

Implementasi Deep Learning Dengan Feature Selection Untuk Prediksi Cacat Perangkat Lunak Pada Dataset Nasa Metrics Data Program

Dwi Agung Septian¹, Bobi Agustian²

^{1,2} Program Studi Teknik Informatika, Universitas Pamulang
e-mail: ¹dwiagungseptian01@gmail.com, ²dosen00679@unpam.ac.id

Abstrak

Prediksi cacat perangkat lunak sangat penting untuk menjaga kualitas sistem, tetapi sering kali menghadapi masalah dimensi atribut yang tidak relevan dan ketidakseimbangan kelas data yang ekstrem. Penelitian ini bertujuan menganalisis pengaruh Feature Selection menggunakan Information Gain dan Synthetic Minority Over-sampling Technique (SMOTE) untuk mengatasi ketidakseimbangan data terhadap performansi model Multi Layer Perceptron (MLP). Dataset yang digunakan berasal dari NASA Metrics Data Program (NASA MDP) yang menggabungkan enam dataset dengan total 28.258 data. Tahap preprocessing data meliputi standardisasi label, median imputation untuk data hilang di bawah 30 persen, eliminasi atribut dengan data hilang di atas 30 persen, standardisasi StandardScaler, dan SMOTE. Feature Selection Information Gain digunakan untuk memilih 10 atribut terbaik. Model MLP dilatih dan dievaluasi menggunakan Stratified 10-Fold Cross Validation. Hasil penelitian menunjukkan bahwa penggunaan SMOTE secara signifikan meningkatkan kemampuan model mendeteksi kelas cacat (Defective), ditandai dengan kenaikan recall dari 0,1897 menjadi 0,7112. Skenario terbaik diperoleh pada kombinasi MLP dengan seleksi fitur dan SMOTE yang menghasilkan akurasi sebesar 0,7549, presisi 0,2368, recall mencapai nilai tertinggi sebesar 0,8267, F1-Score 0,3681, dan Area Under Curve (AUC) sebesar 0,8670. Kombinasi ini berhasil meminimalkan risiko kegagalan sistem akibat cacat yang tidak terdeteksi (false negative). Penelitian ini berkontribusi memberikan panduan metodologis yang efektif bagi pengembang perangkat lunak untuk mengidentifikasi potensi cacat sejak tahap awal pengembangan guna meningkatkan efisiensi pengujian.

Kata Kunci: Prediksi Cacat Perangkat Lunak, NASA Metrics Data Program, Multi Layer Perceptron, Information Gain, SMOTE.

Abstract

Software defect prediction is crucial for maintaining system quality, but often faces issues related to irrelevant attribute dimensions and extreme data class imbalance. This study aimed to analyze the effect of feature selection using Information Gain and the Synthetic Minority Over-sampling Technique (SMOTE) to handle class imbalance on the performance of the Multi

Layer Perceptron (MLP) model. The dataset was sourced from the NASA Metrics Data Program (NASA MDP), which merged six datasets with a total of 28,258 records. Data preprocessing stages included label standardization, median imputation for missing values under 30 percent, elimination of attributes with missing values above 30 percent, data standardization using StandardScaler, and SMOTE. Information Gain feature selection was utilized to select the 10 best attributes. The MLP model was trained and evaluated using Stratified 10-Fold Cross Validation. The results show that the implementation of SMOTE significantly improves the model's ability to detect defective classes, as indicated by a recall increase from 0.1897 to 0.7112. The best scenario is obtained from the combination of MLP with feature selection and SMOTE, which yields an accuracy of 0.7549, precision of 0.2368, a maximum recall of 0.8267, an F1-Score of 0.3681, and an Area Under Curve (AUC) of 0.8670. This combination successfully minimizes system failure risks caused by undetected defects (false negatives). This study contributes a highly effective methodological guideline for software developers to identify potential defects at early development stages, thereby enhancing testing efficiency.

Keywords : Software Defect Prediction, NASA Metrics Data Program, Multi Layer Perceptron, Information Gain, SMOTE.

LATAR BELAKANG

Pesatnya perkembangan teknologi informasi mendorong penggunaan perangkat lunak (software) di berbagai sektor vital. Kualitas perangkat lunak menjadi aspek yang sangat krusial karena adanya cacat perangkat lunak (software defect) dapat mengakibatkan kesalahan fungsi sistem, menurunkan kinerja aplikasi, hingga memicu kegagalan sistem yang merugikan pengguna (Pachouly et al., 2022). Sebagai upaya preventif untuk meminimalkan risiko kegagalan operasional tersebut, pendekatan prediksi cacat perangkat lunak atau Software Defect Prediction (SDP) diimplementasikan sejak tahap awal siklus pengembangan sistem. SDP bertujuan untuk mengidentifikasi potensi cacat pada modul perangkat lunak dengan memanfaatkan metrik kualitas perangkat lunak secara historis (Khalid et al., 2023). Melalui identifikasi dini ini, alokasi sumber daya pengujian dapat difokuskan pada modul-modul yang rentan, sehingga proses perbaikan berjalan lebih efisien. Dalam pengembangannya, keandalan model SDP sangat bergantung pada kemampuan algoritma klasifikasi yang digunakan untuk mendeteksi data cacat. Pendekatan berbasis Deep Learning memiliki kemampuan yang unggul dalam menangkap pola representasi data non-linear dan kompleks dari kode sumber perangkat lunak dibandingkan metode tradisional (Liang, 2024). Salah satu metode Deep Learning yang terbukti andal dalam memproses pola data yang rumit adalah Multi Layer Perceptron (MLP) yang memiliki efektivitas tinggi jika dikonfigurasi dengan tepat (Mustaqeem et al., 2024).

