

**PREDIKSI CACAT PADA PERANGKAT LUNAK UNTUK KELAS
TIDAK SEIMBANG DENGAN MENGGUNAKAN
RESAMPLE ADABOOST DAN BAGGING J48**



TESIS

Diajukan sebagai salah satu syarat untuk memperoleh gelar
Ilmu Komputer (S2) (M.Kom)

SUGIONO

14002160

PROGRAM STUDI ILMU KOMPUTER (S2)
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
NUSA MANDIRI
JAKARTA
2019

PERSETUJUAN DAN PENGESAHAN TESIS

Tesis ini diajukan oleh:

Nama : Sugiono
NIM : 14002160
Program Studi : Ilmu Komputer
Jenjang : Program STRATA-2
Judul Tesis : Prediksi Cacat Pada Perangkat Lunak Untuk Kelas Tidak Seimbang Dengan Menggunakan Resample Adaboost Dan Bagging J48

Telah dipertahankan pada periode 2019-1 dihadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh MAGISTER ILMU KOMPUTER (M.Kom) pada Program STRATA-2 Program Studi Ilmu Komputer di Sekolah Tinggi Manajemen Informatika dan Komputer Nusa Mandiri (STMIK Nusa Mandiri).

Jakarta, 21 Agustus 2019

PEMBIMBING TESIS

Dosen Pembimbing : Risnandar, Ph.D.



DEWAN PENGUJI

Penguji I : Dr. Dwiza Riana, S.Si, MM, M.Kom



Penguji II : Dr. Tr. Lindung Parningotan, M.T.I



Penguji III / Dosen Pembimbing : Risnandar, Ph.D.



DAFTAR ISI

1. PENDAHULUAN	4
2. LANDASAN/ KERANGKA PEMIKIRAN	4
3. METODOLOGI PENELITIAN.....	5
4. HASIL DAN PEMBAHASAN.....	7
3.1.Preprocessing Data	
3.2.Eksperimen dan Pengujian	
3.3.Evaluasi Hasil Kinerja Model Usulan	
3.4.Uji Statistik Hasil Kinerja Model Usulan	
3.5.Perbandingan Penelitian Terkait	
5. KESIMPULAN	12

1. PENDAHULUAN

Kualitas suatu software dapat ditemukan pada saat fase pemeriksaan dan fase pengujian (Fitriani & Wahono, 2015). Cacat pada perangkat lunak merupakan kesalahan, kegagalan atau kesalahan pada sistem komputer tersebut sehingga menghasilkan kesalahan lainnya yang tak terduga serta dapat menurunkan kualitas suatu perangkat lunak (Wahono, Dian, Semarang, & Suryana, 2013).

Prediksi kesalahan pada perangkat lunak merupakan langkah penting untuk meningkatkan kualitas perangkat lunak. Kesalahan yang terdapat pada perangkat lunak di masa depan merupakan kesalahan pada data yang salah sebelumnya (Sathyaraj & Prabu, 2015). Oleh karena itu diperlukan adanya penelitian yang matang dalam memprediksi cacat pada perangkat lunak

Banyak metode yang dapat digunakan untuk memprediksi modul cacat *software* dari potensi rawan kegagalan, salah satunya yang efektif adalah dengan menggunakan reka teknik data mining yang diaplikasikan pada *software matrix* yang diperoleh selama proses pengembangan *software* (Khoshgoftaar, 2010). Salah satu algoritma yang dapat diterapkan untuk memprediksi modul cacat *software* adalah algoritma *decision tree*.

Dalam penelitian mengenai software defect prediction data metrik yang digunakan sebagai acuan adalah dataset NASA MDP. Penggunaan NASA dataset merupakan pilihan yang tepat karena mudah diperoleh dan dapat dibandingkan kinerjanya dengan penelitian yang sebelumnya (Saifudin & Wahono, 2015). Dataset NASA berasal dari NASA Metric Data Program (MDP) repository dan Predictor Models in Software Engineering (PROMISE) repository. (Gray, Bowes, Davey, Sun, & Christianson, 2011).

Untuk mengatasi ketidak seimbangan kelas dapat dilakukan melalui pendekatan level data. Pendekatan level data yang digunakan berupa metode Resampling termasuk Random Over Sampling (ROS), Random Under Sampling (RUS) dan Synthetic Minority Over-sampling Technique (SMOTE) untuk menangani ketidak seimbangan kelas dengan cara menduplikasi kelas minoritas agar setara dengan kelas mayoritas sehingga dataset menjadi lebih seimbang (Yap et al., 2014).

