Journal of Information Technology and Computer Science (INTECOMS)
Volume 7 Nomor 4, Tahun 2024
e-ISSN : 2614-1574
p-ISSN : 2621-3249

PREDIKSI CACAT PERANGKAT LUNAK KELAS TIDAK SEIMBANG MENGGUNAKAN RESAMPLE J48 DAN J48 CONSOLIDATED

PREDICTION OF UNBALANCED CLASS SOFTWARE DEFECTS USING RESAMPLE J48 AND J48 CONSOLIDATED

Ilham Nurjabar[1], Lindung Parningotan Manik[2]
[1,2]Program Studi Ilmu Komputer Universitas Nusa Mandiri
14002451@nusamandiri.ac.id[1], lindung.lpm@nusamandiri.ac.id[2]

ABSTRACT

Software defect prediction is a crucial aspect of software quality assurance, aiming to identify and address potential defects before they manifest in a production environment. This research presents an innovative approach to tackle the issue of imbalanced class distribution in software defect prediction using resampling techniques and the J48 and J48 Consolidated algorithms. Additionally, this study introduces a new variant of J48, referred to as J48 Consolidated, which combines multiple decision trees into a single ensemble model to enhance prediction performance. The J48 Consolidated model is compared to the traditional J48 algorithm in the context of software defect prediction with imbalanced class distribution. The dataset used in this research is sourced from the PROMISE repository. The research findings indicate that the integration of the RUS + J48 Consolidated algorithm is suitable for predicting software defects with an average accuracy of 78% and an AUC value of 0.783. This study evaluates the performance of J48 with ROS and RUS approaches using the J48 and J48 Consolidated algorithms. The research results show that the RUS + J48 Consolidated model outperforms the RUS + J48 model with average accuracies of 78% and 77%, along with AUC values of 0.783 and 0.766, respectively.

Keywords*: Resample, J48, J48 Consolidated, Software Defect, Imbalanced Class, Random Over Sampling, Random Under Sampling.*

ABSTRAK

Prediksi cacat perangkat lunak merupakan aspek penting dalam jaminan kualitas perangkat lunak, dengan tujuan mengidentifikasi dan mengatasi potensi cacat sebelum mereka muncul dalam lingkungan produksi. Penelitian ini menyajikan pendekatan inovatif untuk mengatasi masalah distribusi kelas yang tidak seimbang dalam prediksi cacat perangkat lunak menggunakan teknik resampling dan algoritma J48 dan J48 Consolidated. Selain itu, penelitian ini memperkenalkan varian baru dari J48, yang disebut sebagai J48 Consolidated, yang menggabungkan beberapa pohon keputusan menjadi satu model ensemble tunggal untuk meningkatkan kinerja prediksi. Model J48 Consolidated dibandingkan dengan algoritma J48 tradisional dalam konteks prediksi cacat perangkat lunak dengan distribusi kelas yang tidak seimbang. Dataset yang yang digunakan pada penelitian ini menggunakan dataset PROMISE repository. Hasil penelitian menunjukan bahwa integrase Algoritma RUS + J48 Consolidated layak digunakan untuk memprediksi cacat software dengan rata-rata akurasi 78% dengan nilai AUC 0.783. Penelitian ini menguji kinerja J48 dengan pendekata ROS dan RUS menggunakan Algoritma J48 dan J48 Consolidated. Hasil penelitian menunjukan model RUS+J48 Consolidated lebih baik dari model RUS+J48 dengan nilai rata-rata akurasi 78% dan 77% serta nilai AUC 0.783 dan 0.766.

Kata Kunci: Resample, J48, J48 Consolidated, Cacat Software, Imbalanced Class, Random Over Sampling, Random Under Sampling.