Penelitian SDP umumnya dievaluasi menggunakan dataset standar dari industri penerbangan dan antariksa, salah satunya adalah dataset NASA Metrics Data Program (NASA MDP) yang bersumber dari repositori terbuka PROMISE. Dataset ini menyediakan beragam metrik kualitas perangkat lunak seperti metrik ukuran kode (Lines of Code), metrik

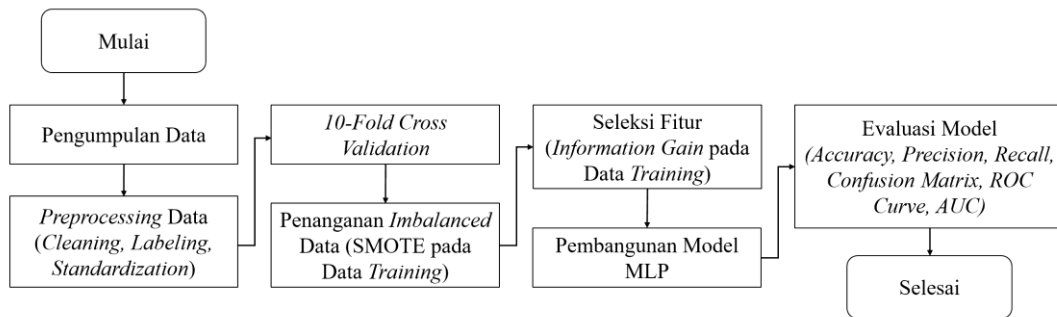
Halstead untuk mengukur volume kognitif, metrik McCabe untuk menilai kompleksitas alur logika, serta metrik tambahan lainnya (Ramadhani et al., 2024). Namun, pemanfaatan dataset NASA MDP secara langsung menghadapi kendala besar terkait dimensi atribut yang tinggi serta ketidakseimbangan kelas (imbalanced data) yang sangat ekstrem. Distribusi kelas yang sangat timpang, di mana kelas modul tidak cacat (Non-Defective) jauh lebih mendominasi dibandingkan kelas modul cacat (Defective), menyebabkan model klasifikasi cenderung bias terhadap kelas mayoritas. Akibatnya, model sering kali menghasilkan nilai akurasi umum yang sangat tinggi tetapi gagal mendeteksi modul yang sebenarnya cacat, yang dicerminkan oleh nilai Recall yang sangat rendah.

Keterbatasan tersebut dicoba diselesaikan oleh beberapa penelitian terdahulu dengan berbagai fokus metodologi. Sapta Nugraha et al. (2021) menerapkan Deep Neural Network (DNN) pada dataset MC1 dan PC1 dengan hasil performa AUC yang baik, namun penelitian tersebut belum mengintegrasikan teknik seleksi fitur untuk mereduksi dimensi atribut yang redundan. Siswanto dan Yuhana (2023) menyimpulkan bahwa pemilihan fitur dan optimasi parameter sangat memengaruhi performa klasifikasi, tetapi mereka belum mengombinasikannya dengan model klasifikasi berbasis Deep Learning. Penelitian oleh Khalid et al. (2023) menganalisis performa Support Vector Machine (SVM) untuk prediksi cacat, namun belum memanfaatkan keunggulan representasi otomatis dari arsitektur MLP berbasis Deep Learning. Goyal dan Sinha (2023) berfokus pada penanganan ketidakseimbangan data menggunakan algoritma boosting, tetapi belum menguji efektivitas penggunaan teknik over-sampling yang lebih stabil seperti Synthetic Minority Over-sampling Technique (SMOTE). Secara kritis, analisis terhadap literatur terdahulu menunjukkan adanya kesenjangan penelitian (research gap) yang nyata. Masih sangat terbatas penelitian yang mengkaji integrasi kekuatan arsitektur MLP dengan penanganan ketidakseimbangan kelas menggunakan SMOTE dan seleksi fitur berbasis teori informasi secara bersamaan.

Guna mengatasi kesenjangan tersebut, artikel ini mengusulkan sebuah pendekatan integratif yang menggabungkan model MLP dengan teknik seleksi fitur Information Gain (IG) dan penyeimbangan data menggunakan SMOTE pada dataset gabungan NASA MDP. Penggunaan Information Gain berfungsi untuk menyaring dan memilih sepuluh atribut metrik perangkat lunak yang paling relevan dan berkontribusi signifikan terhadap variabel target, sehingga mampu mengurangi noise, kompleksitas komputasi, dan risiko overfitting (Pudjihartono et al., 2022). Di sisi lain, metode SMOTE diterapkan untuk menyeimbangkan sebaran data kelas minoritas dengan cara membangkitkan data sintesis baru, sehingga model MLP dapat mempelajari karakteristik kelas Defective secara optimal tanpa kehilangan data penting dari kelas mayoritas (Jude & Uddin, 2024). Kebaruan (novelty) dari penelitian ini terletak pada formulasi integrasi antara MLP, seleksi fitur Information Gain, dan SMOTE yang diterapkan khusus secara internal pada setiap lipatan dalam proses Stratified 10-Fold Cross Validation guna mencegah kebocoran data (data leakage). Penelitian ini bertujuan untuk mengimplementasikan algoritma MLP dalam prediksi cacat perangkat lunak, menerapkan seleksi fitur Information Gain untuk meningkatkan performansi model pada dataset NASA MDP, serta menganalisis pengaruh gabungan seleksi fitur dan SMOTE terhadap kinerja model MLP dalam memprediksi potensi kesalahan kode sejak tahap awal.

METODE

Penelitian ini menerapkan jenis penelitian kuantitatif eksperimental untuk menguji dan membandingkan performansi model klasifikasi dalam memprediksi cacat perangkat lunak. Sumber data yang digunakan adalah data sekunder yang dikumpulkan melalui teknik observasi non-partisipan dari repositori terbuka Open Science tera-PROMISE Data Repository yang dapat diakses secara publik pada tautan resmi (<https://openscience.us/repo/defect/>). Dataset yang dianalisis berasal dari NASA Metrics Data Program (NASA MDP) yang terdiri atas enam dataset perangkat lunak mandiri dengan karakteristik metrik kode yang bervariasi, yaitu: CM1 (sistem instrumen pesawat ruang angkasa), JM1 (sistem prediksi darat waktu nyata), KC3 (sistem pengumpul data satelit), PC1 (instrumen pesawat ruang angkasa orbit bumi), PC5 (sistem operasi instrumen pesawat ruang angkasa), dan MW1 (sistem kontrol gravitasi nol). Penggabungan keenam dataset tersebut menghasilkan satu dataset utama yang komprehensif dengan total populasi data sebanyak 28.258 record. Tahapan sistematis yang dilakukan mulai dari pengumpulan data hingga evaluasi model dirangkum secara visual pada Gambar 1.