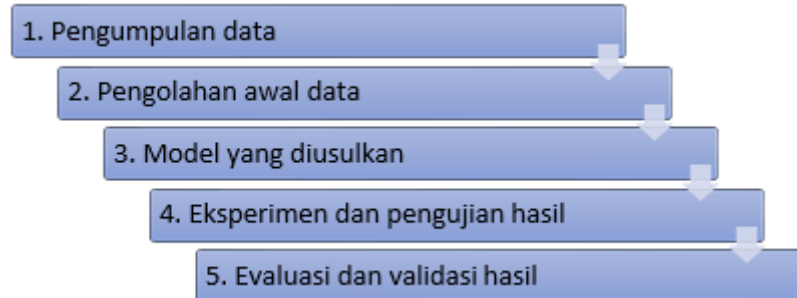
Tujuan penelitian ini adalah untuk menemukan pendekatan level data yang sesuai untuk menangani ketidak seimbangan kelas serta pendekatan algoritma yang dapat meningkatkan kinerja algoritma pengklasifikasian dengan menerapkan model usulan yaitu integrasi seleksi fitur PSO dengan pendekatan level level data ROS, RUS dan SMOTE serta pendekatan level algoritma AdaBoost dan Bagging terhadap algoritma pengklasifikasian berbasis decision tree J48.

2. LANDASAN/ KERANGKA PEMIKIRAN

Berdasarkan studi yang dilakukan terhadap beberapa penelitian yang telah dilakukan terkait model prediksi cacat software yang membahas mengenai ketidak seimbangan kelas, berbagai metode telah digunakan oleh para peneliti. State of the art yang dapat diambil dari beberapa penelitian terkait, secara umum studi mengenai ketidak seimbangan kelas pada model prediksi cacat software dapat dibagi menjadi dua pendekatan. Pendekatan pertama yaitu pendekatan level data, yang diimplementasikan untuk memperbaiki kualitas data dengan cara menyeimbangkannya dengan menggunakan teknik resampling seperti ROS dan RUS, atau mensintesa data menggunakan teknik SMOTE serta menyeleksi fitur yang tepat (feature selection). Sedangkan pendekatan yang kedua yaitu pendekatan level algoritma dengan memilih algoritma yang tepat untuk mengklasifikasi cacat software serta meningkatkan kemampuan algoritma tersebut dengan metode AdaBoost dan Bagging.

3. METODOLOGI PENELITIAN

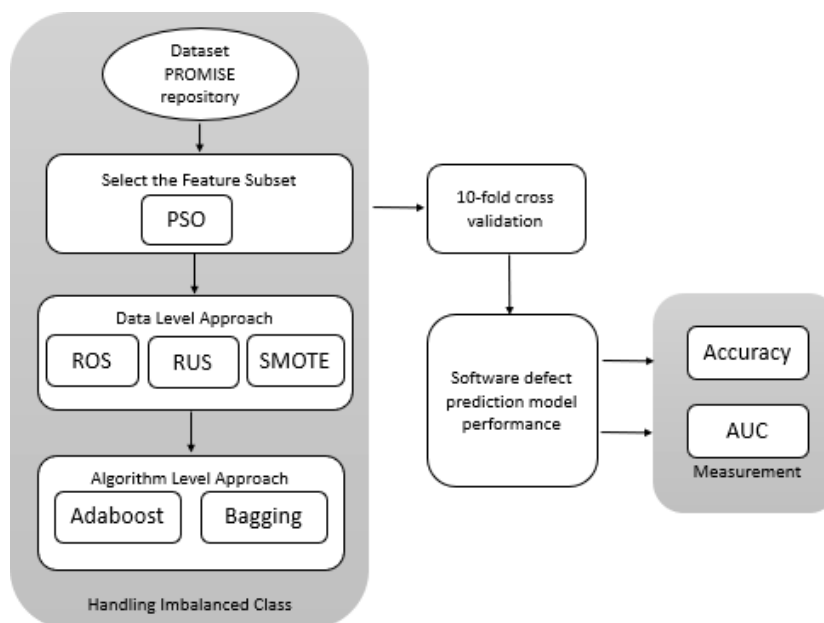
Tujuan dari penelitian ini untuk mencari solusi mengenai ketidak seimbangan kelas pada dataset model prediksi cacat software serta menemukan model yang dapat mengoptimalkan kinerja algoritma dalam melakukan pengklasifikasian. Berikut ini tahapan-tahapan yang dilakukan pada penelitian ini.



Gambar 3.1.

Tahapan Penelitian

Kerangka pemikiran pada penelitian ini terdiri dari beberapa tahapan. Pembahasan permasalahan pada penelitian ini meliputi data yang mengandung noise, tidak relevan dan redundansi yang dapat mempengaruhi metaheuristik data serta jumlah kelas yang tidak sama sehingga dapat mengurangi kinerja pengklasifikasian J48. Untuk mengatasi permasalahan metaheuristik data digunakan pendekatan level data Partition Swarm Optimatation(PSO), untuk mengurangi pengaruh ketidak seimbangan data terhadap model pengklasifikasian J48 maka dilakukan pendekatan level data resampling (ROS, RUS dan SMOTE). Sedangkan untuk lebih meningkatkan kinerja pengklasifikasian J48 pada pendekatan level algoritma digunakan metode AdaBoost dan Bagging. Kemudian dilakukan pengukuran kinerja model menggunakan teknik cross validation, selanjutnya dilakukan perbandingan untuk mengetahui pendekatan level data dan algoritma yang terbaik.

