

Prediksi Cacat *Software* Menggunakan *Class Balancer Bagging C4.5* dan Analisis Statistik SPSS dalam Konteks Akuntansi

Nurul Ichsan^{1*}, Haerul Fatah², Tri Wahyuni³, Erni Ermawati⁴, Indriyanti⁵, Suhardi⁶

^{1,2,3,4,5}Universitas Bina Sarana Informatika

Jl. Kramat Raya No.98, Kwitang, Kec.Senen, Kota Jakarta Pusat, 10450, Indonesia

email korespondensi: nurul.nrc@bsi.ac.id (*)

Artikel Info : Diterima : 10-04-2025 | Direvisi : 10-05-2025 | Disetujui : 10-06-2025

Abstrak Dalam dunia pengembangan perangkat lunak, prediksi terhadap kemungkinan cacat (defect) bukan sekadar kebutuhan teknis, melainkan menjadi pilar penting dalam menjamin akurasi sistem informasi, khususnya di bidang akuntansi yang menuntut presisi tinggi. Namun, tantangan besar hadir dalam bentuk ketidakseimbangan kelas data (class imbalance), yang kerap melemahkan kemampuan model dalam melakukan prediksi secara andal. Penelitian ini menawarkan solusi melalui pendekatan inovatif yang menggabungkan teknik Class Balancer, metode ensemble Bagging, dan algoritma C4.5. Hasil pengujian pada dataset PC4.arff menunjukkan performa yang menjanjikan: nilai AUC meningkat menjadi 0,834 dengan akurasi mencapai 83,35%, masuk dalam kategori *Good Classification*. Lebih lanjut, peningkatan rata-rata AUC dari 0,599 menjadi 0,672 menandai transisi performa dari *Failure* ke *Poor Classification* yang lebih baik. Uji statistik T-Test dan Wilcoxon memperkuat temuan ini, dengan nilai signifikansi masing-masing sebesar 0,014 dan 0,003, menandakan perbedaan yang signifikan antara model C4.5 murni dan model kombinasi yang diusulkan. Dengan demikian, pendekatan Class Balancer + Bagging + C4.5 tidak hanya menjawab tantangan ketidakseimbangan data, tetapi juga membuka peluang baru dalam meningkatkan keandalan sistem prediktif dalam konteks akuntansi digital.

Kata Kunci: Prediksi Cacat Software, Class Balancer, C4.5 dan Bagging, Analisis Statistik SPSS

Abstract In the world of software development, prediction of possible defects is not just a technical requirement, but also an important pillar in ensuring the accuracy of information systems, especially in the field of accounting that demands high precision. However, a major challenge comes in the form of class imbalance data, which often increases the model's ability to make reliable predictions. This study offers a solution through an innovative approach that combines the Class Balancer technique, the Bagging ensemble method, and the C4.5 algorithm. The test results on the PC4.arff dataset show promising performance: the AUC value increases to 0.834 with an accuracy of 83.35%, falling into the *Good Classification* category. Furthermore, the increase in the average AUC from 0.599 to 0.672 marks a transformation in performance from *Failure* to *Poor Classification* which is better. The T-Test and Wilcoxon statistical tests strengthen this finding, with significance values of 0.014 and 0.003, respectively, indicating a significant difference between the pure C4.5 model and the proposed combination model. Thus, the Class Balancer + Bagging + C4.5 approach not only addresses the challenges of data performance, but also opens up new opportunities in improving the performance of predictive systems in the context of digital accounting.

Keywords: Defect Prediction Software, Class Balancer, C4.5 and Bagging, SPSS Statistical Analysis

PENDAHULUAN

Dalam dinamika pengembangan perangkat lunak, prediksi terhadap kemungkinan cacat bukan sekadar tahapan teknis semata, melainkan elemen strategis dalam rangkaian Software Development Life Cycle (SDLC). Tahap pengujian ini memiliki peran vital dalam menilai integritas dan reliabilitas perangkat lunak, sekaligus menjadi garda terdepan dalam mengidentifikasi adanya ketidaksempurnaan sistem. Tanpa adanya proses pengujian yang sistematis dan menyeluruh, mustahil untuk menjamin lahirnya perangkat lunak yang berkualitas dan sesuai dengan standar yang diharapkan. (Ichsan et al., 2023). Prediksi terhadap cacat perangkat lunak telah menjadi salah satu topik utama yang banyak dikaji dalam bidang rekayasa perangkat lunak. Hal ini tidak lepas dari urgensi untuk menyelesaikan berbagai permasalahan terkait ketidakefisienan dan ketidakefektifan dalam proses pengujian serta peninjauan perangkat lunak (Wahono et al., 2014). Mengatasi cacat perangkat lunak setelah sistem dikirimkan ke pengguna terbukti memerlukan biaya yang jauh lebih besar dibandingkan jika dilakukan selama tahap pengembangan (Saifudin & Wahono, 2015). Bahkan, sebagian besar anggaran dalam proses pengembangan perangkat lunak terserap untuk mendeteksi dan memperbaiki cacat (Wahono & Suryana, 2013). Diperkirakan, anggaran tahunan yang dihabiskan untuk memperbaiki perangkat lunak yang mengandung cacat dapat mencapai angka fantastis hingga 60 miliar rupiah (Fitriyani & Wahono, 2015). Oleh karena itu, metode prediksi cacat



