KOMPARASI INTEGRASI TEKNIK *RESAMPLING* PADA *NAÏVE BAYES* (NB) DAN *LOGISTIC REGRESSION* (LR) BERBASIS *PARTICLE SWARM OPTIMIZATION* UNTUK KLASIFIKASI CACAT PERANGKAT LUNAK

OLEH:

ANDRE HARDONI

09042611822003

PROGRAM STUDI MAGISTER ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SRIWIJAYA
2021

KOMPARASI INTEGRASI TEKNIK *RESAMPLING* PADA *NAÏVE BAYES* (NB) DAN *LOGISTIC REGRESSION* (LR) BERBASIS *PARTICLE SWARM OPTIMIZATION* UNTUK KLASIFIKASI CACAT PERANGKAT LUNAK

Laporan Tesis

Diajukan untuk Melengkapi Salah Satu Syarat
Memperoleh Gelar Magister

OLEH:

ANDRE HARDONI

09042611822003

PROGRAM STUDI MAGISTER ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SRIWIJAYA

2021

LEMBAR PENGESAHAN

KOMPARASI INTEGRASI TEKNIK *RESAMPLING* PADA *NAÏVE BAYES* (NB) DAN *LOGISTIC REGRESSION* (LR) BERBASIS *PARTICLE SWARM OPTIMIZATION* UNTUK KLASIFIKASI CACAT PERANGKAT LUNAK

TESIS

Diajukan Untuk Melengkapi Salah Satu Syarat
Memperoleh Gelar Magister

OLEH:
ANDRE HARDONI
09042611822003

Palembang, Desember 2021

Pembimbing I

Dian Palupi Rini, M.Kom., Ph.D.
NIP 197802232006042002

Pembimbing II

Dr. Ir. Sukemi, M.T.
NIP 196612032006041001

Mengetahui,
Koordinator Program Studi Magister Ilmu Komputer

Dian Palupi Rini, M.Kom., Ph.D.
NIP 197802232006042002

HALAMAN PERSETUJUAN

Pada hari Jumat tanggal 30 Juli 2021 telah dilaksanakan ujian sidang tesis II secara daring oleh Program Studi Magister Ilmu Komputer Fakultas Ilmu Komputer Universitas Sriwijaya.

Nama : Andre Hardoni
NIM : 09042611822003
Judul : Komparasi Integrasi Teknik *Resampling* Pada *Naïve Bayes* (NB) Dan *Logistic Regression* (LR) Berbasis *Particle Swarm Optimization* Untuk Klasifikasi Cacat Perangkat Lunak

1. Pembimbing I

   Dian Palupi Rini, M.Kom., Ph.D ……
   NIP 197802232006042002

2. Pembimbing II

   Dr. Ir. Sukemi, M.T. ……
   NIP 196612032006041001

3. Penguji I

   Dr. Ermatita, M.Kom. ……
   NIP 196709132006042001

4. Penguji II

   Dr. Yusuf Hartono, M.Sc. ……
   NIP 196411161990031002

Mengetahui,
Koordinator Program Studi Magister Ilmu Komputer

Dian Palupi Rini, M.Kom., Ph.D.
NIP 197802232006042002

LEMBAR PERNYATAAN

Yang bertanda tangan di bawah ini :

Nama	: Andre Hardoni
NIM	: 09042611822003
Program Studi	: Magister Ilmu Komputer
Judul Tesis	: Komparasi Integrasi Teknik *Resampling* pada *Naïve Bayes* (NB) dan *Logistic Regression* (LR) Berbasis *Particle Swarm Optimization* untuk Klasifikasi Cacat Perangkat Lunak

Hasil Pengecekan Software iThenticate/Turnitin : **17 %**

Menyatakan bahwa laporan tesis saya merupakan hasil karya sendiri dan bukan hasil penjiplakan/plagiat. Apabila ditemukan unsur penjiplakan/plagiat dalam laporan tesis ini, maka saya bersedia menerima sanksi akademik dari Universitas Sriwijaya sesuai dengan ketentuan yang berlaku.

Demikian, pernyataan ini saya buat dengan sebenarnya dan tidak ada paksaan oleh siapapun.

Palembang, Desember 2021

Andre Hardoni

NIM. 09042611822003

KATA PENGANTAR

بسم الله الرحمن الرحيم

Assalamu'alaikum Wr. Wb.

Alhamdulillahirobbilalamin, penulis ucapkan kepada Allah SWT atas berkat, rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Tesis dengan judul "Komparasi Integrasi Teknik *Resampling* pada *Naïve Bayes* (NB) dan *Logistic Regression* (LR) Berbasis *Particle Swarm Optimization* Untuk Klasifikasi Cacat Perangkat Lunak". Penyusunan Tesis ini ditujukan untuk melengkapi salah satu syarat memperoleh gelar Master pada Program Studi Magister Ilmu Komputer Universitas Sriwijaya.

Dalam penulisan Tesis ini, tentunya penulis ingin dan berharap agar Tesis ini dapat bermanfaat bagi banyak orang sekalipun masih jauh dari kata sempurna. Hal ini dikarenakan keterbatasnya pengetahuan yang dimiliki. Oleh karena itu dalam rangka melengkapi kesempurnaan dari penulisan Tesis ini diharapkan adanya saran dan kritik yang diberikan yang bersifat membangun.