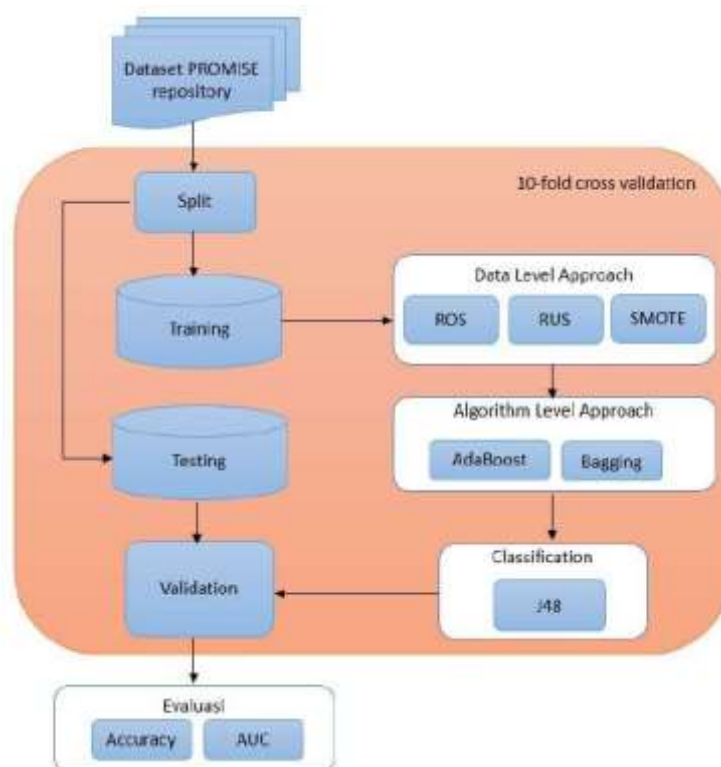


Gambar 3.2

Kerangka Pemikiran Penelitian

Pada penelitian ini data yang digunakan adalah NASA dataset yang diunduh dari PROMISE repository yang sudah banyak digunakan para peneliti dalam meneliti cacat software. Data ini tersedia secara umum di internet, oleh karena itu metode pengumpulan data yang digunakan yaitu dengan mengunduh NASA dataset dari PROMISE repository dengan url <http://promise.site.uottawa.ca/SERepository/>.

Model yang diusulkan pada penelitian ini menggunakan dua pendekatan yaitu pendekatan level data dan pendekatan level algoritma. Pada pendekatan level data diusulkan metode resampling (ROS dan RUS) dan SMOTE untuk mengatasi ketidak seimbangan kelas pada dataset software metric. Pendekatan yang kedua yaitu level algoritma, pada pendekatan ini dilakukan teknik assembling dengan menggabungkan pendekatan level algoritma AdaBoost dan Bagging dengan algoritma pengklasifikasian J48. Berikut ini merupakan kerangka model penelitian yang diusulkan.



Gambar 3.3
Kerangka kinerja yang diusulkan

4. HASIL DAN PEMBAHASAN

4.1. Preprocessing Data

Dataset PROMISE repository merupakan data bersih yang sudah terbebas dari inkonsistensi serta *anomaly* sehingga tidak perlu untuk dilakukan proses pengolahan data. *Dataset* yang digunakan pada penelitian ini yaitu *dataset* CM1, JM1, KC1, dan PC1. Tahap *preporcessing* dilakukan dengan melihat beberapa ketentuan awal yang perlu menjadi perhatian diantaranya jumlah atribut, jumlah modul dan jumlah cacat pada setiap *dataset*. Spesifikasi *dataset* yang digunakan dalam penelitian ini sebagai berikut.

Tabel 4.1
Spesifikasi dataset PROMISE repository yang digunakan dalam penelitian

Dataset	Atribut	Modul	Cacat	Tidak Cacat
CM1	22	498	49	449
JM1	22	10885	2106	8779
KC1	22	2109	326	1783
PC1	22	1109	77	1032

4.2. Eksperimen dan Pengujian

Eksperimen dilakukan dengan melakukan uji coba pengklasifikasian model usulan terhadap *dataset*. Pengujian yang dilakukan meliputi penerapan model seleksi fitur PSO dengan pendekatan level data *resampling* (ROS, RUS) dan SMOTE dan pendekatan level algoritma AdaBoost dan Bagging. Semua *dataset* diujikan secara bergantian dengan teknik *10-cross validation*. Pengujian dilakukan dengan aplikasi WEKA

4.3. Evaluasi Hasil Kinerja Model Usulan

Setelah dilakukan pengujian pada setiap model yang diusulkan langkah selanjutnya adalah melakukan evaluasi terhadap model usulan dengan melakukan perbandingan kinerja model pengklasifikasian yang diujikan. Perbandingan kinerja model pengklasifikasian disajikan dalam diagram perbandingan berdasarkan akurasi, sensitivitas, specificity, G-means, F-measure, precision dan AUC.

