MJI

Media Jurnal Informatika
Vol. 14, no.2, Desember 2022
http://jurnal.unsur.ac.id/mjinformatika

**p-issn : 2088-2114**
**e-issn : 2477-2542**

# Integrasi *Distribution Based Balance* dan Teknik *Ensemble Bagging Naive Bayes* Untuk Prediksi Cacat *Software*

Nurul Ichsan [1], Haerul Fatah [2], Erni Ermawati [3], Indriyanti [4], Tri Wahyuni [5]
*Sistem Informasi (D3) Universitas Bina Sarana Informatika [1]*
*Sistem Informasi Kampus Kota Tasikmalaya (D3) Universitas Bina Sarana Informatika [2]*
*Sistem Informasi Kampus Kabupaten Karawang (D3) Universitas Bina Sarana Informatika [3]*
*Sistem Informasi Kampus Kabupaten Banyumas (D3) Universitas Bina Sarana Informatika [4]*
*Ilmu Komputer (S1) Universitas Bina Sarana Informatika [5]*
nurul.nrc@bsi.ac.id, haerul.hef@bsi.ac.id, erni.ert@bsi.ac.id, indriyanti.iyt@bsi.ac.id, tri.twy@bsi.ac.id

Abstract

*High-quality software is software that is not found defects (defects) during the inspection or testing process. The cost of repairing software defects is much more expensive than the costs during development. Prediction of software defects is proposed to determine the priority of software modules to be tested, so as to improve software quality and reduce costs. The main problems in software defect prediction are redundant data, correlation, irrelevant features and missing samples. 62 studies out of 208 studies in the development of software defect prediction models using the NASA (National Aeronautics and Space Administration) MDP Repository dataset. The handling of the imbalance class is done by using the Distribution Based Balance sampling technique and the algorithm level approach with the ensemble learning technique using Bagging. The classification algorithm used in this study is the Naïve Bayes classifier. The results showed that the proposed model achieved higher classification accuracy and AUC. In the Distribution Based Balance + Bagging + Naïve Bayes model the average accuracy value reaches 98.71%, the average AUC value is 0.987 with an average AUC percentage increase of 0.373. While in the comparison model, namely SMOTE+Bagging+Naïve Bayes the average accuracy value reaches 73.30%, the average AUC value is 0.639 with an average AUC percentage increase of 0.025. From these results, it can be concluded that the proposed model is Distribution Based Balance+Bagging+Nave Bayes is the best model to handle imbalance class.*

*Keywords: Distribution Based Balance, Bagging, Naïve Bayes, Imbalance Class*

**Abstrak**

*Software* berkualitas tinggi adalah *software* yang tidak ditemukan cacat (*defect*) selama proses pemeriksaan atau pengujian. Biaya perbaikan cacat *software* jauh lebih mahal dibanding biaya saat pengembangan. Prediksi cacat *software* diusulkan untuk menentukan prioritas modul *software* yang akan diuji, sehingga mampu meningkatkan kualitas *software* serta mungurangi biaya. Masalah utama dalam *software defect prediction* adalah *redundant data*, korelasi, fitur yang tidak relevan dan *missing samples*. 62 penelitian dari 208 penelitian dalam pengembangan model prediksi cacat *software* menggunakan dataset NASA (*National Aeronautics and Space Administration*) MDP *Repository*. Penanganan *imbalance class* dilakukan dengan menggunakan *sampling technique Distribution Based Balance* dan pendekatan level algoritma dengan teknik *ensemble learning* menggunakan *Bagging*. Algoritma klasifikasi yang digunakan dalam penelitian ini yaitu *classifier Naïve Bayes*. Hasil penelitian menunjukkan bahwa model yang diusulkan mencapai akurasi dan AUC klasifikasi yang lebih tinggi. Pada Model Integrasi *Distribution Based Balance dan Teknik Ensemble Bagging Naive Bayes* rata-rata nilai akurasi mencapai 98,71%, rata-rata nilai AUC 0.987 dengan nilai rata-rata peningkatan presentase AUC mencapai 0.373. Sedangkan pada model pembanding yaitu Integrasi *SMOTE dan Teknik Ensemble Naïve Bayes* rata-rata nilai akurasi mencapai 73,30%, rata-rata nilai AUC 0,639 dengan nilai rata-rata peningkatan presentase AUC mencapai 0,025. Dari hasil tersebut dapat disimpulkan bahwa model yang diusulkan yaitu *Integrasi Distribution Based Balance dan Teknik Ensemble Bagging Naive Bayes* merupakan model terbaik untuk menangani *imbalance class*.