Pada kesempatan yang baik ini, tak lupa pula penulis menghaturkan terima kasih kepada semua pihak yang telah memberikan bimbingan, pengarahan, nasehat dan pemikiran dalam penulisan Tesis ini, terutama kepada :

1. Orang tua, Bapak Joko dan Mamak Dina yang selalu mendukung dan memberikan doa yang tiada putus untuk penulis.
2. Saudara penulis kakak dan adik yang selalu memberikan semangat.
3. Ibu Dian Palupi Rini, M.Kom., Ph.D selaku pembimbing I, pembimbing akademik dan juga sekaligus Koordinator Program Studi Magister Ilmu Komputer Fakultas Ilmu Komputer Universitas Sriwijaya yang selalu memberikan nasihat dan dukungan selama pembuatan tesis dan publikasi-publikasi untuk syarat menyelesaikan program magister.

4. Bapak Dr. Ir. Sukemi, M.T. selaku pembimbing II yang juga selalu memberikan dukungan, nasihat dan bantuan dalam menyelesaikan program magister ini.
5. Ibu Dr. Ermatita, M.Kom. dan Bapak Dr.Yusuf Hartono, M.Sc. selaku penguji I dan penguji II baik pada sidang proposal maupun sidang Tesis selalu baik dan tidak merumitkan penulis ketika memberikan revisi untuk penulisan lebih baik.
6. Teman-teman satu angkatan 2018 genap dan juga teman mahasiswa program Magister Ilmu Komputer.
7. Bapak/Ibu dosen Program Studi Magister Ilmu Komputer Fakultas Ilmu Komputer Universitas Sriwijaya.

Akhir kata, dengan segala kerendahan hati dan keterbatasan, penulis berharap Tesis ini nantinya dapat bermanfaat bagi semua pihak, baik bagi penulis sendiri maupun bagi Jurusan Ilmu Komputer Fakultas Ilmu Komputer Universitas Sriwijaya.

Wassalamu'alaikum Wr.Wb.

Palembang, Desember 2021

Penulis,
Andre Hardoni

COMPARASION OF RESAMPLING TECHNIQUE INTEGRATION ON NAIVE BAYES (NB) AND LOGISTIC REGRESSION (LR) BASED ON PARTICLE SWARM OPTIMIZATION FOR SOFTWARE DEFECTS CLASSIFICATION

Andre Hardoni

ABSTRACT

Software quality is generally used as a condition that must exist in a software developer or builder company to make the company more competitive. Software quality in general can be seen from the number of defects contained in the resulting software. Improving software quality can be done in various ways, but the best approach is to prevent defects, one of which is done by predicting the possibility of defects through the classification method. The NASA MDP dataset is public and is widely used by researchers for defect classification cases, but the lack of this dataset is class imbalance and noise attribute. This experiment introduces a combination method between SMOTE (Synthetic Minority Over-sampling Technique) and PSO (Particle Swarm Optimization) to deal with imbalance and noise attribute problems which are integrated into the basic method like Naive Bayes classification method and logistic regression. The results of experiments that have been carried out on 9 (nine) NASA MDP datasets show that overall SMOTE + PSO integration can improve classification performance with the highest average AUC (Area Under Curve) value in the logistic regression method, which is 0.853 and in the nave Bayes method, with 0.831 and its also better than without combining the two.

Keywords : Software Defect, Classification, Naïve Bayes, Logistic Regression, SMOTE, PSO

KOMPARASI INTEGRASI TEKNIK *RESAMPLING* PADA *NAÏVE BAYES* (NB) DAN *LOGISTIC REGRESSION* (LR) BERBASIS *PARTICLE SWARM OPTIMIZATION* UNTUK KLASIFIKASI CACAT PERANGKAT LUNAK

ABSTRAK

Kualitas perangkat lunak pada umumnya dijadikan suatu syarat yang harus ada pada suatu perusahaan pengembang atau pembangun perangkat lunak untuk menjadikan perusahaan lebih kompetitif. Kualitas perangkat lunak pada umumnya dapat dilihat dari jumlah cacat yang terdapat pada perangkat lunak yang dihasilkan. Meningkatan kualitas perangkat lunak bisa dilakukan melalui bermacam-macam cara, namun untuk pendekatan terbaik adalah dengan melakukan pencegahan terjadinya cacat yang salah satunya dilakukan dengan memprediksi kemungkinan terjadinya cacat melalui metode klasifikasi. Dataset NASA MDP bersifat publik dan banyak digunakan peneliti untuk kasus klasifikasi cacat, namun kekurangan dataset tersebut yaitu ketidakseimbangan kelas dan *noise attribute*. Penelitian ini memperkenalkan metode penggabungan antara SMOTE (*Synthetic Minority Over-sampling Technique*) dan PSO (*Particle Swarm Optimization*) untuk menangani masalah ketidakseimbangan dan *noise attribute* yang diintegrasikan pada metode klasifikasi dasar *naïve bayes* dan *logistic regression*. Hasil percobaan yang telah dilakukan pada 9 (sembilan) dataset NASA MDP diperoleh hasil bahwa secara keseluruhan integrasi SMOTE +PSO dapat meningkatkan kinerja pengklasifikasian dengan nilai AUC (*Area Under Curve*) tertinggi rata-rata pada metode *logistic regression* yaitu 0,853 dan pada metode *naïve bayes* yaitu 0,831 juga lebih baik dibanding dengan tanpa meng-kombinasikan keduanya.