Tabel 4.2
Perbandingan nilai akurasi model terhadap dataset

Komparasi Akurasi	Dataset			
Model Usulan	CM1	JM1	KC1	PC1
PSO+ROS+J48	92,57%	85,50%	88,56%	95,85%
PSO+ROS+AdaBoost+J48	95,38%	90,29%	90,79%	97,56%
PSO+ROS+Bagging+J48	94,78%	88,14%	89,65%	96,93%
PSO+RUS+J48	63,27%	66,14%	66,56%	68,83%
PSO+RUS+AdaBoost+J48	61,22%	65,27%	69,32%	76,62%
PSO+RUS+Bagging+J48	55,10%	66,76%	71,16%	71,43%
PSO+SMOTE+J48	88,19%	83,18%	87,49%	92,34%
PSO+SMOTE+AdaBoost+J48	92,43%	85,71%	89,17%	95,45%
PSO+SMOTE+Bagging+J48	90,31%	87,04%	90,15%	94,23%

Tabel 4.3
Perbandingan nilai sensitivitas model terhadap dataset

Komparasi Sensitivitas	Dataset			
Model Usulan	CM1	JM1	KC1	PC1

PSO+ROS+J48	0,98	0,89	0,91	1,00
PSO+ROS+AdaBoost+J48	0,99	0,94	0,93	1,00
PSO+ROS+Bagging+J48	0,99	0,91	0,92	1,00
PSO+RUS+J48	0,63	0,66	0,69	0,78
PSO+RUS+AdaBoost+J48	0,61	0,66	0,71	0,83
PSO+RUS+Bagging+J48	0,57	0,68	0,72	0,77
PSO+SMOTE+J48	0,92	0,80	0,85	0,94
PSO+SMOTE+AdaBoost+J48	0,94	0,84	0,89	0,96
PSO+SMOTE+Bagging+J48	0,93	0,84	0,89	0,96

Tabel 4.4
Perbandingan nilai Specificity model terhadap dataset

Komparasi Specificity	Dataset			
Model Usulan	CM1	JM1	KC1	PC1
PSO+ROS+J48	0,88	0,82	0,86	0,92
PSO+ROS+AdaBoost+J48	0,92	0,87	0,89	0,95
PSO+ROS+Bagging+J48	0,90	0,85	0,87	0,94
PSO+RUS+J48	0,63	0,67	0,64	0,60
PSO+RUS+AdaBoost+J48	0,61	0,65	0,68	0,70
PSO+RUS+Bagging+J48	0,53	0,66	0,70	0,66
PSO+SMOTE+J48	0,85	0,86	0,90	0,91
PSO+SMOTE+AdaBoost+J48	0,91	0,87	0,90	0,95
PSO+SMOTE+Bagging+J48	0,88	0,90	0,91	0,92

Tabel 4.5
Perbandingan nilai G-Means model terhadap dataset

Komparasi G-Means	Dataset			
Model Usulan	CM1	JM1	KC1	PC1
PSO+ROS+J48	0,92	0,85	0,89	0,96
PSO+ROS+AdaBoost+J48	0,95	0,90	0,91	0,98
PSO+ROS+Bagging+J48	0,95	0,88	0,90	0,97
PSO+RUS+J48	0,63	0,66	0,67	0,68
PSO+RUS+AdaBoost+J48	0,61	0,65	0,69	0,76
PSO+RUS+Bagging+J48	0,55	0,67	0,71	0,71
PSO+SMOTE+J48	0,88	0,83	0,87	0,92
PSO+SMOTE+AdaBoost+J48	0,92	0,86	0,89	0,95
PSO+SMOTE+Bagging+J48	0,90	0,87	0,90	0,94

Tabel 4.6
Perbandingan nilai F-Measure model terhadap dataset

Komparasi F-Measure	Dataset			
Model Usulan	CM1	JM1	KC1	PC1
PSO+ROS+J48	0,93	0,86	0,89	0,96

PSO+ROS+AdaBoost+J48	0,96	0,91	0,91	0,98
PSO+ROS+Bagging+J48	0,95	0,88	0,90	0,97
PSO+RUS+J48	0,63	0,66	0,67	0,71
PSO+RUS+AdaBoost+J48	0,61	0,65	0,70	0,78
PSO+RUS+Bagging+J48	0,56	0,67	0,72	0,73
PSO+SMOTE+J48	0,89	0,83	0,87	0,92
PSO+SMOTE+AdaBoost+J48	0,93	0,86	0,89	0,95
PSO+SMOTE+Bagging+J48	0,91	0,87	0,90	0,94

Tabel 4.7
Perbandingan nilai Precision model terhadap dataset

Komparasi Precision	Dataset			
Model Usulan	CM1	JM1	KC1	PC1
PSO+ROS+J48	0,89	0,83	0,87	0,92
PSO+ROS+AdaBoost+J48	0,92	0,88	0,89	0,95
PSO+ROS+Bagging+J48	0,91	0,86	0,88	0,95
PSO+RUS+J48	0,63	0,66	0,66	0,66
PSO+RUS+AdaBoost+J48	0,61	0,65	0,69	0,74
PSO+RUS+Bagging+J48	0,55	0,66	0,71	0,69
PSO+SMOTE+J48	0,86	0,85	0,90	0,91
PSO+SMOTE+AdaBoost+J48	0,92	0,87	0,90	0,95
PSO+SMOTE+Bagging+J48	0,88	0,89	0,91	0,92