Gambar 1. Alur Diagram Tahapan Metodologi Penelitian

Sebelum digunakan untuk melatih model, data terlebih dahulu melewati tahap preprocessing data guna menjamin konsistensi dan kualitas informasi yang diolah. Langkah pertama dimulai dengan mengonversi format dokumen asli dari Attribute-Relation File Format (.arff) menjadi Comma-Separated Values (.csv) menggunakan bantuan perangkat lunak Weka guna mempermudah manipulasi data menggunakan bahasa pemrograman Python (Ratra et al., 2021). Selanjutnya, dilakukan transformasi dan standarisasi label target. Mengingat adanya perbedaan penamaan variabel target di beberapa dataset (dataset JM1 menggunakan atribut label, PC5 menggunakan c, dan dataset lainnya menggunakan Defective), seluruh variabel tersebut disatukan secara konsisten menjadi nama atribut target Defective. Label target ini dikodekan secara numerik menjadi 1 untuk merepresentasikan modul yang cacat (Defective) dan 0 untuk modul yang tidak cacat (Non-Defective). Tahap berikutnya adalah pembersihan data (data cleaning) dan penanganan data hilang (missing value). Untuk menjaga kualitas data, dilakukan eliminasi terhadap 18 atribut yang memiliki persentase missing value di atas 30 persen akibat penggabungan dataset, seperti atribut CALL_PAIRS, NODE_COUNT, PARAMETER_COUNT, dan sejenisnya. Langkah ini menghasilkan dataset akhir dengan 21 atribut fitur numerik dan 1 atribut target biner. Sementara itu, penanganan

sis missing value di bawah 30 percent diatasi dengan metode median imputation, di mana nilai kosong diganti dengan nilai tengah dari atribut numerik bersangkutan untuk menjaga stabilitas sebaran data dari pengaruh data pencilan (outlier). Seluruh atribut fitur numerik tersebut kemudian diselaraskan skalanya menggunakan standardisasi StandardScaler agar proses pelatihan model jaringan saraf tiruan dapat berjalan optimal tanpa bias skala. Untuk mencegah terjadinya kebocoran data (data leakage), proses standardisasi skala ini hanya diterapkan pada data latih (training data) di setiap iterasi validasi silang.

Dataset utama ini dicirikan oleh masalah ketidakseimbangan data (imbalanced data) yang sangat ekstrem, dengan sebaran kelas mayoritas Non-Defective sebanyak 25.817 record dan kelas minoritas Defective hanya sebanyak 2.441 record. Kondisi ini diatasi dengan menerapkan metode Synthetic Minority Over-sampling Technique (SMOTE). SMOTE bekerja secara aktif menyeimbangkan sebaran kelas target dengan cara membangkitkan data sintesis baru pada kelas minoritas (Defective) menggunakan prinsip interpolasi berdasarkan kedekatan jarak antardata terdekat (k-nearest neighbors), sehingga jumlah sampel kedua kelas menjadi seimbang masing-masing sebanyak 25.817 record (Jude & Uddin, 2024; Altalhan et al., 2025). Demi mempertahankan objektivitas pengujian dan mencegah data leakage, penerapan SMOTE hanya dijalankan secara khusus pada data latih di setiap lipatan (fold) dalam proses validasi silang, sedangkan data uji (testing data) dibiarkan tetap dalam kondisi distribusi aslinya. Untuk mereduksi dimensi data yang redundan dan meminimalkan noise yang dapat menurunkan kinerja klasifikasi, diterapkan metode seleksi fitur (Feature Selection) menggunakan Information Gain (IG) (Pudjihartono et al., 2022). Information Gain bekerja dengan mengukur seberapa besar suatu atribut metrik perangkat lunak mampu mengurangi tingkat ketidakpastian (entropy) terhadap variabel target Defective (Pratama & Armansyah, 2024). Nilai kontribusi atau kepentingan metrik dihitung dan dirata-ratakan (average Information Gain) di setiap lipatan data latih untuk menghasilkan peringkat kontribusi yang stabil. Dari pemeringkatan tersebut, dipilih 10 atribut terbaik dengan skor Information Gain tertinggi sebagai masukan utama bagi model klasifikasi. Sama seperti metode standardisasi dan penyeimbangan data, tahapan seleksi fitur ini hanya diaplikasikan pada data latih di setiap fold validasi silang.

Pembangunan model prediktif dalam penelitian ini mengimplementasikan algoritma Deep Learning yaitu Multi Layer Perceptron (MLP) yang terdiri atas lapisan masukan, lapisan tersembunyi (hidden layer), dan lapisan keluaran (Desai & Shah, 2021). Model MLP dilatih menggunakan parameter model terkonfigurasi secara spesifik, yang meliputi: ukuran satu lapisan tersembunyi sebanyak 100 neuron; fungsi aktivasi Rectified Linear Unit (ReLU) untuk mempercepat komputasi dan meminimalkan masalah vanishing gradient; algoritma optimasi bobot menggunakan solver Adam yang stabil dan efisien; jumlah maksimum iterasi pelatihan sebanyak 300 iterasi; tingkat pembelajaran yang adaptif (adaptive learning rate) untuk menjaga stabilitas konvergensi; serta penguncian parameter acak (random state) pada nilai 42 demi menjamin konsistensi hasil pengujian. Pembaruan nilai bobot jaringan dilakukan secara berulang melalui algoritma pembelajaran mundur (Backpropagation) berdasarkan tingkat kesalahan prediksi (error). Untuk memperoleh estimasi performansi yang solid dan objektif, seluruh skenario pemodelan dievaluasi secara ketat menggunakan metode Stratified 10-Fold Cross Validation (Malakouti et al., 2023; Sweet et al., 2023). Dataset utama dibagi ke

dalam 10 bagian (fold) berukuran sama dengan tetap mempertahankan proporsi sebaran kelas Defective dan Non-Defective yang konsisten di setiap lipatan. Eksperimen klasifikasi diulang sebanyak 10 kali, di mana pada setiap iterasi, 9 bagian data digunakan sebagai data latih dan 1 bagian sisanya dialokasikan sebagai data uji. Kinerja model diukur menggunakan metrik evaluasi yang komprehensif, yaitu: Accuracy (mengukur ketepatan prediksi umum), Precision (menilai ketepatan prediksi modul cacat), Recall (mengukur sensitivitas model dalam mendeteksi seluruh modul cacat riil), F1-Score (rata-rata harmonis penyeimbang nilai Precision dan Recall), Confusion Matrix (untuk merekam jumlah prediksi benar dan salah), serta Area Under the ROC Curve (AUC) untuk mengevaluasi kekuatan model dalam memisahkan kedua kelas pada berbagai ambang batas klasifikasi (Rainio et al., 2024; Sathyanarayanan, 2024).

