Penanganan Data Tidak Seimbang Menggunakan *Hybrid Method Resampling* Pada Algoritma *Naive Bayes* Untuk *Software Defect Prediction*

Moch. Lutfi[*], Arief Tri Arsanto[**], Muhammad Faishol Amrulloh [***], Ummi Kulsum[****]
Program Studi Teknik Informatika, Universitas Yudharta Pasuruan
*moch.lutfi@yudharta.ac.id,**arief_inf@yudharta.ac.id , ***faishol@yudharta.ac.id, ****ummi1289@gmail.com

ABSTRACT

Software defect prediction is *software* data that is used to identify a *software* module and can also be used to predict *software* defects. Before carrying out further trials, it is necessary to carry out special handling, especially by using algorithm models as predictions of *software* defects with the aim of obtaining information from the device being developed. Therefore, it is necessary to predict *software* defects using appropriate classification and prediction methods, so that the resulting accuracy results are better. In this study, the *naïve bayes* algorithm was used as a classification with a resampling technique approach to handle unbalanced data, including SMOTEENN and SMOTETomek. The best accuracy results in the research conducted were 92.5% on the Nasa Repository PC4 dataset.

Keyword: *Software defect, Hybrid Resampling, Naive Bayes*

1. Pendahuluan

Software atau perangkat lunak yang dikembangan tentu didalamnya memiliki *bug*, *bug* tersebut tidak bisa terlihat dengan kasat mata atau fungsi program tidak bisa dijalankan sesuai fungsinya. Peningkatan kualitas suatu program yang dikembang dapat dilakukan dengan perbaikan terus menurus agar *software* yang dikembangan berjalan dengan baik. *software* yang berkualitas apabila *software* tersebut tidak ditemukan cacat pada tahap pengujian[1]. Prediksi cacat *software* dapat digunakan sebagai identifikasi suatu modul perangkat lunak yang rentan terhadap kecacatan, untuk meningkatkan kualitas pengujian perangkat lunak maka perlu adanya *tools* bantu yang dapat digunakan untuk prediksi cacat dari perangkat lunak. Namun melakukan perawatan perangkat lunak yang dikembangan membutuhkan biaya yang cukup tinggi[2]. Oleh karena itu diperlukan solusi untuk meminimalisir biaya pengembangan agar lebih cepat dan murah salah satunya dengan menggunakan pengujian algoritma.

Meminimalisir biaya pengembangan perangkat lunak pada proyek yang dihasilkan maka dapat meningkatkan kualitas proyek yang dikerjakan. Secara tradisional, proyek pengembangan perangkat lunak berkualitas tinggi adalah proyek yang tidak memakan biaya *cost drivers*[3] selama pengembangan perangkat lunak yang meliputi dari produk, perangkat keras, sumberdaya manusia dan proyek. Selama proses pengujian serta dapat memberikan penilaian kepada *user* sebagai dasar untuk memenuhi kebutuhan meraka[1]. Proses pemeriksaan dan pengujian dilakukan dengan alur input output dari perangkat lunak sehingga dapat diukur dengan subjektif berdasarkan harapan dan kebutuhan *customer*.

Pengujian *software* merupakan bagian proses tahap development perangkat lunak yang banyak membutuhkan waktu dan juga karena tingginya biaya, oleh sebab itu kurang lebih dari 50% jadwal project pengembangan perangkat lunak digunakan untuk pengujian maupun pemeriksaan[4]. Proses pengembangan *software* dan proses *management project* adalah ranah penelitian dalam bidang *software engineering*[5]. Kesempurnaan perangkat lunak yang dikembangan merupakan hal yang penting, maka perlu adanya prosedur pengujian yang baik untuk menghindari over time maupun over cost selama proses pengembangan perangkat lunak. Prosedur yang baik adalah strategi pengujian yang efektif maupun efisien untuk mengurangi perkiraan biaya pengembangan perangkat lunak, pada prediksi modul cacat lebih membutuhkan fokus dari pada prediksi modul tidak cacat[6].

Sebelum dilakukan uji coba lebih lanjut perlu dilakukan penanganan khusus terutama dengan menggunakan model-model algoritma sebagai prediksi cacat *software* dengan tujuan untuk mendapatkan informasi dari perangkat yang dikembangkan. Oleh karena itu prediksi cacat *software* perlu dilakukan dengan model klasifikasi maupun prediksi dengan harapan hasil akurasi yang terbaik. Pada penelitian *software* defect prediction ini difokuskan pada model klasifikasi dengan menggunakan algoritma *naïve bayes* pada dataset tidak seimbang (*imbalanced*), hal ini didasarkan penyesuaian attribut dan label pada penerapan algoritma *naïve bayes* dan juga penelitian sebelumnya memakai algoritma yang sama memiliki hasil akurasi yang tinggi seperti