Tabel 4.8
Perbandingan nilai AUC model terhadap dataset

Komparasi AUC	Dataset			
Model Usulan	CM1	JM1	KC1	PC1
PSO+ROS+J48	0,93	0,86	0,89	0,96
PSO+ROS+AdaBoost+J48	0,95	0,90	0,91	0,98
PSO+ROS+Bagging+J48	0,95	0,88	0,90	0,97
PSO+RUS+J48	0,63	0,66	0,67	0,69
PSO+RUS+AdaBoost+J48	0,61	0,65	0,69	0,77
PSO+RUS+Bagging+J48	0,55	0,67	0,71	0,71
PSO+SMOTE+J48	0,88	0,83	0,87	0,92
PSO+SMOTE+AdaBoost+J48	0,92	0,86	0,89	0,95
PSO+SMOTE+Bagging+J48	0,90	0,87	0,90	0,94

4.4. Uji Statistik Hasil Kinerja Model Usulan

Berdasarkan hasil uji t yang pertama diketahui bahwa nilai signifikansi dari model PSO+ROS+J48 lebih kecil dari nilai p value sebesar 0,05. Dengan demikian dapat disimpulkan bahwa model PSO+ROS+J48 memberikan pengaruh yang signifikan terhadap model dasar PSO+J48. Hal ini sekaligus menjawab hipotesis pertama (H1).

Pengujian t yang ke dua diketahui bahwa nilai signifikansi dari model PSO+RUS+J48

lebih besar dari nilai p value sebesar 0,05. Dengan demikian dapat disimpulkan bahwa model PSO+RUS+J48 tidak memberikan pengaruh yang signifikan terhadap model dasar PSO+J48. Hal ini sekaligus menjawab hipotesis pertama (H2).

Pengujian t yang ke tiga diketahui bahwa nilai signifikansi dari model PSO+SMOTE+J48 lebih kecil dari nilai p value sebesar 0,05. Dengan demikian dapat disimpulkan bahwa model PSO+SMOTE+J48 memberikan pengaruh yang signifikan terhadap model dasar PSO+J48. Hal ini sekaligus menjawab hipotesis pertama (H3).

Pengujian t yang ke empat diketahui bahwa nilai signifikansi dari model PSO+ROS+AdaBoost+J48 lebih kecil dari nilai p value sebesar 0,05. Dengan demikian dapat disimpulkan bahwa model PSO+ROS+AdaBoost+J48 memberikan pengaruh yang signifikan terhadap model dasar PSO+J48. Hal ini sekaligus menjawab hipotesis pertama (H4).

Pengujian t yang ke lima diketahui bahwa nilai signifikansi dari model PSO+RUS+AdaBoost+J48 lebih besar dari nilai p value sebesar 0,05. Dengan demikian dapat disimpulkan bahwa model PSO+RUS+AdaBoost+J48 tidak memberikan pengaruh yang signifikan terhadap model dasar PSO+J48. Hal ini sekaligus menjawab hipotesis pertama (H5).

Pengujian t yang ke enam diketahui bahwa nilai signifikansi dari model PSO+SMOTE+AdaBoost+J48 lebih kecil dari nilai p value sebesar 0,05. Dengan demikian dapat disimpulkan bahwa model PSO+SMOTE+AdaBoost+J48 memberikan pengaruh yang signifikan terhadap model dasar PSO+J48. Hal ini sekaligus menjawab hipotesis pertama (H6).

Pengujian t yang ke tujuh diketahui bahwa nilai signifikansi dari model PSO+ROS+Bagging+J48 lebih kecil dari nilai p value sebesar 0,05. Dengan demikian dapat disimpulkan bahwa model PSO+ROS+Bagging+J48 memberikan pengaruh yang signifikan terhadap model dasar PSO+J48. Hal ini sekaligus menjawab hipotesis pertama (H7).

Pengujian t yang ke delapan diketahui bahwa nilai signifikansi dari model PSO+RUS+Bagging+J48 lebih besar dari nilai p value sebesar 0,05. Dengan demikian dapat disimpulkan bahwa model PSO+RUS+Bagging+J48 tidak memberikan pengaruh yang signifikan terhadap model dasar PSO+J48. Hal ini sekaligus menjawab hipotesis pertama (H8).