HASIL DAN PEMBAHASAN

Hasil Penelitian

1. Hasil Pengolahan Data

Eksperimen diawali dengan pengolahan dataset yang bersumber dari repositori terbuka NASA Metrics Data Program (NASA MDP). Penelitian ini mengintegrasikan enam dataset perangkat lunak mandiri, yaitu CM1, JM1, KC3, MW1, PC1, dan PC5. Integrasi ini bertujuan untuk menciptakan basis data yang lebih luas dan bervariasi demi meningkatkan kemampuan generalisasi model. Karakteristik rinci dari setiap dataset penyusun sebelum digabungkan disajikan pada Tabel 1. Sesuai dengan format penulisan jurnal ilmiah IJIMR, seluruh tabel disajikan dengan lebar garis 1 pt, tanpa menggunakan garis pembatas vertikal, dan keterangan tabel diletakkan di atas tabel.

Tabel 1. Karakteristik Spesifik Dataset Perangkat Lunak NASA MDP

Dataset	Sistem	Bahasa	Jumlah Atribut	Total Modul	Modul Cacat	Modul Tidak Cacat	Persentase Cacat
CM1	Instrumen pesawat ruang angkasa	C	38	327	42	285	12,84%
JM1	Sistem prediksi darat (real-time ground system)	C	22	9.593	1.759	7.834	18,34%
KC3	Sistem pengumpul data satelit	Java	40	194	36	158	18,56%

	(satellite storage management)						
PC1	Instrumen pesawat ruang angkasa orbit bumi	C	38	705	61	644	8,65%
PC5	Sistem operasi instrumen pesawat ruang angkasa	C++	39	17.186	516	16.670	3,00%
MW1	Sistem kontrol gravitasi nol (zero gravity experiment)	C	38	253	27	226	10,67%

Penggabungan keenam dataset tersebut menghasilkan satu dataset utama dengan populasi sebanyak 28.258 record data. Proses standardisasi label target dilakukan karena adanya perbedaan nama variabel keputusan pada berkas asli masing-masing dataset. Dataset JM1 menggunakan nama atribut label, dataset PC5 menggunakan c, sedangkan dataset lainnya menggunakan Defective. Seluruh atribut target ini diselaraskan menjadi satu atribut keputusan yang konsisten dengan nama Defective. Nilai label dikodekan secara biner, yaitu nilai 1 untuk merepresentasikan modul yang cacat (Defective) dan nilai 0 untuk modul yang tidak cacat (Non-Defective).

Berdasarkan hasil penggabungan awal, diperoleh distribusi kelas target yang sangat tidak seimbang. Sebaran kelas target sebelum dilakukan penanganan ketidakseimbangan data disajikan pada Tabel 2.

Tabel 2. Distribusi Kelas Target Sebelum Penerapan SMOTE

Kelas Target	Jumlah Data	Persentase
<i>Non-Defective</i> (0)	25.817	91,36%
<i>Defective</i> (1)	2.441	8,64%
Total	28.258	100,00%

Data pada Tabel 2 menunjukkan ketidakseimbangan kelas yang sangat ekstrem, di mana proporsi modul tidak cacat mencapai lebih dari sembilan puluh persen dari seluruh populasi data. Pada tahap preprocessing data, dilakukan proses pembersihan data (data cleaning) untuk mengatasi nilai yang kosong (missing value). Berdasarkan analisis persentase kehilangan data pada setiap metrik, ditemukan 18 atribut yang memiliki persentase missing value di atas 30 percent akibat dari perbedaan ketersediaan atribut pada berkas asli masing-

masing dataset perangkat lunak. Atribut yang dieliminasi dari dataset adalah CALL_PAIRS, CONDITION_COUNT, CYCLOMATIC_DENSITY, DECISION_COUNT, DECISION_DENSITY, DESIGN_DENSITY, EDGE_COUNT, ESSENTIAL_DENSITY, PARAMETER_COUNT, MAINTENANCE_SEVERITY, MODIFIED_CONDITION_COUNT, MULTIPLE_CONDITION_COUNT, NODE_COUNT, NORMALIZED_CYCLOMATIC_COMPLEXITY, NUMBER_OF_LINES, PERCENT_COMMENTS, GLOBAL_DATA_COMPLEXITY, dan GLOBAL_DATA_DENSITY.

Langkah eliminasi ini menyisakan 21 atribut fitur numerik dan 1 atribut target biner. Penanganan sisa missing value dengan persentase di bawah 30 percent dilakukan menggunakan metode median imputation, di mana nilai kosong diganti dengan nilai median dari atribut terkait untuk menghindari sensitivitas terhadap pencilan (outlier). Setelah itu, standardisasi skala menggunakan StandardScaler diterapkan pada seluruh fitur numerik untuk menyamakan skala rentang nilai. Demi menjaga integritas evaluasi, standardisasi ini hanya diaplikasikan pada data latih (training data) di setiap lipatan (fold) validasi silang.

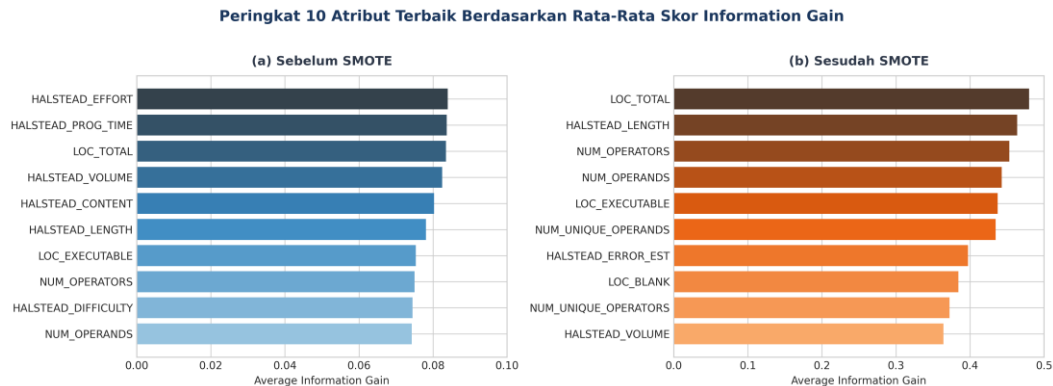
Untuk menangani masalah ketidakseimbangan kelas pada Tabel 2, diterapkan metode Synthetic Minority Over-sampling Technique (SMOTE) khusus pada data latih di setiap fold validasi silang. Penerapan SMOTE berhasil menyetarakan sebaran kelas target, di mana masing-masing kelas target memiliki jumlah data yang seimbang seperti yang disajikan pada Tabel 3.