Kata Kunci : Cacat Perangkat Lunak, Klasifikasi, *Naïve Bayes*, *Logistic Regression*, SMOTE, PSO.

DAFTAR ISI

DAFTAR GAMBAR

Halaman

DAFTAR TABEL

Halaman

LAMPIRAN

BAB I. PENDAHULUAN

Pendahuluan BAB ini menjelaskan tentang latar belakang penelitian yang berjudul: “Komparasi Integrasi Teknik Resampling Pada *Naïve Bayes* (NB) dan *Logistic Regression* (LR) Berbasis *Particle Swarm Optimization* (PSO) Untuk Klasifikasi Cacat Perangkat Lunak”. Permasalahan klasifikasi cacat perangkat lunak dipilih menjadi topik dikarenakan perangkat lunak jika diperbaiki yang cacat setelah selesai ke tangan *user* membutuhkan biaya jauh lebih mahal dan menyita waktu dari pada dilakukakan pada saat pengembangan.

1.1 Latar Belakang

Kualitas perangkat lunak apabila dijadikan suatu syarat yang harus ada pada suatu perusahaan pengembang atau pembangun perangkat lunak, maka dapat menjadikan suatu perusahaan lebih kompetitif dalam setiap pembuatan ataupun pembangunan suatu perangkat lunak, namun dengan melihat tingkat standar kualitas yang tinggi diperlukan pengembangan dan pemantauan secara *continue* (Bergmane, Grabis dan Žeiris, 2018). Kualitas perangkat lunak pada umumnya dapat dilihat dari jumlah cacat yang terdapat pada perangkat lunak yang dihasilkan (Turhan dan Bener, 2007). Cacat adalah salah satu yang berperan dalam penyumbang limbah teknologi informasi dan hal ini menjadikan pengerjaan ulang (*rework*) pada suatu projek sehingga menbuang banyak waktu dan biaya (Sedano, Ralph dan Peraire, 2017). Terjadinya cacat pada suatu produk dapat ditimbulkan oleh manusia (*human error*) seperti kesalahan dalam pengetikan suatu *coding* program, jika ditemukan cacat pada saat eksekusi kode, sistem mungkin tidak bisa berfungsi seperti yang diiniginkan sehingga menyebabkan kegagalan (Shenvi, 2009), namun jika cacat ditemukan setelah akhir dari pekerjaan maka secara tidak langsung dapat menyebabkan proyek selesai melewati waktu yang ditentukan (Lehtinen *dkk.*, 2014). Perbaikan cacat perangkat lunak mahal dan biaya akan meningkat pada setiap tahap pengerjaannya dan hal itu juga dapat mempengaruhi siklus pengembangan waktu perangkat lunak (Gupta, Ganeshan dan Singhal, 2015). Pada umumnya, perhitungan biaya untuk masa depan dalam mengkoreksi

cacat yang tidak dapat dideteksi setelah produk tekah selesai dapat menghabiskan sebagian dari dana pemeliharaan yang dianggarkan (In *dkk.*, 2006), sehingga pengkoreksian cacat harus segera dilakukan sebelum produk diserahkan pada *user*. Cacat perangkat lunak pada umumnya diketahui sebagai *bugs*, maka untuk mencari cacat perangkat lunak dan memperbaikinya biasanya dilakukan dengan *debugging* (Wambugu dan Njeru, 2017), namun hal ini dapat menyebabkan kebutuhan sumber daya yang tidak sedikit dan menjadi tidak efisien karena mengasumsikan jika kode program yang dibuat oleh pembuat *coding* tidak bisa dipercaya.

Meningkatan kualitas perangkat lunak bisa dilakukan melalui bermacam-macam cara, namun untuk pendekatan terbaik adalah dengan melakukan pencegahan terjadinya cacat, karena manfaatnya dapat diterapkan kembali untuk yang akan datang (McDonald, Musson dan Smith, 2008). Untuk pencegahan cacat dapat dilakukan, maka kemungkinan terjadinya cacat harus dapat diprediksi.

Prediksi cacat suatu perangkat lunak merupakan masalah klasifikasi biner yang mana modul tertentu harus dapat diklasifikasikan sebagai cacat atau tidak cacat (Munir *dkk.*, 2017). Metode klasifikasi adalah pendekatan yang paling sering digunakan untuk memprediksi cacat suatu perangkat lunak dan metode klasifikasi dapat digunakan untuk menentukan kelas sebagai cacat atau tidak cacat (Iqbal, Aftab, Ali, *dkk.*, 2019). untuk dapat melakukan klasifikasi maka perlu adanya data yang didapat dari riwayat pengembangan yang sudah dilakukan sebelumnnya.