yang pernah dilakukan oleh Menzies [7], Lessman [5]. Penelitian yang dilakukan [8] mengumpulkan 527 paper yang membahas tentang ketidak seimbangan kelas dan metode yang dibahas meliputi teknik seperti data preprocessing, algoritma klasifikasi dan evaluasi model berbasis *ensemble*. Pada penelitian yang dilakukan [9] resampling terhadap data yang tidak seimbang. Melalui kombinasi prosedur *oversampling* dan *undersampling* yang baru, memungkinkan dapat meningkatkan akurasi klasifikasi. Metode yang diusulkan pada penelitian ini bernama sundo, dimana metode ini telah diuji pada contoh kasus yang berbeda, di mana hasil yang diperoleh dibandingkan dengan metode lain dan hasil dari metode yang diusulkan dapat mengatasi data tidak seimbang secara signifikan.

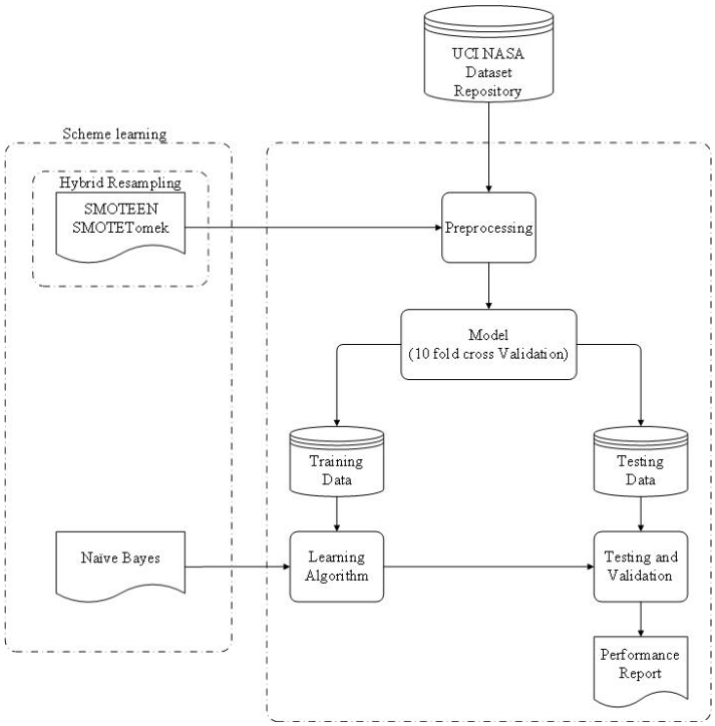
Díez-Pastor dkk [10] mengusulkan pendekatan baru untuk mengklasifikasikan dua kelas dataset yang tidak seimbang yang disebut *Random Balance*. Setiap anggota *ensamble Random Balance* dilatih dengan data latih dan ditambah dengan contoh buatan yang diperoleh dengan menggunakan *SMOTE*. Dalam penelitian [11] mengusulkan algoritma pengambilan sampel hibrid yang disebut *RFMSE*, yang menggabungkan orientasi Misklasifikasi Teknik pengambilan sampel berlebih minoritas sintetis (*M-SMOTE*) dan tetangga terdekat yang diedit (*ENN*) berdasarkan *Random Forest* (*RF*). *M-SMOTE* digunakan untuk meningkatkan jumlah sampel pada kelas minoritas, sedangkan tingkat over-sampling *M-SMOTE* adalah tingkat kesalahan klasifikasi dari RF. Kemudian *ENN* digunakan untuk menghilangkan *noise* dari sampel mayoritas. Dan RF digunakan untuk melakukan prediksi klasifikasi untuk sampel setelah pengambilan *sample hybrid*, dan kriteria penghentian untuk iterasi ditentukan sesuai dengan perubahan indeks klasifikasi (*Matthews Correlation Coefficient* (*MCC*)) dan ketika nilai *MCC* terus turun, proses iterasi akan dihentikan.