Tabel 3. Distribusi Kelas Target Setelah Penerapan SMOTE

Kelas Target	Jumlah Data	Persentase
<i>Non-Defective</i> (0)	25.817	50,00%
<i>Defective</i> (1)	25.817	50,00%
Total	51.634	100,00%

2. Hasil Seleksi Fitur (*Feature Selection*)

Proses seleksi fitur berbasis teori informasi dengan metode Information Gain (IG) dijalankan untuk mengukur tingkat kontribusi setiap metrik terhadap target variabel Defective. Skor Information Gain dihitung secara terpisah pada dua kondisi data latih, yaitu sebelum dan sesudah penerapan SMOTE. Hal ini bertujuan untuk mengukur dampak langsung dari penyeimbangan data terhadap nilai penting dari masing-masing fitur. Pemeringkatan metrik dihitung berdasarkan rata-rata nilai penting (average Information Gain) dari seluruh fold validasi silang. Hasil visualisasi perbandingan peringkat atribut terbaik disajikan pada Gambar 2.



Gambar 2. Perbandingan 10 atribut terbaik berdasarkan Information Gain sebelum dan sesudah SMOTE

Hasil Feature Selection Tanpa SMOTE

Pada kondisi data asli yang tidak seimbang, hasil perhitungan average Information Gain menunjukkan tingkat relevansi fitur yang cenderung rendah. Nilai kontribusi tertinggi dicatat oleh metrik HALSTEAD_EFFORT (0,083957) dan HALSTEAD_PROG_TIME (0,083731), diikuti oleh metrik ukuran kode fisik LOC_TOTAL (0,083533). Karakteristik sepuluh metrik terpilih pada Gambar 2a didominasi oleh kelompok Halstead Metrics dan ukuran fisik baris kode (Lines of Code). Hal ini mengindikasikan bahwa tanpa penyeimbangan kelas, model sangat mengandalkan metrik usaha kognitif (HALSTEAD_EFFORT) dan estimasi waktu (HALSTEAD_PROG_TIME) sebagai indikator utama kemunculan cacat.

Hasil Feature Selection Menggunakan SMOTE

Setelah dataset latih diseimbangkan menggunakan SMOTE, dilakukan perhitungan ulang average Information Gain. Hasil perhitungan menunjukkan lonjakan nilai kontribusi yang sangat signifikan pada seluruh fitur terbaik. Sebagaimana disajikan pada Gambar 2b, metrik ukuran fisik kode keseluruhan, yaitu LOC_TOTAL, bergeser menempati peringkat teratas dengan nilai kontribusi sebesar 0,479769. Metrik ini diikuti oleh kompleksitas internal baris program seperti HALSTEAD_LENGTH (0,463875) dan metrik jumlah operator NUM_OPERATORS (0,453139).

Perbandingan Hasil Feature Selection Sebelum dan Sesudah SMOTE

Perbandingan antara Gambar 2a dan Gambar 2b menunjukkan perubahan yang signifikan dalam hal nilai kontribusi maupun peringkat metrik terbaik. Sebelum penerapan SMOTE, nilai Information Gain tertinggi hanya berkisar pada angka 0,083957, sedangkan setelah penerapan SMOTE, nilai penting tersebut meningkat tajam hingga mencapai 0,479769. Perubahan ini membuktikan bahwa penyeimbangan sebaran data dengan membangkitkan data sintesis pada kelas minoritas mampu memperjelas batasan informasi keputusan pada masing-masing metrik kualitas perangkat lunak. Hal ini membantu algoritma seleksi fitur untuk mengidentifikasi atribut-atribut yang benar-benar memiliki korelasi kuat dengan kelas cacat (Defective). Meskipun terjadi pergeseran peringkat, metrik yang terpilih pada kedua kondisi tetap didominasi oleh kelompok Lines of Code dan Halstead Metrics, yang

menunjukkan bahwa kompleksitas fisik dan kognitif secara konsisten menjadi faktor penentu kemunculan cacat perangkat lunak.

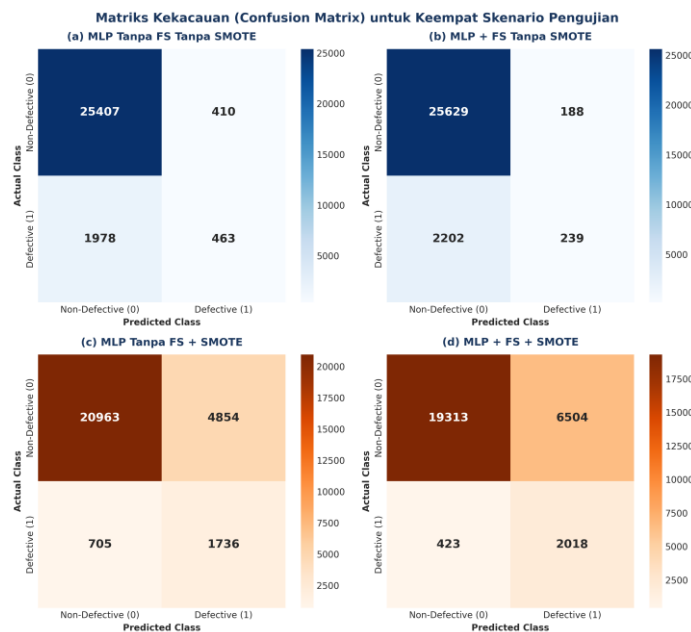
3. Hasil Pengujian Model

Evaluasi performansi model klasifikasi dibangun menggunakan algoritma Deep Learning Multi Layer Perceptron (MLP). Seluruh pengujian dievaluasi secara objektif menggunakan metode Stratified 10-Fold Cross Validation dengan membagi data ke dalam sepuluh lipatan berimbang secara proporsional. Pengujian ini terbagi menjadi empat skenario pemodelan untuk menguji pengaruh individu maupun kombinasi dari seleksi fitur dan penanganan ketidakseimbangan data. Rangkuman hasil performansi dari keempat skenario pengujian tersebut disajikan secara komprehensif pada Tabel 4.