Software metrics adalah kumpulan dari data yang bisa dipakai untuk mendeteksi suatu modul perangkat lunak apakah mempunyai cacat atau tidak (Chiş, 2008). *Data* mining merupakann salah satu metode yang efektif dalam mengidentifikasi suatu modul perangkat lunak dari potensi adanya rawan cacat yang diimplikasikan pada *software metrics* yang didapatkan dari pengembangan (Khoshgoftaar, Gao dan Seliya, 2010) dan *software metrics* yang telah didapatkan selama pengembangan disimpan menjadi dataset.

Dataset NASA (*National Aeronautics and Space Administration*) yang sudah bersifat publik berbentuk data metrik perangkat lunak yang terkenal karena sudah banyak penelitian telah menggunakan dataset NASA (Aries Saifudin, 2014). Dataset NASA sendiri dapat ditemukan dari dua sumber salah satunya yaitu bersumber dari NASA MDP (*Metrics Data Program*).

Banyak penelitian yang telah menggunakan dataset NASA MDP pada penelitiannya dan metode klasifikasi menjadi pusat topik penelitiannya dengan algoritma klasifikasi, seperti penggunaan algoritma *Logistic Regression* (LR) dan *Naïve Bayes* (NB). *Logistic Regression* pada metode klasifikasi terbukti membuat hasil dari klasifikasi yang *powerful* dalam hal penanganan masalah klasifikasi kelas yang banyak (Canu dan Smola, 2006), sedangkan untuk *naïve bayes* merupakan metode klasifikasi yang dapat dikatakan efektif, karena dapat memperlakukan klasifikasi cacat sebagai klasifikasi biner, melatih dan membuat prediktor dengan menganalisis data historis pada modul perangkat lunak. Dari prediktor ini dapat membuat keputusan modul baru terdapat cacat atau tidak (Wang dan Li, 2010), namun perlu pengembangan suatu prosedur penelitian yang lebih bisa untuk diandalkan sebelum terdapat keyakinan dalam menyimpulkan komparasi penelitian dari metode klasifikasi cacat perangkat lunak (Myrtveit, Stensrud dan Shepperd, 2005).

Dataset yang tidak seimbang (*imbalance*) dan *noise attribute* merupakan beberapa masalah pada klasifikasi cacat perangkat lunak yang ditemukan (Wahono dan Suryana, 2013). Dataset NASA MDP yang sudah banyak digunakan para peneliti dalam penelitian ditemukan jika jumlah data yang tidak cacat (*not defect*) lebih banyak dari pada jumlah data yang cacat (*defect*) dan menjadikan data tidak seimbang, sehingga dapat membuat hasil klasifikasi cenderung dapat menghasilkan kelas yang lebih banyak yang dalam hal ini kelas tidak cacat (*not defect*) (Khoshgoftaar *dkk.*, 2014), selain itu pada dataset MDP dataset yang memiliki ukuran besar dan multi kelas dapat memiliki *noise* atau terdapat *error*, sehingga hal ini menyebabkan berkurangnya kinerja pada klasifikasi (Han, Kamber dan Pei, 2012). Untuk itu diperlukannya melakukan modifikasi metode klasifikasi dengan menambahkan metode lain atau mengombinasikan metdoe lain untuk dapat mengatasi masalah tersebut sehingga membuat kinerja klasifikasi akan semkin lebih baik.

Penelitian dengan memanfaatkan teknik pendekatan level data *resample* khususnya SMOTE telah dilakukan, namun hal ini hanya terbatas pada penaganan dataset yang tidak seimbang dan belum pada penanganan masalah *noise attribute* yang terdapat pada dataset NASA MDP yang juga dapat mempengaruhi hasil

kinerja klasifikasi, sehingga perlu untuk ditambahkan teknik lain untuk menangani masalah tersebut. PSO adalah metode komputasi yang memecahkan masalah dengan mencoba mencari solusi terbaik secara berulang-ulang dengan menggunakan pengukuran kualitas tertentu. PSO mengoptimasikan setiap pemecahan masalah dengan membuat kandidat populasi yang disebut partikel dan setiap perpindahan partikel akan dicari menggunakan rumus matematika sederhana melalui *update* posisi dan kecepatan partikel (Sethuramalingam dan Nagaraj, 2016). Penggunaan PSO sebagai seleksi fitur/atribut dapat membuat model klasifikasi menjadi lebih baik dan menghasilkan kinerja yang meningkat pada model klasifikasi (Brezočnik dan Podgorelec, 2019).

SMOTE sebagai *resampling* ketika dikombinasikan dengan PSO dapat membuat penyeleksian fitur atau atribut terpilih dapat memilih parameter terbaik dan mengindari pemilihan parameter secara acak dan tidak teratur, serta memiliki tujuan tidak hanya menangani masalah ketidakseimbangan kelas pada dataset tetapi juga menangani masalah *noise atribut* pada multi kelas sehingga menghasikanl kinerja yang lebih baik (Li, Fong dan Zhuang, 2016). Dari alasan tersebut maka pada penelitian ini yang akan dilakukan yaitu dangan mengkombinasikan antara teknik *resample* SMOTE dan seleksi atribut PSO pada *naïve bayes* dan *logistic regression* untuk memecahkan masalah ketidak-seimbangan kelas (*class imbalance*) dan *noise attribute* dalam klasifikasi cacat perangkat lunak pada 9 dataset NASA yang memiliki jumlah data dan atribut yang berbeda-beda sehingga diharapkan mendapatkan hasil klasifikasi yang terbaik.

