

Penerapan PSO Over Sampling Dan Adaboost Random Forest Untuk Memprediksi Cacat Software

Richky Faizal Amir¹, Irwan Agus Sobari², Rousyati³

¹Teknologi Komputer, Universitas Bina Sarana Informatika

²Teknik Informatika, STMIK Nusa mandiri

³Sistem Informasi Kampus Kota Tegal, Universitas Bina Sarana Informatika

¹richky.cfj@bsi.ac.id, ²irwan.igb@nusamandiri.ac.id, ³rousyati.rou@bsi.ac.id

Abstract: The dataset of software metrics, in general, are not balanced (Imbalanced). Class imbalance in Dataset can reduce the performance of software defect prediction models, because it tends to produce majority class predictions from minority classes, the dataset used in this study uses the National Aeronautics and Space Administration (NASA) Metrics Data Program (MDP), dataset From Stages Pre-processing proposed the Particle Swarm Optimization (PSO). method to overcome the problem of attributes in the training data and the Random Over Sampling (ROS) Resampling method. to deal with class imbalances. This study proposes that the Random Forest method combined with Adaboost can estimate the level of disability of software through training data. The results of this study indicate that the Resampling + Adaboost + Random Forest algorithm can be used to predict software defects with an average accuracy of 94.70% and a value of AUC 0.939. While the PSO + Random Forest algorithm only has an average accuracy of 89.60% and AUC 0.636 the difference in the accuracy of the two models is 5.10% and AUC 0.303. Statistical tests show that there is a significant influence between the proposed model and the Random Forest model with a p-value (0.036) smaller than the alpha value (0.05), which means there is a significant difference between the two models.

Keywords: Imbalanced Class, Resample, Particle Swarm Optimization, Random Forest, Adaboost, Software Defect

Abstrak: Dataset dari software matrik secara umum bersifat tidak seimbang (Imbalanced). Ketidak seimbangan kelas yang ada dalam dataset dapat menurunkan kinerja model prediksi cacat software, karena cenderung menghasilkan prediksi kelas mayoritas dari kelas minoritas. Dataset yang digunakan pada penelitian ini menggunakan dataset National Aeronautics and Space Administration (NASA) Metrics Data Program (MDP). Dari tahapan pra pemrosesan diusulkan metode Particle Swarm Optimization (PSO) untuk mengatasi masalah attribute pada data training dan metode Resampling Random Over Sampling (ROS). untuk menangani ketidak seimbangan kelas. Penelitian ini mengusulkan metode Random Forest yang dikombinasikan dengan Adaboost dapat mengestimasi tingkat kecacatan suatu Software melalui data training, Dari Hasil penelitian ini menunjukkan bahwa algoritma Resampling+Adaboost+Random Forest dapat digunakan untuk memprediksi cacat software dengan rata-rata akurasi 94,70% dan nilai AUC 0,939. Sementara algoritma PSO+Random Forest hanya memiliki rata-rata akurasi 89,60% dan AUC 0,636 perbedaan akurasi dari kedua model tersebut 5,10% dan AUC 0,303. Uji statistik menunjukkan bahwa adanya pengaruh yang signifikan antara model usulan dengan model Random Forest dengan nilai p (0,036) lebih kecil dari nilai alpha (0,05) yang artinya terdapat perbedaan yang signifikan antara kedua model.

Kata kunci: Imbalanced Class, Resample, Particle Swarm Optimization, Random Forest, Adaboost, Kecacatan Software



This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. ©2019 by author and IJSE-Indonesian Journal on Software Engineering.

A. PENDAHULUAN

Prediksi cacat software merupakan salah satu fase atau tahap pengujian dalam *Software Development Life Cycle* (Saifudin et al., 2015). Kualitas sebuah software dapat diketahui pada

tahap pengujian ketika software dibangun. Proses pengujian software dapat mengidentifikasi apakah sebuah software mengandung cacat atau tidak. 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.

Potensi cacat tertinggi terjadi pada tahap pengkodean (Hofman, 2011). Dibandingkan dengan tahap lainnya. Maka diperlukan sebuah prosedur dalam tahap pengembangan perangkat lunak yang dapat digunakan di masa depan dalam memprediksi cacat software sehingga dapat menghasilkan software yang berkualitas. Saat ini penelitian tentang prediksi cacat software berfokus pada metode klasifikasi sebanyak 77.46%, 14.08% menggunakan metode estimasi dan sebanyak 1.41% menggunakan metode clustering dan asosiasi (Wahono, 2007). Maka diperlukan data yang diperoleh dari riwayat pengembangan software sebelumnya untuk mengklasifikasikan cacat atau tidak.