PENDAHULUAN

Cacat pada suatu pernagkat lunak merupakan kegagalan atau kesalahan dalam sistem komputer tersebut yang mengakibatkan dihasilkanya kesalahan lainya yang tidak terduga dan dapat menurunkan kualitas pada suatu perangkat lunak (Wahono & Suryana, 2013). Software defect, juga dikenal sebagai bug, adalah kesalahan atau kegagalan dalam perangkat lunak yang menyebabkan perangkat lunak tersebut tidak berfungsi sesuai dengan yang diharapkan atau tidak memenuhi spesifikasi yang telah ditetapkan. Defect dapat terjadi pada berbagai tahap siklus hidup pengembangan perangkat lunak, mulai dari perencanaan, desain, implementasi, hingga pengujian

dan penggunaan. Kualitas suatu *software* dapat ditemukan pada saat fase pemeriksaan dan fase pengujian (Fitriyani & Wahono, 2015)

Penting untuk mendeteksi dan memperbaiki defect sejak awal dalam siklus pengembangan perangkat lunak, karena semakin lama defect ditemukan, semakin mahal dan sulit untuk memperbaikinya. Proses pengujian perangkat lunak sangat penting untuk mengidentifikasi defect sebelum perangkat lunak dilepaskan kepada pengguna akhir. Ada berbagai metode dan alat pengujian yang dapat membantu dalam mengidentifikasi dan memperbaiki defect sebelum perangkat lunak diluncurkan.

Kajian prediksi cacat software dalam software engineering , juga dikenal sebagai prediksi cacat perangkat lunak, adalah pendekatan untuk mengidentifikasi dan memprediksi potensi defect atau bug dalam perangkat lunak sebelum perangkat lunak tersebut dirilis ke pengguna akhir(Batuwita & Palade, 2010). Tujuan utama dari kajian ini adalah untuk mengurangi jumlah defect yang mungkin muncul setelah perangkat lunak diluncurkan, sehingga dapat menghemat waktu, biaya, dan upaya yang diperlukan untuk memperbaiki defect di kemudian hari.Kajian prediksi cacat *software* dalam *software* engineering terutama ketika mengatasi efektifitas dan efisiensi diluar pengujian perangkat lunak dan *review*, sehingga ketepatan prediksi cacat *software* dapat memudahkan pengujian, mengurangi biaya dan bisa meningkatkan kualitas suatu *software*(Wahono et al., 2014)

Prediksi kesalahan pada perangkat lunak ialah langkah penting untuk tingkatkan mutu perangkat lunak. Kesalahan yang ada pada perangkat lunak di masa depan ialah kesalahan pada informasi yang salah sebelumnya(Sathyaraj & Prabu, 2015). Oleh sebab itu dibutuhkan adanya riset yang matang dalam memprediksi cacat pada perangkat lunak.

Studi mengenai cacat pada perangkat lunak upaya untuk menganalisis, memahami, dan mengidentifikasi jenis-jenis cacat yang muncul dalam perangkat lunak, serta faktor-faktor yang menyebabkan timbulnya cacat tersebut. Tujuan dari studi ini adalah untuk meningkatkan pemahaman tentang alasan di balik cacat perangkat lunak dan mencari cara-cara untuk mencegah, mendeteksi, dan mengatasi cacat-cacat tersebut. (Gorunescu, 2011) Dalam riset mengenai prediksi cacat pada aplikasi, pemrediksian model bersumber pada pada atribut kode buat memperhitungkan kemungkinan modul software yang memiliki cacat (Ma et al., n.d.).

Studi mengenai cacat pada perangkat lunak merupakan bagian penting dari upaya keseluruhan untuk meningkatkan kualitas perangkat lunak dan mengurangi risiko cacat (Wang & Yao, n.d.). Hasil studi ini dapat memberikan wawasan berharga bagi para pengembang perangkat lunak untuk membuat keputusan yang lebih baik dalam hal desain, pengujian, dan pengelolaan kualitas perangkat lunak. Fokus pada riset prediksi cacat aplikasi pada saat ini lebih pada memperkirakan jumlah cacat pada suatu software, menciptakan bagian yang cacat pada komponen software serta mengelompokannya ke dalam klasifikasi rawan dan tidak rawan(Song et al., 2011)

Banyak metode yang bisa digunakan guna memprediksi modul cacat software dari potensi rawan kegagalan, salah satunya yang efisien yaitu dengan menggunakan reka metode data mining yang diaplikasikan pada software matrix yang diperoleh sepanjang proses pengembangan aplikasi (Khoshgoftaar et al., 2010). Salah satu algoritma yang bisa diterapkan buat memprediksi modul cacat software yaitu algoritma decision tree.