*Kata kunci: Distribution Based Balance, Bagging, Naïve Bayes, Imbalance Class*

## I. PENDAHULUAN

Software berkualitas tinggi adalah *software* yang tidak ditemukan cacat (*defect*) baik selama proses pemeriksaan atau pengujian. Kualitas *software* biasanya diukur dari jumlah cacat yang ada pada produk yang dihasilkan [1]. Pengujian menjadi standar dalam menghasilkan perangkat lunak yang berkualitas [2]. Proses pengujian *software* dapat mengidentifikasi apakah sebuah *software* mengandung cacat atau tidak. Sejumlah cacat yang ditemukan di akhir proyek secara sistematis menyebabkan penyelesaian proyek dapat melebihi jadwal yang sudah ditentukan [3]

Biaya perbaikan cacat *software* jauh lebih mahal dibanding biaya saat pengembangan, oleh karena itu prediksi cacat *software* menjadi topik yang penting dalam proses pengembangan perangkat lunak, kondisi ini menjadi tantangan bagi praktisi dan peneliti untuk mecari cara yang paling efektif dan efesien pada saat proses pengujian kualitas *software* [4]

Prediksi cacat *software* diusulkan untuk membantu menentukan prioritas modul *software* yang akan diuji, sehingga mampu meningkatkan kualitas *software* serta mungurangi biaya. Hasil prediksi cacat *software* dapat digunakan oleh praktisi *software* untuk menilai secara efisien modul perangkat lunak yang rusak, seperti jumlah cacat dalam modul atau informasi lain yang terkait dengan cacat *software* sebelum pengujian dilakukan [5].

Masalah utama dalam *software defect prediction* adalah *redundant data*, korelasi, fitur yang tidak relevan, *missing samples* dan masalah ini dapat membuat dataset tidak seimbang karena sulit untuk memastikan antara data cacat atau tidak cacat [6]. Selain itu dalam dataset *software metrics* juga terdapat masalah *imbalance class* yang membuat data menjadi tidak seimbang karena data yang cacat (kelas minoritas) jumlahnya lebih sedikit dibandingkan dengan data yang tidak cacat (kelas mayoritas), masalah ini dapat menurunkan kinerja klasifikasi [7]

Terdapat dua pendekatan yang dapat menangani *imbalance class* yaitu pendekatan level data (*sampling technique*) dan pendekatan level algoritma dengan teknik *ensemble learning* [3]. Sedangkan klasifikasi adalah pendekatan yang paling populer untuk menangani *software defect prediction* [8]. Data penelitian yang didapat 77.46% dalam menyelesaikan masalah prediksi cacat *software* diselesaikan dengan menggunakan metode klasifikasi [9].

Berbagai *software metrics* yang berbeda ditemukan setelah proses ekstraksi *software* dan digunakan untuk memprediksi kecenderungan cacat. [10]. 62 penelitian dari 208 penelitian dalam pengembangan model prediksi cacat *software* menggunakan dataset NASA (*National Aeronautics and Space Administration*) MDP (*Metrics Data Program*) *repository* sebagai *software metrics* [11]. CRISP-DM merupakan model proses *data mining* dengan 6 tahapan yaitu: *Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation,* dan *Deployment* dalam menentukan perbandingan algoritma klasifikasi untuk memprediksi cacat *software* [12].

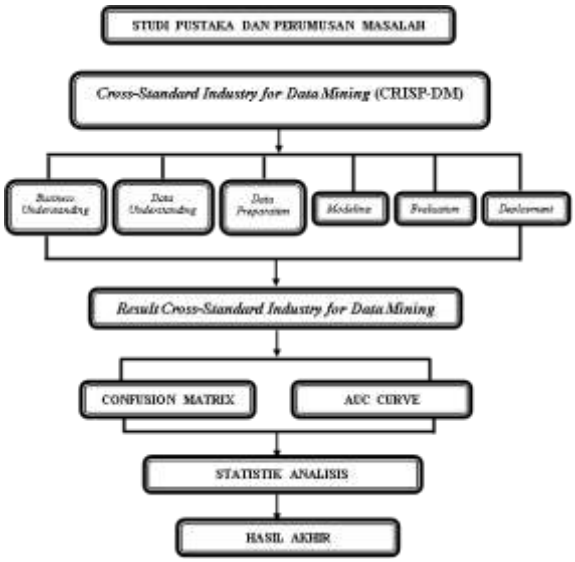
Berdasarkan dari masalah diatas terkait *software defect prediction*, pada penelitian ini eksperimen yang digunakan penulis yaitu desain eksperimen *Cross-Standard Industry for Data Mining* (CRISP-DM). Hasil dari penelitian ini bertujuan untuk mengurangi dan mengatasi permasalahan *imbalance class* pada model prediksi cacat *software*. *Data understanding* yang digunakan merupakan data *public*, dataset ini tersedia secara umum di internet yaitu 12 dataset NASA (*National Aeronautics and Space Administration*) MDP (*Metrics Data Program*) *repository* yang dapat diunduh di https://github.com/klainfo/NASADefectDataset. Dataset NASA MDP *Repository* mudah diperoleh dan tersedia untuk umum. 64.79% penelitian terkait prediksi cacat *software* menggunakan dataset publik dan 35.21% penelitian menggunakan dataset privat. Dataset NASA MDP *Repository* memiliki kelemahan kelas yang tidak seimbang (*Imbalance Class).*