perangkat lunak menjadi sangat relevan, khususnya dalam membantu menentukan prioritas modul yang perlu diuji lebih dahulu, demi efisiensi biaya serta peningkatan kualitas produk perangkat lunak (Ichsan et al., 2022).

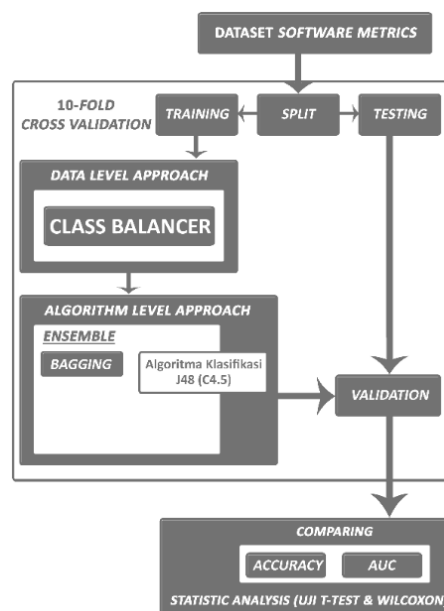
Dalam perkembangannya, sekitar 77,46% dari studi yang mengangkat tema ini memanfaatkan metode klasifikasi, sementara 14,08% menggunakan pendekatan estimasi, dan hanya 1,41% yang memanfaatkan teknik pengelompokan maupun asosiasi. Selain itu, data yang digunakan dalam penelitian tersebut sebagian besar berasal dari sumber publik (64,79%), sementara sisanya menggunakan data privat (35,21%) (Wahono, 2015). Salah satu tantangan mendasar yang sering dihadapi dalam pemrosesan dataset metrik perangkat lunak adalah fenomena ketidakseimbangan kelas (*class imbalance*), di mana jumlah data minoritas—yakni modul yang mengandung cacat—sangat sedikit dibandingkan dengan data mayoritas yang bebas dari cacat. Ketidakseimbangan ini menjadi penghambat utama dalam pengembangan model klasifikasi yang andal karena dapat menurunkan akurasi dan mengaburkan kemampuan model dalam mengenali cacat yang jarang terjadi (Ichsan, 2019). Salah satu dataset yang sering digunakan untuk penelitian ini adalah NASA MDP, yang dikenal memiliki skala besar serta dimensi tinggi (Putri, 2017).

Untuk mengatasi ketidakseimbangan kelas tersebut, pendekatan dapat dilakukan pada dua level: data dan algoritma. Pada level data, strategi seperti *Class Balancer* digunakan untuk menyamakan distribusi antara kelas minoritas dan mayoritas sebelum proses pelatihan dimulai (Aries & Wahono, 2015). Teknik ini bertujuan agar model memiliki persepsi yang proporsional terhadap kedua kelas, sehingga menghasilkan generalisasi yang lebih baik, terutama terhadap kelas minoritas. Sementara itu, pada level algoritma, metode *ensemble* seperti *Bagging* (*Bootstrap Aggregating*) diterapkan guna memperkuat performa model. *Bagging* bekerja dengan menggabungkan beberapa model klasifikasi untuk membentuk satu model akhir yang lebih stabil dan akurat. Dalam konteks ini, algoritma *decision tree* C4.5 dipilih sebagai *base classifier*, yang kemudian diperkuat melalui teknik *bagging* guna mengurangi variansi dan meningkatkan akurasi prediksi, khususnya terhadap data minoritas yang sebelumnya kurang teridentifikasi.

Eksperimen dalam penelitian ini dilakukan menggunakan perangkat lunak WEKA, yang merupakan platform *open-source* populer dalam bidang *data mining* dan *machine learning*. WEKA menyediakan beragam fitur untuk kebutuhan *pre-processing* data, pelatihan model, dan evaluasi performa secara menyeluruh. Dengan mengintegrasikan teknik *Class Balancer* pada level data dan metode *Bagging* berbasis C4.5 pada level algoritma, penelitian ini bertujuan untuk mengembangkan model prediksi cacat perangkat lunak yang tidak hanya akurat, tetapi juga efisien. Solusi yang ditawarkan diharapkan mampu mengatasi kendala ketidakseimbangan kelas sekaligus meningkatkan kinerja prediksi dalam penerapan nyata di dunia pengembangan perangkat lunak.