Decision tree ialah salah satu metode penambangan data dengan memprediksi nilai sasaran bersumber pada satu atau lebih nilai atribut yang tersedia (Dr. Bhargava N., Sharma G., Dr. Bhargava R.,

2013). Tujuan Decision tree J48 adalah untuk menciptakan pohon keputusan terus menerus hingga memperoleh penyeimbang fleksibilitas serta ketepatan (Daud et al., 2019).

Algoritma J48 adalah algoritma pengambilan keputusan yang digunakan dalam pembelajaran mesin untuk membangun pohon keputusan. Pohon keputusan adalah struktur berhirarki yang digunakan untuk mengambil keputusan berdasarkan serangkaian kondisi atau atribut.Algoritma J48 merupakan suatu jenis *classifier* pada metode klasifikasi dalam data mining dan merupakan bagian dari C4.5 *decision tree* yang sederhana (Dr. Bhargava N., Sharma G., Dr. Bhargava R., 2013). Pada algoritma C4.5 membangun suatu pohon keputusan berdasarkan pada perangkat input data yang berlabel (Daud et al., 2019). J48 adalah implementasi dari algoritma C4.5 yang dikembangkan oleh Ross Quinlan. Algoritma ini cocok untuk masalah klasifikasi, di mana tujuannya adalah untuk mengklasifikasikan entitas menjadi salah satu dari beberapa kelas yang telah ditentukan sebelumnya. Pohon keputusan ialah bentuk prediksi dengan menggunakan struktur pohon ataupun hirarki. Rancangan dari model keputusan adalah dengan mengganti data menjadi suatu pohon keputusan serta aturan- aturan keputusan (Diwandri & Setiawan, 2015).

Dalam penelitian mengenai *software defect prediction* data metrik yang digunakan sebagai acuan adalah *dataset* NASA MDP. Penggunaan NASA *dataset* merupakan pilihan yang tepat karena mudah diperoleh dan dapat dibandingkan kinerjanya dengan penelitian yang sebelumnya (Vasco, 2014).*Dataset* NASA berasal dari NASA *Metric Data Program* (MDP) *repository* dan *Predictor Models in Software Engineering* (PROMISE) *repository*. (Saifudin & Wahono, 2015)

Tujuan dalam penelitian ini adalah untuk miningkatkan kualitas *software* oleh karena itu dibutuhkanya model pemprediksi cacat *software* yang bias menangani ketidakseimbangan kelas serta memiliki kinerja yang tinggi dalam mengkasifikasi cacat software dan mengetahui pengaruh teknik *Resample*terhadap metode kasifikasi J48 dan J48 *consolidated.*

METODE

Guna meningkatkan kualiatas software dibutuhkan model pemprediksi cacat software yang mempunyai kemampuan besar dalam mengklasifikasi cacat software, namun saat ini belum terdapat model prediksi cacat software yang berlaku secara umum.

Kumpulan data metric perangkat lunak NASA umumnya tidak seimbang,sehingga dapat menurunkan kinerja model prediktif dalam mengklasifikasikan kegagalan perangkat lunak karena cenderung menghasilkan tipemayoritas(Gray et al., 2011).

Dataset yang dipakai pada penelitian ini bersumber dari dataset PROMISE NASA MDP berupa dataset JM1, CM1, KC1, dan KC2.

Untuk alat atau tools yang digunakan dalam proses pengujian dataset PROMISE NASA MDP menggunakan Weka (Weikato Environment for Knowledge Analyisis) Version 3.9.5.

Pengolahan data awal menggunakan teknik Resample serta algoritma yang digunakan yaitu J48 dan J48 Consolidated(Yap et al., 2014).

