKAN METODE RANDOM FOREST DAN FEATURE SELECTION," 2023. [Online]. Available: http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/indexHenriTantyoko|http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/index|

[18] N. Wuryani, S. Agustiani, I. Komputer, and N. Mandiri, "Random Forest Classifier untuk Deteksi Penderita COVID-19 berbasis Citra CT Scan," Jurnal Teknik Komputer AMIK BSI, vol. 7, no. 2, 2021, doi: 10.31294/jtk.v4i2.

[19] U. Sunarya and T. Haryanti, "Perbandingan Kinerja Algoritma Optimasi pada Metode Random Forest untuk Deteksi Kegagalan Jantung," Jurnal Rekayasa Elektrika, vol. 18, no. 4, Dec. 2022, doi: 10.17529/jre.v18i4.26981.

[20] L. Hakim, Z. Sari, A. Rizaldy Aristyo, and S. Pangestu, "Optimzing Android Program Malware Classification Using GridSearchCV Optimized Random Forest," Computer Network, Computing, Electronics, and Control Journal, vol. 9, no. 2, pp. 173–180, 2024.