METODE PENELITIAN

Penelitian ini menerapkan pendekatan kuantitatif sebagai landasan analisis, dengan memanfaatkan data sekunder sebagai sumber utama dalam pengumpulan informasi. Metodologi yang digunakan mengacu pada eksperimen ilmiah, yang mencakup proses pengujian terhadap model yang diusulkan, serta tahapan evaluasi dan validasi guna menilai efektivitasnya. Dalam upaya menjawab permasalahan ketidakseimbangan kelas pada dataset metrik perangkat lunak, penulis merancang sebuah model yang dikembangkan secara khusus dan disajikan dalam penelitian ini sebagai solusi alternatif yang potensial.

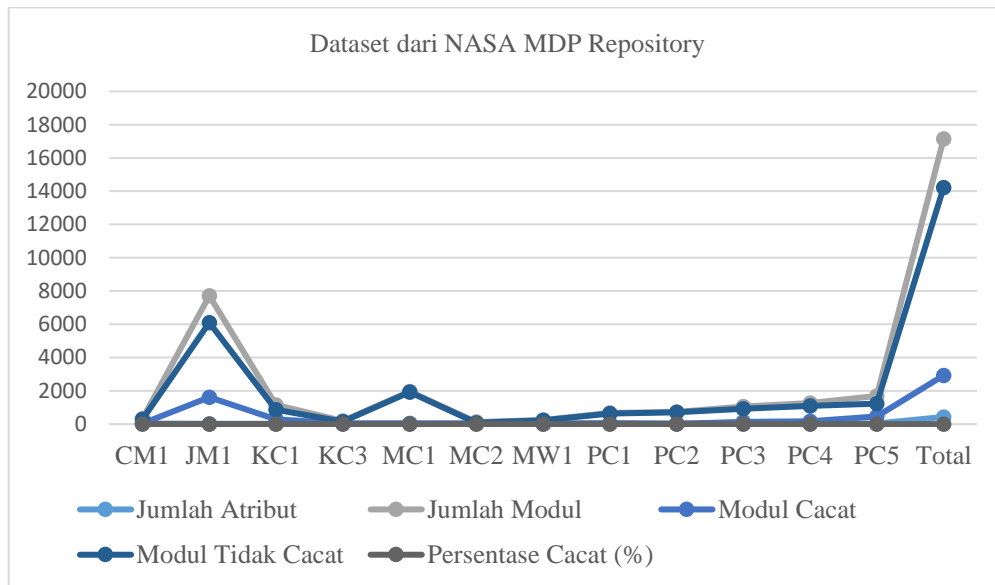


Gambar 1. Kerangka Konseptual Model yang diusulkan

HASIL DAN PEMBAHASAN

Penelitian ini mengadopsi pendekatan eksperimental dengan mengacu pada kerangka kerja Cross-Industry Standard Process for Data Mining (CRISP-DM) sebagai landasan metodologis, sebagaimana telah dijelaskan dalam bagian sebelumnya. Tujuan utama dari implementasi eksperimen ini adalah untuk merumuskan strategi yang efektif dalam mengatasi permasalahan *class imbalance* pada model prediksi cacat perangkat lunak. Untuk mendukung proses ini, digunakan perangkat lunak Waikato Environment for Knowledge Analysis (WEKA), sebuah platform *open-source* berbasis Java yang telah terbukti luas penggunaannya dalam berbagai studi *data mining*.

Eksperimen dilakukan dengan memanfaatkan 12 dataset yang bersumber dari NASA Metrics Data Program (MDP) Repository, yang dapat diakses melalui tautan berikut: <https://github.com/klainfo/NASADefectDataset>. Tabel 1 menyajikan detail komposisi data yang digunakan dalam penelitian ini.



