Pengaruh Penanganan Ketidakseimbangan Kelas pada Prediksi Cacat Perangkat Lunak dengan Teknik Oversampling

I Gusti Agung Ramananda Wira Dharma[a1], I Wayan Santiyasa[a2]

Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Udayana
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia
[1]agungramananda@gmail.com
[2]santiyasa@unud.ac.id

Abstract

Software defect prediction plays a vital role in SDLC testing by identifying modules prone to defects. However, imbalanced class distributions, where defect (minority) samples are outnumbered by non-defect ones, can hinder model performance. This study investigates the impact of oversampling techniques (SMOTE, ADASYN) on Naive Bayes classification for defect prediction. While the base Naive Bayes model achieved good overall accuracy (83%), it struggled with defect class recall (30%). Applying SMOTE and ADASYN improved recall (40% and 38%, respectively) but slightly lowered accuracy (77% and 80%). Future work will explore feature selection and deep learning approaches for potentially better performance.

Keywords: *Software Defect Prediction, Classification, Naïve Bayes, Oversampling, SMOTE, ADASYN*

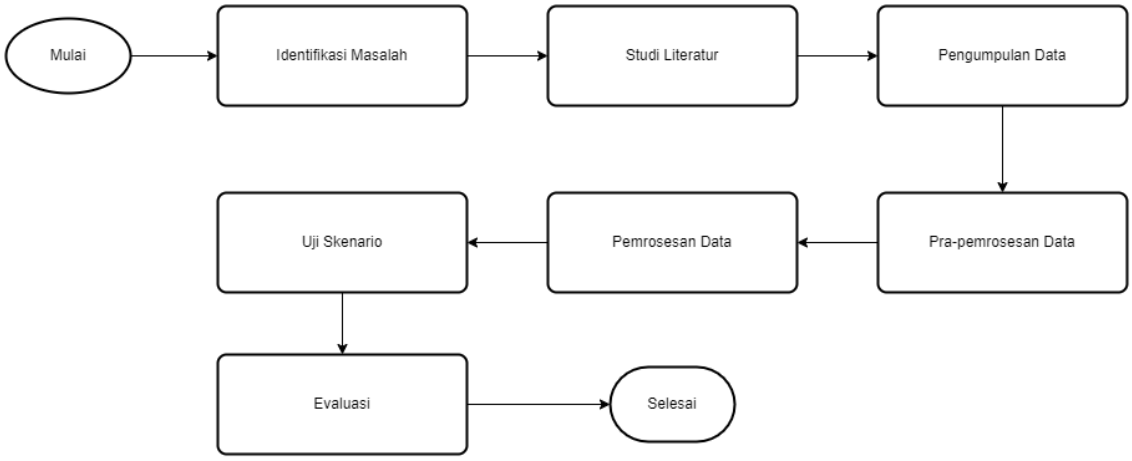
1. Pendahuluan

Prediksi cacat perangkat lunak merupakan salah satu hal yang paling membantu dalam fase pengujian SDLC (Software Development Life Cycle) dengan mengidentidikasi modul-modul yang rawan cacat dan memerlukan pengujian ekstesif. Dengan cara ini, sumber daya pada pengujian dapat digunakan secara efisien tanpa melanggar batasan-batasan [1]. Saat ini, algoritma machine learning yang umum digunakan adalah K-Means, K Nearest Neighbors (KNN), Naïve Bayes, Decision Tree, Support Vector Machine (SVM), dan Random Forest (RF) [2]. Efisiensi model prediksi pada cacat perangkat lunak sangat ditentukan oleh distribusi kelas dari data pelatihan [1]. Distribusi kelas digambarkan sebagai jumlah instance dari setiap kelas dalam datset pelatihan. Jika jumlah instance yang termasuk dalam satu kelas jauh lebih banyak daripada jumlah instance yang termasuk dalam kelas lain, maka masalah tersebut dikenal sebagai masalah ketidakseimbangan kelas [3]. Kelas dengan lebih banyak instance disebut kelas mayoritas dan kelas dengan instance yang lebih sedikit disebut kelas minoritas [1]. Pada prediksi cacat perangkat lunak kelas non-cacat merupakan kelas mayoritas dan kelas cacat merupakan kelas minortias. Ketidakseimbangan kelas dapat menyebabkan model yang tidak praktis dalam prediksi cacat perangkat lunak, karena kebanyakan kasus akan diprediksi sebagai non-cacat rawan [4]. Terdapat beberapa metode yang diusulkan untuk menangani masalah ketidakseimbangan kelas dari berbagai penelitian diantaranya, yakni pada penelitian Liu et al [5] menggunakan tiga jenis teknik sampling seperti resampling, spread subsampling dan SMOTE, dan lima jenis pengklasifikasi machine