1.2 Perumusan Masalah

Dari latar belakang yang telah disampaikan di atas terdapat beberapa masalah yang dapat mempengaruhi hasil dari kinerja model klasifikasi pada dataset NASA seperti ketidakseimbangan kelas (*class imbalance*) dan *noise attribute*, sehingga pada penelitian ini dapat di rumuskan masalah yang tuangkan dalam beberapa pertanyaan sebagai berikut :

1. Bagaimana mengembangkan kerangka kerja penyeimbangan data dan pemilihan fitur pada cacat perangkat lunak dengan teknik resampling SMOTE dan PSO pada naïve bayes dan logistic regression ?

2. Apakah dari *balancing* dan fitur-fitur terpilih dapat meningkatkan hasil kinerja model klasifikasi ?
3. Dari semua model, baik sebelum di kombinasi maupun setelah dikombinasi, manakah yang memiliki hasil kinerja terbaik dan bagaimana pengaruh jumlah dataset yang memiliki jumlah data berbeda-beda terhadap hasil klasifikasi ?

1.3 Batasan Masalah

Dari penelitian yang akan dilakukan agar tidak terlalu melebar pembahasanya meka dibuat beberapa batasan masalah pada tesis ini, antara lain yaitu:

1. Data yang digunakan merupakan dataset dari NASA MDP *repository* dengan menggunakan sembilan (9) dataset.
2. Sistem yang dibangun merupakan bentuk simulasi untuk mengklasifikasi cacat perangkat lunak.
3. Pengolahan data awal untuk *balancing* data menggunakan teknik *resampling* SMOTE.
4. Teknik klasifikasi *Naïve Bayes* menggunakan fungsi *densitas gauss*.
5. Hasil yang akan digunakan sebagai perbandingan yaitu nilai AUC (*Area Under the ROC (Receiver Operating Characteristic) Curve*) .

1.4 Tujuan

Pada penelitian (*Research Objective*) ini memiliki tujuan yaitu membangun sistem dalam menerapkan teknik resampling sebagai *balancing* data serta PSO sebagai seleksi fitur untuk yang diharapkan dapat mengurangi pengaruh ketidakseimbangan kelas dan *noise attruibute* dalam dataset, agar kinerja pengklasifikasi (*Naïve Bayes* dan *logistic regression*) pada prediksi cacat perangkat lunak dapat meningkat dalam memprediksi kerawanan cacat. Berikut ini adalah penjabaran dari tujuan penelitian :

1. Mengembangkan kerangka kerja *balancing* data dengan SMOTE dan pemilihan fitur dengan PSO pada klasifikasi cacat perangkat lunak.
2. Menentukan bagaimana *balancing* data dan seleksi fitur dibangun sehingga dapat meningkatkan hasil dari kinerja metode klasifikasi.

3. Menganalisis pengaruh *balancing* menggunakan SMOTE dan seleksi fitur menggunakan PSO dan mengukur hasil kinerja masing masing model kemudian mengkomparasi hasil model tersebut dan melihat pengaruh dari jumlah data pada dataset terhadap hasil penelitian.

1.5 Manfaat

Dalam penelitian tesis ini penulis berharap dapat memiliki manfaat antara lain :

1. Kerangka kerja penelitian dapat menjadi acuan dalam pengklasifikasian cacat perangkat lunak untuk penelitian selanjutnya yang berkaitan dengan metode klasifikasi penelitian ini.
2. Untuk memberi gambaran mengenai bagaimana sistem dibangun.
3. Memberikan pengetahuan bagaimana pengaruh penerapan SMOTE dan PSO terhadap hasil kinerja klasifikasi.

1.6 Metodologi Penulisan

Agar tesis ini mudah dipelajari dan dipahami, maka pada penjelasan penelitian ini akan dibagi dengan menjadikannya lima bab dan dengan setiap bab dibagi lagi menjadi beberapa subbab sesuai topik pembahasan. Sistematika penulisan pada penulisan tesisini adalah:

1. **BAB I** **Pendahuluan**

   Pendahuluan, membahas mengenai latar belakang penelitian ini akan dilakukan, rumusan masalah penelitian sebagai acuan penelitian, batasan masalah dalam penelitian, tujuan dan manfaat dalam penelitian, serta metodologi penelitian.

2. **BAB II** **Tinjauan Pustaka**

   Berisi tentang tinjauan studi, yaitu membahas tentang penelitian sebelumnya yang mendasari penelitian ini. Dan tinjauan pustaka, yaitu membahas tentang landasan secara teoritis yang diambil dari jurnal, tesis, maupun buku.

3. **BAB III** **Metodologi Penelitian**

Pada motodologi penelitian ini berisi mengenai tahapan-tahapan yang digunakan dalam penelitian yang lebih terperinci yang digunakan sebagai landasan dalam pembuatan kerangka berfikir dan kerangka kerja untuk menyelesaikan penelitian.