Pendekatan yang dipakai pada riset ini yakni pendekatan yang bersifat kuantitatif. Aspek kuantitatif memberikan penekanan bahwasannya pengukuran ialah dasarnya sebab membagikan hubungan antara observasi serta formalisasi bentuk, filosofi dan hipotesis.

Penelitian eksperimen melingkupi investigasi ikatan sebab- akibat memakai pengujian yang dikontrol sendiri. Metode yang dipakai pada penelitian ini merupakan metode penelitian. Riset eksperimen umumnya dilakukan dalam pengembangan penilaian serta pemecahan permasalahan proyek. Tujuan dari riset ini untuk mencari solusi mengenai ketidak

seimbangan kelas pada dataset bentuk prediksi cacat software serta menciptakan model yang bisa memaksimalkan kemampuan algoritma dalam melaksanakan pengklasifikasian (McDonald et al., 2007). Berikut ini tahapan- tahapan yang dilakukan pada penelitian ini.

Dalam penelitian ini, data yang digunakan adalah dataset NASA yang diunduh dari repositori Promise, yang banyak digunakan para peneliti dalam menemukan bug perangkat lunak. Data ini biasanya tersedia di internet, sehingga metode pengumpulan data yang digunakan adalah dengan mendownload dataset NASA dari repositori Promise.dengan.url http://promise.site.uottawa.ca/SERepository/.

Keseluruhan akurasi pada percobaan dataset biasanya dipakai guna menilai kemampuan pengklasifikasi (Philip & Gray, 2012). Namun untuk data yang tidak seimbang, akurasi yang mendalam didominasi oleh kelas minoritas, akibatnya pengganti penilaian metrik dipakai. Metrik penilaian yang pas tercantum Zona Under the ROC(Receiver Operating Characteristic) Curve(AUC), F- Measure, Geometric Mean(G- Mean), seluruh akurasi dan umumnya akurasi untuk kelas minoritas. Untuk melaksanakan penilaian serta pengesahan terhadap bentuk yang diusulkan, hingga dilakukan sebagian pengetesan memanfaatkan confusion matrix(Bienvenido-Huertas et al., 2020).

Pada penelitian ini digunakan metode Correlation Feature Selection (CFS) seperti Feature Selection, Random Over Sampling untuk menangani data yang tidak seimbang, dan AdaBoost untuk meningkatkan kinerja algoritma J48 agar memperoleh hasil yang lebih optimal.

Berdasarkan hasil penelitian yang dilakukan, pemilihan fitur korelasi untuk pemilihan atribut dan random oversampling untuk mengatasi ketidakseimbangan kelas menggunakan algoritma J48 berbasis Adaboost terbukti meningkatkan hasil klasifikasi diabetes dengan akurasi 92,3. %.

Pada penelitian ini digunakan metode Particle Swarm Optimization (PSO) sebagai optimasi metaheuristic untuk mengatasi permasalahan redudansi data serta data yang tidak relevan. Pada pendekatan level algoritma digunakan algoritma Bagging untuk meningkatkan kinerja model pengklasifikasian(Ibarguren et al., 2015).

Uji coba pada penelitian ini dengan mengklasifikasi 9 dataset NASA MDP dengan 11 model algoritma yang dibagi menjadi 5 tipe model klasifikasi yaitu : statistic tradisional (Logistic Regression (LR), Linear Discriminant Analysis (LDA), Naïve bayes (NB)), tipe algoritma Nearest Neighbors (K-Nearest Neighbor (K-NN) dan K*), tipe algoritma Neural Network (Back Propagation (BP)), tipe algoritma Support Vector Machine (SVM dan LibSVM) dan tipe algoritma Decission Tree (C4.5 Classification and Regression Tree (CART), dan Random Forest (RF)).

HASIL DAN PEMBAHASAN

Preprocessing Data

Sistem intelijen dan model matematis untuk pengambilan keputusan dapat mencapai hasil yang akurat dan efektif hanya jika data yang digunakan dapat diandalkan [22]. Untuk mengenali, menghilangkan anomaly dan inkonsistensi serta penurunan ukuran dan diskritisasi informasi, serta untuk mendapatkan dataset dengan jumlah atribut dan record yang lebih rendah namun informatif digunakan sebagian teknik validasi data.

