

**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****Nurul Ichsan<sup>1)</sup>, Suhardi<sup>2)</sup>, Robi Sopandi<sup>3)</sup>, Hananda Priyandaru<sup>4)</sup>, Muhamad Tabrani<sup>5)</sup>**<sup>1)</sup>Sistem Informasi, Fakultas Teknik dan Informatika, Universitas Bina Sarana Informatika<sup>2)</sup>Sistem Informasi Akuntansi PSDKU Karawang, Universitas Bina Sarana Informatika<sup>3)</sup>Informatika, Fakultas Teknologi dan Informasi, Universitas Nusa Mandiri<sup>4)</sup>Manajemen, Fakultas Ekonomi dan Bisnis, Universitas Bina Sarana Informatika<sup>5)</sup>Sistem Informasi PSDKU Karawang, Universitas Bina Sarana Informatika<sup>1)</sup>nurul.nrc@bsi.ac.id**ABSTRAK**

Prediksi cacat software merupakan salah satu fase atau tahap pengujian dalam Software Development Life Cycle. Proses pengujian software dapat mengidentifikasi apakah sebuah software mengandung cacat atau tidak. Pengujian menjadi standar dalam menghasilkan software yang berkualitas. Masalah utama dalam dataset software metrics adalah *imbalance class* yang menjadikan data tidak seimbang karena data yang cacat (kelas minoritas) jumlahnya lebih sedikit dibandingkan dengan data yang tidak cacat (kelas mayoritas). Pendekatan level data *Synthetic Minority Over-sampling Technique* dan *Bagging* diusulkan untuk meningkatkan kinerja classifier C4.5 pada prediksi cacat software untuk menangani *imbalance class*. Model yang diusulkan diterapkan pada 12 NASA (*National Aeronautics and Space Administration*) MDP (*Metrics Data Program*) repository sebagai *software metrics* pada prediksi cacat software. Model yang diusulkan mencapai akurasi dan AUC klasifikasi yang baik. Pada Model Synthetic Minority Over-sampling Technique (SMOTE)+Bagging+C4.5. Rata-rata Akurasi dan AUC meningkat dari model sebelumnya dimana akurasi meningkat sebesar 2.77% dan AUC meningkat sebesar 0.143 dan hasil tersebut berpengaruh terhadap tingkatan level kinerja klasifikasi yang didapat yang pada awalnya berada pada level "Failure Classification" karena berada pada rata-rata nilai AUC 0.50-0.60. Setelah dilakukannya integrasi pengujian model SMOTE+Bagging+C4.5 kinerja klasifikasi meningkat pada level "Fair Classification" dengan nilai rata-rata AUC 0.741.

Kata kunci: Synthetic Minority Over sampling Technique; Bagging; C4.5

**ABSTRACT**

*Software defect prediction is one of the phases or testing stages in the Software Development Life Cycle. The software testing process can identify whether a software contains defects or not. Testing is the standard for producing quality software. The main problem in software metrics datasets is imbalance class which makes the data unbalanced because the number of defective data (minority class) is less than that of non-defective data (majority class). Synthetic Minority Over-sampling Technique and Bagging data level approaches are proposed to improve the performance of the C4.5*

*classifier in software defect prediction to handle imbalance classes. The proposed model is applied to 12 NASA (National Aeronautics and Space Administration) MDP (Metrics Data Program) repositories as software metrics in software defect prediction. the proposed model achieves good classification accuracy and AUC. In the Synthetic Minority Over-sampling Technique (SMOTE)+Bagging+C4.5 model. The average accuracy and AUC increased from the previous model where accuracy increased by 2.77% and AUC increased by 0.143 and these results affected the level of classification performance obtained which was initially at the "Failure Classification" level because it was at the average AUC value 0.50-0.60. After integrating the SMOTE+Bagging+C4.5 model testing, classification performance increased to the "Fair Classification" level with an average AUC value of 0.741.*

*Keywords: synthetic minority over sampling technique; bagging; C4.5*

## PENDAHULUAN

Prediksi cacat *software* merupakan salah satu fase atau tahap pengujian dalam *Software Development Life Cycle* (Amir et al., 2020). Prediksi cacat *software* menjadi topik penelitian paling penting dalam *software engineering* yang telah menarik banyak minat peneliti besar dari berbagai komunitas, akademisi dan industri (Hidayati et al., 2021). *Software* berkualitas tinggi adalah *software* yang tidak ditemukan cacat (*defect*) baik selama proses pemeriksaan atau pengujian (Wahono, 2015)

Proses pengujian *software* dapat mengidentifikasi apakah sebuah *software* mengandung cacat atau tidak (Sugiono et al., 2020). Sejumlah cacat yang ditemukan di akhir proyek secara sistematis menyebabkan penyelesaian proyek dapat melebihi jadwal yang sudah ditentukan (Ichsan, 2019). Menurut NIST (*National Institute of Standards and Technology*) biaya untuk estimasi dalam memperbaiki *software* yang cacat mencapai \$60 *billion* atau sekitar 60 miliar pertahun (Putri, 2019). Pengujian menjadi standar dalam menghasilkan *software* yang berkualitas (Ichsan et al., 2022).