Penanganan *imbalance class* dilakukan dengan menggunakan *sampling technique* dan pendekatan level algoritma atau penggabungan dengan teknik *ensemble learning* menggunakan *Bagging*. *sampling technique* yang digunakan dibagi dengan dua pendekatan berbeda yaitu untuk teknik *oversampling* menggunakan SMOTE (*Synthetic Minority Over-sampling Technique*) dan teknik *hybrid sampling* menggunakan *Distribution Based Balance*. algoritma klasifikasi yang digunakan dalam penelitian ini yaitu *classifier Naïve Bayes*.

Penelitian ini diharapkan akan memperoleh nilai akurasi dan AUC (*Area Under ROC Curve*) yang siginifikan untuk mengatasi permasalahan *imbalance class* yang terdapat pada dataset *software metrics* NASA (*National Aeronautics and Space Administration*) MDP (*Metrics Data Program*) *repository* dalam penelitian prediksi cacat *software*.

## II. METODE PENELITIAN

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




Gambar 1. Tahapan Penelitian

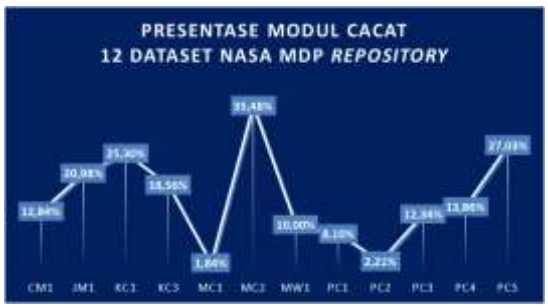
Penelitian ini menggunakan pendekatan kuantitatif menggunakan data sekunder. Data yang digunakan dalam penelitian ini adalah dataset NASA (*National Aeronautics and Space Administration*) MDP (*Metrics Data Program*) *repository*. Metode penelitian yang digunakan pada eksperimen ini adalah model *Cross-Standard Industry for Data Mining* (CRISP-DM) yang terdiri dari 6 fase

Tabel 1. Penerapan CRSIP-DM

| NO | *Cross-Standard Industry for Data Mining* | PENERAPAN DALAM PENELITIAN |
|---|---|---|
| 1 | *Business Understanding* | Pada fase ini bisa disebut sebagai tahap pemahaman penelitian, menentukan tujuan proyek penelitian dalam perumusan mendefinisikan masalah *data mining* untuk penanganan ketidakseimbangan kelas pada *software defect prediction*. |
| 2 | *Data Understanding* | Pada fase ini penelitian lebih difokuskan pada tahap pemilihan dan pengumpulan data. Data yang akan digunakan merupakan data public yaitu 12 Dataset Nasa MDP *Repository* yang terdiri dari kelas (CM1, JM1, KC1, KC3, MC1, MC2, MW1, PC1 PC2, PC3, PC4, PC5). |
| 3 | *Data Preparation* | Fase pengolahan, Dataset NASA MDP yang akan digunakan dalam penelitian ini adalah yang telah ditransformasi (D"). Teknik validasi yang digunakan adalah 10-*fold cross validation*, sehingga dataset dibagi menjadi 10, kemudian diambil satu bagian untuk dijadikan sebagai data uji dan yang lainnya dijadikan data latih. |
| 4 | *Modeling* | Pada tahap ini dilakukan pemrosesan *data training*. Model yang akan digunakan model algoritma klasifikasi yang di integrasikan dengan pendekatan level data dan *ensemble learning* untuk pengujiannya. |
| 5 | *Evaluation* | Pada tahap evaluasi, disebut tahap klasifikasi karena pada tahap ini akan ditentukan pengujian untuk akurasi. Tahap pengujiannya adalah melihat hasil akurasi dari berbagai model. Penjelasan secara lengkap terdapat pada bab IV. |
| 6 | *Deployment* | - |