4. **BAB IV** **Hasil dan Analisis**

Hasil dan analisis ini berisi mengenai hasil dari penelitian dan uraian serta analisis terhadap hasil penelitian yang disajikan lebih terperinci baik dengan bentuk uraian-uraian singkat hasil penelitian maupun dengan tabel dan gambar.

5. **BAB V** **Kesimpulan dan Saran**

Kesimpulan dan saran berisi mengenai simpulan dari hasil penelitian mengenai bagaimana hasil dari penelitian dan berdasarkan hasil dari kesimpulan pada bab ini didapatkan saran untuk penelitian selanjutnya mengenai topik dan metode yang sama.

DAFTAR PUSTAKA

Akbar, M. F., Kurniawan, I. dan Fauzi, A. (2019) "Mengatasi Imbalanced Class Pada Software Defect Prediction Menggunakan Two-Step Clustering-Based Undersampling dan Bagging Tehcnique," *Jurnal Informatika*, 6(1), hal. 107–113. doi: 10.31311/ji.v6i1.5448.

Alberto Fernandez *dkk*. (2018) "SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary," *Journal of Artificial Intelligence Research*.

Alfian, A. N. (2016) *IMPLEMENTASI REGRESI LOGISTIK UNTUK MENDETEKSI IKAN BERFORMALIN BERBASIS ANDROID BERDASARKAN CITRA DAN SIFAT FISIK IKAN*, *Skripsi*. MAULANA MALIK IBRAHIM STATE ISLAMIC UNIVERSITY.

Aries Saifudin (2014) *Pendekatan Level Data dan Algoritma untuk Penanganan Ketidakseimbangan Kelas pada Prediks Cacat Software Berbasis Naïve Bayes*. doi: 10.13140/RG.2.1.3363.9445.

Arora, I., Tetarwal, V. dan Saha, A. (2015) "Open issues in software defect prediction," in *Procedia Computer Science*. doi: 10.1016/j.procs.2015.02.161.

Bennin, K. E. *dkk*. (2017) "The Significant Effects of Data Sampling Approaches on Software Defect Prioritization and Classification," *International Symposium on Empirical Software Engineering and Measurement*, 2017-Novem, hal. 364–373. doi: 10.1109/ESEM.2017.50.

Bergmane, L., Grabis, J. dan Žeiris, E. (2018) "A Case Study: Software Defect Root Causes," *Information Technology and Management Science*. doi: 10.1515/itms-2017-0009.

Bowes, D. *dkk*. (2011) "A Systematic Review of Fault Prediction Performance in Software Engineering," *IEEE Transactions on Software Engineering*.

Brezočnik, L. dan Podgorelec, V. (2019) "Applying Weighted Particle Swarm Optimization to Imbalanced Data in Software Defect Prediction," in *Lecture Notes in Networks and Systems*. doi: 10.1007/978-3-319-90893-9_35.

Canu, S. dan Smola, A. (2006) "Kernel methods and the exponential family," *Neurocomputing*. doi: 10.1016/j.neucom.2005.12.009.

Chen, R. M. dan Shih, H. F. (2013) “Solving university course timetabling problems using constriction particle swarm optimization with local search,” *Algorithms*, 6(2), hal. 227–244. doi: 10.3390/a6020227.

Chiş, M. (2008) “Evolutionary Decision Trees and Software Metrics for Module Defects Identification,” *World Academy of Science, Engineering and Technology*, 2(2), hal. 25–29.

Gorunescu, F. (2011) “Data mining: Concepts, models and techniques,” *Intelligent Systems Reference Library*. doi: 10.1007/978-3-642-19721-5.

Gray, D. *dkk.* (2011) “The misuse of the NASA Metrics Data Program data sets for automated software defect prediction,” in *IET Seminar Digest*. doi: 10.1049/ic.2011.0012.

Gupta, V., Ganeshan, N. dan Singhal, T. K. (2015) “Determining the root causes of various software bugs throughsoftware metrics,” in *2015 International Conference on Computing for Sustainable Global Development, INDIACom 2015*.

H Bahtiar, A Sudianto, R. A. (2016) “PREDIKSI CACAT SOFTWARE MENGGUNAKAN NEURAL NETWORK BERBASIS PSO,” *Jurnal Informatika Hamzanwadi*, I(1), hal. 1–26.

Hall, T. *dkk.* (2012) “A systematic literature review on fault prediction performance in software engineering,” *IEEE Transactions on Software Engineering*. doi: 10.1109/TSE.2011.103.

Han, J., Kamber, M. dan Pei, J. (2012) *Data Mining: Concepts and TechniquesHan, J., Kamber, M., & Pei, J. (2012). Data Mining: Concepts and Techniques. San Francisco, CA, itd: Morgan Kaufmann. https://doi.org/10.1016/B978-0-12-381479-1.00001-0, San Francisco, CA, itd: Morgan Kaufmann*. doi: 10.1016/B978-0-12-381479-1.00001-0.