Masalah utama dalam dataset *software metrics* adalah *imbalance class* yang menjadikan data tidak seimbang karena data yang cacat (kelas minoritas) jumlahnya lebih sedikit dibandingkan dengan data yang tidak cacat (kelas mayoritas). Terdapat dua pendekatan yang dapat menangani *imbalance class* yaitu pendekatan level data (*sampling technique*) dan pendekatan level algoritma dengan teknik *ensemble learning*.

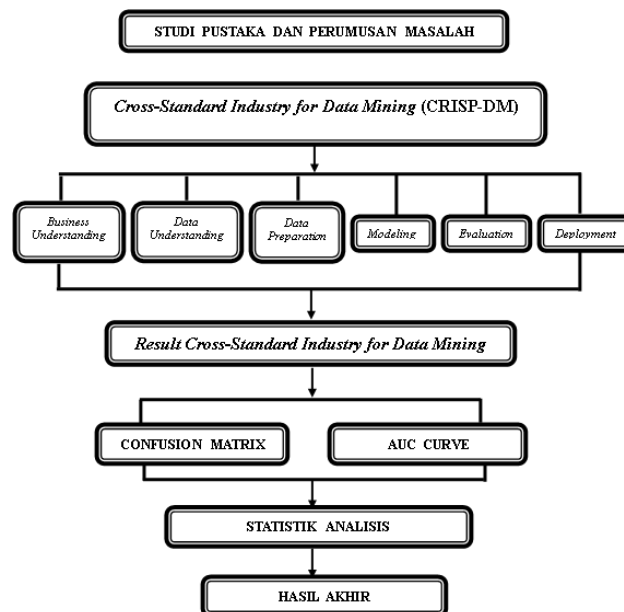
Sedangkan klasifikasi adalah pendekatan yang paling populer untuk menangani *software defect prediction* dimana data penelitian yang didapat 77.46% dalam menyelesaikan masalah prediksi cacat *software* diselesaikan dengan menggunakan metode klasifikasi (Hall et al., 2012). Dari banyaknya penelitian yang sudah dilakukan mengenai *software defect prediction* oleh peneliti sebelumnya baik dari akademisi maupun industri, tidak ditemukan satu metode yang mutlak terbaik untuk mengklasifikasikan *software* berbasis metrik yang secara umum dan selalu konsisten dalam semua penelitian yang berbeda (Hardoni et al., 2021). Hal ini disebabkan karena untuk pengklasifikasian metrik hasil metode klasifikasi terbaik tergantung dari dataset *software metrics* yang digunakan (Prasetyo et al., 2021).

Pada penelitian ini untuk menangani masalah *imbalance class* dilakukan dengan menggunakan pendekatan level data (*sampling technique*) dan pendekatan level algoritma atau penggabungan dengan teknik *ensemble (ensemble learning)*. Pendekatan level data yang akan dilakukan menggunakan teknik *oversampling* menggunakan SMOTE. Teknik *ensemble* yang digunakan adalah *Bagging*. Sedangkan algoritma klasifikasi yang digunakan dalam penelitian ini yaitu *classifier C4.5* Dataset yang digunakan dalam penelitian ini adalah dataset NASA MDP *Repository* sebagai *software metrics*. Dengan melakukan penelitian tersebut diharapkan akan memperoleh nilai akurasi dan AUC (*Area Under ROC Curve*) yang signifikan dan dapat mengatasi permasalahan seperti yang dijelaskan sebelumnya yaitu mengenai ketidakseimbangan kelas (*imbalance class*) yang terdapat pada dataset *software metrics* dalam fokus penelitian prediksi cacat *software*.