learning seperti C4.5, Naïve Bayes, Logistic Regression, Support Vector Machine and Deep Learning untuk prediksi cacat. Diambil kesimpulan bahwa sampling sangat berpengaruh besar terhadap hasil prediksi cacat karena menghasilkan distribusi data yang berbeda untuk pelatihan model. Dan pada penelitian [6] mengusulkan metode yaitu Naïve Bayes dengan integrasi teknik oversampling SMOTE dengan seleksi fitur Information Gain untuk solusi dari ketidakseimbangan kelas. Metode yang diusulkan meningkatkan kinjerja dari Naïve Bayes pada prediksi cacat perangkat lunak dengan diperoleh hasil nilai rata-rata AUC pada

model adalah 0.798, sedangkan model Naïve Bayes yang memiliki nilai rata-rata AUC adalah 0.753.

Dari hasil penelitian-penelitian diatas, penelitian ini dilakukan dengan tujuan untuk mengetahui pengaruh dari penanganan ketidakseimbangan kelas dengan menggunakan teknik oversampling. Teknik oversampling yang diterapkan diantaranya SMOTE dan ADASYN dengan menguji performa dari model klasifikasi Naïve Bayes. Dilakukan pengujian beberapa skenario pada data yang telah dilakukan oversampling dan yang tidak dilakukan oversampling. Evaluasi menggunakan beberapa metriks dan dilakukan perbandingan pada beberapa pengujian berdasarkan skenario dan metode-metode yang diterapkan pada model klasifikasi yang digunakan.

2. Metode Penelitian



Gambar 1. Alur Penelitian

Dalam penelitian yang dilakukan untuk membandingkan kinerja model klasifikasi antara menggunakan SMOTE dengan tanpa melakukan oversampling data pada software defect prediction ini dibagi menjadi 6 tahapan seperti yang dapat dilihat pada Gambar 1. Berikut penjelasan dari alur penelitian tersebut:

a. Identifikasi Masalah
   Pada tahapan dilakukan identifikasi latar belakang permasalahan yang akan diangkat pada penelitian. Tahapan ini sangat menentukan tahapan-tahapan selanjutnya seperti topik studi literatur yang akan dipelajari, data yang akan digunakan, dan metode yang akan digunakan pada penelitian.
b. Studi Literatur
   Tahapan ini mencari dan mempelajari literatur dari artikel pada jurnal seperti penelitian-penelitian terkait atau penelitian sebelumnya tentang topik yang sama dan buku yang berkaitan dengan persoalan dan metode yang digunakan untuk pendekatan teori yang akan digunakan pada penelitian.
c. Pengumpulan Data
   Pada tahapan ini bertujuan untuk mencari dan mengumpulkan data yang digunakan untuk membandingkan kinerja model klasifikasi pada penelitian ini. data yang digunakan adalah data sekunder yang diperoleh dari NASA Software Defect Dataset yang sudah dibersihkan. Dataset tersebut memiliki ketidakseimbangan kelas antara kelas minoritas (cacat) dan kelas mayoritas (non-cacat).
d. Pra-pemrosesan Data
   Pada tahapan ini dilakukan pembersihan pada data yang kosong sebelum dilakukan pemrosesan data dan normalisasi data untuk mengubah nilai menjadi kisaran 0 sampai 1.