Pada penelitian yang diusulkan J. A. Sáez dkk, [12] perluasan *SMOTE* melalui elemen baru, *filter noise* berbasis ensemble iteratif atau Filter Partisi Iteratif (*IPF*), yang dapat mengatasi masalah yang dihasilkan oleh contoh *noise* dan *borderline* dalam kumpulan data yang tidak seimbang. Eksperimen dilakukan pada kumpulan data tidak seimbang sintetis dengan berbagai bentuk kelas minoritas dan rasio ketidakseimbangan. Penelitian yang dilakukan [13] memaparkan bahwa penelitian ini difokuskan pada pengujian kualias *Software* dengan menggunakan teknik sampling seperti *RUS* (*Random Undersampling*) dan juga *SMOTE* (*Synthetic Minority Over-sampling Technique*). Pada Dataset yang diuji ditemukan data tidak seimbang juga *noise* sehingga peneliti menggunakan algoritma *Chi Square* dan juga *Information Gain* sebagai *preprocessing* data. Data yang digunakan dalam penelitian ini merupakan dataset *public* yang bersumber dari NASA MDP dengan menggunakan 4 Dataset sekaligus, seperti CM1, MW1, PC1 dan juga PC4. Kemudian data tersebut dibagi kedalam 2 bagian besar untuk dijadikan data sampel dan training. Dalam pengujian validasi peneliti menggunakan *confusion matrix* dengan 10 *fold cross validation*. Sedangkan penelitian yang dilakukan [14] identifikai dengan menggunakan *multi objective naive bayes*. Dataset *Software Defect* memiliki masalah ketidakseimbangan kelas, yang menunjukkan bahwa kelas cacat memiliki lebih sedikit *instance* dari pada kelas tidak cacat. Sehingga diusulkan dalam penelitian ini teknik *Learning Multi-Object Naive Bayes* yang dimodelkan oleh algoritma *Harmony Search meta-heuristic*. Dan penelitian yang dilakukan oleh[15] melakukan independensi fitur dari metode *Naive Bayes* dan mengusulkan sebuah pengklasifikasiam yang dimodifikasi metode FDNB. Hasil yang diperoleh menunjukkan kinerja yang lebih baik dari metode yang diusulkan di atas *Naive Bayes* standar dengan fitur subset seleksi *preprocessing* dan variasi *Naive Bayes* lainnya dari fitur berbobot karakteristik. Selain itu, metode FDNB (*Feature Dependent Naive Bayes*) menunjukkan kinerja yang kompetitif dengan PCA-NB yang diproses sebelumnya.

Pada penelitian ini dilakukan klasifikasi dengan metode *naïve bayes* dan juga menerapkan teknik *hybrid resampling* (*SMOTEENN* dan *SMOTETomek*) untuk menangani data tidak seimbang (*imbalanced*). Metode *SMOTEENN* [8]merupakan gabungan metode *SMOTE* dan *ENN* yang digunakan untuk menghasilkan data seimbang, pada bagian metode ini dilakukan *undersampling* pada class *majority* yang memiliki label berbeda pada data yang berdekatan. Sedangkan metode *SMOTETomek* [8] merupakan metode proses pengurangan *sample* data terhadap data yang saling berdekatan antara *class minority* dan *class majority*. Dalam penelitian ini diusulkan metode *hybrid resampling* untuk menangani ketidak seimbangan class pada dataset *nasa repository*, sedangkan untuk klasifikasi hasil digunakan algoritma *naïve bayes*.

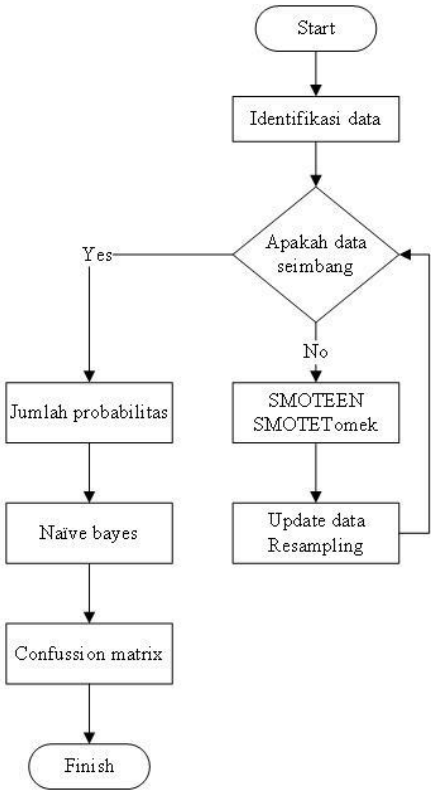
2. Metode Penelitian

Penelitian ini dilakukan dengan mengusulkan metode *Resampling* pada *NASA Promise Repository*. Untuk menangani data yang tidak seimbang digunakan teknik resampling (*SMOTEENN* dan *SMOTETomek*) dan algoritma *Naive Bayes*. Validasi menggunakan 10-*fold cross validation*. Hasil pengukuran dilakukan dengan *confusion matrix*. Gambar 1 kerangka kerja metode yang diusulkan.



Gambar 1. kerangka kerja metode yang diusulkan

Pada metode yang diusulkan dataset yang diinput dibagi menjadi data latih dan data uji. Pengujian data digunakan sebanyak 10 kali untuk data uji dan sebanyak 10 kali untuk data latih dari keseluruhan dataset. Kemudian dilakukan praproses dengan menggunakan *SMOTEENN* dan *SMOTETomek* untuk mendapatkan hasil terbaik. Berikut ini *flowchart* metode yang diusulkan ditunjukan gambar 2.