## METODE PENELITIAN

### 1. Tahapan Penelitian

Tahapan-tahapan yang dilakukan dalam penelitian ini dapat dilihat pada gambar di bawah ini :



**Gambar 1.** Tahapan penelitian

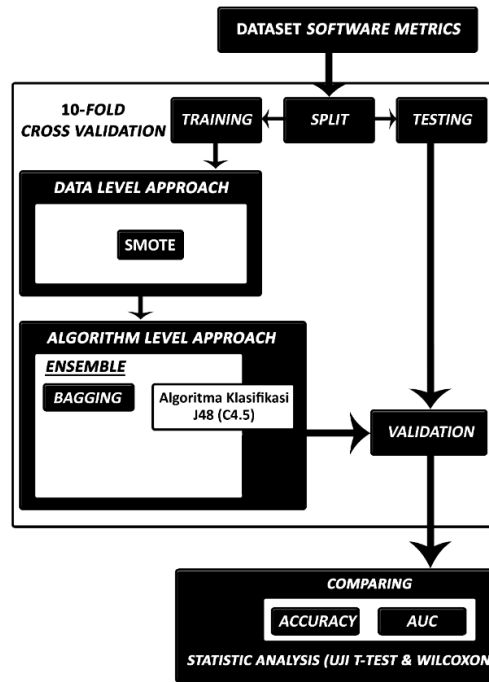
Metode yang digunakan pada penelitian ini adalah eksperimen dengan menguji model yang diusulkan, evaluasi dan validasi. Tujuan dari penelitian ini untuk mengurangi pengaruh ketidakseimbangan kelas pada model prediksi cacat *software*. Metode penelitian yang digunakan pada eksperimen ini adalah model *Cross-Standard Industry for Data Mining (CRISP-DM)* yang terdiri dari 6 fase yaitu *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modeling*, *Evaluation* dan *Deployment*.

## 2. Metode Pengumpulan Data

Penelitian ini menggunakan pendekatan kuantitatif, sedangkan untuk pemilihan datanya menggunakan data sekunder. Data yang digunakan dalam penelitian ini yaitu dataset NASA (*National Aeronautics and Space Administration*) MDP (*Metrics Data Program*) repository. Tools yang digunakan untuk pengujian algoritma adalah aplikasi WEKA (*Waikato Environment for Knowledge Analysis*), yang merupakan aplikasi data mining open source berbasis Java.

## 3. Model yang diusulkan

Untuk mencari solusi masalah ketidakseimbangan kelas pada dataset *software metrics*, model yang diusulkan oleh penulis dalam penelitian ini yaitu sebagai berikut:



Gambar 2. Kerangka kerja model yang diusulkan

## HASIL PENELITIAN DAN PEMBAHASAN

### 1. Dataset (NASA MDP Repository)

Data yang digunakan merupakan data *public* yaitu 12 dataset NASA (*National Aeronautics and Space Administration*) MDP (*Metrics Data Program*) *repository* yang dapat diunduh di <https://github.com/klainfo/NASADefectDataset>. Jumlah data yang digunakan sebagai berikut:

Tabel 1. Jumlah data penelitian

DATA NASA MDP REPOSITORY (D <sup>n</sup> )	12 DATASET												
	CM1	JM1	KC1	KC3	MC1	MC2	MW1	PC1	PC2	PC3	PC4	PC5	TOTAL
Attribut	38	22	22	40	39	40	38	38	37	38	38	39	429
Jumlah Modul	327	7720	1162	194	1952	124	250	679	722	1053	1270	1694	17147
Modul Cacat	42	1612	294	36	36	44	25	55	16	130	176	458	2924
Modul Tidak cacat	285	6108	868	158	1916	80	225	624	706	923	1094	1236	14223
Presentase Cacat (%)	12.84	20.98	25.30	18.56	1.84	35.48	10	8.10	2.21	12.34	13.86	27.03	17.05

Perbandingan jumlah presentase cacat sangat bervariasi dari 12 dataset *software metrics* yang akan diuji cobakan mulai dari 1.84% s/d 35.48% untuk

tingkat presentase cacatnya. Maka diperlukan model untuk menangani ketidakseimbangan kelas yang dapat menurunkan kinerja pengklasifikasi. Untuk menangani *imbalance class* teknik pengujiannya akan mencoba dengan model pengujian baru yaitu mengintegrasikan perbandingan kinerja model pendekatan level data SMOTE dan pendekatan level algoritma (*ensemble*) *Bagging* berbasis *classifier* C4.5 untuk mencari kinerja model yang paling memberikan hasil yang terbaik dan signifikan baik untuk akurasi ataupun nilai AUC nya.

## 2. Pengujian Model SMOTE+BAGGING+C4.5

Sebelum diterapkan dengan berbagai model pengujian, terlebih dahulu dilakukan pengujiannya dengan kinerja asli dari model *classifier* secara original yaitu C4.5