Dataset National Aeronautics and Space Administration (NASA) merupakan data metrik perangkat lunak yang sering digunakan dalam penelitian prediksi cacat software. Dataset (NASA) mudah diperoleh dan tersedia untuk umum karena sebanyak 64.79% penelitian menggunakan public dataset dan 35.21% penelitian menggunakan dataset privat. Menggunakan dataset NASA merupakan pilihan yang baik, karena mudah diperoleh dan kinerja dari metode yang digunakan menjadi mudah untuk dibandingkan dengan penelitian sebelumnya.

Masalah 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 (Laradji et al., 2014). Selain masalah redundansi data, fitur-fitur yang tidak relevan juga terdapat masalah ketidak seimbangan kelas (*imbalance class*) karena data yang cacat jumlahnya lebih sedikit dibandingkan dengan data yang tidak cacat, sehingga data yang termasuk kelas mayoritas adalah tidak cacat dan data yang termasuk kelas minoritas adalah cacat.

Feature selection (pemilihan fitur atau atribut) dapat menangani masalah redundant data dan fitur yang tidak relevan. Seleksi fitur merupakan langkah yang penting dalam mesin pembelajaran (*machine learning*). Tujuan dari seleksi fitur diantaranya adalah menyederhanakan dan meningkatkan kualitas dataset dengan memilih atribut-atribut yang relevan. Pada penelitian yang dilakukan menggunakan metode optimasi particle swarm optimization diterapkan untuk menangani pemilihan fitur (Wahono et al., 2014).

Dalam menangani dataset yang tidak seimbang, terdapat 3 pendekatan yang dapat dilakukan, yaitu pendekatan level data, level algoritma dan pendekatan dengan menggabungkan metode. Pada pendekatan level data, mencakup berbagai teknik resampling untuk memperbaiki distribusi kelas pada data. Pada level algoritma, dilakukan proses penyesuaian operasi algoritma yang ada untuk menjalankan pengklasifikasian (*classifier*) agar lebih kondusif terhadap klasifikasi kelas minoritas. Sedangkan pada pendekatan gabungan (*ensemble*), terdapat 2 *ensemble-learning* paling populer, yaitu *boosting* dan *bagging*. Pada pendekatan algoritma dan *ensemble* memiliki tujuan yang sama, yaitu memperbaiki algoritma pengklasifikasi tanpa mengubah data, sehingga dapat dianggap ada 2 pendekatan saja, yaitu pendekatan level data dan pendekatan level algoritma. Dengan membagi menjadi 2 pendekatan dapat mempermudah dalam proses perbaikan.

Random Forest (RF) adalah metode *ensemble learning* yang diusulkan oleh Breiman tahun 2001 yang merupakan kombinasi dari pohon klasifikasi sedemikian rupa sehingga setiap pohon bergantung pada nilai acak vektor sampel secara mandiri dan dengan distribusi yang sama untuk semua pohon di hutan (Biau & Scornet, 2016).

Teknik *AdaBoost* digunakan untuk menangani ketidak seimbangan kelas. Hasilnya menunjukkan bahwa metode yang diusulkan dapat memberikan peningkatan yang mengesankan dalam hasil perbandingan. Teknik *AdaBoost* juga digunakan untuk meningkatkan nilai akurasi pada *dataset bank direct marketing* menghasilkan nilai akurasi 92,25% (Rais & Subekti, 2019).

Sedangkan pada klasifikasi data time series arus lalu lintas dengan menggunakan algoritma AdaBoost dan *random forest* 87,8% (Muslikh et al., 2018).

Dengan menggunakan metode seleksi fitur *Particle Swarm Optimization* (PSO) dapat memperbaiki attribute pada dataset sehingga meningkatkan hasil dari pengklasifikasian. Penggunaan PSO dapat dikombinasikan dengan algoritma lain, seperti mengkombinasikan dengan algoritma SVM yang digunakan untuk memprediksi data keuangan sehingga dapat mencegah perusahaan dari kebangkrutan. Penggunaan PSO dapat meningkatkan nilai akurasi terhadap algoritma SVM sehingga menunjukkan nilai akurasi menjadi 99,6% (Amalia et al., 2017).

Penanganan ketidak seimbangan kelas secara umum terdapat tiga pendekatan untuk menangani dataset tidak seimbang (Imbalanced), yaitu pendekatan pada level data, level algoritmik dan menggabungkan atau memasang (ensemble) metode. Pendekatan pada level data mencakup berbagai teknik resampling, memanipulasi data latih untuk memperbaiki kecondongan distribusi kelas, seperti Random Over-Sampling (ROS) dan Random Under-Sampling (RUS), dan Synthetic Minority Over-sampling Technique (SMOTE), Pendekatan level algoritma menggunakan teknik ensemble Adaboost dengan algoritma pengklasifikasi yang digunakan adalah Random Forest. Penelitian menggunakan dataset National Aeronautics and Space Administration (NASA) Metrics Data Program (MDP) (Tang & Chen, 2019).