e. Pemrosesan Data
Tahapan ini melakukan resampling data, yakni dengan oversampling data untuk penanganan untuk ketidakseimbangan data antara kelas minoritas (cacat) dan kelas mayoritas (non-cacat) menggunakan SMOTE (Synthetic Minority Over-sampling Technique), dan ADASYN (Adaptive Synthenic Sampling).
f. Uji Skenario
Skenario pengujian diatur menjadi beberapa skenario, yakni menerapkan oversampling dan yang tidak menerapkan overampling dengan menguji performa model klasifikasi Naïve Bayes.
g. Evaluasi
Perhitungan performa model Naïve Bayes yang digunakan berfokus pada masalah ketidakseimbangan kelas. Pada penelitian ini menggunakan Confusion Matrix sebagai evaluasi dengan melihat hasil recall dari kelas minoritas.

2.1. Data Penelitian

Dataset yang digunakan dalam penelitian ini adalah dataset NASA MDP Repository. Dataset tersebut memliki ketidakseimbangan kelas antara kelas mayoritas (non-cacat) dan kelas minoritas (cacat). Dataset terdiri dari dua belas dataset dan dipilih 7 dataset diantaranya, yaitu CM1, JM1, KC1, dan PC1. Penjelasan dataset dapat dilihat dari tabel berikut.

Tabel 1. Dataset NASA MDP

Dataset	Sistem	Bahasa	Jumlah Atribut	Jumlah Instance
CM1	Instrumen pesawat ruang angkasa	C++	22	327
JM1	Sistem prediksi pendaratan secara real-time	C	22	10885
KC1	Manajemen penyimpanan data Lapangan	C++	22	2109
PC1	Perangkat lunak penerbangan satelit yang mengorbit bumi	C	38	1109

Dibawah ini merupakan contoh isi dataset CM1 :

Tabel 2. Isi Dataset CM1

Dataset	No	LOC_BLANK	BRANCH_COUNT	CALL_PAIRS	…	Defective
CM1	1	9	5	3	…	False
	2	19	3	2	…	False
	3	0	9	0	…	False
	…	…	…	…	…	…
	327	10	17	1	…	False

2.2. SMOTE (Synthetic Minority Over-sampling Technique)

SMOTE [7] berdasar bahwa ruang fitur dari instance kelas minoritas serupa. Untuk setiap x^i instance yang terdapat pada kelas minoritas, SMOTE mencari tetangga tederkatnya dengan K-Nearest Negihbors dan satu tetangga secara acak dipilih sebagai x'. Selanjutnya nomer acak δ diantara 0 dengan 1 dihasilkan. Sample baru x_{new} dibuat :

$$x_{new} = x^i + (x' - x^i) \times \delta \quad (1)$$

2.3. ADASYN (Adaptive Synthetic)

ADASYN [8] menggunakan bobot distribusi untuk instance yang terdapat pada kelas minoritas berdasarkan pada tingkat kesulitan pembelajaran model. Sample yang dihasilkan dari instance pada kelas minoritas yang sulit dipahami dibandingkan dengan instance kelas minoritas yang lebih mudah dipahami. Algoritma ADASYN adalah sebagai berikut:

a. Menentukan nilai parameter dari ADASYN, yaitu nilai dari maksimal toleran d_{th} ketidakseimbangan kelas dan nilai level keseimbangan β
b. Menghitung derajat keseimbangan

$$d = m_{minority}/m_{majority} \quad (2)$$

c. Menghitung banyaknya instance sample sintetis yang akan dibuat untuk kelas minoritas