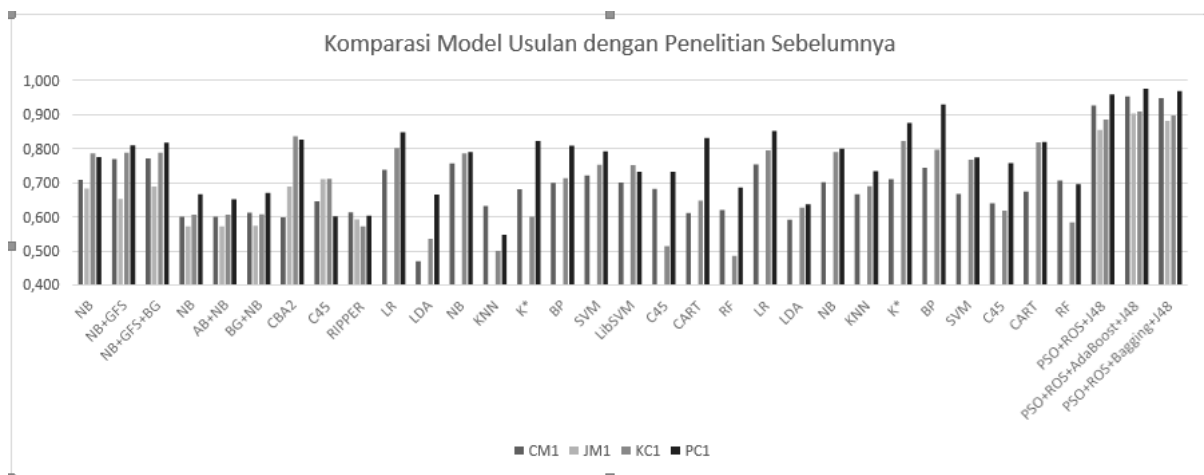
Pengujian t yang ke sembilan diketahui bahwa nilai signifikansi dari model PSO+SMOTE+Bagging+J48 lebih kecil dari nilai p value sebesar 0,05. Dengan demikian dapat disimpulkan bahwa model PSO+SMOTE+Bagging+J48 memberikan pengaruh yang signifikan terhadap model dasar PSO+J48. Hal ini sekaligus menjawab hipotesis pertama (H9).

4.5. Perbandingan Penelitian Terkait

Tabel 4.8
Komparasi model usulan dengan penelitian terkait

Penelitian	METODE	CM1	JM1	KC1	PC1	Rata-rata
Fitriyani, Wahono RS, 2015	NB	0,708	0,683	0,786	0,775	0,738
	NB+GFS	0,769	0,653	0,788	0,810	0,755
	NB+GFS+BG	0,771	0,688	0,788	0,817	0,766
Saifudin A, Wahono RS, 2015	NB	0,600	0,572	0,607	0,666	0,611
	AB+NB	0,600	0,572	0,607	0,652	0,608
	BG+NB	0,612	0,574	0,608	0,671	0,616
Ma B, Dejaeger K, Vanthienen J, Baesens B, 2011	CBA2	0,598	0,688	0,836	0,827	0,737
	C45	0,645	0,710	0,711	0,601	0,667
	RIPPER	0,613	0,593	0,572	0,604	0,596

Wahono RS, Suryana N, 2013	LR	0,738		0,801	0,848	0,796
	LDA	0,469		0,536	0,665	0,557
	NB	0,756		0,786	0,790	0,777
	KNN	0,632		0,500	0,547	0,560
	K*	0,681		0,600	0,822	0,701
	BP	0,700		0,713	0,809	0,741
	SVM	0,721		0,752	0,792	0,755
	LibSVM	0,701		0,751	0,732	0,728
	C45	0,682		0,515	0,732	0,643
	CART	0,611		0,648	0,831	0,697
	RF	0,620		0,485	0,686	0,597
RS, Suryana N, 2014	LR	0,753		0,795	0,852	0,800
	LDA	0,592		0,627	0,637	0,619
	NB	0,702		0,790	0,799	0,764
	KNN	0,666		0,689	0,734	0,696
	K*	0,710		0,822	0,876	0,803
	BP	0,744		0,797	0,929	0,823
	SVM	0,667		0,767	0,774	0,736
	C45	0,640		0,618	0,758	0,672
	CART	0,674		0,818	0,819	0,770
	RF	0,706		0,584	0,696	0,662
Metode Usulan	PSO+ROS+J48	0,926	0,855	0,886	0,958	0,906
	PSO+ROS+AdaBoost+J48	0,954	0,903	0,908	0,976	0,935
	PSO+ROS+Bagging+J48	0,948	0,881	0,897	0,969	0,924
Hasil Nilai AUC Tertinggi		0,954	0,903	0,908	0,976	0,935



Gambar 4.1
Grafik perbandingan metode usulan terhadap penelitian terkait

5. KESIMPULAN

Hasil pengujian menunjukan metode dengan integrasi seleksi fitur PSO, pendekatan level data ROS, pendekatan level algoritma AdaBoost dan algoritma pengklasifikasi J48 menunjukan kinerja pengklasifikasian yang tinggi dengan tingkat rata-rata akurasi 93,5075% dan AUC 0,935. Sehingga metode PSO+ROS+J48+AdaBoost layak digunakan sebagai metode pemrediksian cacat software.