## III. HASIL PENELITIAN

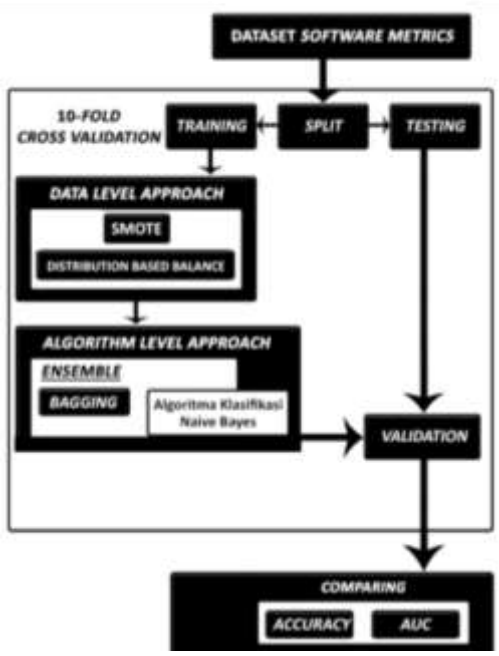
Spesifikasi jumlah data penelitian diambil dari 12 Dataset Nasa MDP *Repository* dengan memperhatikan jumlah atribut, modul, modul cacat (kelas minoritas), modul tidak cacat (kelas mayoritas) dan presentase modul cacat. Perbandingan jumlah presentase cacat sangat bervariasi dari 12 dataset *software metrics* yang akan diuji cobakan mulai dari 1.84% s/d 35.48% untuk tingkat presentase cacatnya. Dari hasi pengolahan data, nilai presentese terendah 1.84% untuk dataset MC1 dipengaruhi oleh ketidakseimbangan kelas antara modul cacat dan modul tidak cacat. Dataset MC1 memiliki spesifikasi jumlah attribute 39, jumlah modul 1952, modul cacat 36 dan modul tidak cacat 1916. Sedangkan untuk nilai presentase tertinggi dataset MC2 memiliki spesifikasi jumlah attribute 40, jumlah modul 124, modul cacat 44 dan modul tidak cacat 80.




Gambar 2. Presentase Modul Cacat

Dari analisa Gambar 2. hasil presentase modul cacat tersebut diambil dari atribut, jumlah modul, modul cacat dan modul tidak cacat dari setiap dataset. Tinggi atau rendahnya hasil presentase dipengaruhi oleh ketidakseimbangan kelas (*Imbalance Class*) dari modul cacat dan tidak cacat, dari Analisa tersebut 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 *Distribution Based Balance)* dan pendekatan level algoritma (*ensemble) Bagging* berbasis *classifier* Naïve Bayes

untuk mencari kinerja model yang paling memberikan hasil yang terbaik dan signifikan baik untuk akurasi ataupun nilai AUC nya.




Gambar 3. Kerangka Kerja Model yang Diusulkan

Pada penelitian yang dilakukan, *tools* yang digunakan untuk pengujian algoritma adalah aplikasi WEKA (*Waikato Environment for Knowledge Analysis*), yang merupakan aplikasi *data mining open source* berbasis Java. Hal yang paling penting pada model desain eksperimen *Cross-Standard Industry for Data Mining* (CRISP-DM) adalah mendapatkan data untuk diteliti yang disebut dengan dataset. Secara umum data yang digunakan pada penelitian prediksi cacat *software* ini yaitu menggunakan format ARFF (*Attribute-Relation File Format*).

Sebelum diterapkan dengan berbagai model pengujian, terlebih dahulu dilakukan pengujiannya dengan kinerja asli dari *classifier* secara original yaitu Naïve Bayes yang belum di integrasikan dengan model apapun dengan maksud dan tujuan untuk melihat sejauh mana perbedaan hasil kinerja model yang didapat sebelum dan sesudahnya dilakukan integrasi pengujian model yang diusulkan.

Tabel 2. Pengujian 12 Dataset dengan *Classifier* Naïve Bayes (Original)

| Dataset | Naïve Bayes (Original) | | | Kinerja Klasifikasi |
|---|---|---|---|---|
| | Akurasi | AUC | ROC Area | |
| CM1 | 81.35% | 0.599 | 0.645 | *Failure Classification* |
| JM1 | 78.91% | 0.570 | 0.650 | *Failure Classification* |
| KC1 | 73.58% | 0.599 | 0.681 | *Failure Classification* |
| KC3 | 78.87% | 0.634 | 0.661 | *Poor Classification* |
| MC1 | 90.98% | 0.600 | 0.746 | *Failure Classification* |
| MC2 | 70.16% | 0.620 | 0.707 | *Poor Classification* |
| **MW1** | **81.60%** | **0.720** | **0.780** | ***Fair Classification*** |
| PC1 | 89.10% | 0.667 | 0.793 | *Poor Classification* |
| PC2 | 90.30% | 0.553 | 0.710 | *Failure Classification* |
| PC3 | 39.41% | 0.602 | 0.749 | *Poor Classification* |
| PC4 | 85.35% | 0.624 | 0.814 | *Poor Classification* |
| PC5 | 74.97% | 0.582 | 0.719 | *Failure Classification* |
| **Rata-rata** | **77.88%** | **0.614** | **0.721** | ***Poor Classification*** |