**Tabel 2.** Pengujian 12 dataset dengan *classifier* C4.5 (Original)

Dataset	C4.5 (Original)			Kinerja Klasifikasi
	Akurasi	AUC	ROC Area	
CM1	81.04%	0.536	0.570	<i>Failure Classification</i>
JM1	77.53%	0.582	0.653	<i>Failure Classification</i>
KC1	74.18%	0.591	0.604	<i>Failure Classification</i>
KC3	79.38%	0.616	0.653	<i>Poor Classification</i>
MC1	97.90%	0.540	0.566	<i>Failure Classification</i>
MC2	60.48%	0.576	0.589	<i>Failure Classification</i>
MW1	90.40%	0.662	0.503	<i>Poor Classification</i>
PC1	91.46%	0.597	0.598	<i>Failure Classification</i>
PC2	97.51%	0.499	0.463	<i>Failure Classification</i>
PC3	84.71%	0.599	0.591	<i>Failure Classification</i>
<b>PC4</b>	<b>86.93%</b>	<b>0.726</b>	<b>0.789</b>	<b><i>Fair Classification</i></b>
PC5	73.91%	0.662	0.673	<i>Poor Classification</i>
<b>Rata-rata</b>	<b>82.95%</b>	<b>0.599</b>	<b>0.604</b>	<b><i>Failure Classification</i></b>

Rata-rata performa hasil yang didapatpun sangat buruk yaitu 82.95% untuk akurasi, AUC 0.599 dan ROC Area 0.604 dengan tingkatan level klasifikasi berada pada level *Failure Classification*. Jika mengacu pada Pedoman umum yang digunakan untuk klasifikasi yaitu:

**Tabel 3.** Klasifikasi akurasi

NO	NILAI AUC	KLASIFIKASI
1	0.90-1.00	<i>Excellent Classification</i>
2	0.80-0.90	<i>Good Classification</i>
3	0.70-0.80	<i>Fair Classification</i>
4	0.60-0.70	<i>Poor Classification</i>
5	0.50-0.60	<i>Failure</i>

Hasil pengujian awal didapat performa yang tidak begitu bagus dari *classifier C4.5* tersebut hal ini dikarenakan pada 12 dataset tersebut terdapat *imbalance class* atau ketidakseimbangan kelas, karena kelas mayoritas (*not defect*) lebih dominan dibandingkan dengan kelas minoritas (*defect*), hal ini dapat menurunkan kinerja dari klasifikasi prediksi cacat *software*. Berikut adalah data *imbalance class* yang terdapat pada 12 dataset Nasa MDP Repository (*D''*).

**Tabel 4.** Data *imbalance class* pada 12 dataset NASA

DATASET	SPESIFIKASI			
	Modul	Defect	Not Defect	Imbalance Class
CM1	327	42	285	243
JM1	7720	1612	6108	4496
KC1	1162	294	868	574
KC3	194	36	158	122
MC1	1952	36	1916	1880
MC2	124	44	80	36
MW1	250	25	225	200
PC1	679	55	624	569
PC2	722	16	706	690
PC3	1053	130	923	793
PC4	1270	176	1094	918
PC5	1694	458	1236	778

Dari data tersebut yang disajikan pada **Tabel 4.** bisa dilihat bahwa nilai ketidakseimbangan kelas pada 12 dataset *Software Metrics* Nasa MDP Repository sangat begitu besar dan ini yang menyebabkan kinerja awal pengklasifikasi tidak bekerja begitu maksimal. perbedaan antara data *defect* dan *not defect* terlihat sangat jelas, maka dari itu perlu adanya tahapan *preprocessing data* atau pendekatan level data untuk mengatasi *imbalance class* dan pendekatan level algoritma *ensemble (Bagging)* untuk mengatasi *imbalance class* dan membantu meningkatkan kinerja *classifier* utama yaitu *C4.5*. Tahapan *Preprocessing Data* yang akan dilakukan menggunakan teknik *oversampling (SMOTE)* yaitu dengan cara menggabungkan teknik *oversampling* (menaikan kelas minoritas) dan *undersampling* (menurunkan kelas mayoritas).

*Synthetic Minority Over-sampling Technique (SMOTE)* Menaikan Jumlah Modul sesuai dengan nilai pertambahan kelas minoritas, dan menaikan Kelas Minoritas (Kelas Cacat) sebesar 100% dari jumlah kelas minoritas sebelumnya, sedangkan kelas mayoritas (kelas tidak cacat) tetap dengan nilai yang sama.

### 3. Pengujian Model **SMOTE+BAGGING+C4.5**

*Classifier* yang digunakan dalam penelitian ini yaitu algoritma klasifikasi C4.5. Pada model pengujian C4.5 akan dilakukan model pengujian dengan integrasi teknik pendekatan level data SMOTE dengan Algoritma *ensemble Bagging* untuk meningkat kinerja klasifikasi C4.5 dalam mengatasi *imbalance class* pada prediksi cacat *software*. Berikut adalah hasil dari dataset yang belum melalui tahapan *preprocessing data* (Original) dan dataset yang sudah melakukan tahapan *preprocessing* yang berpengaruh signifikan terhadap keseimbangan kelas yang berhasil meningkatkan kelas minoritas.