Gambar 1. Jumlah Dataset dari NASA MDP Repository

Berdasarkan hasil analisis terhadap tabel persentase modul cacat yang telah disajikan, Analisis awal terhadap proporsi data cacat dalam masing-masing dari 12 dataset menunjukkan adanya disparitas yang cukup mencolok, yang mengindikasikan bahwa tantangan ketidakseimbangan kelas merupakan aspek penting yang harus ditangani secara strategis dalam upaya menghasilkan model prediksi yang akurat dan dapat diandalkan. Dengan kisaran persentase mulai dari 1,84% hingga mencapai 35,48%. Variasi ini mencerminkan tingkat ketidakseimbangan yang beragam dalam masing-masing dataset, yang tentunya berdampak pada kompleksitas proses klasifikasi. Penelitian ini mengusulkan pendekatan pengujian inovatif dengan mengintegrasikan dua strategi penanganan ketidakseimbangan data: pendekatan pada level data melalui metode *Class Balancer*, serta pendekatan pada level algoritma menggunakan metode *ensemble Bagging* yang berbasis *classifier C4.5*.

Pengujian Classifier C4.5

Sebagai tahap awal sebelum implementasi berbagai skenario pengujian, penelitian ini terlebih dahulu melakukan evaluasi terhadap kinerja asli dari classifier C4.5 dalam bentuknya yang murni, tanpa adanya integrasi dengan pendekatan atau metode tambahan. Langkah ini bertujuan untuk memperoleh gambaran awal mengenai performa dasar dari model tersebut, sehingga memungkinkan adanya perbandingan yang objektif dan terukur terhadap hasil yang diperoleh setelah integrasi dengan model-model pengujian yang diusulkan. Perbedaan kinerja sebelum dan sesudah integrasi dapat diidentifikasi secara jelas, baik dari sisi efektivitas klasifikasi maupun kontribusinya terhadap peningkatan akurasi dan nilai AUC.

Tabel 1. Hasil Pengujian Menggunakan Algoritma C4.5

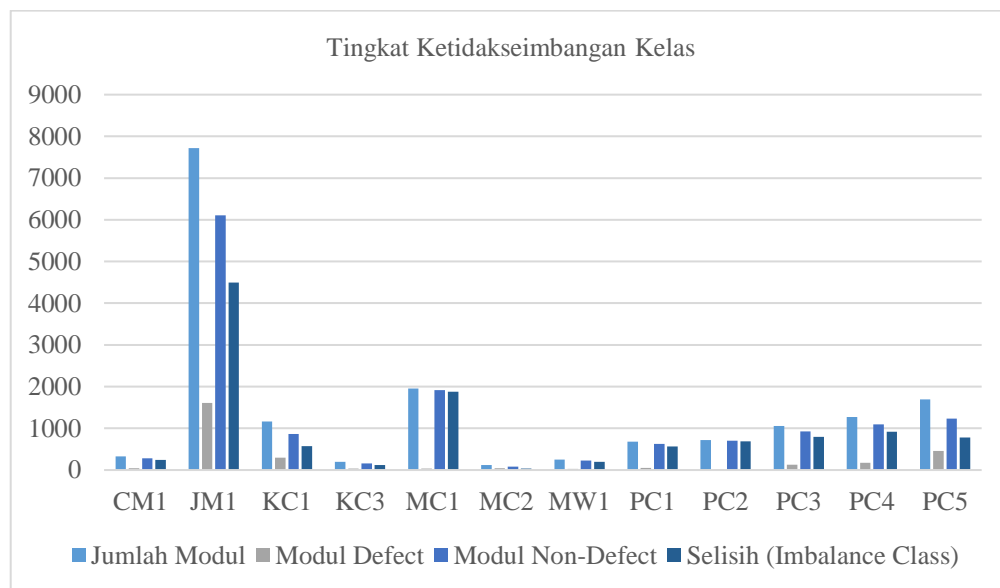
Dataset	Akurasi (%)	AUC	ROC Area
CM1	81.04	0.536	0.570
JM1	77.53	0.582	0.653
KC1	74.18	0.591	0.604
KC3	79.38	0.616	0.653

MC1	97.90	0.540	0.566
MC2	60.48	0.576	0.589
MW1	90.40	0.662	0.503
PC1	91.46	0.597	0.598
PC2	97.51	0.499	0.463
PC3	84.71	0.599	0.591
PC4	86.93	0.726	0.789
PC5	73.91	0.662	0.673
Rata-rata	82.95	0.599	0.604

Secara umum, dari model klasifikasi menunjukkan hasil yang kurang memuaskan. Rata-rata akurasi yang tercapai hanya sebesar 82,95%, dengan nilai AUC sebesar 0,599 dan ROC Area sebesar 0,604. Capaian ini mengindikasikan bahwa model berada pada kategori *Failure Classification* dalam tingkat kualitas klasifikasi. Sebagai standar evaluasi kinerja model klasifikasi, maka capaian ini belum memenuhi kriteria yang dapat dianggap optimal atau andal dalam konteks pemodelan prediktif.