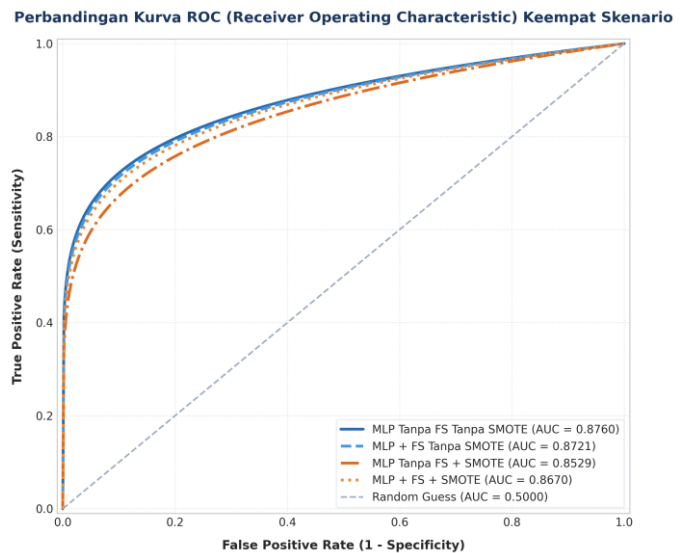
Tabel 6. Perbandingan Hasil Performansi Model MLP pada Empat Skenario Pengujian

Skenario Pengujian	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>AUC</i>
MLP Tanpa FS Tanpa SMOTE	0,9155	0,5304	0,1897	0,2794	0,8760
MLP + FS Tanpa SMOTE	0,9154	0,5597	0,0979	0,1667	0,8721
MLP Tanpa FS + SMOTE	0,8033	0,2634	0,7112	0,3845	0,8529
MLP + FS + SMOTE	0,7549	0,2368	0,8267	0,3681	0,8670

Untuk memperjelas analisis sebaran kesalahan prediksi rill dari masing-masing skenario secara verbal, hasil pencatatan matriks kekacauan (Confusion Matrix) untuk keempat skenario digabungkan ke dalam Gambar 3. Sementara itu, kemampuan model dalam memisahkan kelas keputusan divisualisasikan melalui perbandingan Kurva ROC pada Gambar 4.



Gambar 3. Matriks Kekacauan untuk Keempat Skenario Pengujian Model MLP



Gambar 4. Perbandingan Kurva ROC untuk Keempat Skenario Pengujian Model MLP Tanpa Feature Selection dan Tanpa SMOTE

Pada skenario pertama, model MLP dilatih menggunakan seluruh 21 atribut fitur numerik hasil preprocessing tanpa mengaplikasikan seleksi fitur dan penyeimbangan data menggunakan SMOTE. Berdasarkan pengujian validasi silang pada Tabel 4, skenario pertama ini menghasilkan nilai Akurasi sebesar 0,9155, Presisi sebesar 0,5304, Recall sebesar 0,1897,

F1-Score sebesar 0,2794, dan AUC sebesar 0,8760. Hasil pencatatan matriks kekacauan (Gambar 3a) menunjukkan nilai True Negative (TN) sebesar 25.407, False Positive (FP) sebesar 410, False Negative (FN) sebesar 1.978, dan True Positive (TP) sebesar 463. Angka AUC yang cukup tinggi ini menunjukkan kemampuan pemisahan kelas yang baik, namun nilai Recall yang sangat rendah (0,1897) menegaskan bahwa model ini gagal mendeteksi mayoritas modul yang sebenarnya mengalami cacat akibat distribusi data yang tidak seimbang.

MLP Menggunakan Feature Selection Tanpa SMOTE

Pada skenario kedua, model MLP dikembangkan dengan menerapkan seleksi fitur Information Gain untuk memilih sepuluh atribut metrik terbaik tanpa melakukan penyeimbangan data menggunakan SMOTE. Hasil pengujian pada Tabel 4 mencatat nilai Akurasi sebesar 0,9154, Presisi sebesar 0,5597, Recall sebesar 0,0979, F1-Score sebesar 0,1667, dan AUC sebesar 0,8721. Hasil pencatatan matriks kekacauan (Gambar 3b) memperoleh nilai TN sebesar 25.629, FP sebesar 188, FN sebesar 2.202, dan TP sebesar 239. Reduksi atribut menjadi sepuluh fitur terbaik tanpa disertai penanganan ketidakseimbangan kelas justru menurunkan nilai Recall hingga di bawah sepuluh persen, meskipun mencatat nilai Presisi tertinggi sebesar 0,5597.

MLP Tanpa Feature Selection Menggunakan SMOTE

Pada skenario ketiga, model MLP dilatih menggunakan seluruh 21 atribut fitur numerik dengan menerapkan metode SMOTE khusus pada data training di setiap fold validasi silang. Hasil pengujian pada Tabel 4 menunjukkan nilai Akurasi sebesar 0,8033, Presisi sebesar 0,2634, Recall sebesar 0,7112, F1-Score sebesar 0,3845, dan AUC sebesar 0,8529. Hasil pencatatan matriks kekacauan (Gambar 3c) mendeteksi TN sebesar 20.963, FP sebesar 4.854, FN sebesar 705, dan TP sebesar 1.736. Penerapan metode SMOTE pada skenario ketiga ini berhasil meningkatkan nilai Recall secara tajam dari 0,1897 menjadi 0,7112, yang mengonfirmasi efektivitas penyeimbangan kelas dalam melatih kepekaan jaringan saraf tiruan.