**Tabel 5.** Hasil *preprocessing data*

DATASET	ORIGINAL			SMOTE		
	Modul	Defect	Not Defect	Modul	Defect	Not Defect
CM1	327	42	285	369	84	285
JM1	7720	1612	6108	9332	3224	6108
KC1	1162	294	868	1456	588	868
KC3	194	36	158	230	72	158
MC1	1952	36	1916	1988	72	1916
MC2	124	44	80	168	88	80
MW1	250	25	225	275	50	225
PC1	679	55	624	734	110	624
PC2	722	16	706	738	32	706
PC3	1053	130	923	1183	260	923
PC4	1270	176	1094	1446	352	1094
PC5	1694	458	1236	2152	916	1236

Setelah melalui tahapan *preprocessing data* dataset terlihat ada berbagai perbedaan diantaranya ada yang meningkat kelas minoritasnya, menurun kelas mayoritasnya dan ada yang membuat kelas mayoritas menurun secara drastis dan membuat dataset tampil secara seimbang antara kelas minoritas dan mayoritas. Selanjutnya dilakukan pengujian dengan *classifier Bagging* berbasis C4.5 untuk mengatasi masalah *imbalance class* dengan mencari perbandingan akurasi dan AUC yang didapat dari model pengujian. Tabel 6 adalah hasil pengujian model SMOTE+Bagging+C4.5 beserta penjelasannya.



**Tabel 6** Hasil kinerja SMOTE+Bagging+C4.5

Dataset	C4.5 (Original)			Kinerja	SMOTE+BG+C4.5			Kinerja
	Akurasi	AUC	ROC Area		Akurasi	AUC	ROC Area	
CM1	81.04%	0.536	0.57	Failure	82.93%	0.705	0.848	Fair
JM1	77.53%	0.582	0.653	Failure	80.72%	0.763	0.838	Fair
KC1	74.18%	0.591	0.604	Failure	78.57%	0.768	0.842	Fair
KC3	79.38%	0.616	0.653	Poor	83.04%	0.775	0.822	Fair
MC1	97.90%	0.54	0.566	Failure	97.03%	0.604	0.941	Poor
MC2	60.48%	0.576	0.589	Failure	79.17%	0.791	0.861	Fair
MW1	90.40%	0.662	0.503	Poor	86.91%	0.733	0.808	Fair
PC1	91.46%	0.597	0.598	Failure	89.92%	0.746	0.922	Fair
PC2	97.51%	0.499	0.463	Failure	96.48%	0.609	0.919	Poor
PC3	84.71%	0.599	0.591	Failure	85.04%	0.756	0.897	Fair
<b>PC4</b>	<b>86.93%</b>	<b>0.726</b>	<b>0.789</b>	<b>Fair</b>	<b>88.45%</b>	<b>0.849</b>	<b>0.948</b>	<b>Good</b>
PC5	73.91%	0.662	0.673	Poor	80.44%	0.797	0.872	Fair
<b>Rata-rata</b>	<b>82.95%</b>	<b>0.599</b>	<b>0.604</b>	<b>Failure</b>	<b>85.73%</b>	<b>0.741</b>	<b>0.877</b>	<b>Fair</b>

Dari hasil pengujian model SMOTE+Bagging+C4.5 hampir pada semua dataset mengalami peningkatan baik dari akurasi, AUC dan ROC Area. Hasil terbaik dari didapat pada dataset PC4.arff dengan nilai akurasi 88.45%, AUC 0.849, ROC Area 0.948 dengan hasil tingkatan level akurasi "Good Classification". Untuk rata-rata dari pengujian model ini yaitu akurasi meningkat sebesar 2.77%, AUC meningkat 0.142, ROC Area meningkat 0.273 dengan nilai rata-ratanya yaitu 85.73% untuk akurasi, 0.741 untuk nilai AUC, 0.877 untuk nilai ROC Area dan rata-rata pengujian kinerja model ini berada pada level "Fair Classification" lebih baik dari model C4.5 yang berada pada level "Fair Classification". Hasil akurasi, AUC dan ROC Area pada **Tabel 6** didapat dengan perhitungan kinerja model menggunakan *Confusion Matrix* untuk mencari akurasi, sensitivitas/*recall*/TPrate, *specificity*/TNrate, FPrate, FNrate, *Precision*/PPV, *F-Measure*, *G-Mean* dan AUC. Hasil perhitungan diperoleh probabilitas sebagai berikut:

**Tabel 7.** Hasil *Confusion Matrix* Pada Dataset PC4.arff

Class		Actual	
		Defective	Not Defective
Prediction	Defective	True Positive(TP) <b>275</b>	False Negative(FN) <b>77</b>
	Not Defective	False Positive(FP) <b>90</b>	True Negative(TN) <b>1004</b>