$$G = (m_{majority} - m_{minority}) \times \beta \quad (3)$$

d. Menghitung rasio berdasarkan K-Nearest Neighbors menggunakan Euclidean distance

$$r_i = \frac{\Delta_i}{K}, i = 1, \dots, m_{minority} \quad (4)$$

e. Normalisasi r_i sehingga r'_i adalah distribusi kerapatan

$$r'_i = \frac{r_i}{\sum_{i=1}^{m_{minority}} r_i} \quad (5)$$

f. Menghitung banyaknya instance sample sintetis yang perlu dibangkitkan untuk setiap instance kelas minoritas

$$g_i = r_i \times G \quad (6)$$

g. Pembangkitan sample sintetis

$$s_i = x_i + (x_{zi} - x_i) \times \lambda \quad (7)$$

Setiap instance pada kelas minoritas x_i, dipilih satu instance secara acak x_{zi} menggunakan K-Nearest Neighbors dan λ adalah bilangan acak antara 0 dan 1.

2.4. Naïve Bayes

Naïve Bayes [9] merupakan metode dengan teknik prediksi probabilitas dengan penerapan teorema bayes yang mana diantara suatu fitur dengan fitur lain dalam satu data itu tidak saling berikatan. Metode ini merupakan salah satu bentuk sederhana untuk klasifikasi, persamaan dari Naïve Bayes dapat dilihat pada persamaan (8).

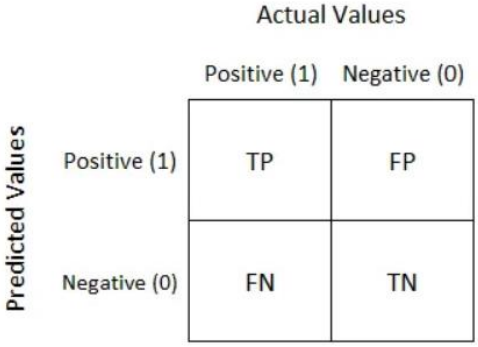
$$P_{(x|y)} = \frac{P_{(y|x)}P_{(x)}}{P_{(y)}} \quad (8)$$

Keterangan :

x = Hipotesis data y
y = Data kelas yang belum diketahui
$P_{(x|y)}$ = Probabilitas hipotesis x berdasarkan kondisi y
$P_{(x)}$ = Probabilitas hipotesis x
$P_{(y|x)}$ = Probabilitas hipotesis y berdasarkan kondisi x
$P_{(y)}$ = Probabilitas hipotesis y

2.5. Confusion Matrix

Confusion matrix adalah matrix yang menyatakan klasifikasi jumlah data uji yang benar dan jumlah data uji yang salah [10].



Gambar 1. Confusion Matrix

Keterangan:

TP (True Positive) = Jumlah instance dari kelas 1 yang benar diklasifikasikan sebagai kelas 1
TN (True Negative) = Jumlah instance dari kelas 0 yang benar diklasifikasikan sebagai kelas 0
FP (False Positive) = Jumlah instance dari kelas 0 yang salah diklasifikasikan sebagai kelas 1
FN (False Negative) = Jumlah instance dari kelas 1 yang salah diklasifikasikan sebagai kelas 0

Persamaan confussion matrix untuk menghitung accuracy, recall, precision, dan f1-score dapat dilihat pada persamaan (9) dan (10).