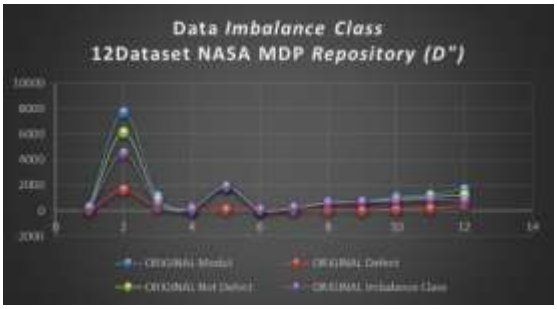
Pada model Naïve Bayes (Original) hasil terbaik didapatkan pada dataset MW1.arff dengan nilai akurasi 81.60%, AUC 0.720 dan ROC Area 0.780 dengan tingkatan level klasifikasi "*Fair Classification*". Hasil penelitian diatas didapat dari pengujian *Classifier* Naïve Bayes yang belum di integrasikan dengan model pengujian yang diusulkan yaitu Integrasi *Distribution Based Balance* dan *Teknik Ensemble Bagging Naive Bayes.* Melihat dari hasil yang ada tentunya hasilnya diluar dari ekspetasi yang diharapkan karena nilai AUC yang didapat sangat rendah, karena nilai AUC tersebut dapat menunjukan sejauh mana kualitas dari *classifier.* Rata-rata performa hasil yang didapat sangat buruk yaitu 77.88% untuk akurasi, AUC 0.614 dan ROC Area 0.721 dengan tingkatan level klasifikasi berada pada level *Poor Classification.* Jika mengacu pada Pedoman umum yang digunakan untuk klasifikasi yaitu:

**Tabel 3.** Klasifikasi Akurasi

| NO | NILAI AUC | KLASIFIKASI |
|---|---|---|
| 1 | 0.90-1.00 | *Excellent Classification* |
| 2 | 0.80-0.90 | *Good Classification* |
| 3 | 0.70-0.80 | *Fair Classification* |
| 4 | 0.60-0.70 | *Poor Classification* |
| 5 | 0.50-0.60 | *Failure* |

**Sumber :** (Gao, Khoshgoftaar, & Wald , 2014)

Performa yang tidak begitu bagus dari *classifier* tersebut dikarenakan pada 12 dataset terdapat *imbalance class*, 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").*




Gambar 4.
Data *Imbalance Class* Pada 12 Dataset NASA (D")

Melihat grafik yang ditunjukan pada Gambar 4. perbedaan antara data *defect* dan *not defect* terilihat 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 Naïve Bayes. Tahapan *Preprocessing Data* yang akan dilakukan menggunakan teknik *oversampling* (SMOTE) dan *hybrid* (*Distribution Based* Balance) yaitu dengan cara menggabungkan teknik *oversampling (*menaikan kelas minoritas) dan *undersampling (*menurunkan kelas mayoritas).

Tabel 4. Hasil *Preprocessing Data*

| DATA SET | ORIGINAL | | | SMOTE | | | DBB | | |
|---|---|---|---|---|---|---|---|---|---|
| | M | D | ND | M | D | ND | M | D | ND |
| CM1 | 327 | 42 | 285 | 369 | 84 | 285 | 369 | 184.5 | 184.5 |
| JM1 | 7720 | 1612 | 6108 | 9332 | 3224 | 6108 | 9332 | 4666 | 4666 |
| KC1 | 1162 | 294 | 868 | 1456 | 588 | 868 | 1456 | 728 | 728 |
| KC3 | 194 | 36 | 158 | 230 | 72 | 158 | 230 | 115 | 115 |
| MC1 | 1952 | 36 | 1916 | 1988 | 72 | 1916 | 1988 | 994 | 994 |
| MC2 | 124 | 44 | 80 | 168 | 88 | 80 | 168 | 84 | 84 |
| MW1 | 250 | 25 | 225 | 275 | 50 | 225 | 275 | 137.5 | 137.5 |
| PC1 | 679 | 55 | 624 | 734 | 110 | 624 | 734 | 367 | 367 |
| PC2 | 722 | 16 | 706 | 738 | 32 | 706 | 738 | 369 | 369 |
| PC3 | 1053 | 130 | 923 | 1183 | 260 | 923 | 1183 | 591.5 | 591.5 |
| PC4 | 1270 | 176 | 1094 | 1446 | 352 | 1094 | 1446 | 723 | 723 |
| PC5 | 1694 | 458 | 1236 | 2152 | 916 | 1236 | 2152 | 1076 | 1076 |