Perhitungan *confusion matrix* menggunakan persamaan berikut:

$$\begin{aligned} \text{Akurasi} &= \frac{TP+TN}{TP + TN + FP + FN} \\ &= (275+1004) / (275+1004+90+77) \\ &= 1279 / 1446 \\ &= 0.8845 \rightarrow 88.45\% \end{aligned}$$

$$\begin{aligned} \text{Sensitivitas} = \text{recall} = TP_{\text{rate}} &= \frac{TP}{TP + FN} \\ &= (275) / (275+77) \\ &= 275 / 352 \\ &= 0.7812 \rightarrow 78.12\% \end{aligned}$$

$$\begin{aligned} \text{Specificity} = TN_{\text{rate}} &= \frac{TN}{TN + FP} \\ &= (1004) / (1004+90) \\ &= 1004 / 1094 \\ &= 0.9177 \rightarrow 91.77\% \end{aligned}$$

$$\begin{aligned} FP_{\text{rate}} &= \frac{FP}{FP + TN} \\ &= (90) / (90+1004) \\ &= 90 / 1094 \\ &= 0.0822 \rightarrow 8.22\% \end{aligned}$$

$$\begin{aligned} FN_{\text{rate}} &= \frac{FN}{TP + FN} \\ &= (77) / (275+77) \\ &= 77 / 352 \\ &= 0.2187 \rightarrow 21.87\% \end{aligned}$$

$$\begin{aligned} \text{Precision} = PPV &= \frac{TP}{TP + FP} \\ &= (275) / (275+90) \\ &= 275 / 365 \\ &= 0.7534 \rightarrow 75.34\% \end{aligned}$$

$$\begin{aligned} NPV &= \frac{TN}{TN + FN} \\ &= (1004) / (1004+77) \\ &= 1004 / 1081 \\ &= 0.9287 \rightarrow 92.87\% \end{aligned}$$

P-ISSN 2580 - 7781

E-ISSN 2615 - 3238

$$\begin{aligned}
 F - Measure &= \frac{2 \times recall \times precision}{(recall + precision)} \\
 &= (2 \times 0.781 \times 0.753) / (0.781 + 0.753) \\
 &= 1.176 / 1.534 \\
 &= 0.766
 \end{aligned}$$

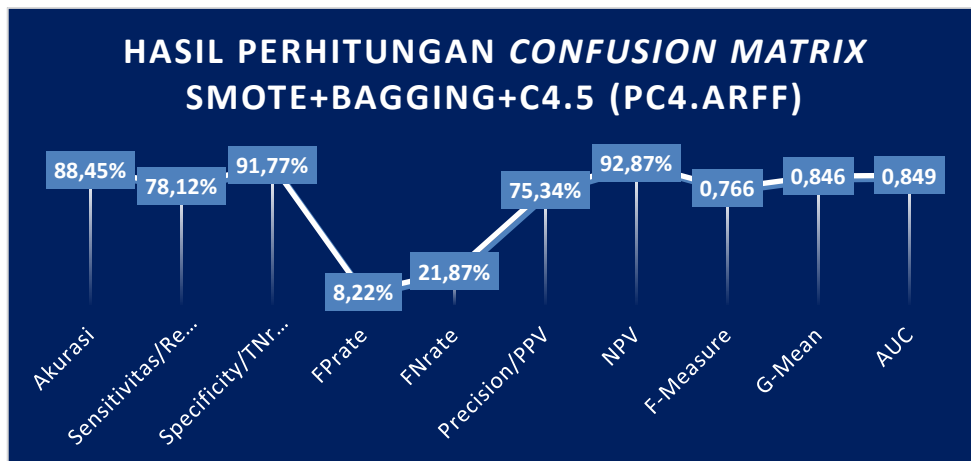
$$\begin{aligned}
 G - Mean &= \sqrt{Sensitivitas \times Specificity} \\
 &= \sqrt{0.781 \times 0.917} \\
 &= \sqrt{0,716177} \\
 &= 0.846
 \end{aligned}$$

$$\begin{aligned}
 AUC &= \frac{1 + TP_{rate} - FP_{rate}}{2} \\
 &= (1 + 0.781 - 0.082) / 2 \\
 &= 1.699 / 2 \\
 &= 0.849
 \end{aligned}$$

Hasil keseluruhan perhitungan *confusion matrix* pada dataset PC4.arff diperoleh nilai sebagai berikut

**Tabel 8.** Hasil perhitungan *confusion matrix* pada dataset PC4.arff

DATASET PC4.Arff				
SMOTE+BAGGING+C4.5	TP	FP	FN	TN
	275	90	77	1004
Akurasi	88.45%			
Sensitivitas/Recall/TPrate	78.12%			
Specificity/TNrate	91.77%			
FPrate	8.22%			
FNrate	21.87%			
Precision/PPV	75.34%			
NPV	92.87%			
F-Measure	0.766			
G-Mean	0.846			
AUC	0.849			