Penelitian ini menggunakan dataset dari NASA MDP repository menggunakan 4 dataset yaitu JM1, CM1, KC1, KC2. Tahap preprocessing dilakukan dengan melihat beberapa ketentuan awal yang perlu dijadikan perhatian diantaranya jumlah atribut, jumlah modal dan jumlah cacat pada setiap dataset. Spesifikasi dataset yang digunakan dalam penelitian ini sebagai berikut.

Tabel 1. Spesifikasi Dataset PROMISE Repository Yang Digunakan Dalam Penelitian

Dataset	Atribut	Modul	Cacat	Tidak Cacat
JM1	22	10855	2106	8779
CM1	22	498	49	449
KC1	22	2109	326	1783
KC2	22	522	105	415

Pengujian Model Random Over SamplngDecission Tree J48

Pertama dilakukan pengujian dengan menggunakan model Pengklasifikasi pada keempat dataset. Hasil pengujian tersebut dihitung dengan menggunakan Confusion Metrix untuk mencari akurasi, sensitifitas, soecificity, FPrate, FNrate, Precision, F-Measure, G-Mean dan AUC (Vercellis, 2009). Hasil diperoleh probabilitas sebagai berikut.

Tabel 2. Hasil Confusion Matrix pada dataset JM1

Kelas	Predicted : NO	Predicted : YES	
Actual NO	TN = 8302	FP =477	8779
Actual YES	FN =997	TP =1109	2106

Perhitungan menggunakan persamaan berikut :

Akurasi = (TP+TN / TP+TN+FP+FN)
= (1109+8302 / 1109+8302+477+997
= (9411 / 10885)
= 0.8645

Sensitifitas = (TP/TP+FN)
= (1109/1109+997)
= (1109/2106
=0.5265

Specificity = (TN/TN+FP)
= (8302/8302+477)
= 8302/8779
= 0.9456

FPrate = (FP/FP+TN)
= 477/477+8302
= 477/8779
= 0.0543

FNrate = (FN/TP+FN)
= (997/1109+997)
= 997/2106
= 0.4734

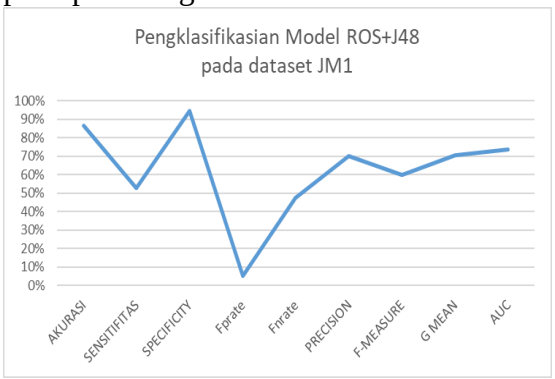
Precision = (TP/TP+FP)
= (1109/1109+477)
= 1109/1586
= 0.6992

F-Measure = (2 x Sensitivitas x Precision) / (Sensitivitas + Precision)
= (2 x 05265. x 0.6992) / (0.5265 + 0.6992)
= 0.7362/ 1.2257
= 0.6007

G Mean = √Sensitivitas x specificity
= √0.5265 x 0.9456
= √0.4978
= 0.7056

AUC = (1+TPrate-FPrate)/2
= (1+ 0.5265- 0.0543)/2
= (1+0.4723)/2
= 0.7361

Perhitungan diatas menunjukan bahwa kinerja pengkalsifikasian berbasin ROS+J48 pada dataset JM1 menghasilkan nilai akurasi mencapai 86%, Sensitifitas 53%, Specificity 95%, Precision 70%, F-measure 60%, G-mean 71% dan nilai AUC 74%. Diagram berikut ini menjelaskan perolehan nilai dari 9 variabel yang dicari pada perhitungan diatas.



Perhitungan yang sama dilakukan terhadap confusion matrix hasil pengujian pada dataset JM1, CM1, KC1, dan KC2.