DBB=Distribution Based Balance, M=Modul, D=Defect, ND=Not Defect

Setelah melalui tahapan *preprocessing data* dataset terlihat ada berbagai perbedaan dari 2 model *preprocessing* yang digunakan 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 Naïve Bayes untuk mengatasi masalah *imbalance class* dengan mencari perbandingan akurasi dan AUC yang didapat dari berbagai model pengujian.

## IV. PEMBAHASAN

*Classifier* yang yang akan digunakan dalam penelitian ini yaitu algoritma klasifikasi Naïve Bayes. Pada model pengujian Naïve Bayes akan dilakukan 3 pengujian model yang berbeda terhadap 12 dataset Nasa MDP *Repository.* Diantaranya model pengujian dengan integrasi teknik pendekatan level data SMOTE dan *Distribution Based Balance* dengan Algoritma *ensemble Bagging* untuk meningkat kinerja klasifikasi Naïve Bayes dalam mengatasi *imbalance class* pada prediksi cacat *software*.

Tabel 5. Model Pengujian Algoritma

| NO | MODEL PENGUJIAN ALGORITMA |
|---|---|
| 1 | Naïve Bayes |
| 2 | *SMOTE + BAGGING + NAÏVE BAYES* |
| 3 | *DISTRIBUTION BASED BALANCE + BAGGING + NAÏVE BAYES* |

*Data Level Approach* :
SMOTE dan *Distribution Based Balance*
*Algorithm Level Approach* : *Bagging*
*Classification Algorithm* : *Naïve Bayes*

1. Naïve Bayes

Hasil pengujian Naïve Bayes bisa dilihat pada Tabel.2, dari hasil pengujian yang didapat nilai akurasi 81.60%, AUC 0.720 dan ROC Area 0.780. Rata-rata performa hasil yang didapat sangat buruk yaitu 77.88% untuk akurasi, AUC 0.614 dan ROC Area 0.721 dengan tingkatan level klasifikasi berada pada level *Poor Classification.*

*2. SMOTE + Bagging + Naïve Bayes*

Pengujian yang kedua didapat dari hasil integrasi pendekatan level data *SMOTE* dengan pendekatan level algoritma bagging berbasis *classifier Naïve Bayes*, berikut adalah hasil pengujian datanya:

Tabel 6. Hasil Kinerja SMOTE+*Bagging*+Naïve Bayes

| Dataset | SMOTE + Bagging + Naïve Bayes | | | Kinerja Klasifikasi |
|---|---|---|---|---|
| | Akurasi | AUC | ROC Area | |
| CM1 | 74,25% | 0,598 | 0,71 | *Failure* |
| JM1 | 68,58% | 0,572 | 0,650 | *Failure* |
| KC1 | 65,80% | 0,610 | 0,690 | *Poor* |
| KC3 | 71,74% | 0,643 | 0,683 | *Poor* |
| MC1 | 89,34% | 0,657 | 0,779 | *Poor* |
| MC2 | 62,50% | 0,638 | 0,741 | *Poor* |
| MW1 | 80,36% | 0,748 | 0,806 | *Fair* |
| PC1 | 84,20% | 0,664 | 0,812 | *Poor* |

| | | | | |
|---|---|---|---|---|
| PC2 | 89,43% | 0,617 | 0,771 | *Poor* |
| PC3 | 47,59% | 0,632 | 0,788 | *Poor* |
| PC4 | 81,40% | 0,693 | 0,827 | *Poor* |
| PC5 | 64,36% | 0,592 | 0,728 | *Failure* |
| Rata-rata | 73,30% | 0,639 | 0,749 | *Poor* |