DAFTAR PUSTAKA

- Afza, a J. M. A., Farid, D., & Rahman, C. M. (2011). *A Hybrid Classifier using Boosting , Clustering , and Naïve Bayesian Classifier*. 1(3), 105–109.
- Alfaro, E., Gáamez, M., & García, N. (2013). Adabag: An R package for classification with boosting and bagging. *Journal of Statistical Software*, 54(2). <https://doi.org/10.18637/jss.v054.i02>
- Askari, M. M., & Bardsiri, V. K. (2014). *Software Defect Prediction using a High Performance Neural Network*. 8(12), 177–188.
- Batuwita, R., & Palade, V. (2010). Efficient resampling methods for training support vector machines with imbalanced datasets. *Proceedings of the International Joint Conference on Neural Networks*.
<https://doi.org/10.1109/IJCNN.2010.5596787>
- Bhargava, N., & Sharma, G. (2013). *International Journal of Advanced Research in Decision Tree Analysis on J48 Algorithm for Data Mining*. 3(6), 1114–1119.
- Bienvenido-huertas, D., Nieto-julián, J. E., Moyano, J. J., Macías-bernal, J. M., Castro, J., Nieto-julián, J. E., ... Macías-, J. M. (2019). Implementing Artificial Intelligence in H-BIM Using the J48 Algorithm to Manage Historic Buildings Implementing Artificial Intelligence in H-BIM Using the J48 Algorithm to Manage. *International Journal of Architectural Heritage*, 0(0), 1–13. <https://doi.org/10.1080/15583058.2019.1589602>
- Blaxter, L., Hughes, C., & Tight, M. (2010). *How to Research* (4th ed.). Retrieved from
https://books.google.co.id/books/about/How_to_Research.html?id=d1VvQgAACAAJ&redir_esc=y
- Cao, Y., Miao, Q., Liu, J., & Gao, L. (n.d.). Advance and Prospects of AdaBoost Algorithm. *Acta Automatica Sinica*, 39(6), 745–758.
[https://doi.org/10.1016/S1874-1029\(13\)60052-X](https://doi.org/10.1016/S1874-1029(13)60052-X)

- Daud, A., Mara, U. T., Mara, U. T., Mara, U. T., Mara, U. T., & Mara, U. T. (2018). *Predictive Analytics : The Application of J48 Algorithm on Grocery Data to Predict Obesity*. 1–6.
- Dawson, C. W. (2009). Projects in Computing and Information Systems. In *Information Systems Journal* (Vol. 2). Retrieved from http://www.sentimentaltoday.net/National_Academy_Press/0321263553.Addison.Wesley.Publishing.Company.Projects.in.Computing.and.Information.Systems.A.Students.Guide.Jun.2005.pdf
- Diwandari¹, S., & Setiawan, N. A. (2015). *PERBANDINGAN ALGORITME J48 DAN NBTREE UNTUK KLASIFIKASI*. 2015(Sentika), 205–212.
- Faruk, Ö. (2015). Software defect prediction using cost-sensitive neural network. *Elsevier*, 33, 263–277. <https://doi.org/10.1016/j.asoc.2015.04.045>
- Fitriani, & Wahono, R. S. (2015). Integrasi Bagging dan Greedy Forward Selection pada Prediksi Cacat Software dengan Menggunakan Naïve Bayes. *Journal of Software Engineering*, 1(2), 101–108.
- García, V., Sánchez, J. S., Martín-Félez, R., & Mollineda, R. A. (2012). Surrounding neighborhood-based SMOTE for learning from imbalanced data sets. *Progress in Artificial Intelligence*, 1(4), 347–362. <https://doi.org/10.1007/s13748-012-0027-5>
- Gray, D., Bowes, D., Davey, N., Sun, Y., & Christianson, B. (2011). *The Misuse of the NASA Metrics Data Program Data Sets for Automated Software Defect Prediction*. 96–103.
- Jia, J., Xiao, X., Liu, B., & Jiao, L. (2011). Bagging-based spectral clustering ensemble selection. *Pattern Recognition Letters*, 32(10), 1456–1467. <https://doi.org/10.1016/j.patrec.2011.04.008>
- Jiawei Han, M. K. and J. P. (2012). Data Mining: Concepts and Techniques, Third Edition - Books24x7. *Morgan Kaufmann Publishers*, 745. Retrieved from <http://library.books24x7.com/toc.aspx?bkid=44712>
- Khoshgoftaar, T. M. (2010). *Attribute Selection and Imbalanced Data : Problems*
Program Pascasarjana Magister Ilmu Komputer STMIK Nusa Mandiri