Pengujian Model Random Over Sampling + J48 Consolidated

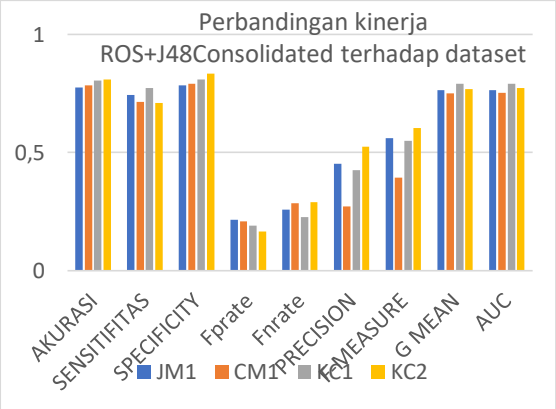
Model yang digabungkan pertama dengan teknik Random over Sampling (ROS) sebagai pendekatan level data untuk menangani ketidakseimbangan kelas. Pengklasifikasian dengan algoritma J48

Consolidated dengan menggunakan confusion matrix.

Tabel 3. Hasil kinerja model ROS+J48 Consolidated

DATASET	JM1	CM1	KC1	KC2	Rata-rata
AKURASI	0.775655	0.783133	0.804173	0.808429	0.792847
SENSITIFITAS	0.74264	0.714286	0.773006	0.71028	0.735053
SPECIFICITY	0.783574	0.790646	0.809871	0.833735	0.804457
Fprate	0.216426	0.209354	0.190129	0.166265	0.195543
Fnrate	0.25736	0.285714	0.226994	0.28972	0.264947
PRECISION	0.451501	0.271318	0.426396	0.524138	0.418338
F-MEASURE	0.56158	0.393258	0.549618	0.603175	0.526908
G MEAN	0.762833	0.751497	0.791224	0.769536	0.768772
AUC	0.763107	0.752466	0.791439	0.772008	0.769755

Dari tabel diatas terlihat bahwa nilai akurasi tertinggi yang dihasilkan oleh model fusion ROS+J48 terdapat pada dataset KC1 dengan nilai sensitivitas sebesar 0.773 dan nilai AUC tertinggi juga terdapat pada dataset KC1. dengan nilai AUC mencapai 0,7914 dan nilai mean G tertinggi juga dimiliki oleh dataset KC1 dengan nilai mean G mencapai 0,7912. Bagan berikut membandingkan performa model ROS+J48 terpadu terhadap kumpulan data JM1, CM1, KC1, dan KC2.



Dari hasil uji coba diatas ditunjukan nilai rata-rata dari kinerja model ROS+J48Consolidated terhadap dataset JM1,CM1,KC1, dan KC2 meliputi akurasi sebesar 0.7928, Sensitifitas 0.7350, Specificity 0.8044, precision 0.4183, F-measure 0.5269,G-mean 0.7687 dan AUC 0.7697. pada pengujian kali ini terlihat adanya peningkatan signifikan, terutama pada Sensitifitas, G-Mean dan AUC.

Pengujian Model Random Under Sampling + J48

Model yang digabungkan pertama dengan teknik Random Under Sampling (RUS) sebagai pendekatan level data untuk menangani ketidakseimbangan kelas. Pengklasifikasian dengan algoritma J48 dengan menggunakan confusion matrix.

Pengujian Model Random Under Sampling + J48 Consolidated

Model yang digabungkan pertama dengan teknik Random Under Sampling (RUS) sebagai pendekatan level data untuk menangani ketidakseimbangan kelas. Pengklasifikasian dengan algoritma J48 Consolidated dengan menggunakan confusion matrix.

Nilai kebenaran tertinggi yang dihasilkan oleh model fusi RUS+J48 terdapat pada dataset KC1 dengan nilai sensitivitas sebesar 0,7797 dan juga ditampilkan nilai AUC tertinggi, pada dataset KC2 memiliki nilai AUC sebesar 0,8284 dan nilai mean G tertinggi juga dimiliki oleh dataset KC2 dengan nilai mean G sebesar 0,8249. Bagan berikut membandingkan performa model RUS+J48 terpadu dengan kumpulan data JM1, CM1, KC1, dan KC2.