B. TINJAUAN PUSTAKA.

Istilah cacat, kesalahan dan bug biasa digunakan untuk kesalahan dalam kode sumber (source code), di mana kesalahan adalah suatu tindakan yang keliru yang dibuat oleh pengembang. Cacat software dapat menjadi penyebab kegagalan software, yang terjadi ketika pengguna mengalami perilaku sistem yang tidak diinginkan. Cacat software adalah segala kekurangan yang menyebabkan software tidak dapat memenuhi apa yang diharapkan. Misalnya user ingin software yang digunakan dapat memberikan hasil penghitungan yang tepat, tetapi software memberikan hasil yang salah. Pencarian dan perbaikan cacat software membutuhkan biaya yang paling besar pada saat pengembangan software dibandingkan dengan biaya lainnya (Singleton, 2016). Banyaknya kesalahan pada software mengakibatkan penurunan kualitas dan kepercayaan pengguna. Maka perbaikan cacat pada software dianggap sebagai biaya termahal dibanding biaya pengemabngan dan mengakibatkan penambahan jadwal proyek.

Tabel 1. Contoh umum cacat *software*

Harapan	Cacat
Software dapat membantu menyelesaikan pekerjaan	Fungsionalitas software tidak ada
Mengklik tombol untuk mengerjakan suatu proses	Mengklik tombol, tetapi tidak ada atau tidak sesuai dengan proses yang diinginkan
File dapat di-copy ke lokasi lain	Terjadi kerusakan file selama proses peng-copy-an

Metrik kode statis (static code metrics) adalah pengukuran langsung kode sumber (source code) yang dapat digunakan dalam upaya untuk mengukur berbagai properti software. Metrik kode statik yang paling dikenal adalah metrik yang didasarkan pada jumlah baris kode (LoC: Line of Code), dan memberikan indikasi ukuran perangkat lunak. Panjang metrik adalah jumlah dari semua operator dan operan, dan merupakan alternatif pengukuran ukuran untuk mereka yang berbasis hitungan LoC (Lines Of Code). Volume metrik menggambarkan kandungan informasi dalam bit, dan ukuran terkait lain. Metrik/tingkat kesulitan telah diklaim untuk mengukur betapa sulitnya kode itu ditulis dan menggambarkan seberapa rawan kesalahan itu mungkin terjadi. Kebalikan dari tingkat metrik ini adalah tingkat yang lebih rendah menunjukkan sedikit rawan terjadi kesalahan penulisan kode.

Ketidak Seimbangan kelas (class imbalance) mengacu pada dataset yang menunjukkan ketidak Seimbangan yang signifikan di antara atau di dalam kelas tersebut. Pemahaman umum tentang “ketidak Seimbangan” dalam literatur berkaitan dengansituasi, di mana beberapa kelas data sangat kurang terwakili dibandingkan dengan kelas lainnya. Secara konvensi, kelas yang memiliki lebih banyak disebut kelas mayoritas, dan yang memiliki lebih sedikit disebut kelas minoritas. Pengenalan kelas ini penting, karena kegagalan menemukan cacat dapat sangat menurunkan kualitas perangkat lunak.

Banyak metode yang telah diusulkan untuk mengatasi masalah ketidak Seimbangan kelas di level data dan level algoritma. Metode level data mencakup berbagai teknik resampling, memanipulasi data latih untuk memperbaiki kecondongan distribusi kelas, seperti Random Over Sampling (ROS).

Salah satu metode untuk seleksi fitur atau atribut adalah menggunakan particle swarm optimization (PSO). Dengan pengoptimalan berbasis populasi yang dikembangkan oleh Eberhart dan Kennedy pada tahun 1995, masing-masing partikel menyesuaikan posisinya di ruang pencarian dari waktu ke waktu sesuai pengalaman terbangnya sendiri dan tetangganya (Chopard & Tomassini, 2018).

```
1 public class Metrik{
2     public static void main(String[] args){
3         int jumlahParameter = args.length;
4
5         if (jumlahParameter < 1){
6             System.out.println("Tidak ada parameter yang diberikan");
7         } else {
8
9             //Menampilkan parameter
10            for (int i=0; i<jumlahParameter.length; i++){
11                System.out.println(args[i]);
12            }
13
14        }
15    }
16 }
```

Gambar 1. Contoh kode sumber java

Resampling merupakan konsep penting dalam statistik inferensial. Hal ini digunakan untuk menarik sejumlah besar sampel dari yang asli untuk mencapai perkiraan distribusi teoritis yang mendasari. Hal ini didasarkan pada perulangan sampel dalam kumpulan data yang sama. Pendekatan sampel adalah salah satu pendekatan untuk memecahkan masalah ketidak seimbangan kelas dalam dataset. Pendekatan sampel yang umum digunakan adalah Random Over Sampling (Li et al., 2015).