MLP Menggunakan Feature Selection dan SMOTE

Pada skenario keempat, model MLP dikembangkan dengan mengintegrasikan seleksi fitur Information Gain (sepuluh atribut terbaik) dan penyeimbangan kelas menggunakan SMOTE secara bersamaan. Hasil pengujian pada Tabel 4 menunjukkan nilai Akurasi sebesar 0,7549, Presisi sebesar 0,2368, Recall sebesar 0,8267, F1-Score sebesar 0,3681, dan AUC sebesar 0,8670. Hasil pencatatan matriks kekacauan (Gambar 3d) menunjukkan TN sebesar 19.313, FP sebesar 6.504, FN sebesar 423, dan TP sebesar 2.018. Integrasi kombinasi seleksi fitur dan SMOTE pada skenario keempat ini berhasil menghasilkan kemampuan deteksi modul cacat (Recall) tertinggi dari seluruh skenario pengujian, yaitu sebesar 0,8267.

4. Pembahasan

Hasil eksperimen menunjukkan adanya trade-off antara Accuracy dan Recall. Pada skenario pertama dan kedua, model yang dilatih menggunakan data asli menghasilkan nilai akurasi di atas 91%. Namun, berdasarkan Confusion Matrix (Gambar 3a dan Gambar 3b), tingginya akurasi tersebut disebabkan oleh dominasi prediksi pada kelas Non-Defective,

sehingga nilai Recall sangat rendah, yaitu 0,1897 dan 0,0979. Kondisi ini menunjukkan bahwa model belum mampu mendeteksi sebagian besar modul yang benar-benar cacat. Setelah diterapkan SMOTE pada skenario ketiga dan keempat, nilai Recall meningkat menjadi 0,7112 dan 0,8267, yang menunjukkan peningkatan kemampuan model dalam mengenali kelas Defective.

Pengaruh Information Gain berbeda sebelum dan sesudah penerapan SMOTE. Pada data asli tanpa SMOTE, seleksi fitur menurunkan nilai Recall dari 0,1897 menjadi 0,0979. Sebaliknya, setelah data diseimbangkan menggunakan SMOTE, seleksi fitur meningkatkan nilai Recall menjadi 0,8267 dan AUC dari 0,8529 menjadi 0,8670. Hasil ini menunjukkan bahwa kombinasi SMOTE dan Information Gain mampu mengurangi atribut yang kurang relevan sehingga model dapat mempelajari pola yang lebih representatif.

Meskipun skenario MLP + Information Gain + SMOTE menghasilkan lebih banyak False Positive, model ini tetap menjadi yang terbaik karena mampu memberikan nilai Recall tertinggi. Dalam prediksi cacat perangkat lunak, kesalahan False Negative lebih berisiko karena modul yang cacat dapat lolos ke tahap implementasi dan berpotensi menyebabkan kegagalan sistem. Oleh karena itu, model dengan kemampuan deteksi cacat yang lebih tinggi lebih sesuai untuk diterapkan.

Hasil penelitian ini juga menunjukkan bahwa metrik LOC_TOTAL, HALSTEAD_LENGTH, dan HALSTEAD_VOLUME merupakan atribut yang paling berpengaruh dalam prediksi cacat perangkat lunak. Temuan ini mengindikasikan bahwa semakin besar ukuran dan kompleksitas kode, semakin tinggi kemungkinan munculnya cacat.

Penelitian ini menunjukkan bahwa kombinasi MLP, Information Gain, dan SMOTE mampu meningkatkan kemampuan deteksi cacat perangkat lunak dibandingkan penggunaan MLP tanpa penanganan ketidakseimbangan data. Namun, penggunaan SMOTE masih menyebabkan penurunan Precision akibat meningkatnya jumlah False Positive. Oleh karena itu, penelitian selanjutnya dapat mengombinasikan SMOTE dengan teknik pembersihan data setelah over-sampling untuk meningkatkan keseimbangan performa model.

SIMPULAN

Penelitian ini menyimpulkan bahwa model Multi Layer Perceptron (MLP) dapat diterapkan secara efektif untuk prediksi cacat perangkat lunak menggunakan dataset NASA Metrics Data Program (NASA MDP). Penggabungan metode seleksi fitur Information Gain dan penanganan ketidakseimbangan kelas menggunakan SMOTE (skenario MLP + FS + SMOTE) terbukti menghasilkan performa pendeteksian cacat (Recall) paling optimal sebesar 0,8267 dan AUC 0,8670, yang berhasil meminimalkan risiko False Negative secara signifikan. Pengembang perangkat lunak disarankan mengimplementasikan kombinasi metode ini untuk menyaring dan menguji modul-modul yang kompleks secara terarah guna menghemat biaya operasional pemeliharaan sistem.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Program Studi Teknik Informatika Universitas Pamulang atas dukungan dan fasilitas akademik yang diberikan selama penyusunan penelitian ini. Ucapan terima kasih juga disampaikan kepada Bapak Bobi

Agustian, S.Kom., M.Kom., selaku dosen pembimbing, atas bimbingan, arahan, dan masukan yang diberikan selama proses penyusunan hingga konversi skripsi menjadi artikel jurnal ilmiah. Selain itu, penulis mengucapkan terima kasih kepada Juniawati atas dukungan dan motivasi yang diberikan selama penyusunan manuskrip ini.