Huda, S. *dkk.* (2018) “An Ensemble Oversampling Model for Class Imbalance Problem in Software Defect Prediction,” *IEEE Access*, 6(March), hal. 24184–24195. doi: 10.1109/ACCESS.2018.2817572.

In, H. P. *dkk.* (2006) “A quality-based cost estimation model for the product line life cycle,” *Communications of the ACM*. doi: 10.1145/1183236.1183273.

Iqbal, A., Aftab, S., Ullah, I., *dkk.* (2019) “A Feature Selection based Ensemble

Classification Framework for Software Defect Prediction," *International Journal of Modern Education and Computer Science*, 11(9), hal. 54–64. doi: 10.5815/ijmecs.2019.09.06.

Iqbal, A., Aftab, S., Ali, U., *dkk.* (2019) "Performance analysis of machine learning techniques on software defect prediction using NASA datasets," *International Journal of Advanced Computer Science and Applications*, 10(5), hal. 300–308. doi: 10.14569/ijacsa.2019.0100538.

Khoshgoftaar, T. M. *dkk.* (2014) "A comparative study of iterative and non-iterative feature selection techniques for software defect prediction," *Information Systems Frontiers*, 16(5), hal. 801–822. doi: 10.1007/s10796-013-9430-0.

Khoshgoftaar, T. M., Gao, K. dan Seliya, N. (2010) "Attribute selection and imbalanced data: Problems in software defect prediction," *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, 1, hal. 137–144. doi: 10.1109/ICTAI.2010.27.

Korb, K. B. dan Nicholson, A. E. (2010) *Bayesian artificial intelligence, second edition, Bayesian Artificial Intelligence, Second Edition*. doi: 10.1201/b10391.

Kusmarna, I., Wardhani, L. K. dan Safrizal, M. (2015) "Aplikasi Penjadwalan Mata Kuliah Menggunakan Algoritma Particle Swarm Optimization (Pso)," *Jurnal Teknik Informatika*, 8(2), hal. 1–8. doi: 10.15408/jti.v8i2.2441.

Laradji, I. H., Alshayeb, M. dan Ghouti, L. (2015) "Software defect prediction using ensemble learning on selected features," *Information and Software Technology*. doi: 10.1016/j.infsof.2014.07.005.

Lehtinen, T. O. A. *dkk.* (2014) "Perceived causes of software project failures - An analysis of their relationships," *Information and Software Technology*. doi: 10.1016/j.infsof.2014.01.015.

Li, J., Fong, S. dan Zhuang, Y. (2016) "Optimizing SMOTE by Metaheuristics with Neural Network and Decision Tree," in *Proceedings - 2015 3rd International Symposium on Computational and Business Intelligence, ISCBI 2015*. doi: 10.1109/ISCBI.2015.12.

Liu, X. Y. dan Zhou, Z. H. (2013) "Ensemble methods for class imbalance learning," *Imbalanced Learning: Foundations, Algorithms, and Applications*,

hal. 61–82. doi: 10.1002/9781118646106.ch4.

López, V., Fernández, A. dan Herrera, F. (2014) "On the importance of the validation technique for classification with imbalanced datasets: Addressing covariate shift when data is skewed," *Information Sciences*. doi: 10.1016/j.ins.2013.09.038.

Luque, A. *dkk.* (2019) "The impact of class imbalance in classification performance metrics based on the binary confusion matrix," *Pattern Recognition*. doi: 10.1016/j.patcog.2019.02.023.

M Reza Faisal (2017) *Menghitung kinerja algoritma klasifikasi: Pilih ROC Curve atau Precision-Recall Curve?*, *ulm.ac.id*. Tersedia pada: https://staf.ulm.ac.id/rezafaisal/2017/01/12/menghitung-kinerja-algoritma-klasifikasi-pilih-roc-curve-atau-precision-recall-curve/.

McDonald, M., Musson, R. dan Smith, R. (2008) *The Practical Guide to Defect Prevention, Control*.

Minabari, F., Titaley, J. dan Nainggolan, N. (2019) "Pengaruh Pelayanan Di Fakultas Matematika dan Ilmu Pengetahuan Alam Terhadap Kepuasan Mahasiswa Fmipa Unsrat Menggunakan Logistik Ordinal," *Jurnal Matematika Dan Aplikasi*, 8(2), hal. 153–160.

Muhamad, H. *dkk.* (2017) "Optimasi Naïve Bayes Classifier Dengan Menggunakan Particle Swarm Optimization Pada Data Iris," *Jurnal Teknologi Informasi dan Ilmu Komputer*, 4(3), hal. 180. doi: 10.25126/jtiik.201743251.

Munir, A. *dkk.* (2017) "Hybrid Tools and Techniques for Sentiment Analysis: A Review," *International Journal of Multidisciplinary Sciences and Engineering*, 8(4), hal. 28–33.

Myrtveit, I., Stensrud, E. dan Shepperd, M. (2005) "Reliability and validity in comparative studies software prediction models," *IEEE Transactions on Software Engineering*. doi: 10.1109/TSE.2005.58.

Patil, T. R. (2013) "Performance Analysis of Naive Bayes and J48 Classification Algorithm for Data Classification," *International Journal Of Computer Science And Applications, ISSN: 0974-1011*.