$$\text{accuracy} = \frac{\text{TP+TN}}{\text{Total}} \quad (9)$$

$$\text{recall} = \frac{\text{TP}}{\text{TP+FN}} \quad (10)$$

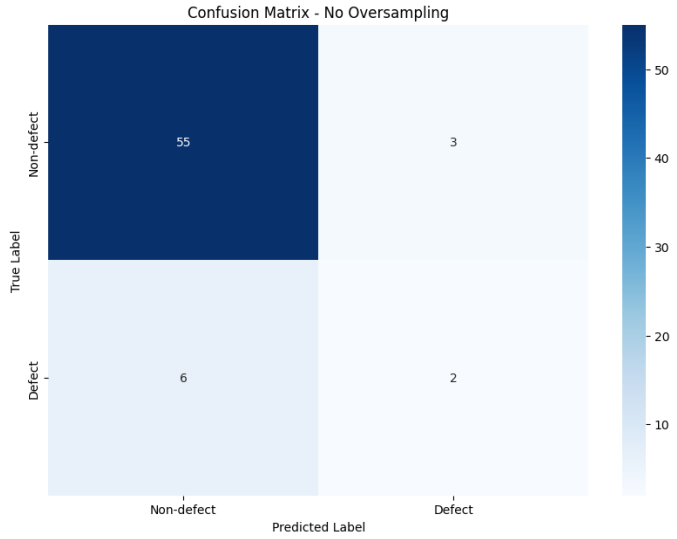
3. Hasil dan Diskusi

Penelitian ini menggunakan algoritma Naïve Bayes sebagai model klasifikasi. Dataset yang digunakan akan dibagi terlebih dahulu menjadi data latih dan data uji dengan pembagian skema 80% untuk data latih dan 20% untuk data uji. Dilakukan tiga scenario pengujian, diantaranya menggunakan model Naïve Bayes tanpa oversampling, Naïve Bayes dengan SMOTE, dan Naïve Bayes dengan ADASYN. Berikut merupakan hasil performa dari masing-masing model.

3.1. Hasil Performa Naïve Bayes

Berikut merupakan confusion matrix dari model Naïve Bayes tanpa menerapkan teknik oversampling pada salah satu dataset yang digunakan.

p-ISSN: 2986-3929
e-ISSN: 3032-1948



Gambar 2. Confusion Matrix dari Naive Bayes Tanpa Oversampling pada Dataset CM1

Dari confusion matrix diatas dapat dilihat bahwa model kurang sensitif dalam mengidentifikasi kelas minoritas atau cacat dan cenderung memberikan prediksi yang salah pada sampel yang sebenarnya cacat. Berikut merupakan tabel hasil akurasi dan hasil recall pada kelas cacat dari model pada seluruh dataset yang digunakan.

Tabel 3. Hasil Performa Model Naïve Bayes

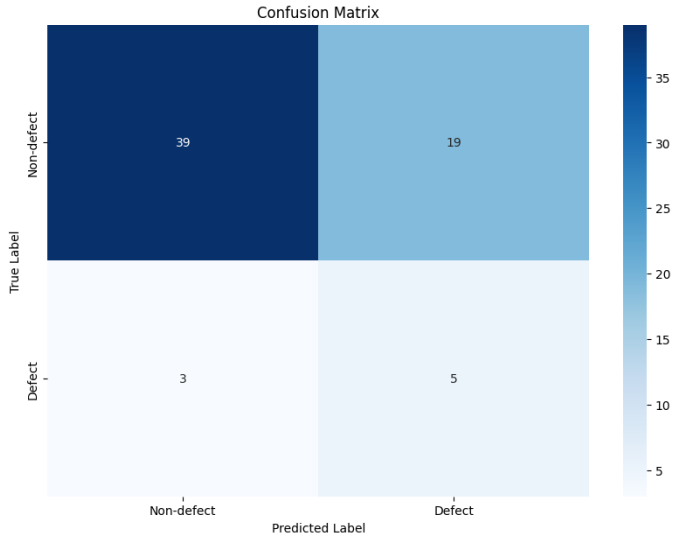
Dataset	Akurasi	Recall
CM1	0.86	0.25
JM1	0.78	0.19
KC1	0.78	0.34
PC1	0.89	0.43
Rata-Rata	0.83	0.30

Dari hasil dapat dilihat bahwa model memiliki akurasi yang tinggi dengan rata-rata 82%, akan tetapi model memiliki recall terhadap kelas minoritas yang rendah dengan rata-rata 30%.