Dari hasil pengujian model SMOTE+Bagging+Naïve Bayes hampir pada semua dataset tidak mengalami kenaikan yang signifikan, untuk hasil akurasi mengalami penurunan, sedangkan untuk hasil AUC dan ROC Area mengalami kenaikan namun tidak signifikan. Hasil terbaik dari didapat pada dataset MW1.arff dengan nilai akurasi 80.36%, AUC 0.748, ROC Area 0.806 dengan hasil tingkatan level akurasi "*Fair Classification*". Untuk rata-rata dari pengujian model ini yaitu akurasi menurun sebesar 4,59%, AUC meningkat 0.025, ROC Area meningkat 0.028 dengan nilai rata-ratanya yaitu 73,30% untuk akurasi, 0,639 untuk nilai AUC, 0,749 untuk nilai ROC Area dan rata-rata pengujian kinerja model ini berada pada level "*Poor Classification*", hasil yang didapat masih sama dengan model Naïve Bayes yang berada pada level "*Poor Classification*". Hasil akurasi, AUC dan ROC Area pada Tabel 6. didapat dengan perhitungan kinerja model menggunakan *Confusion Metrix* untuk mencari akurasi, sensitivitas/*recall*/TPrate, *specificity*/TNrate, FPrate, FNrate, *Precision*/PPV, *F-Measure*, *G-Mean* dan AUC.

*3. DBB + Bagging + Naïve Bayes*

Pengujian model terakhir yaitu pendekatan level data *Distribution Based Balance* yang akan di integrasikan dengan pendekatan level algoritma bagging berbasis *classifier Naïve Bayes*. Model ini menjadi model yang diusulkan oleh penulis untuk mengatasi permasalahan *imbalance class*. Berikut adalah hasil pengujian modelnya :

Tabel 7. Hasil Kinerja DBB+*Bagging*+Naïve Bayes

| Dataset | DBB + Bagging + Naïve Bayes | | | Kinerja Klasifikasi |
|---|---|---|---|---|
| | Akurasi | AUC | ROC Area | |
| CM1 | 97,83% | 0,978 | 0,997 | *Excellent* |
| JM1 | 99,53% | 0,995 | 1,000 | *Excellent* |
| KC1 | 95,54% | 0,955 | 0,989 | *Excellent* |
| KC3 | 98,70% | 0,987 | 0,999 | *Excellent* |
| MC1 | 99,35% | 0,993 | 1,000 | *Excellent* |
| MC2 | 100,00 % | 1,000 | 1,000 | *Excellent* |
| MW1 | 100,00 % | 1,000 | 1,000 | *Excellent* |
| PC1 | 100,00 % | 1,000 | 1,000 | *Excellent* |
| PC2 | 99,19% | 0,992 | 1,000 | *Excellent* |
| PC3 | 96,19% | 0,962 | 0,996 | *Excellent* |
| PC4 | 98,89% | 0,989 | 1,000 | *Excellent* |
| PC5 | 99,35% | 0,993 | 0,999 | *Excellent* |
| Rata-rata | 98,71% | 0,987 | 0,998 | *Excellent* |

Hasil dari model integrasi pengujian ini memberikan hasil melampaui ekspetasi yang diproyeksikan artinya kinejanya sangat bagus sekali untuk menagani *imbalance class* pada prediksi cacat *software* dengan rata-rata nilai akurasi mencapai 98.71%, rata-rata nilai AUC 0.987 dan rata-rata peningkatan nilai AUC jika dibandingkan dengan model Naïve Bayes (original) mencapai pada nilai 0.373 untuk presentase peningkatannya. Model ini memberikan kinerja klasifikasi *Excellent Classification.* Diantara model *preprocessing* integrasi lainnya yaitu SMOTE, model ini tampil sangat baik dan prima untuk menangani masalah *Imbalance Class* serta bisa dikategorikan sebagai model integrasi terbaik untuk tingkatan *classifier* Naïve Bayes.

Hasil akurasi, AUC dan ROC Area pada Tabel7. didapat dengan perhitungan kinerja model menggunakan *Confusion Metrix* untuk mencari akurasi, sensitivitas / *recall*/ TPrate, *specificity* / TNrate, FPrate, FNrate, *Precision*/PPV, *F-Measure*, *G-Mean* dan AUC.

Tabel 8.
Perbandingan Evaluasi Kinerja Naïve Bayes

| MODEL | NAÏVE BAYES | |
|---|---|---|
| | AKURASI | AUC |
| ORI | 77.88% | 0.614 |
| SMOTE+BG | 73,30% | 0,639 |
| DBB+BG | 98,71% | 0,987 |
| RATA-RATA | 86,01% | 0,813 |

Hasil tabel diatas menunjukan bahwa integrasi model Distribution Based Balance+Bagging+Naïve Bayes *classifier* yang diusulkan penulis dalam penelitian ini memberikan hasil *classifier* terbaik secara keseluruhan dari berbagai model pengujian dengan rata-rata akhir kinerja model klasifikasi dengan akurasi 98,71% dan AUC 0,987 lebih baik dibandingkan kinerja algoritma pembanding Naïve Bayes (Original) ataupun dengan kinerja SMOTE+Bagging+Naïve Bayes.