Masalah ketidak seimbangan kelas juga dapat menyebabkan masalah lain karena ukuran kecil dari kelas minoritas. Hal Ini sering disebabkan oleh perbedaan dalam kelas. Ketika mempertimbangkan masalah ketidak seimbangan kelas, di mana data kelas minoritas yang kecil, kemampuan algoritma pembelajaran (*learning algorithms*) untuk menemukan pola dalam data kelas minoritas tidak dapat diandalkan. Namun dengan meningkatkan jumlah kelas minoritas dapat meningkatkan kemampuan algoritma pembelajar menjadi lebih baik, karena bisa mengenali sampel kelas minoritas dari mayoritas.

Resampling digunakan sebagai sarana mengubah distribusi kelas minoritas sehingga tidak kurang terwakili ketika pelatihan algoritma pembelajar. untuk mengatasi masalah ketidak seimbangan kelas, yaitu oversampling kelas minoritas, undersampling kelas mayoritas, atau penggunaan pendekatan hybrid didasarkan pada keduanya. Maka penelitian ini menggunakan Random Under-Sampling (RUS) dan Random Over-Sampling (ROS).

1. Algoritma Random Over-Sampling (ROS)

Pada algoritma ROS, data kelas minoritas dipilih secara acak kemudian ditambahkan ke dalam data latih. Maka Proses pemilihan dan penambahan ini diulang-ulang sampai jumlah data kelas minoritas sama dengan jumlah kelas mayoritas. Pertama dihitung selisih antara kelas mayoritas dengan kelas minoritas. Kemudian dilakukan perulangan sebanyak hasil penghitungan selisih sambil membaca data kelas minoritas secara acak, dan ditambahkan ke dalam data latih.

2. Algoritma Random Under Sampling (RUS)

Hampir sama dengan ROS, pertama dihitung selisih antara kelas mayoritas dengan kelas minoritas. Kemudian dilakukan perulangan sejumlah selisih hasil penghitungan, selama perulangan data kelas mayoritas dihapus secara acak, sehingga jumlah kelas mayoritas sama dengan jumlah kelas minoritas.

Pendekatan level algoritma yang digunakan adalah teknik ensemble Adaboost berbasis Random Forest. Boosting (Adaboost) adalah pendekatan pada *machine learning* untuk meningkatkan peraturan prediksi yang akurat dengan menggabungkan banyak peraturan yang relatif lemah dan tidak akurat. Adaptive boosting (adaboost) merupakan salah satu dari beberapa varian pada algoritma boosting. Boosting juga bisa dikombinasikan dengan classifier algoritma yang lain untuk meningkatkan performa klasifikasi. Tentunya secara intuitif, penggabungan beberapa model akan membantu jika model tersebut berbeda satu sama lain (Onan et al., 2016).

Sampel adaboost diperoleh dengan melakukan teknik resampling dengan penggantian (with replacements) dari dataset asli sehingga menghasilkan jumlah elemen yang sama dari dataset asli. AdaBoost merupakan salah satu algoritma boosting yang telah menunjukkan dan dapat memperbaiki kinerja pengklasifikasi. Maka penelitian ini mengusulkan penerapan teknik ensemble. Boosting telah menunjukkan dapat meningkatkan kinerja pengklasifikasi dalam banyak situasi, termasuk ketika data tidak seimbang (Adegoke et al., 2017).

Random Forest (RF) merupakan salah satu metode yang digunakan untuk klasifikasi dengan membangun banyak pohon klasifikasi. Random Forest adalah sebuah pendekatan pembelajaran ensemble untuk klasifikasi dimana “pelajar yang lemah” berkolaborasi untuk membentuk “pelajar yang kuat” dengan menggunakan banyak koleksi dari dekorasi pohon keputusan. Sedangkan pendapat lainnya Random Forest adalah metode ensemble learning yang pertama kali diusulkan oleh Breiman tahun 2001 yang merupakan kombinasi dari pohon klasifikasi sedemikian rupa sehingga setiap pohon bergantung pada nilai acak vektor sampel secara mandiri dan dengan distribusi yang sama untuk semua pohon di hutan (Xiao, Xie, He, & Jiang, 2012). Random Forest telah banyak digunakan pada penelitian sebelumnya untuk klasifikasi maupun untuk regresi karena kinerjanya yang unggul dan strukturnya yang sederhana (Lin, Wang, Xie, & Zhong, 2017).