3.2. Hasil Performa Naïve Bayes + SMOTE

Berikut merupakan confusion matrix dari model Naïve Bayes dengan menerapkan teknik oversampling SMOTE pada salah satu dataset yang digunakan.

p-ISSN: 2986-3929
e-ISSN: 3032-1948



Gambar 3. Confusion Matrix dari Naive Bayes+SMOTE pada Dataset CM1

Dari confusion matrix diatas dapat dilihat bahwa model masih kurang sensitif dalam mengidentifikasi kelas minoritas atau cacat dengan 5 sampel yang diprediksi cacat dengan benar dan 3 sampel yang seharusnya cacat tapi diprediksi tidak cacat. Berikut merupakan tabel hasil akurasi dan hasil recall pada kelas cacat dari model pada seluruh dataset yang digunakan.

Tabel 4. Hasil Performa Model Naïve Bayes+SMOTE

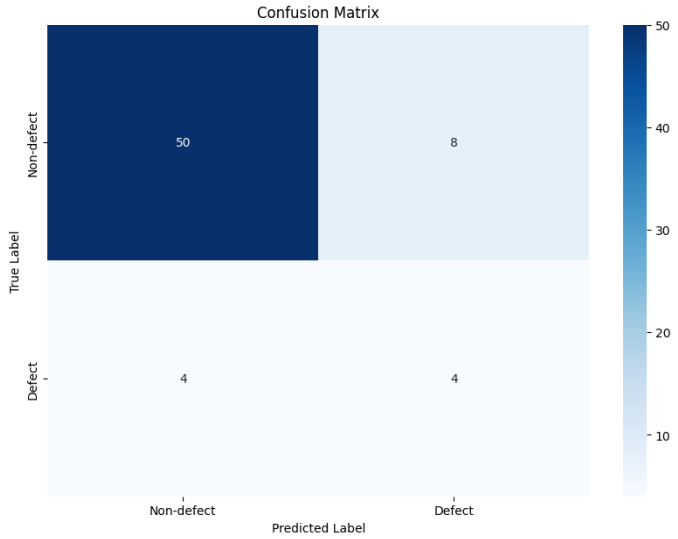
Dataset	Akurasi	Recall
CM1	0.67	0.62
JM1	0.78	0.22
KC1	0.76	0.36
PC1	0.88	0.43
Rata-Rata	0.77	0.40

Dari hasil dapat dilihat bahwa model memiliki akurasi yang tinggi dengan rata-rata 82%, akan tetapi model memiliki recall terhadap kelas minoritas yang rendah dengan rata-rata 40%.

3.3. Hasil Performa Naïve Bayes + ADASYN

Berikut merupakan confusion matrix dari model Naïve Bayes dengan menerapkan teknik oversampling ADASYN pada salah satu dataset yang digunakan.

p-ISSN: 2986-3929
e-ISSN: 3032-1948



Gambar 4. Confusion Matrix dari Naive Bayes+ADASYN pada Dataset CM1

Dari confusion matrix diatas dapat dilihat bahwa model kurang baik dalam mengidentifikasi kelas minoritas atau cacat dengan 5 sampel yang diprediksi cacat dengan benar dan 5 sampel yang seharusnya cacat tapi diprediksi tidak cacat. Berikut merupakan tabel hasil akurasi dan hasil recall pada kelas cacat dari model pada seluruh dataset yang digunakan.

Tabel 5. Hasil Performa Model Naïve Bayes+ADASYN

Dataset	Akurasi	Recall
CM1	0.82	0.50
JM1	0.78	0.24
KC1	0.75	0.38
PC1	0.88	0.43
Rata-Rata	0.80	0.38

Dari hasil dapat dilihat bahwa model memiliki akurasi yang tinggi dengan rata-rata 80%, akan tetapi model memiliki recall terhadap kelas minoritas yang rendah dengan rata-rata 38%.