## V. KESIMPULAN

Integrasi *Distribution Based Balance* dan Bagging diusulkan untuk meningkatkan kinerja *classifier* Naïve Bayes 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 lebih tinggi. Pada Model *Distribution Based Balance*+Bagging+Naïve Bayes rata-rata nilai akurasi mencapai 98,71%, rata-rata nilai AUC 0.987 dengan nilai rata-rata peningkatan presentase AUC mencapai 0.373. Sedangkan pada model pembanding yaitu *SMOTE*+Bagging+Naïve Bayes rata-rata nilai akurasi mencapai 73,30%, rata-rata nilai AUC 0,639 dengan nilai rata-rata peningkatan presentase AUC mencapai 0,025. Dari hasil tersebut dapat disimpulkan bahwa model yang diusulkan yaitu *Distribution Based Balance*+Bagging+Naïve Bayes merupakan model terbaik dalam penelitian prediksi cacat *software* untuk menangani *imbalance class.*

## VI. REFERENSI


[1] A. Hardoni, D. P. Rini, and S. Sukemi, "Integrasi SMOTE pada Naive Bayes dan Logistic Regression Berbasis Particle Swarm Optimization untuk Prediksi Cacat Perangkat Lunak," *J. Media Inform. Budidarma*, vol. 5, no. 1, p. 233, 2021, doi: 10.30865/mib.v5i1.2616.

[2] R. Prasetyo, I. Nawawi, A. Fauzi, and G. Ginabila, "Komparasi Algoritma Logistic Regression dan Random Forest pada Prediksi Cacat Software," *J. Tek. Inform. UNIKA St. Thomas*, vol. 06, no. Siringoringo 2017, pp. 275–281, 2021, doi: 10.54367/jtiust.v6i2.1522.

[3] N. Ichsan, "Metoda Distribution Based Balance dan Bagging C4 . 5," *Indones. J. Comput. Inf. Technol.*, vol. 4, no. 2, pp. 215–224, 2019.

[4] Sugiono, A. Taufik, and R. Faizal Amir, "Penerapan Penerapan Teknik Pso Over Sampling Dan Adaboost J48 Untuk Memprediksi Cacat Software," *J. Responsif Ris. Sains dan Inform.*, vol. 2, no. 2, pp. 198–203, 2020, doi: 10.51977/jti.v2i2.249.

[5] 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.

[6] R. F. Amir, I. A. Sobari, and R. Rousyati, "Penerapan PSO Over Sampling Dan Adaboost Random Forest Untuk Memprediksi Cacat Software," *Indones. J. Softw. Eng.*, vol. 6, no. 2, pp. 230–239, 2020, doi: 10.31294/ijse.v6i2.9258.

[7] M. F. Akbar, I. Kurniawan, and A. Fauzi, "Mengatasi Imbalanced Class Pada Software Defect Prediction Menggunakan Two-Step Clustering-Based Undersampling dan Bagging Tehcnique," *J. Inform.*, vol. 6, no. 1, pp. 107–113, 2019, doi: 10.31311/ji.v6i1.5448.

[8] A. Hardoni and D. P. Rini, "Integrasi Pendekatan Level Data Pada Logistic Regression Untuk Prediksi Cacat Perangkat Lunak," *JIKO (Jurnal Inform. dan Komputer)*, vol. 3, no. 2, pp. 101–106, 2020, doi: 10.33387/jiko.v3i2.1734.

[9] R. S. Wahono, "A Systematic Literature Review of Software Defect Prediction : Research Trends , Datasets , Methods and Frameworks," *J. Softw. Eng.*, vol. 1, no. 1, 2015.

[10] A. Fauzi and Ginabila, "Komparasi Algoritma Dengan Pendekatan Random," *J. PILAR Nusa Mandiri*, vol. 15, no. 1, pp. 27–34, 2019.

[11] T. Hall, S. Beecham, D. Bowes, D. Gray, and S. Counsell, "A Systematic Literature Review on Fault Prediction Performance in Software Engineering," *IEEE Trans. Softw. Eng.*, vol. 38, no. 6, pp. 1276–1304, 2012.

[12] N. Hidayati, J. Suntoro, and G. G. Setiaji, "Perbandingan Algoritma Klasifikasi untuk Prediksi Cacat Software dengan Pendekatan CRISP-DM," *J. Sains dan Inform.*, vol. 7, no. 2, pp. 117–126, 2021, doi: 10.34128/jsi.v7i2.313.

[13] 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.