- in Software Defect Prediction*. <https://doi.org/10.1109/ICTAI.2010.27>
- Kim, J. J., Ja, W. K., Lee, H. J., & Han, J. S. (2005). Amygdalar inactivation blocks stress-induced impairments in hippocampal long-term potentiation and spatial memory. *Journal of Neuroscience*, 25(6), 1532–1539. <https://doi.org/10.1523/JNEUROSCI.4623-04.2005>
- Korb, K. B., & Nicholson, A. E. (2005). Bayesian Artificial Intelligence. In *Technometrics* (Vol. 47). <https://doi.org/10.1198/tech.2005.s836>
- Kunang, Y. N., Darma, U. B., Andri, A., & Darma, U. B. (2013). *IMPLEMENTASI TEKNIK DATA MINING UNTUK MEMPREDIKSI TINGKAT*. (June 2016). <https://doi.org/10.13140/RG.2.1.4212.1845>
- 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>
- Lee, J. J., Lee, P. H., Lee, S. W., Yuille, A., & Koch, C. (2011). AdaBoost for text detection in natural scene. *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 429–434. <https://doi.org/10.1109/ICDAR.2011.93>
- López, V., Fernández, A., & Herrera, F. (2014). On the importance of the validation technique for classification with imbalanced datasets: Addressing covariate shift when data is skewed. *Information Sciences*, 257, 1–13. <https://doi.org/10.1016/j.ins.2013.09.038>
- Ma, B., Dejaeger, K., Vanthienen, J., & Baesens, B. (n.d.). *Software Defect Prediction Based on Association Rule Classification (2011)*. Available at SSRN: <https://ssrn.com/abstract=1785381> or <http://dx.doi.org/10.2139/ssrn.1785381>. 0–7.
- Nickabadi, A., Ebadzadeh, M. M., & Safabakhsh, R. (2011). A novel particle swarm optimization algorithm with adaptive inertia weight. *Applied Soft Computing Journal*, 11(4), 3658–3670. <https://doi.org/10.1016/j.asoc.2011.01.037>

- Niranjan, A., Prakash, A., Veena, N., Geetha, M., Shenoy, P. D., & Venugopal, K. R. (2017). EBJRV : An Ensemble of Bagging , J48 and Random Committee by Voting for Efficient Classification of Intrusions. *2017 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)*, (December), 51–54.
- Pandey, P., & Prabhakar, R. (2016). *An Analysis of Machine Learning Techniques (J48 & AdaBoost) -for Classification*. 0–5.
- Philip, D., & Gray, H. (2012). *Software Defect Prediction Using Static Code Metrics : Formulating a Methodology*. (December).
- Putri, S. A., & Wahono, R. S. (2015). Integrasi SMOTE dan Information Gain pada Naive Bayes untuk Prediksi Cacat Software. *Journal of Software Engineering*, 1(2), 86–91.
- Rini, D. P. (2011). *Particle Swarm Optimization : Technique , System and Challenges Particle Swarm Optimization : Technique , System and Challenges*. (May 2014). <https://doi.org/10.5120/ijais-3651>
- Saifudin, A., & Wahono, R. S. (2015). *Penerapan Teknik Ensemble untuk Menangani Ketidakseimbangan Kelas pada Prediksi Cacat Software*. 1(1).
- Sathyaraj, R., & Prabu, S. (2015). *An Approach for Software Fault Prediction to Measure the Quality of Diferent Prediction Methodologies using Software Metrics*. 8(December). <https://doi.org/10.17485/ijst/2015/v8i35/73717>
- Song, Q., Jia, Z., Shepperd, M., Ying, S., & Liu, J. (2011). *A General Software Defect-Proneness Prediction Framework*. 37(3), 356–370.
- Sumadhi, T., & Hemalatha, M. (2013). An Enhanced Approach for Solving Class Imbalance Problem in Automatic Image Annotation. *International Journal of Image, Graphics and Signal Processing*, 5(2), 9–16.
<https://doi.org/10.5815/ijigsp.2013.02.02>
- Vercellis, C. (2009). Business Intelligence: Data Mining and Optimization for Decision Making. In *Business Intelligence: Data Mining and Optimization for Decision Making* (1st ed.). <https://doi.org/10.1002/9780470753866>
- Program Pascasarjana Magister Ilmu Komputer STMIK Nusa Mandiri**

- Wahono, R. S., Dian, U., Semarang, N., & Suryana, N. (2013). *Combining Particle Swarm Optimization based Feature Selection and Bagging Technique for Software Defect Prediction Combining Particle Swarm Optimization based Feature Selection and Bagging Technique for Software Defect Prediction*. (September). <https://doi.org/10.14257/ijseia.2013.7.5.16>
- Wahono, R. S., Suryana, N., & Ahmad, S. (2014). *Metaheuristic Optimization based Feature Selection for Software Defect Prediction*. 9(5), 1324–1333. <https://doi.org/10.4304/jsw.9.5.1324-1333>
- Wang, S., & Yao, X. (2013). Using class imbalance learning for software defect prediction. *IEEE Transactions on Reliability*, 62(2), 434–443. <https://doi.org/10.1109/TR.2013.2259203>
- Witten, I. H., Frank, E., & Hall, M. a. (2011). Data Mining: Practical Machine Learning Tools and Techniques (Google eBook). In *Complementary literature None*. Retrieved from <http://books.google.com/books?id=bDtLM8CODsQC&pgis=1>
- Yap, B. W., Rani, K. A., Aryani, H., Rahman, A., Fong, S., Khairudin, Z., & Abdullah, N. N. (2014). *An Application of Oversampling , Undersampling , Bagging and Boosting in Handling Imbalanced Datasets*. 13–22. <https://doi.org/10.1007/978-981-4585-18-7>
- Zheng, J. (2010). Expert Systems with Applications Cost-sensitive boosting neural networks for software defect prediction. *Expert Systems With Applications*, 37(6), 4537–4543. <https://doi.org/10.1016/j.eswa.2009.12.056>