4. Kesimpulan

Berdasarkan hasil yang didapat, model Naive Bayes memiliki performa yang cukup baik dalam mengklasifikasikan data cacat perangkat lunak, dengan rata-rata akurasi mencapai 83%. Namun, model ini menunjukkan kelemahan dalam mengidentifikasi kelas minoritas (cacat), dengan rata-rata recall hanya 30%. Hal ini berarti model sering kali gagal memprediksi sampel yang sebenarnya cacat. Penerapan teknik oversampling SMOTE dan ADASYN menunjukkan peningkatan terhadap recall akan tetapi menurunkan akurasi secara keseluruhan. SMOTE meningkatkan rata-rata recall kelas minoritas menjadi 40% dan akurasi rata-rata 77% sedangkan ADASYN, di sisi lain, menghasilkan rata-rata recall 38% dan akurasi rata-rata 80%. Pada penelitian selanjutnya diharapkan dapat menerapkan seleksi fitur atau ekstraksi fitur dan optimasi pada algoritma klasifikasi yang digunakan atau menerapkan deep-learning sehingga mendapatkan performa dari model yang lebih baik.

Daftar Pustaka

[1] I. Arora, V. Tetarwal, and A. Saha, “Open issues in software defect prediction,” in Procedia Computer Science, Elsevier B.V., 2015, pp. 906–912. doi: 10.1016/j.procs.2015.02.161.
[2] X. Dong, Y. Liang, S. Miyamoto, and S. Yamaguchi, “Ensemble learning based software defect prediction,” Journal of Engineering Research (Kuwait), vol. 11, no. 4, pp. 377–391, Dec. 2023, doi: 10.1016/j.jer.2023.10.038.
[3] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, “A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches,” IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, vol. 42, no. 4. pp. 463–484, Jul. 2012. doi: 10.1109/TSMCC.2011.2161285.
[4] T. M. Khoshgoftaar and K. Gao, “Feature selection with imbalanced data for software defect prediction,” in 8th International Conference on Machine Learning and Applications, ICMLA 2009, 2009, pp. 235–240. doi: 10.1109/ICMLA.2009.18.
[5] Y. Liu, W. Zhang, G. Qin, and J. Zhao, “A comparative study on the effect of data imbalance on software defect prediction,” in Procedia Computer Science, Elsevier B.V., 2022, pp. 1603–1616. doi: 10.1016/j.procs.2022.11.349.
[6] S. A. Putri and R. S. Wahono, “Integrasi SMOTE dan Information Gain pada Naive Bayes untuk Prediksi Cacat Software,” Journal of Software Engineering, vol. 1, no. 2, 2015, [Online]. Available: http://journal.ilmukomputer.org
[7] Z. Zheng, Y. Cai, Y. Li, Z. Zheng, Y. Cai, and Y. Li, “Oversampling Method For Imbalanced Classification,” 2015.
[8] H. He, Y. Bai, E. A. Garcia, and S. Li, “ADASYN: Adaptive synthetic sampling approach for imbalanced learning,” in Proceedings of the International Joint Conference on Neural Networks, 2008, pp. 1322–1328. doi: 10.1109/IJCNN.2008.4633969.
[9] M. E. Lasulika, “Komparasi Naïve Bayes, Support Vector Machine Dan K-Nearest Neighbor Untuk Mengetahui Akurasi Tertinggi Pada Prediksi Kelancaran Pembayaran Tv Kabel,” ILKOM Jurnal Ilmiah, vol. 11, no. 1, pp. 11–16, May 2019, doi: 10.33096/ilkom. v11i1.408.11-16.
[10] D. Normawati and S. A. Prayogi, “Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter,” 2021.

p-ISSN: 2986-3929
e-ISSN: 3032-1948

Halaman ini sengaja dibiarkan kosong