Berdasarkan hasil pengujian awal terhadap algoritma klasifikasi C4.5, performa model masih tergolong rendah dan belum mencapai tingkat klasifikasi yang diharapkan. Salah satu penyebab utama dari hasil ini adalah adanya ketidakseimbangan kelas (*class imbalance*) yang cukup signifikan pada seluruh dataset yang digunakan. Ketimpangan tersebut ditandai oleh dominasi data dari kelas mayoritas (*non-defect*) yang jauh lebih banyak dibandingkan kelas minoritas (*defect*), sehingga mempengaruhi efektivitas model dalam mengenali dan memprediksi modul yang benar-benar mengalami cacat.

Dalam konteks pembelajaran mesin, kondisi seperti ini merupakan tantangan klasik yang dapat menurunkan performa model, terutama pada metrik penting seperti akurasi, AUC, dan kemampuan generalisasi. Ketidakseimbangan ini perlu diperhatikan secara serius, karena berdampak langsung pada keandalan hasil klasifikasi. Untuk memberikan gambaran yang lebih rinci, berikut disajikan data tingkat ketidakseimbangan kelas pada 12 dataset yang diambil dari NASA Repository:



Gambar 2. Tingkat Ketidakseimbangan Kelas

Pengujian Model yang diusulkan

Eksperimen atau pengujian proses penghitungan terhadap model yang diusulkan dalam penelitian ini menggunakan model integrasi data dan *algorithm approach* menggunakan *classifier C4.5*. berikut adalah langkah-langkah pengujiannya menggunakan *tools weka* 3.8 :

1. Buka dan jalankan aplikasi WEKA, maka akan muncul tampilan awal pada aplikasi weka.
2. Klik tombol *Explorer* untuk membuka antarmuka WEKA Explorer
3. Klik *Button Open File* kemudian pilih dataset → klik *open*
4. Klik *Button Choose* pada area *Filter* → klik tombol panah *filters* → pilih *supervised* → pilih *instance* → pilih *Class Balancer* → klik *button apply*
5. Selanjutnya klik *button classify* → pilih *button choose* pada *classifier* → pilih *meta* → pilih *Bagging* → pastikan *test options* berada pada pilihan *cross validation* dengan *folds=10* → klik pada *textbox* pilihan

bagging yang berwarna putih → pilih *choose* pada *classifier* → pilih *tress* → pilih *C4.5* → klik OK → klik *button start*

Sebelum memaparkan hasil pengujian, Proses evaluasi dilakukan pada dua kondisi data: sebelum dan sesudah dilakukan tahap *preprocessing*. Data awal (original) menunjukkan ketimpangan signifikan antara kelas mayoritas dan minoritas. Setelah diterapkan teknik *class balancing*, distribusi data menjadi lebih seimbang, yang secara signifikan membantu meningkatkan proporsi kelas minoritas. Berikut ini adalah hasil distribusi data sebelum dan sesudah dilakukan proses *class balancing*:

Tabel 2. Distribusi Data Sebelum dan Sesudah Preprocessing

Dataset	Original Modul	Defect	Non-Defect	After Class Balancing Modul	Defect	Non-Defect
CM1	327	42	285	327	163.5	163.5
JM1	7720	1612	6108	7720	3860	3860
KC1	1162	294	868	1162	581	581
KC3	194	36	158	194	97	97
MC1	1952	36	1916	1952	976	976
MC2	124	44	80	124	62	62
MW1	250	25	225	250	125	125
PC1	679	55	624	679	339.5	339.5
PC2	722	16	706	722	361	361
PC3	1053	130	923	1053	526.5	526.5
PC4	1270	176	1094	1270	635	635
PC5	1694	458	1236	1694	847	847

Penerapan class balancer mempertahankan jumlah modul yang sama, namun berhasil mengatur ulang distribusi kelas dengan meningkatkan jumlah data pada kelas minoritas dan menurunkan proporsi kelas mayoritas. Hasil akhir menunjukkan bahwa distribusi antara dua kelas menjadi lebih seimbang. Tahap selanjutnya adalah menguji performa algoritma Bagging berbasis C4.5 dengan mengombinasikan teknik class balancing untuk mengetahui peningkatan akurasi dan AUC dibanding model awal. Berikut ini disajikan hasil pengujian model Class Balancer + Bagging + C4.5 pada 12 dataset:

Tabel 3. Performa Model

Kinerja	CM1	JM1	KC1	KC3	MC1	MC2	MW1	PC1	PC2	PC3	PC4	PC5
Akurasi (%)	58.84	63.42	64.40	72.33	60.41	67.95	65.78	72.76	57.68	69.55	83.35	69.74
TP Rate (%)	31.10	46.46	50.00	61.11	22.22	65.90	44.00	52.73	18.75	52.31	78.41	60.92
TN Rate (%)	86.59	80.37	78.80	83.55	98.59	70.00	87.55	92.79	96.60	86.78	88.30	78.56
FP Rate (%)	13.41	19.63	21.20	16.45	1.41	30.00	12.45	7.21	3.40	13.22	11.70	21.44
FN Rate (%)	68.90	53.54	50.00	38.89	77.78	34.10	56.00	47.27	81.25	47.69	21.59	39.08
Precision (%)	69.86	70.30	70.23	78.79	94.04	68.72	77.95	87.97	84.65	79.83	87.02	73.97
NPV (%)	55.69	60.02	61.18	68.24	55.90	67.25	60.99	66.25	54.32	64.53	80.35	66.78
F-Measure	0.430	0.559	0.584	0.688	0.359	0.673	0.562	0.659	0.307	0.632	0.825	0.668
G-Mean	0.519	0.611	0.628	0.715	0.468	0.679	0.621	0.699	0.426	0.674	0.832	0.692
AUC	0.588	0.634	0.644	0.723	0.604	0.680	0.658	0.728	0.577	0.695	0.834	0.697
ROC Area	0.703	0.692	0.709	0.767	0.818	0.706	0.725	0.858	0.625	0.812	0.913	0.792

Hasil terbaik diperoleh pada dataset **PC4.arff**, dengan nilai akurasi sebesar **83.35%** dan nilai AUC mencapai **0.834**. Berdasarkan kategori klasifikasi AUC, nilai tersebut berada pada rentang **0.80–0.90**, yang termasuk dalam tingkat **“Good Classification”**. Hal ini menunjukkan bahwa penerapan strategi kombinasi *Class Balancer + Bagging + C4.5* dapat secara efektif mengatasi masalah *class imbalance* dan meningkatkan kemampuan prediksi terhadap modul yang mengalami cacat. Hasil kinerja model yang didapat lebih baik dibandingkan model C4.5 original meskipun peningkatan yang didapat tidak terlalu signifikan. Pada model *Class Balancer+Bagging+C4.5* terdapat peningkatan dari nilai variable AUC yang dicari yaitu mengalami peningkatan sebesar 0.073 tetapi mengalami penurunan dari hasil variable akurasinya sebesar 15.77%. Rata-rata akurasi yang didapat yaitu 67.18% tidak lebih baik dari model C4.5 dengan rata-rata akurasi 82.95%, sedangkan untuk nilai arata-rata AUC berada pada nilai 0.672 lebih baik dari model C4.5 yang mendapat nilai rata-rata AUC 0.599. Tingkatan level kinerja *classifier* pun meningkat dari level *Failure Classification* menjadi *Poor Classification* setelah dilakukannya model integrasi dengan *Class Balancer+Bagging+C4.5*.

Studi ini turut menyajikan analisis perbandingan terhadap performa model klasifikasi yang memanfaatkan teknik praproses dengan menerapkan metode Class Balancer. Dua skenario konfigurasi model yang dievaluasi mencakup gabungan dari Class Balancer, Bagging, dan algoritma C4.5, serta kombinasi Class Balancer, Bagging, dan algoritma Naïve Bayes. Berdasarkan hasil evaluasi, meskipun performa kedua model tergolong setara, model yang memanfaatkan C4.5 secara konsisten menunjukkan hasil yang lebih unggul.

Tabel 4. Perbandingan Performa Model

Dataset	CB+BG+C4.5 Akurasi	CB+BG+C4.5 AUC	Kinerja	CB+BG+NB Akurasi	CB+BG+NB AUC	Kinerja	Selisih Akurasi	Selisih AUC	Model Terbaik
CM1	58.84%	0.588	Failure	58.54%	0.585	Failure	0.30%	0.003	-
JM1	63.42%	0.634	Poor	57.12%	0.571	Failure	6.30%	0.063	C4.5
KC1	64.40%	0.644	Poor	60.43%	0.604	Poor	3.97%	0.040	-
KC3	72.33%	0.723	Fair	62.80%	0.628	Poor	9.53%	0.095	C4.5
MC1	60.41%	0.604	Poor	64.30%	0.643	Poor	-3.89%	-	-
MC2	67.95%	0.680	Poor	62.05%	0.620	Poor	5.90%	0.060	-
MW1	65.78%	0.658	Poor	70.89%	0.709	Fair	-5.11%	-	Naïve Bayes
PC1	72.76%	0.728	Fair	66.03%	0.660	Poor	6.73%	0.068	C4.5
PC2	57.68%	0.577	Failure	57.05%	0.570	Failure	0.63%	0.007	-
PC3	69.55%	0.695	Poor	63.46%	0.635	Poor	6.09%	0.060	-
PC4	83.35%	0.834	Good	65.01%	0.650	Poor	18.34%	0.184	C4.5
PC5	69.74%	0.697	Poor	58.58%	0.586	Failure	11.16%	0.111	C4.5
Rata-rata	67.18%	0.672	Poor	62.19%	0.622	Poor	5.00%	0.050	C4.5