Gambar 2. *flowchart* metode yang diusulkan

Tujuan digunakan *Naive Bayes* adalah untuk mengklasifikasikan dan juga memprediksi dataset agar mampu mendapatkan hasil yang lebih baik. Pada metode yang diusulkan didapatkan data probabilitas terdekat sehingga akan menunjukkan *f-measure* yang rendah. Proses klasifikasi *Naive Bayes* dengan *hybrid resampling* yaitu :

1. Memulai dengan mengidentifikasi data
2. Menyeimbangkan dataset dengan metode *resampling* dan melakukan *update* data
3. Jika data sudah seimbang maka dataset akan dibentuk tabel probabilitas dengn jumlah tabel probabilitas pada dataset yang seimbang
4. Melakukan proses learning dengan algoritma *naive bayes* dengan menghitung jumlah *Mean* dan standar deviasi setiap parameter
5. Melakukan validasi dengan menggunakan *confusion matrix*

Teknik *resampling* menggunakan *SMOTEEN* menggunakan rumus sebagai berikut :

$$D_{new} = D_i + (D_1 + D_i) \times \delta \quad (1)$$

dimana :

$D_i \in S_{min}$ = minority class example

D_i = salah satu tetangga K terdekat D_i dimana

$\delta \epsilon [0,1]$ = angka random

persamaan dapat dinyatakan dengan :

$$D_{new} = D_i + (D_1\delta + D_i\delta) = D_i - D_1\delta + D_i\delta = (1-\delta)\, D_i + \delta D_1 \quad (2)$$

Jika hanya ada dua persamaan tetangga terdekat maka:

$D_{new2} = \delta D_i + \delta D_i + \delta_1 D_1 + \delta_2 D_2$

$D_{new2} = (1\text{-}\delta_1 - \delta_2) D_i + \delta_1 D_1 \ldots\ldots \delta_k D_k$

$D_{new;k} = D_i + \sum_{j=1}^{k} (D_k\text{-}D_1)\delta_k$

$dimana\ \delta_1, \delta_2 \delta_3 \ldots\ldots \delta_k$ adalah angka random dengan $\sum \delta_i = 1\ maka$:

$\boldsymbol{\delta \in [0, 1]}$

$\boldsymbol{\delta_1 \in [0, 1 - \delta]}$

$\boldsymbol{\delta_2 \in [0, 1 - \delta - \delta_1\ ]}$

$\boldsymbol{\ldots}$

$\boldsymbol{\delta_k \in [0, 1 - \delta - \delta_k}$

SMOTEENN (*Synthetic Minority Oversampling Thechnique Edited Nearest Neighbours*) merupakan teknik *resampling* yang mampu menangani kelas yang *noisy* dan tidak seimbang. Proses yang dilakukan dalam metode ini yaitu dengan menggabungkan metode *oversampling SMOTE* yang digunakan untuk menangani *imbalanced* data, kemudian akan dikurangi dengan menggunakan metode *undersampling* ENN. Metode ini dinilai efektif dalam menangani *noisy* [16] dan juga imbalanced data [17].

SMOTETomek (*Synthetic Minority Oversampling Thechnique Tomek*) merupakan teknik *resampling* yang digunakan untuk menangani kelas tidak seimbang dan *noisy* atribut. Jika setiap tetangga data terdekat memiliki kelas atau label yang berbeda dari data, maka data mayor dihapus dan dianggap *noise* atau *misclassification*. Diberikan dua sampel x dan z milik kelas yang berbeda, dan d(x,z) adalah jarak antara x dan z. Sepasang (x,z) disebut *Tomek Links* jika tidak ada sampel z*,sehingga d( ) < d( ) atau d( ) < d( ) [18] . Jika dua sampel membentuk *links Tomek*, maka salah satu dari dua sampel tersebut adalah data yang noise atau kedua sampel adalah borderline.

Pengklasifikasian *Naive Bayes* dapat dihitung terlebih dahulu dengan mencari nilai rata-rata (μ) dan standar *deviasi* (σ) untuk setiap attribut. Rumus *naive bayes* adalah sebagai berikut :

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (3)$$

dimana μ adalah nilai rata-rata, σ adalah standar deviasi, $\sum_{i=1}^{n} X_i$ adalah nilai jumlah rata-rata , *n* adalah jumlah keseluruhan data. Untuk mengukur kinerja digunakan *confusion matrix*. *Confusion matrix* memberikan keputusan yang diperoleh dalam pelatihan dan pengujian [19]. Pengukuran Kinerja untuk permasalahan dalam klasifikasi, pengukuran yang biasa digunakan adalah *precision*, *recall* dan *accuracy* [20]. Seperti pada penelitian yang dilakukan [21] jenis klasifikasi *binary* yang hanya memiliki 2 keluaran kelas, *confusion matrix* dapat disajikan seperti pada berikut ini :

Table 1. Tabel *Confusion Matrix*

Class	**Klasifikasi Positif**	**Klasifikasi Negatif**
Positif	TP (*True Positif*)	FN *(False Negatif*)
Negatif	FP (*False Positif*)	TN (*True Negatif*)