Pourbahrami, S. (2018) "Improving PSO Global Method for Feature Selection According to Iterations Global Search and Chaotic Theory," hal. 1–17. Tersedia

pada: http://arxiv.org/abs/1811.08701.

Putri, S. A. (2019) “Prediksi Cacat Software Dengan Teknik Sampel Dan Seleksi Fitur Pada Bayesian Network,” *Jurnal Kajian Ilmiah*, 19(1), hal. 17. doi: 10.31599/jki.v19i1.314.

Ridwan, M., Suyono, H. dan Sarosa, M. (2013) “Penerapan Data Mining Untuk Evaluasi Kinerja Akademik Mahasiswa Menggunakan Algoritma Naive Bayes Classifier,” *Eeccis*, 7(1), hal. 59–64. doi: 10.1038/hdy.2009.180.

Rosandy, T. (2016) “PERBANDINGAN METODE NAIVE BAYES CLASSIFIER DENGAN METODE DECISION TREE (C4.5) UNTUK MENGANALISA KELANCARAN PEMBIAYAAN (Study Kasus : KSPPS / BMT AL-FADHILA,” *Jurnal Teknologi Informasi Magister Darmajaya*, 2(01), hal. 52–62.

Saleh, A. (2015) “Klasifikasi Metode Naive Bayes Dalam Data Mining Untuk Menentukan Konsentrasi Siswa,” *KeTIK*, hal. 200–208.

Salim, A. (2019) “Optimalisasi Regresi Logistik Pada Proses Klasifikasi Menggunakan Algoritma Genetika,” 6(2), hal. 50–55. doi: 10.25047/jtit.v6i2.109.

Sedano, T., Ralph, P. dan Peraire, C. (2017) “Software Development Waste,” in *Proceedings - 2017 IEEE/ACM 39th International Conference on Software Engineering, ICSE 2017*. doi: 10.1109/ICSE.2017.20.

Sethuramalingam, T. K. dan Nagaraj, B. (2016) “A Proposed System of Ship Trajectory Control Using Particle Swarm Optimization,” in *Procedia Computer Science*. doi: 10.1016/j.procs.2016.05.164.

Shenvi, A. A. (2009) “Defect prevention with orthogonal defect classification,” in *Proceedings of the 2nd India Software Engineering Conference, ISEC 2009*. doi: 10.1145/1506216.1506232.

Shepperd, M. *dkk*. (2013) “Data quality: Some comments on the NASA software defect datasets,” *IEEE Transactions on Software Engineering*. doi: 10.1109/TSE.2013.11.

Suryadi, A. (2019) “Integration of Feature Selection with Data Level Approach for Software Defect Prediction,” *SinkrOn*. doi: 10.33395/sinkron.v4i1.10137.

Turhan, B. dan Bener, A. (2007) “Software defect prediction: Heuristics for

weighted naïve bayes," in *ICSOFT 2007 - 2nd International Conference on Software and Data Technologies, Proceedings*.

Ulhaq, Z. dan Adji, T. B. (2017) "Technique (SMOTE) dengan Correlated Naïve Bayes pada Klasifikasi Siswa Berkesulitan Belajar," *Citee*, hal. 201–205.

Wahono, R. S. dan Suryana, N. (2013) "Combining particle swarm optimization based feature selection and bagging technique for software defect prediction," *International Journal of Software Engineering and its Applications*, 7(5), hal. 153–166. doi: 10.14257/ijseia.2013.7.5.16.

Wambugu, G. M. dan Njeru, K. M. (2017) "Automatic Debugging Approaches : A literature Review.," *International Journal of Applied computer Science (IJACS)*, 1(I), hal. 1–5.

Wang, T. dan Li, W. H. (2010) "Naïve bayes software defect prediction model," in *2010 International Conference on Computational Intelligence and Software Engineering, CiSE 2010*. doi: 10.1109/CISE.2010.5677057.

Webb, A. R. dan Copsey, K. D. (2011) *Statistical Pattern Recognition: Third Edition, Statistical Pattern Recognition: Third Edition*. doi: 10.1002/9781119952954.

Witten, I. H., Frank, E. dan Hall, M. a. (2011) *Data Mining: Practical Machine Learning Tools and Techniques, Third Edition, Annals of Physics*. doi: 10.1002/1521-3773(20010316)40:6<9823::AID-ANIE9823>3.3.CO;2-C.

Yusup, M., Panjaitan, E. S. dan Yunis, R. (2020) "Analisis Kinerja dalam Mendeteksi Student Loses Berdasarkan Nilai Gain dengan Splite Feature Reduction Model pada Algoritma C4,5," *CESS (Journal of Computer Engineering, System and Science)*, 5(2), hal. 267. doi: 10.24114/cess.v5i2.17667.

Zhang, H. dan Wang, Z. (2011) "A normal distribution-based over-sampling approach to imbalanced data classification," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. doi: 10.1007/978-3-642-25853-4_7.

Zhou, L. *dkk*. (2018) "Imbalanced Data Processing Model for Software Defect Prediction," *Wireless Personal Communications*. Springer US, 102(2), hal. 937–950. doi: 10.1007/s11277-017-5117-z.