**Gambar 3.** Perhitungan *confusion matrix* pada dataset PC4.arff

Hasil yang didapat pada *Line Diagram* perolehan nilai dari 10 variabel yang dicari pada perhitungan diatas menunjukkan bahwa kinerja model SMOTE+Bagging+C4.5 dengan *sample* perhitungan dataset PC4.arff mencapai nilai akurasi 88.45% dan AUC 0.849 yang artinya dari 12 dataset yang ada, performa terbaik didapatkan pada dataset ini dengan tingkatan level klasifikasi berada pada level “*Good Classification*” karena nilai AUC yang didapat berada pada *range* nilai 0.80-0.90, sedangkan untuk hasil perhitungan *confusion matrix* secara keseluruhan dari 12 dataset Nasa MDP *Repository*.

**Tabel 9.** Hasil Kinerja SMOTE+Bagging+C45 Pada 12 Dataset (*Confusion Matrix*)

KINERJA	DATASET											
	CM1	JM1	KC1	KC3	MC1	MC2	MW1	PC1	PC2	PC3	PC4	PC5
<b>Akurasi</b>	82.93%	80.72%	78.57%	83.04%	97.03%	79.17%	86.91%	89.92%	96.48%	85.04%	<b>88.45%</b>	80.44%
TPrate	47.62%	61.94%	67.52%	62.50%	20.83%	80.68%	52.00%	52.73%	21.88%	58.85%	78.13%	75.11%
TNrate	93.33%	90.64%	86.06%	92.41%	99.90%	77.50%	94.67%	96.47%	99.86%	92.42%	91.77%	84.39%
FPrate	6.67%	9.36%	13.94%	7.59%	0.10%	22.50%	5.33%	3.53%	0.14%	7.58%	8.23%	15.61%
FNrate	52.38%	38.06%	32.48%	37.50%	79.17%	19.32%	48.00%	47.27%	78.13%	41.15%	21.88%	24.89%
<i>Precision</i>	67.80%	77.73%	76.64%	78.95%	88.24%	79.78%	68.42%	72.50%	87.50%	68.61%	75.34%	78.09%
NPV	85.81%	81.86%	79.64%	84.39%	97.11%	78.48%	89.87%	92.05%	96.58%	88.85%	92.88%	82.06%
<i>F-Measure</i>	0.559	0.689	0.718	0.698	0.337	0.802	0.591	0.611	0.350	0.634	0.767	0.766
<i>G-Mean</i>	0.667	0.749	0.762	0.760	0.456	0.791	0.702	0.713	0.467	0.737	0.847	0.796
<b>AUC</b>	0.705	0.763	0.768	0.775	0.604	0.791	0.733	0.746	0.609	0.756	<b>0.849</b>	0.797
ROC Area	0.848	0.838	0.842	0.822	0.941	0.861	0.808	0.922	0.919	0.897	0.948	0.872

Dari hasil 11 variabel yang didapat, penelitian ini hanya berfokus pada perbandingan nilai akurasi dan AUC untuk menangani *imbalance class* pada prediksi cacat *software*. Untuk lebih memudahkan hasil evaluasi yang didapat dari setiap hasil model pengujian akan dibandingkan kinerja perbandingannya antara model C4.5 original dan model integrasi pengujian untuk melihat sejauh mana

grafik dan hasil peningkatan yang didapat sebelum dan sesudah dilakukannya integrasi model pengujian. Berikut adalah evaluasi model dari pengujian C4.5 dan model integrasi SMOTE+Bagging+C4.5.

**Tabel 10.** Evaluasi Kinerja C4.5 dan SMOTE+Bagging+C4.5

NILAI	C4.5		SMOTE+BAGGING+C4.5		(%) PENINGKATAN		KINERJA	
	AKURASI	AUC	AKURASI	AUC	AKURASI	AUC	C4.5	SMOTE+BAGGING+C4.5
Minimum	60.48%	0.499	78.57%	0.604	18.09%	0.105	<i>Failure Classification</i>	<i>Fair Classification</i>
Q1	76.69%	0.567	80.65%	0.726	3.96%	0.159		
Median	82.88%	0.594	85.73%	0.760	2.85%	0.166		
Q3	90.67%	0.6275	89.55%	0.779	-1.11%	0.152		
Maximum	97.90%	0.726	97.03%	0.849	-0.87%	0.123		
<b>Mean</b>	<b>82.95%</b>	<b>0.599</b>	<b>85.73%</b>	<b>0.741</b>	<b>2.77%</b>	<b>0.143</b>		
Range	37.42%	0.227	18.46%	0.245	-18.96%	0.018		