Hasil pengujian menunjukkan bahwa kombinasi model *Class Balancer + Bagging + C4.5* menghasilkan rata-rata akurasi sebesar 67,18% dan rata-rata nilai AUC (Area Under Curve) sebesar 0,672. Di sisi lain, model perbandingan yang menggunakan *Naïve Bayes* memiliki performa yang sedikit lebih rendah. Meskipun kedua model termasuk dalam kategori klasifikasi yang sama, yakni *Poor Classification*, temuan ini menunjukkan bahwa penerapan algoritma C4.5 yang digabungkan dengan metode balancing dan teknik ensemble mampu memberikan hasil yang lebih unggul, terutama saat diaplikasikan pada data dengan distribusi kelas yang tidak seimbang.

Pengujian T-Test (Independent Sample T-Test)

Hipotesis nol (H₀): Tidak terdapat perbedaan rata-rata nilai AUC antara model C4.5 dengan model gabungan *Class Balancer + Bagging + C4.5*.

Hipotesis alternatif (H₁): Terdapat perbedaan rata-rata nilai AUC antara model C4.5 dengan model *Class Balancer + Bagging + C4.5*.

Tabel 5. Hasil Uji T antara Model C4.5 dan Model Class Balancer + Bagging + C4.5

Nama Model	Jumlah Data (N)	Rata-rata (Mean)	Simpangan Baku (Std. Deviation)	Galat Baku (Std. Error Mean)
Model C4.5	12	0.5988	0.06230	0.01798
Model Class Balancer + Bagging + C4.5	12	0.6718	0.07156	0.02066

Independent Samples Test										
		Levene's Test for Equality of Variances		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
Hasil Pengujian Model	Equal variances assumed	.285	.599	-2.665	22	.014	-.07300	.02739	-.12980	-.01620
	Equal variances not assumed			-2.665	21.591	.014	-.07300	.02739	-.12986	-.01614

Nilai rata-rata AUC dari model Class Balancer + Bagging + C4.5 tercatat lebih tinggi, yaitu 0,6718, dibandingkan dengan model C4.5 murni. Dalam pengujian perbedaan statistik, digunakan tingkat signifikansi (alpha) sebesar 0,05. Jika nilai p lebih kecil dari alpha (p < 0,05), maka hipotesis nol (H₀) ditolak dan hipotesis alternatif (H₁) diterima, yang berarti terdapat perbedaan yang signifikan antara kedua model. Sebaliknya, jika nilai p melebihi alpha (p > 0,05), maka H₀ diterima dan H₁ ditolak, menunjukkan tidak adanya perbedaan yang berarti.

Hasil pengujian menunjukkan bahwa nilai Sig. (2-tailed) sebesar 0,014, yang lebih kecil dari alpha (0,014 < 0,05). Oleh karena itu, H₀ ditolak dan H₁ diterima, yang mengindikasikan bahwa terdapat perbedaan yang

signifikan antara model C4.5 dan model Class Balancer + Bagging + C4.5. Temuan ini menunjukkan bahwa penerapan teknik balancing dan ensemble pada algoritma C4.5 memberikan peningkatan performa yang signifikan dibandingkan model C4.5 tanpa kombinasi tersebut.

Uji Wilcoxon

H₀: Tidak terdapat perbedaan nilai rata-rata AUC antara model C4.5 dan model Class Balancer + Bagging + C4.5.

H₁: Terdapat perbedaan nilai rata-rata AUC antara model C4.5 dan model Class Balancer + Bagging + C4.5

Tabel 6. Hasil Uji Wilcoxon untuk Model C4.5 dan Model Class Balancer + Bagging + C4.5

		Ranks		
		N	Mean Rank	Sum of Ranks
CB_BG_C45 - C45	Negative Ranks	1 ^a	1.00	1.00
	Positive Ranks	11 ^b	7.00	77.00
	Ties	0 ^c		
	Total	12		

a. CB_BG_C45 < C45

b. CB_BG_C45 > C45

c. CB_BG_C45 = C45

Test Statistics^a

	CB_BG_C45 - C45
Z	-2.982 ^b
Asymp. Sig. (2-tailed)	.003

a. Wilcoxon Signed Ranks Test

b. Based on negative ranks.