C. METODE PENELITIAN

Penelitian ini menggunakan pendekatan kuantitatif sebagaimana diketahui bahwa metode kuantitatif atau disebut juga metode discovery, karena dengan metode ini dapat ditemukan dan dikembangkan iptek baru dan disebut metode kuantitatif karena data penelitian berupa angka-angka dan analisis menggunakan statistik (Sugiyono, 2016).

Pada Objek data penelitian ini menggunakan beberapa sample dataset dari Nasa MDP repository bentuk pertama yang dapat diunduh dari NASA MDP *repository* pada halaman <https://github.com/klainfo/NASADefectDataset>.

Pada penelitian ini teknik pengolahan data terbagi menjadi dua tahapan yakni tahapan *pre-processing* dan tahapan *processing*. Pada tahapan *pre-processing* dilakukan pendekatan pada level data dengan menggunakan teknik PSO dan ROS. Teknik PSO digunakan untuk menyeleksi *attribute* data dan teknik ROS digunakan untuk mengatasi ketidak seimbangan kelas pada *dataset software metrics*. Pada tahapan *processing* data penelitian ini menggunakan teknik *ensemble adaboost* untuk mengoptimalkan kinerja algoritma *decision tree* Random Forest. Dengan Pengujian yang dilakukan menggunakan *10-fold cross validation* dengan mengambil 10% *dataset* untuk dijadikan sebagai data *testing* dan dilakukan secara berulang sehingga didapatkan tingkat akurasi dan AUC kinerja model.

D. HASIL DAN PEMBAHASAN

Pada penelitian ini dilakukan eksperimen untuk melakukan pengujian terhadap model yang diusulkan. Penelitian ini menggunakan komputer dengan spesifikasi perangkat keras prosesor Intel(R) Core (TM) i5-3210M CPU @ 2.10 GHz, memori (RAM) 8,00 GB, monitor standar dan menggunakan sistem operasi Windows 10 Profesional 64bit. Dan Aplikasi mechine learning yang digunakan adalah WEKA versi 3.8 untuk software data mining.

Obyek penelitian yang akan digunakan dalam penelitian ini menggunakan dataset NASA MDP repository yang diunduh pada url berikut ini <https://github.com/klainfo/NASADefectDataset>.

Dataset yang digunakan dalam penelitian ini yaitu dataset yang sudah ditransformasi terlebih dahulu. Model yang diusulkan meliputi penerapan seleksi fitur menggunakan particle swarm optimization (PSO), Algoritma pendekatan level data Random Over-Sampling (ROS), Dan mengkombinasi algoritma Adaboost dengan algoritma pengklasifikasi Random Forest, Hasil akhir akan dilakukan perbandingan pengujian dan analisa terhadap model prediksi yang memiliki tingkat akurasi tertinggi atau terbaik dalam prediksi cacat software.

Berdasarkan model yang diusulkan terdiri dari dua pendekatan, yaitu pendekatan level data dan pendekatan level algoritma. Kedua pendekatan akan digunakan bergantian, dan gabungan dari keduanya untuk membuat berbagai model prediksi cacat software. Pendekatan level data ditujukan untuk menyeimbangkan kelas dalam dataset kualitas software yang pada umumnya bersifat tidak seimbang. Pada pendekatan level data yang diusulkan yaitu, Random Over-Sampling (ROS) bekerja dengan memilih secara acak data dari kelas minoritas dan menambahkan ke data latih sampai jumlah data kelas minoritas sama dengan data kelas mayoritas. Konsep Random Over Sampling (ROS) dengan memilih kelas minoritas secara acak kemudian ditambahkan ke dalam data latih. dari Proses pemilihan dan penambahan ini diulang-ulang sampai jumlah data kelas minoritas sama dengan jumlah kelas mayoritas.

Pendekatan level algoritma ditujukan untuk memperbaiki kinerja pengklasifikasi dengan teknik ensemble menggunakan algoritma Adaboost. Adaboost menggabungkan sejumlah iteration pengklasifikasi Random Forest yang diberi pelatihan dengan data berbeda yang diambil secara acak dengan pergantian dari data latih dengan ukuran yang sama. Penerapan dataset pada model yang terbentuk divalidasi menggunakan 10 fold cross validation.

Cross validation adalah metode statistik untuk mengevaluasi dan membandingkan algoritma pembelajaran (learning algorithms) dengan membagi data menjadi dua segmen, satu segmen digunakan untuk belajar atau data latih, dan yang lain digunakan untuk memvalidasi. Cross validation kumpulan pelatihan dan validasi harus crossover berturut-turut sehingga setiap data memiliki kesempatan tervalidasi. K-fold cross validation adalah teknik umum untuk memperkirakan kinerja pengklasifikasi. K-fold cross validation dilakukan dengan menggunakan kembali dataset yang sama, sehingga menghasilkan k perpecahan dari kumpulan data menjadi non-overlapping dengan proporsi pelatihan $(k-1)/k$ dan $1/k$ untuk pengujian.