Dari hasil uji coba diatas ditunjukan nilai rata-rata dari kinerja model RUS+J48 Consolidated terhadap dataset JM1,CM1,KC1, dan KC2 meliputi akurasi sebesar 0.8039, Sensitifitas 0.7499, Specificity 0.8174, precision 0.4691, F-measure 0.5631, G-mean 0.7825 dan AUC 0.7836. pada pengujian kali ini terlihat adanya peningkatan signifikan, terutama pada Sensitifitas, G-Mean, dan AUC.

Perbandingan Penelitian Terkait

Perbandingan model usulan terhadap model lain yang diusulkan oleh peneliti lain pada penelitian sebelumnya dilaksanakan untuk menghasilkan kontribusi penelitian ini terhadap bidang ilmu pengetahuan walaupun unsur keberhasilan suatu penelitian tidak selalu diukur melalui angka yang dihasilkan.

Hasil perbandingan model yang diusulkan dengan penelitian sebelumnya terkait prediksi kesalahan perangkat lunak menunjukkan bahwa model prediksi

kesalahan perangkat lunak menunjukkan kinerja yang lebih unggul dibandingkan dengan model prediksi kesalahan perangkat lunak yang diusulkan pada penelitian sebelumnya.

Keunggulan performa tersebut tergambar pada model prediksi kegagalan perangkat lunak berbasis ROS+J48, dimana pada dataset CM1 dapat diperoleh nilai AUC sebesar 0,784. Keunggulan performa diilustrasikan pada model RUS+J48, dimana kumpulan data KC2 dapat memperoleh nilai AUC. adalah 0.829, keunggulan performa pada model fusion RUS+J48 dimana dataset JM1 memperoleh nilai AUC sebesar 0.765 sedangkan dataset KC1 memperoleh nilai AUC sebesar 0.798. Oleh karena itu, dapat disimpulkan bahwa model prediksi kegagalan perangkat lunak yang diusulkan, yaitu model fusi RUS+J48, menunjukkan kinerja yang lebih baik dengan rata-rata AUC sebesar 0,783. Hasil ini dapat dijadikan acuan untuk penelitian selanjutnya.

SIMPULAN

Untuk mengatasi masalah ketidakseimbangan kelas pada kumpulan data digunakan metode resampling yaitu Random Over Sampling (ROS), Random Under Sampling (RUS) serta menggunakan algoritma J48 dan J48 Consolidated.

Setelah dilakukan percobaan dengan melakukan pengujian pengklasifikasian dengan aplikasi WEKA. Metode usulan disimulasikan pada dataset PROMISE dengan 4 dataset software matrix yaitu JM1,CM1,KC1, dan KC2. Hasil pengujian metdoe kinerja dapat dilihat pada pembahasan Bab IV. Hasil pengujian metode RUS+J48 Consolidated menunjukan tingkat rata-rata akurasi yang tinggi yaitu 78% dengan AUC 0,783. Sehingga metode RUS +J48 Consolidated layak digunakan sebagai metode pemprediksian cacat perangkat lunak.

DAFTAR PUSTAKA

Batuwita, R., & Palade, V. (2010). Efficient resampling methods for training support vector machines with imbalanced datasets. *Proceedings of the International Joint Conference on Neural Networks*. https://doi.org/10.1109/IJCNN.2010.5596787

Bienvenido-Huertas, D., Nieto-Julián, J. E., Moyano, J. J., Macías-Bernal, J. M., & Castro, J. (2020). Implementing Artificial Intelligence in H-BIM Using the J48 Algorithm to Manage Historic Buildings. *International Journal of Architectural Heritage*, *14*(8), 1148–1160. https://doi.org/10.1080/15583058.2019.1589602

Daud, N., Mohd Noor, N. L., Aljunid, S. A., Noordin, N., & Fahmi Teng, N. I. M. (2019). Predictive Analytics: The Application of J48 Algorithm on Grocery Data to Predict Obesity. *2018 IEEE Conference on Big Data and Analytics, ICBDA 2018*, *November*, 1–6. https://doi.org/10.1109/ICBDAA.2018.8629623

Diwandri, N., & Setiawan, A. (2015). Perbandingan Algoritme J48 Dan Nbtree Untuk Klasifikasi. *Seminar Nasional Teknologi Informasi Dan Komunikasi 2015 (SENTIKA 2015)*, *2015*(Sentika), 205–212.