DAFTAR PUSTAKA

- Altalhan, M., Algarni, A., & Turki-Hadj Alouane, M. (2025). Imbalanced Data Problem In Machine Learning: A Review. *IEEE Access*, 13, 13686-13699. <https://doi.org/10.1109/ACCESS.2025.3531662>
- Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaria, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review Of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions. *Journal Of Big Data*, 8(1). <https://doi.org/10.1186/S40537-021-00444-8>
- Buyukkececi, M., & Okur, M. C. (2023). A Comprehensive Review Of Feature Selection And Feature Selection Stability In Machine Learning. *Gazi University Journal Of Science*, 36(4), 1506-1520. <https://doi.org/10.35378/Gujs.993763>
- Corbacioglu, S. K., & Aksel, G. (2023). Receiver Operating Characteristic Curve Analysis In Diagnostic Accuracy Studies: A Guide To Interpreting The Area Under The Curve Value. *Turkish Journal Of Emergency Medicine*, 23(4), 195-198. https://doi.org/10.4103/Tjem.Tjem_182_23
- Desai, M., & Shah, M. (2021). An Anatomization On Breast Cancer Detection And Diagnosis Employing Multi-Layer Perceptron Neural Network (MLP) And Convolutional Neural Network (CNN). *Clinical Ehealth*, 4, 1-11. <https://doi.org/10.1016/J.Ceh.2020.11.002>
- Goyal, J., & Sinha, R. R. (2023). A New Improved Prediction Of Software Defects Using Machine Learning-Based Boosting Techniques With NASA Dataset. *International Journal On Recent And Innovation Trends In Computing And Communication*, 11(10s), 492-504. <https://doi.org/10.17762/Ijritcc.V11i10s.7659>
- Han, J., Pei, J., & Tong, H. (2022). *Data Mining: Concepts And Techniques*, Fourth Edition. Morgan Kaufmann. <https://doi.org/10.1016/C2013-0-18660-6>
- Jorayeva, M., Akbulut, A., Catal, C., & Mishra, A. (2022). Machine Learning-Based Software Defect Prediction For Mobile Applications: A Systematic Literature Review. *Sensors*, 22(7), 2551. <https://doi.org/10.3390/S22072551>
- Jude, A., & Uddin, J. (2024). Explainable Software Defects Classification Using SMOTE And Machine Learning. *Annals Of Emerging Technologies In Computing*, 8(1), 35-49. <https://doi.org/10.33166/Aetic.2024.01.00>
- Kabir, M. A., Ahmed, F., Islam, M. M., & Ahmed, Md. R. (2024). Python For Data Analytics: A Systematic Literature Review Of Tools, Techniques, And Applications. *Academic Journal On Science, Technology, Engineering & Mathematics Education*, 4(4), 134-154. <https://doi.org/10.69593/Ajsteme.V4i04.146>
- Kemendikbud, T., Wintana, D., Sulaeman, H., & Bahri, S. (2024). Penerapan Multi Layer Perceptron Dan Diskrit Pada Prediksi Cacat Software. *J-INTECH (Journal Of Information And Technology)*, 12(1), 45-53.

- Khalid, A., Badshah, G., Ayub, N., Shiraz, M., & Ghouse, M. (2023). Software Defect Prediction Analysis Using Machine Learning Techniques. *Sustainability (Switzerland)*, 15(6), 5517. <https://doi.org/10.3390/Su15065517>
- Liang, Q. (2024). Research On Software Defect Prediction Model Based On Deep Learning. *Highlights In Science, Engineering And Technology*, 10, 112-120.
- Malakouti, S. M., Menhaj, M. B., & Suratgar, A. A. (2023). The Usage Of 10-Fold Cross-Validation And Grid Search To Enhance ML Methods Performance In Solar Farm Power Generation Prediction. *Cleaner Engineering And Technology*, 15, 100664. <https://doi.org/10.1016/J.Clet.2023.100664>
- Mustaqeem, M., Mustajab, S., & Alam, M. (2024). A Hybrid Approach For Optimizing Software Defect Prediction Using A Gray Wolf Optimization And Multilayer Perceptron. *International Journal Of Intelligent Computing And Cybernetics*, 17(2), 241-260. <https://doi.org/10.1108/IJICC-11-2023-0385>
- Ningsih, D., Sukmasetya, P., & Gustian, D. (2026). Implementasi Algoritma Machine Learning Pada Data Tidak Seimbang Menggunakan SMOTE Untuk Klasifikasi Kemiskinan Di Indonesia. *Jurnal Algoritme*, 6(2), 296-309. <https://doi.org/10.35957/Algoritme.V6i2.13518>
- Pachouly, J., Ahirrao, S., Kotecha, K., Selvachandran, G., & Abraham, A. (2022). A Systematic Literature Review On Software Defect Prediction Using Artificial Intelligence: Datasets, Data Validation Methods, Approaches, And Tools. *Engineering Applications Of Artificial Intelligence*, 111, 104773. <https://doi.org/10.1016/J.Engappai.2022.104773>
- Pudjihartono, N., Fadason, T., Kempa-Liehr, A. W., & O'Sullivan, J. M. (2022). A Review Of Feature Selection Methods For Machine Learning-Based Disease Risk Prediction. *Frontiers In Bioinformatics*, 2, 927312. <https://doi.org/10.3389/Fbinf.2022.927312>
- Rainio, O., Teuho, J., & Klen, R. (2024). Evaluation Metrics And Statistical Tests For Machine Learning. *Scientific Reports*, 14(1), 5670. <https://doi.org/10.1038/S41598-024-56706-X>
- Ramadhani, A. A. P., Nugroho, R. A., Faisal, M. R., Abadi, F., & Herteno, R. (2024). The Impact Of Software Metrics In NASA Metric Data Program Dataset Modules For Software Defect Prediction. *Telkomnika (Telecommunication Computing Electronics And Control)*, 22(4), 846-853. <https://doi.org/10.12928/TELKOMNIKA.V22i4.25787>
- Ratra, R., Gulia, P., & Gill, N. S. (2021). Performance Analysis Of Classification Techniques In Data Mining Using WEKA. *SSRN Electronic Journal*, 12, 1-10. <https://doi.org/10.2139/SSRN.3879610>
- Sapta Nugraha, A., Reza Faisal, M., Abadi, F., Adi Nugroho, R., & Herteno, R. (2021). Deep Neural Network On Software Defect Prediction. *Journal Of Data Science And Software Engineering*, 8(2), 112-121.
- Sarker, I. H. (2021a). Deep Learning: A Comprehensive Overview On Techniques, Taxonomy, Applications And Research Directions. *SN Computer Science*, 2(6), 420. <https://doi.org/10.1007/S42979-021-00815-1>
- Sarker, I. H. (2021b). Machine Learning: Algorithms, Real-World Applications And Research Directions. *SN Computer Science*, 2(3), 160. <https://doi.org/10.1007/S42979-021-00592-Y>

- Sathyanarayanan, S. (2024). Confusion Matrix-Based Performance Evaluation Metrics. *African Journal Of Biomedical Research*, 27(4), 4023-4031.
- Siswantoro, M. Z. F. N., & Yuhana, U. L. (2023). Software Defect Prediction Based On Optimized Machine Learning Models: A Comparative Study. *Teknika*, 12(2), 166-172. <https://doi.org/10.34148/Teknika.V12i2.634>
- Sweet, L., Müller, C., Anand, M., & Zscheischler, J. (2023). Cross-Validation Strategy Impacts The Performance And Interpretation Of Machine Learning Models. *Atmosphere*, 14(10), 1205. <https://doi.org/10.1175/AIES-D-23>