Hasil evaluasi menunjukkan terdapat peningkatan dengan dilakukannya model pengujian integrasi SMOTE+Bagging+C4.5. Rata-rata Akurasi dan AUC meningkat dari model sebelumnya dimana akurasi meningkat sebesar 2.77% dan AUC meningkat sebesar 0.143 dan hasil tersebut berpengaruh terhadap tingkatan level kinerja klasifikasi yang didapat yang pada awalnya berada pada level “*Failure Classification*” karena berada pada rata-rata nilai AUC 0.50-0.60. Setelah dilakukannya integrasi pengujian model SMOTE+Bagging+C4.5 kinerja klasifikasi meningkat pada level “*Fair Classification*” dengan nilai rata-rata AUC 0.741.

**KESIMPULAN**

Pendekatan level data Synthetic Minority Over-sampling Technique dan Bagging diusulkan untuk meningkatkan kinerja *classifier* C4.5 pada prediksi cacat *software* untuk menangani *imbalance class*. Model yang diusulkan diterapkan pada 12 NASA (*National Aeronautics and Space Administration*) MDP (*Metrics Data Program*) *repository* sebagai *software metrics* pada prediksi cacat *software*. Hasil penelitian menunjukkan bahwa model yang diusulkan mencapai akurasi dan AUC klasifikasi yang baik. Pada Model Synthetic Minority Over-sampling Technique (SMOTE)+Bagging+C4.5 rata-rata Akurasi dan AUC meningkat dari model sebelumnya dimana akurasi meningkat sebesar 2.77% dan AUC meningkat sebesar 0.143 dan hasil tersebut berpengaruh terhadap tingkatan level kinerja klasifikasi yang didapat yang pada awalnya berada pada level “*Failure Classification*” karena berada pada rata-rata nilai AUC 0.50-0.60. Setelah

dilakukannya integrasi pengujian model SMOTE+Bagging+C4.5 kinerja klasifikasi meningkat pada level “*Fair Classification*” dengan nilai rata-rata AUC 0.741. Model yang diusulkan yaitu *Synthetic Minority Over-sampling Technique (SMOTE)+Bagging+C4.5* mengalami peningkatan dalam penelitian prediksi cacat *software* untuk menangani *imbalance class*.

#### DAFTAR PUSTAKA

- Amir, R. F., Sobari, I. A., & Rousyati, R. (2020). Penerapan PSO Over Sampling Dan Adaboost Random Forest Untuk Memprediksi Cacat Software. *Indonesian Journal on Software Engineering (IJSE)*, 6(2), 230–239. <https://doi.org/10.31294/ijse.v6i2.9258>
- Hall, T., Beecham, S., Bowes, D., Gray, D., & Counsell, S. (2012). A Systematic Literature Review on Fault Prediction Performance in Software Engineering. *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, 38(6), 1276–1304.
- Hardoni, A., Rini, D. P., & Sukemi, S. (2021). Integrasi SMOTE pada Naive Bayes dan Logistic Regression Berbasis Particle Swarm Optimization untuk Prediksi Cacat Perangkat Lunak. *Jurnal Media Informatika Budidarma*, 5(1), 233. <https://doi.org/10.30865/mib.v5i1.2616>
- Hidayati, N., Suntoro, J., & Setiaji, G. G. (2021). Perbandingan Algoritma Klasifikasi untuk Prediksi Cacat Software dengan Pendekatan CRISP-DM. *Jurnal Sains Dan Informatika*, 7(2), 117–126. <https://doi.org/10.34128/jsi.v7i2.313>
- 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>
- Prasetyo, R., Nawawi, I., Fauzi, A., & Ginabila, G. (2021). Komparasi Algoritma

- Logistic Regression dan Random Forest pada Prediksi Cacat Software. *Jurnal Teknik Informatika UNIKA Santo Thomas*, 06(Siringoringo 2017), 275–281. <https://doi.org/10.54367/jtiust.v6i2.1522>
- Putri, S. A. (2019). Prediksi Cacat Software Dengan Teknik Sampel Dan Seleksi Fitur Pada Bayesian Network. *Jurnal Kajian Ilmiah*, 19(1), 17. <https://doi.org/10.31599/jki.v19i1.314>
- Sugiono, Taufik, A., & Faizal Amir, R. (2020). Penerapan Penerapan Teknik Pso Over Sampling Dan Adaboost J48 Untuk Memprediksi Cacat Software. *Jurnal Responsif: Riset Sains Dan Informatika*, 2(2), 198–203. <https://doi.org/10.51977/jti.v2i2.249>
- Wahono, R. S. (2015). A Systematic Literature Review of Software Defect Prediction : Research Trends , Datasets , Methods and Frameworks. *Journal of Software Engineering*, 1(1).