Rumus-rumus yang digunakan dalam perhitungan adalah sebagai berikut :

$$Accuracy=\frac{TP+TN}{TP+FP+FN+FP} \quad (4)$$

$$Sensitivity\ (TPRate/Recall) = \frac{tp}{tp+fn} \quad (5)$$

$$Specificity\ (\text{TNrate}) = \frac{tn}{tn+fp} \quad (6)$$

$$FPrate = \frac{fp}{fp+tn} \quad (7)$$

$$FNrate = \frac{fn}{tp+fn} \quad (8)$$

$$NPV = \frac{tn}{tn+fn} \quad (9)$$

$$Precision\ (Positive\ Predictive\ Value) = \frac{tp}{tp+fp} \quad (10)$$

$$\text{F-Measure} = \frac{2*(recall*precision)}{recall+precision} \quad (11)$$

$$G\text{-}Mean = \sqrt{sensitivity * specificity} \quad (12)$$

$$Error\ rate = \frac{fn+fp}{tp+tn+fp+fn} \quad (13)$$

3. Hasil Dan Analisis

Pengujian penelitian yang dilakukan menggunakan laptop dell dengan spesifikasi Core i7-6600U, Ram 8GB, SSD NVMe 256GB, dan Sistem Operasi Windows 10 Pro, *software development Annaconda Navigator* 3.8.

Table 2. UCI Dataset NASA Promise Repository dan jumlah attribut

Dataset	Jumlah attribut
CM1 Nasa *Repository*	21
KC1 Nasa *Repository*	21
KC3 Nasa *Repository*	39
MC2 Nasa *Repository*	39
PC3 Nasa *Repository*	42
PC4 Nasa *Repository*	42
Total	204

Pengukuran kinerja dilakukan dengan menggunakan 6 data *PROMISE Repository Dataset* yang digunakan dalam penelitian ini diambil langsung dari situs *NASA PROMISE Repository* (CM1, MC2, PC3, PC4, KC1, KC3). Model yang diuji adalah model klasifikasi algoritma *Naive Bayes* dan model *hybrid resampling* yang sudah dilakukan perbaikan terhadap *Naive Bayes* dan *SMOTENN* (NB+SMOTEENN) juga Naive Bayes + *SMOTETomek* (NB+SMOTETomek). Dalam percobaan ini mendapatkan nilai akurasi tertinggi 0.85 pada model NB+SMOTEENN dan 0.79 menggunakan NB+SMOTETomek dengan beberapa kali pengujian pada dataset CM1, KC1, KC3, PC4, dan MC2 . Rata-rata *Recall*, *Precision* dan *f-measure* memiliki kenaikan rata-rata 0.02 setelah dilakukan *Preprocessing*. Perbandingan hasil metode lain dengan metode penelitian yang diusulkan didapatkan nilai *Specificity* tertinggi pada model *NB+SMOTETomek* (0.925) pada dataset PC4. Namun tidak semua permasalahan imbalanced data dapat diselesaikan dengan tenik ini. Contoh kasus dapat dilihat pada dataset PC3 dimana proses Resampling menggunakan *SMOTEENN* dan *SMOTETomek* menurunkan *Sensitivity*.

Hasil pengukuran penerapan dataset *PROMISE Repository* ditunjukkan pada tabel 3 sampai 8. Sedangkan hasil rekap pengukuran akurasi model pada tabel 9, hasil pengurukan sensistivitas model terdapat pada tabel 10, hasil pengukuran *F-Measure* terdapat pada tabel 11, dan hasil pengukuran *G-Mean* terdapat pada tabel 12.

Table 3. Hasil pengukuran dataset CM1 *Nasa Promise Repository* menggunakan *NB+Hybrid Method Resampling*

Model	**TP**	**FP**	**FN**	**TN**	**Accuracy**	**Sensitivity**	**Specifitcity**	**PPV**	**NPV**	**F-Measure**	**G-Mean**
NB	11	20	10	6	80.00 %	37.5 %	8.0%	23.0 %	91.9 %	0.69 2	0.56 4

NB+SMOTENN	12	7	11	4	85%	26.6 %	93.6 %	36%	90.6 %	1.09 0	0.49 9
NB+SMOTETomek	99	4	9	13	81%	36.6 %	82.1 %	42%	92.0 %	0.83 0	0.52 1

Table 4. Hasil pengukuran dataset MC2 *Nasa Promise Repository* menggunakan *NB+Hybrid Method Resampling*

Model	TP	FP	FN	TN	Accuracy	Sensitivity	Specifitcity	PPV	NPV	F-Measure	G-Mean
NB	27	1	8	3	76.9%	27.2%	96.4%	75%	77.1%	2.25	0.512
NB+SMOTENN	21	0	6	5	81.2 %	45.45 %	100%		77%	3.0	0.674
NB+SMOTETomek	19	1	4	8	84.3 %	66.66%	95%	88%	82.6%	2.66	0.795