Data Set									
Split 1	Split 2	Split 3	Split 4	Split 5	Split 6	Split 7	Split 8	Split 9	Split 10
Test	Training								
Training	Test	Training							
Training	Training	Test	Training						
Training	Training	Training	Test	Training					
Training	Training	Training	Training	Test	Training				
Training	Training	Training	Training	Training	Test	Training			
Training	Training	Training	Training	Training	Training	Test	Training		
Training	Test	Training							
Training	Test	Training							

Gambar 5. Pembagian Dataset untuk 10-Fold Cross Validation

Kinerja model prediksi cacat software diukur berdasarkan akurasi, sensitivitas, F-Measure, G-Mean, dan Area Under ROC Curve (AUC). Nilai AUC dapat dijadikan ukuran untuk melihat model yang terbentuk. AUC digunakan untuk memberikan metrik numerik single untuk dapat membandingkan kinerja dari model, nilai AUC berkisar dari 0 sampai 1 dan model yang lebih baik prediksinya adalah yang mendekati nilai 1.

Confusion matrix merupakan alat yang berguna untuk menganalisa seberapa baik pengklasifikasi dapat mengenali tupel/fitur dari kelas yang berbeda juga dapat membantu

menunjukkan rincian kinerja pengklasifikasi dengan memberikan informasi jumlah fitur suatu kelas yang diklasifikasikan dengan tepat dan tidak tepat. Confusion matrix merupakan matrik 2 dimensi yang menggambarkan perbandingan antara hasil prediksi dengan kenyataan, yang memberikan penilaian kinerja model klasifikasi berdasarkan jumlah objek yang diprediksi dengan benar dan salah.

Tabel 2. Confusion Matrix

Kelas		Nilai Sebenarnya	
		Benar	Salah
Nilai Prediksi	Benar	TP (True Positive)	FP (False Positive)
	Salah	FN (False negative)	TN (True Negative)

Teknik analisis dilakukan dengan praktek uji coba dilakukan untuk membandingkan kinerja semua model yang di kembangkan dengan seleksi fitur particle swarm optimization (PSO) dan pendekatan level data Random Over-Sampling (ROS) dan pendekatan level algoritma (Adaboost) dengan pengklasifikasi Random Forest. Analisis ini ditujukan untuk mengetahui algoritma dan pendekatan mana yang terbaik untuk mengurangi pengaruh ketidak seimbangan kelas, sehingga keakuratan prediksi kelas minoritas dapat ditingkatkan. Untuk mempermudah analisa digunakan aplikasi SPSS versi 22 64 bit. Uji statistik yang digunakan adalah uji test sampel berpasangan (paired-samples t-test).

Penelitian menggunakan dataset dari National Aeronautics and Space Administration (NASA) Metrics Data Program (MDP) repository dengan menggunakan 4 dataset yaitu CM1, MW1, PC1 dan PC4, sebagaimana digunakan pada penelitian sebelum-sebelumnya. Tahap preprocessing dilakukan dengan melihat beberapa ketentuan awal yang perlu menjadi perhatian diantaranya jumlah atribut, jumlah modul dan jumlah cacat pada setiap dataset.

Tabel 3. Spesifikasi 4 Dataset Nasa MDP Repository

Dataset	Atribut	Modul	Cacat	Cacat (%)
CM1	38	327	42	12,84%
MW1	38	250	25	10,00%
PC1	38	679	55	8,10%
PC4	38	1270	176	13,86%

Pengujian dengan menggunakan model pengklasifikasi tunggal Random Forest terhadap keempat dataset, hasil diperoleh sebagai berikut.

Tabel 4. Hasil perhitungan dengan Aplikasi Weka 3,8 menggunakan dataset MW1

	Predicted: NO	Predicted: YES	
Actual YES	TN = 6	FP = 21	42
Actual NO	FN = 9	TP = 277	302
	15	298	

Perhitungan yang sama dilakukan bergantian terhadap keempat dataset yaitu CM1, MW1, PC1 dan PC4. Hasil dari keseluruhan dapat diperlihatkan pada tabel berikut.