Berdasarkan hasil dari tabel "Test Statistic", nilai Asymp. Sig. (2-tailed) tercatat sebesar 0,003. Karena nilai tersebut lebih kecil dari 0,05, maka dapat disimpulkan bahwa hipotesis nol (H₀) ditolak dan hipotesis alternatif (H₁) diterima. Dengan diterimanya H₁, hal ini menunjukkan bahwa terdapat perbedaan yang signifikan antara model C4.5 murni dengan model Class Balancer + Bagging + C4.5, yang berarti model gabungan ini memberikan peningkatan performa klasifikasi secara signifikan dibandingkan model awal.

KESIMPULAN

Ketika ketepatan prediksi menjadi kunci dalam menjamin kualitas perangkat lunak, terutama dalam sistem akuntansi yang sarat akan tuntutan akurasi, maka mengatasi ketidakseimbangan kelas data bukan lagi sebuah opsi melainkan sebuah keharusan. Penelitian ini menunjukkan bahwa kombinasi antara Class Balancer, teknik Bagging, dan algoritma C4.5 mampu menjadi jawaban atas persoalan tersebut. Meski akurasi rata-rata model sedikit menurun, lonjakan nilai AUC yang signifikan menjadi bukti bahwa kualitas klasifikasi mengalami peningkatan nyata. Lebih penting lagi, hasil uji statistik dengan nilai signifikansi di bawah 0,05 (T-Test = 0,014; Wilcoxon = 0,003) memberikan landasan ilmiah yang kuat bahwa perbedaan performa antara model C4.5 murni dan model gabungan bukanlah kebetulan, melainkan hasil dari pendekatan yang terstruktur dan teruji. Dengan kata lain, model Class Balancer + Bagging + C4.5 bukan hanya menawarkan pendekatan teknis, tetapi juga solusi strategis dalam membangun sistem prediksi cacat software yang lebih tangguh dan dapat diandalkan, terlebih dalam ranah akuntansi digital yang semakin kompleks dan dinamis.

REFERENSI

- Aries, S., & Wahono, R. S. (2015). Pendekatan Level Data untuk Menangani Ketidakseimbangan Kelas pada Prediksi Cacat Software. *Journal of Software Engineering*, 1(2), 76–85. [https://doi.org/10.1016/S1896-1126\(14\)00030-3](https://doi.org/10.1016/S1896-1126(14)00030-3)
- Fitriyani, & Wahono, R. S. (2015). Integrasi Bagging dan Greedy Forward Selection pada Prediksi Cacat Software dengan Menggunakan Naïve Bayes. *Journal of Software Engineering*, 1(2), 101–108.
- Ichsan, N. (2019). Metoda Distribution Based Balance dan Bagging C4 . 5. *Indonesian Journal on Computer and Information Technology*, 4(2), 215–224.
- Ichsan, N., Fatah, H., Ermawati, E., Indriyanti, I., & Wahyuni, T. (2022). Integrasi Distribution Based Balance dan Teknik Ensemble Bagging Naive Bayes Untuk Prediksi Cacat Software. *Media Jurnal Informatika*, 14(2), 79. <https://doi.org/10.35194/mji.v14i2.2623>
- Ichsan, N., Sopandi, R., Priyandaru, H., & Tabrani, M. (2023). Pendekatan Level Data Smote Pada Algoritma Bagging C4.5 Untuk Prediksi Cacat Software Smote Data Level Approach of C4.5 Bagging Algorithm for Software Defect Prediction. *CERMIN:JURNAL PENELITIAN*, 7, 402–416.
- Putri, S. A. (2017). Integrasi Teknik Smote Bagging Dengan Information. *Jurnal Ilmu Pengetahuan Dan*

Teknologi Komputer, 2(2), 22–31.

- Saifudin, A., & Wahono, R. S. (2015). Penerapan Teknik Ensemble untuk Menangani Ketidakseimbangan Kelas pada Prediksi Cacat Software. *Journal of Software Engineering*, 1(1).
- Wahono, R. S. (2015). A Systematic Literature Review of Software Defect Prediction : Research Trends , Datasets , Methods and Frameworks. *Journal of Software Engineering*, 1(1).
- Wahono, R. S., & Suryana, N. (2013). Combining particle swarm optimization based feature selection and bagging technique for software defect prediction. *International Journal of Software Engineering and Its Applications*, 7(5), 153–166. <https://doi.org/10.14257/ijseia.2013.7.5.16>
- Wahono, R. S., Suryana, N., & Ahmad, S. (2014). Metaheuristic Optimization based Feature Selection for Software Defect Prediction. *Journal of Software*, 9(5), 1324–1333. <https://doi.org/10.4304/jsw.9.5.1324-1333>