Dr. Bhargava N., Sharma G., Dr. Bhargava R., M. M. (2013). International Journal of Advanced Research in Decision Tree Analysis on J48 Algorithm for Data Mining. *International Journal of Advanced Research in Computer Science and Software Engineering*, *3*(6), 1114–1119.

Fitriyani, & Wahono, R. S. (2015). Integrasi Bagging dan Greedy Forward Selection pada Prediksi Cacat Software dengan Menggunakan Naïve Baye. *Journal*

of Software Engineering, *1*(2).

Gorunescu, F. (2011). *Data Mining: Concepts, models and techniques*.

Gray, D., Bowes, D., Davey, N., Sun, Y., & Christianson, B. (2011). The misuse of the NASA Metrics Data Program data sets for automated software defect prediction. *IET Seminar Digest*, *2011*(1), 96–103. https://doi.org/10.1049/ic.2011.0012

Ibarguren, I., Pérez, J. M., Muguerza, J., Gurrutxaga, I., & Arbelaitz, O. (2015). Coverage-based resampling: Building robust consolidated decision trees. *Knowledge-Based Systems*, *79*, 51–67. https://doi.org/10.1016/j.knosys.2014.12.023

Khoshgoftaar, T. M., Gao, K., & Seliya, N. (2010). Attribute selection and imbalanced data: Problems in software defect prediction. *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, *1*, 137–144. https://doi.org/10.1109/ICTAI.2010.27

Ma, B., Dejaeger, K., Vanthienen, J., & Baesens, B. (n.d.). *Software defect prediction based on asso- ciation rule classification*. 0–7.

McDonald, M., Musson, R., & Smith, R. (2007). *The Practical Guide to Defect Prevention (Best Practices (Microsoft))*. http://www.amazon.com/Practical-Defect-Prevention-Practices-Microsoft/dp/0735622531

Philip, D., & Gray, H. (2012). Software Defect Prediction Using Static Code Metrics : Formulating a Methodology. *Journal*, *December*.

Saifudin, A., & Wahono, R. S. (2015). Penerapan Teknik Ensemble untuk Menangani Ketidakseimbangan Kelas pada Prediksi Cacat Software. *Journal of Software Engineering*, *1*(1).

Sathyaraj, R., & Prabu, S. (2015). An approach for software fault prediction to measure the quality of different prediction methodologies using software metrics. *Indian Journal of Science and Technology*, *8*(35). https://doi.org/10.17485/ijst/2015/v8i35/73717

Song, Q., Jia, Z., Shepperd, M., Ying, S., & Liu, J. (2011). A general software defect-proneness prediction framework. *IEEE Transactions on Software Engineering*, *37*(3), 356–370. https://doi.org/10.1109/TSE.2010.90

Vasco, P. (2014). *An update of the J48Consolidated WEKA ' s class : CTC algorithm enhanced with the notion of coverage*. *June*.

Vercellis, C. (2009). Business Intelligence: Data Mining and Optimization for Decision Making. In *Business Intelligence: Data Mining and Optimization for Decision Making*. https://doi.org/10.1002/9780470753866

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). https://doi.org/10.4304/jsw.9.5.1324-1333

Wang, S., & Yao, X. (n.d.). *Using Class Imbalance Learning for Software Defect Prediction*. 1–11.

Yap, B. W., Rani, K. A., Abd Rahman, H. A., Fong, S., Khairudin, Z., & Abdullah, N. N. (2014). An application of oversampling, undersampling, bagging and boosting in handling imbalanced

datasets. In *Lecture Notes in Electrical Engineering: Vol. 285 LNEE*. https://doi.org/10.1007/978-981-4585-18-7_2