Tabel 5. Hasil Kinerja Random Forest Pada 4 Dataset MDP Repository

Dataset	CM1	MW1	PC1	PC4
<i>Tprate</i>	0	0,2	0,1	0,38
<i>Fprate</i>	1	0,7	0,8	0,61
<i>Precession</i>	0	0,9	0,	0,91
Sensitivitas	0,8	0,9	0,9	0,91
<i>Specificity</i>	0	0,2	0,4	0,72
<i>F-measure</i>	0,8	0,9	0,9	0,91
<i>G-mean</i>	0	0,4	0,6	0,81
<i>Accuracy</i>	0,8	0,8	0,	0,92
AUC	0,4	0,5	0,6	0,82

Pengujian Model Seleksi Fitur PSO, ROS, Adaboost dan Random Forest Model menggunakan teknik seleksi fitur Particle Swarm Optimization (PSO) dengan kategori attribute selection. Hasil dari seleksi fitur menggunakan PSO selanjutnya dilakukan teknik Random Over-sampling (ROS) untuk keseimbangan kelas dan selanjutnya dilakukan teknik ensemble Adaboost dengan pengklasifikasi Random Forest dan teknik validasi menggunakan 10-Fold Cross Validation. Hasil yang diperoleh terhadap model yang diusulkan disajikan pada tabel berikut.

Tabel 6. Hasil Kinerja Model PSO, ROS, Adaboost dan Random Forest

Dataset	CM1	MW1	PC1	PC4
<i>Tprate</i>	1	0,954	1	0,991
<i>Fprate</i>	0,2	0,46	0,2	0,09
<i>Precessio</i>	0,14	0,09	0,057	0,07
Sensitivit	0,86	0,901	0,943	0,93
<i>Specificity</i>	1	0,954	1	1
<i>F-</i>	0,925	0,925	0,97	0,97
<i>G-mean</i>	0,927	0,927	0,97	0,97
<i>Accuracy</i>	0,93	0,927	0,971	0,962
AUC	0,939	0,929	0,927	0,964

Tabel diatas menunjukkan bahwa akurasi tertinggi dari kinerja model usulan PSO+ROS+AB+RF terdapat pada dataset PC1 dengan akurasi mencapai 97%, Namun nilai AUC tertinggi di peroleh pada dataset PC1 dan PC4 sebesar 0,927 dan 0,964. Berikut ini diagram perbandingan dari keseluruhan variabel yang dihitung menggunakan model usulan PSO+ROS+AB+RF pada keempat dataset yang telah diuji coba.

Tabel 7. Perbandingan Kinerja Akurasi Kedua Model

NILAI AKURASI		
MODEL	Random Forest	PSO+ROS+AD+RF
CM1	0,869	0,93
MW1	0,885	0,927
PC1	0,91	0,971
PC4	0,92	0,962
<i>Average</i>	0,896	0,9475

Tabel diatas menerangkan kinerja akurasi dari tiap model terhadap masing-masing dataset. Perbedaan kedua model yang telah dilakukan percobaan dapat dilihat lebih jelas pada diagram berikut

Penelitian terkait membahas tentang penelitian sejenis yang telah dilakukan oleh peneliti lain dengan metode yang berbeda. Perbandingan hasil dari penelitian sebelumnya akan menunjukkan seberapa berpengaruh penelitian ini terhadap kontribusi bidang pengetahuan walaupun unsur suatu keberhasilan penelitian tidak selalu diukur dari seberapa baik angka yang di dihasilkan. Beberapa penelitian sejenis yang telah dilakukan diantaranya menggunakan teknik seleksi fitur Information Gain (IG) dan RLF dengan teknik pendekatan level data SMOTE serta pengklasifikasi Naive Bayes.

Tabel 8. Perbandingan Hasil Penelitian Sebelumnya

Model	Hasil CM1	Hasil MW1	Hasil PC1	Hasil PC4	Rata-Rata
Wahono, et al (2014), NB dengan PSO dan BG	0,75	0,74	0,79	0,85	0,78
Putri, Wahono (2015), NB dengan SMOTE and IG	0,75	0,76	0,817	0,85	0,79
Putri, Friyadi (2017), NB dengan SMOTE dan RLF	0,76	0,77	0,821	0,86	0,80
Richky Faizal Amir PSO+ROS+AB+RF	0,96	0,93	0,92	0,97	0,94

Hasil perbandingan dengan penelitian sebelumnya menunjukkan bahwa kinerja dari model prediksi secara statistik rata-rata lebih unggul.

E. KESIMPULAN

Dari Model yang diusulkan diterapkan pada Dataset dari National Aeronautics and Space Administration (NASA) Metrics Data Program (MDP) sebagai software metrics. Dataset yang digunakan yaitu CM1, MW1, PC1 dan PC4 sesuai dengan penelitian sebelumnya agar hasil penelitian dapat di bandingkan. Hasil penelitian menunjukkan bahwa model yang diusulkan yaitu seleksi fitur swarm optimization (PSO) dengan pendekatan level data Resample dan teknik ensemble Adaboost dengan pengklasifikasi Random Forest mampu meningkatkan klasifikasi secara keseluruhan dengan nilai AUC lebih tinggi 0,94 dari model Naive Bayes yang memiliki rata rata 0,805.