Table 5. Hasil pengukuran dataset KC1 *Nasa Promise Repository* menggunakan *NB+Hybrid Method Resampling*

Model	TP	FP	FN	TN	Accuracy	Sensitivity	Specifitcity	PPV	NPV	F-Measure	G-Mean
NB	48	55	59	36	81.9%	37.8%	89.7%	39.5%	0.89	1.186	0.583
NB+SMOTENN	41	33	59	28	82.9 %	32.1%	89%	87.3%	87%	1.377	0.545
NB+SMOTETomek	42e	37	45	27	84.6%	32.0%	91%	38.4%	0.885	1.153	0.540

Table 6. Hasil pengukuran dataset KC3 *Nasa Promise Repository* menggunakan *NB+Hybrid Method Resampling*

Model	TP	FP	FN	TN	Accuracy	Sensitivity	Specifitcity	PPV	NPV	F-Measure	G-Mean
NB	46	3	7	4	83.3%	36.3%	93.8%	57%	86%	1.712	0.582
NB+SMOTENN	38	4	2	6	88.0 %	75%	90%	60%	95%	1.799	0.823
NB+SMOTETomek	30	3	5	3	88.0 %	50%	93%	50%	93%	1.5	0.86

4.

Table 7. Hasil pengukuran dataset PC3 dari *Promise Repository* menggunakan *NB+Hybrid Method Resampling*

Model	TP	FP	FN	TN	Accuracy	Sensitivity	Specifitcity	PPV	NPV	F-Measure	G-Mean
NB	13	16	5	41	50.8%	89.1%	44.8%	20.3%	96%	0.608	0.632
NB+SMOTENN	19	64	7	22	74.8%	78.5%	74.7%	25.5%	94.5%	0.767	0.752
NB+SMOTETomek	19	52	17	22	79.0 %	56.1%	78.5%	29.7%	91.8%	0.891	0.665

Table 8. Hasil pengukuran dataset PC4 *Nasa Promise Repository* menggunakan *NB+Hybrid Method Resampling*

Model	TP	FP	FN	TN	Accuracy	Sensitivity	Specifitcity	PPV	NPV	F-Measure	G-Mean

NB	11	20	10	6	80.42 %	37.5 %	85.0 %	23.0 %	91.2 %	0.692	0.5648
NB+SMOTENN	29	15	35	9	85.7%	20.4 %	95%	37.5 %	89.2 %	1.125	0.441
NB+SMOTETomek	32	15	35	9	92.57 %	52.3 %	98.0 %	78.5 %	93.7 %	2.357	0.716

Table 9. Hasil Pengukuran Akurasi Pada Dataset *NASA PROMISE Repository*

Model	**PROMISE Repository**					
	CM1	**MC2**	**KC1**	**KC3**	**PC3**	**PC4**
NB	80.0%	76.9%	81.99%	83.33%	50.88%	80.42%
NB+SMOTEENN	85%	81.2 %	82.9 %	88.0 %	74.82%	85.7%
NB+SMOTETomek	81%	84.3 %	84.65%	88.0 %	79.0 %	92.57%

Table 10. Hasil Pengukuran *Sensitivitas* Pada Dataset *NASA PROMISE Repository*

Model	**PROMISE Repository**					
	CM1	**MC2**	**KC1**	**KC3**	**PC3**	**PC4**
NB	37.5%	27.2%	37.8%	36.3%	89.1%	37.5%
NB+SMOTEENN	26.6%	45.4 %	32.1%	75%	78.5%	20.4%
NB+SMOTETomek	36.6%	66.6%	32.0%	50%	56.1%	52.3%

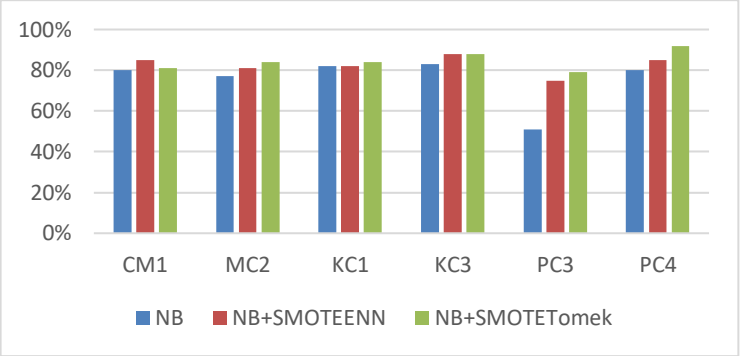
Table 11. Hasil Pengukuran *F-Measure* Pada Dataset *NASA PROMISE Repository*

Model	**PROMISE Repository**					
	CM1	**MC2**	**KC1**	**KC3**	**PC3**	**PC4**
NB	1.712	2.25	1.186	1.712	0.608	0.692
NB+SMOTEENN	1.799	3.0	1.377	1.799	0.767	1.125
NB+SMOTETomek	1.5	2.66	1.153	1.5	0.891	2.357

Table 12. Hasil Pengukuran *G-Mean* Pada Dataset *NASA PROMISE Repository*

Model	**PROMISE Repository**					
	CM1	**MC2**	**KC1**	**KC3**	**PC3**	**PC4**
NB	0.564	0.512	0.583	0.582	0.632	0.5648
NB+SMOTEENN	0.499	0.674	0.545	0.8237	0.752	0.441
NB+SMOTETomek	0.521	0.795	0.540	0.86	0.665	0.716

Berikut ini merupakan Grafik perbandingan akurasi dari beberapa dataset dan metode yang digunakan :



Gambar 3. Grafik Akurasi

Grafik diatas dapat disimpulkan bahwa hasil penelitian dengan metode yang diusulkan *hybrid method resampling* dapat menangani masalah data tidak seimbang (*imbalanced data*) pada metode *naïve bayes*. Uji coba dilakukan pada dataset CM1 didapatkan hasil akurasi 80% menggunakan *Naive Bayes*, sedangkan menggunakan metode yang diusulkan pada *Naive Bayes + SMOTEENN* mendapatkan hasil 85%, serta 81% menggunakan *Naive Bayes+SMOTETomek*. Dataset MC2 didapatkan hasil akurasi 77% menggunakan *Naive Bayes* dan mendapatkan hasil 81% menggunakan *Naive Bayes + SMOTEENN*, serta 84% menggunakan *Naive*

Bayes+SMOTETomek. Dataset KC1 didapatkan hasil akurasi 82% menggunakan *Naive Bayes* dan mendapatkan hasil 82% menggunakan *Naive Bayes + SMOTEENN*, serta 84% menggunakan *Naive Bayes+SMOTETomek*. Dataset KC3 didapatkan hasil akurasi 83% menggunakan *Naive Bayes* dan mendapatkan hasil 88% menggunakan *Naive Bayes + SMOTEENN*, serta 88% menggunakan *Naive Bayes+SMOTETomek*. Dataset PC3 didapatkan hasil akurasi 51% menggunakan *Naive Bayes* dan mendapatkan hasil 75% menggunakan *Naive Bayes + SMOTEENN*, serta 79% menggunakan *Naive Bayes+SMOTETomek*. Pada dataset PC4 didapatkan hasil akurasi 80% menggunakan *Naive Bayes*, dan mendapatkan hasil 85% menggunakan *Naive Bayes + SMOTEENN*, serta 92% menggunakan *Naive Bayes+SMOTETomek.* Hasil rata-rata penelitian diatas adalah 83% dilakukan dengan beberapa uji pada dataset dan hasilnya menunjukan bahwa akurasi metode yang diusulkan lebih baik dari pada metode *naïve bayes*.

4. Kesimpulan

Hasil penelitian dengan metode yang diusulkan didapatkan hasil akurasi tertinggi 0.85 pada model *NB+SMOTEENN* dan 0.79 menggunakan *NB+SMOTETomek* dengan beberapa kali pengujian pada dataset CM1, KC1, KC3, PC4, dan MC2 . Rata-rata *Recall, Precision* dan *f-measure* memiliki kenaikan rata-rata 0.02 setelah dilakukan *Preprocessing*. Perbandingan hasil metode lain dengan metode penelitian yang diusulkan didapatkan nilai *Specificity* tertinggi pada model *NB+SMOTETomek* (0.925) pada dataset PC4. Namun tidak semua permasalahan *imbalanced* data dapat diselesaikan dengan tenik ini. Contoh kasus dapat dilihat pada dataset PC3 dimana proses Resampling menggunakan *SMOTEENN* dan *SMOTETomek* menurunkan *Sensitivity*.

Hasil pengujian diatas dapat disimpulkan bahwa penggunaan metode *Naive Bayes + SMOTETomek* mampu mengatasi ketidak seimbangan data pada dataset prediksi cacat *software* sedangkan pada hasil AUC lebih tinggi dari pada hasil *Naive Bayes* tanpa *SMOTETomek*.

Daftar Pustaka

[1] M. Mcdonald, R. Musson, and R. Smith, *The Practical Guide to Defect Prevention*. Washington: Microsoft Press, 2007.

[2] G. Czibula, Z. Marian, and I. G. Czibula, "Software defect prediction using relational association rule mining," *Inf. Sci. (Ny).*, vol. 264, pp. 260–278, 2014, doi: 10.1016/j.ins.2013.12.031.

[3] R. Litoriya, N. Sharma, and A. Kothari, "Incorporating Cost driver substitution to improve the effort using Agile COCOMO II," *2012 CSI 6th Int. Conf. Softw. Eng. CONSEG 2012*, 2012, doi: 10.1109/CONSEG.2012.6349494.