Kinerja model usulan PSO+ROS+AB+RF dapat dilihat dari rata-rata nilai AUC lebih tinggi 0,94 dari model usulan PSO+RF yang memiliki rata rata AUC 0,636. Kesimpulannya bahwa model usulan dari PSO+ROS+AB+RF dapat meningkatkan kinerja model pengklasifikasi. Hasil ini diperoleh dari nilai AUC yang dapat mencapai 94%. Hasil yang diperoleh dari variabel sensitivitas, spesificity, G-Mean, F-measure dan AUC dapat menjawab identifikasi masalah tentang ketidak seimbangan kelas dapat menurunkan kinerja model prediksi.

REFERENSI

Adegoke, V. F., Chen, D., Banissi, S., & Banissi, E. (2017). Predictive Ensemble Modelling -

- Experimental Comparison of Boosting Implementation Methods. *Proceedings - UKSim-AMSS 11th European Modelling Symposium on Computer Modelling and Simulation, EMS 2017*, 11–16. <https://doi.org/10.1109/EMS.2017.13>
- Amalia, H., Lestari, A. F., & Puspita, A. (2017). Penerapan Metode Svm Berbasis Pso Untuk Penentuan Kebangkrutan Perusahaan. *Jurnal Techno Nusa Mandiri*, 14(2), 131–136.
- Biau, G., & Scornet, E. (2016). A random forest guided tour. *Test*, 25(2), 197–227. <https://doi.org/10.1007/s11749-016-0481-7>
- Chopard, B., & Tomassini, M. (2018). Particle Swarm Optimization. In: An Introduction to Metaheuristics for Optimization. *Natural Computing Series. Springer, Cham*, 23(2), 145–156. <https://doi.org/10.1007/978-3-319-93073-2>
- Hofman, R. (2011). Behavioral economics in software quality engineering. *Empirical Software Engineering*, 16(2), 278–293. <https://doi.org/10.1007/s10664-010-9140-x>
- Laradji, I. H., Alshayeb, M., & Ghouti, L. (2014). Software defect prediction using ensemble learning on selected features. *INFORMATION AND SOFTWARE TECHNOLOGY*. <https://doi.org/10.1016/j.infsof.2014.07.005>
- Li, R. H., Yu, J. X., Qin, L., Mao, R., & Jin, T. (2015). On random walk based graph sampling. *Proceedings - International Conference on Data Engineering, 2015-May*, 927–938. <https://doi.org/10.1109/ICDE.2015.7113345>
- Muslikh, A. R., Santoso, H. A., Marjuni, A., Teknik, P., Universitas, I., & Nuswantoro, D. (2018). *Klasifikasi Data Time Series Arus Lalu Lintas*. 14, 24–38.
- Onan, A., Korukoğlu, S., & Bulut, H. (2016). Ensemble of keyword extraction methods and classifiers in text classification. *Expert Systems with Applications*, 57, 232–247. <https://doi.org/10.1016/j.eswa.2016.03.045>
- Rais, A. N., & Subekti, A. (2019). Integrasi SMOTE Dan Ensemble AdaBoost Untuk Mengatasi Imbalance Class Pada Data Bank Direct Marketing. *Jurnal Informatika*, 6(2), 278–285. <https://doi.org/10.31311/ji.v6i2.6186>
- Saifudin, A., Teknik, F., Pamulang, U., Wahono, R. S., Komputer, F. I., & Nuswantoro, U. D. (2015). *Penerapan Teknik Ensemble untuk Menangani Ketidakseimbangan Kelas pada Prediksi Cacat Software*. 1(1).
- Singleton, A. (2016). The Economics of Microservices. *IEEE Cloud Computing*, 3(5), 16–20. <https://doi.org/10.1109/MCC.2016.109>
- Tang, X., & Chen, L. (2019). Artificial bee colony optimization-based weighted extreme learning machine for imbalanced data learning. *Cluster Computing*, 22, 6937–6952. <https://doi.org/10.1007/s10586-018-1808-9>
- Wahono, R. S. (2007). A Systematic Literature Review of Software Defect Prediction: Research Trends, Datasets, Methods and Frameworks. *Journal of Software Engineering*, 1(1), 1–16. <https://doi.org/10.3923/jse.2007.1.12>
- Wahono, R. S., Suryana, N., & Ahmad, S. (2014). *Metaheuristic Optimization based Feature Selection for Software Defect Prediction*. 9(5), 1324–1333. <https://doi.org/10.4304/jsw.9.5.1324-1333>