[4] S. M. Fakhrahmad and a. Sami, "Effective Estimation of Modules' Metrics in Software Defect Prediction," *World Congr. Eng.*, vol. I, pp. 206–211, 2009.

[5] S. Lessmann, B. Baesens, C. Mues, and S. Pietsch, "Benchmarking classification models for software defect prediction: A proposed framework and novel findings," *IEEE Trans. Softw. Eng.*, vol. 34, no. 4, pp. 485–496, 2008, doi: 10.1109/TSE.2008.35.

[6] F. H. Wattiheluw, S. Rochimah, and C. Fatichah, "Klasifikasi Kualitas Perangkat Lunak Berdasarkan Iso/Iec 25010 Menggunakan Ahp Dan Fuzzy Mamdani Untuk Situs Web E-Commerce," *JUTI J. Ilm. Teknol. Inf.*, vol. 17, no. 1, p. 73, 2019, doi: 10.12962/j24068535.v17i1.a820.

[7] T. Menzies, Z. Milton, B. Turhan, B. Cukic, Y. Jiang, and A. Bener, "Defect prediction from static code features: Current results, limitations, new approaches," *Autom. Softw. Eng.*, vol. 17, no. 4, pp. 375–407, 2010, doi: 10.1007/s10515-010-0069-5.

[8] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Expert Syst. Appl.*, vol. 73, pp. 220–239, 2017, doi: 10.1016/j.eswa.2016.12.035.

[9] S. Cateni, V. Colla, and M. Vannucci, "A method for resampling imbalanced datasets in binary classification tasks for real-world problems," *Neurocomputing*, vol. 135, pp. 32–41, 2014, doi: 10.1016/j.neucom.2013.05.059.

[10] J. F. Díez-Pastor, J. J. Rodríguez, C. García-Osorio, and L. I. Kuncheva, "Random Balance: Ensembles of variable priors classifiers for imbalanced data," *Knowledge-Based Syst.*, vol. 85, no. May, pp. 96–111, 2015, doi: 10.1016/j.knosys.2015.04.022.

[11] Z. Xu, D. Shen, T. Nie, and Y. Kou, "A hybrid sampling algorithm combining M-SMOTE and ENN based on Random forest for medical imbalanced data," *J. Biomed. Inform.*, vol. 107, no. June, p. 103465, 2020, doi: 10.1016/j.jbi.2020.103465.

[12] J. A. Sáez, J. Luengo, J. Stefanowski, and F. Herrera, "SMOTE-IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering," *Inf. Sci. (Ny).*, vol. 291, no. C, pp. 184–203, 2015, doi: 10.1016/j.ins.2014.08.051.

[13] S. A. Putri, "Prediksi Cacat Software Dengan Teknik Sampel Dan Seleksi Fitur Pada Bayesian Network,"

J. Kaji. Ilm., vol. 19, no. 1, p. 17, 2019, doi: 10.31599/jki.v19i1.314.
[14] D. Ryu and J. Baik, "Effective multi-objective naïve Bayes learning for cross-project defect prediction," *Appl. Soft Comput. J.*, vol. 49, pp. 1062–1077, 2016, doi: 10.1016/j.asoc.2016.04.009.
[15] Ö. F. Arar and K. Ayan, "A feature dependent Naive Bayes approach and its application to the software defect prediction problem," *Appl. Soft Comput. J.*, vol. 59, pp. 197–209, 2017, doi: 10.1016/j.asoc.2017.05.043.
[16] M. M. Kabir, M. Shahjahan, and K. Murase, "A new hybrid ant colony optimization algorithm for feature selection," *Expert Syst. Appl.*, vol. 39, no. 3, pp. 3747–3763, 2012, doi: 10.1016/j.eswa.2011.09.073.
[17] R. S. Wahono and N. Suryana, "Combining particle swarm optimization based feature selection and bagging technique for software defect prediction," *Int. J. Softw. Eng. its Appl.*, vol. 7, no. 5, pp. 153–166, 2013, doi: 10.14257/ijseia.2013.7.5.16.
[18] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 20–29, 2004, doi: 10.1145/1007730.1007735.
[19] D. J. Hand, "Principles of Data Mining," *Drug Saf.*, vol. 30, no. 7, pp. 621–622, 2007, doi: 10.2165/00002018-200730070-00010.
[20] J. J. Sheu, "An efficient two-phase spam filtering method based on e-mails categorization," *Int. J. Netw. Secur.*, vol. 9, no. 1, pp. 34–43, 2009.
[21] E. P. K. Orpa, E. F. Ripanti, and T. Tursina, "Model Prediksi Awal Masa Studi Mahasiswa Menggunakan Algoritma Decision Tree C4.5," *J. Sist. dan Teknol. Inf.*, vol. 7, no. 4, p. 272, 2019, doi: 10.26418/justin.v7i4.33163.