BAB II
STUDI LITERATUR

A. KUALITAS

Menurut Kotler (2000) kualitas didefinisikan secara berbeda yaitu keseluruhan ciri serta sifat dari suatu produk atau pelayanan yang berpengaruh pada kemampuan produk atau pelayanan tersebut untuk memuaskan kebutuhan yang dinyatakan (tersirat). Kemudian menurut Tjiptono (2015) bahwa kualitas merupakan suatu kondisi dinamis yang berhubungan dengan produk, jasa, manusia, proses, dan lingkungan yang memenuhi atau melebihi harapan. Dengan demikian sebuah produk dinyatakan berkualitas apabila produk atau pelayanan memenuhi atau melebihi harapan pembeli/pelanggan. Kualitas mencerminkan semua dimensi penawaran produk yang menghasilkan manfaat bagi pelanggan. Kualitas suatu produk baik berupa barang atau jasa ditentukan melalui dimensi-dimensinya. Menurut ISO 9000 (1994) kualitas adalah totalitas karakteristik suatu entitas yang didasarkan pada kemampuannya untuk memenuhi kebutuhan yang dinyatakan dan tersirat. Berdasarkan ulasan definisi kualitas diatas dapat dikatakan bahwa suatu kondisi kompleks pada suatu produk barang dan jasa yang harus menyesuaikan dengan kebutuhan dan keinginan penggunanya.

Dimensi kualitas produk menurut Tjiptono (2015) adalah sebagai berikut :

1. *Performance* (kinerja), berhubungan dengan karakteristik operasi dasar dari sebuah produk.
2. *Durability* (daya tahan), yang berarti berapa lama atau umur produk yang bersangkutan bertahan sebelum produk tersebut harus diganti. Semakin besar frekuensi pemakaian konsumen terhadap produk maka semakin besar pula daya produk.
3. *Conformance to specifications* (kesesuaian dengan spesifikasi), yaitu sejauh mana karakteristik operasi dasar dari sebuah produk memenuhi spesifikasi tertentu dari konsumen atau tidak ditemukannya cacat pada produk.
4. *Features* (fitur), adalah karakteristik produk yang dirancang untuk menyempurnakan fungsi produk atau menambah ketertarikan konsumen terhadap produk.
5. *Reliability* (reliabilitas), adalah probabilitas bahwa produk akan bekerja dengan memuaskan atau tidak dalam periode waktu tertentu. Semakin kecil kemungkinan terjadinya kerusakan maka produk tersebut dapat diandalkan.

6. *Aesthetics* (estetika), berhubungan dengan bagaimana penampilan produk.
7. *Perceived quality* (kesan kualitas), sering dibilang merupakan hasil dari penggunaan pengukuran yang dilakukan secara tidak langsung karena terdapat kemungkinan bahwa konsumen tidak mengerti atau kekurangan informasi atas produk yang bersangkutan.
8. *Serviceability*, meliputi kecepatan dan kemudahan untuk direparasi, serta kompetensi dan keramahtamahan staf layanan.

B. PREDIKSI KUALITAS

Prediksi keadaan di masa mendatang mendorong adanya suatu metode yang dapat memberikan gambaran keadaan tersebut secara akurat. Tidak dapat dipungkiri bahwa sebuah populasi data, ataupun sistem tidak berdiri sendiri secara mutlak, tetapi cenderung memiliki keterkaitan dengan variabel lain Sutono et al., (2017). Definisi dari prediksi itu sendiri menurut Herdianto, 2013 dalam Orpa et al., (2019) adalah suatu proses memperkirakan secara sistematis tentang sesuatu yang paling mungkin terjadi di masa depan berdasarkan informasi masa lalu dan sekarang yang dimiliki, agar kesalahannya (selisih antara sesuatu yang terjadi dengan hasil perkiraan) dapat diperkecil. Prediksi tidak harus memberikan jawaban secara pasti kejadian yang akan terjadi, melainkan berusaha untuk mencari jawaban sedekat mungkin yang akan terjadi. Berdasarkan ulasan definisi prediksi itu sendiri, dapat dikatakan suatu proses untuk memperkirakan sesuatu yang akan terjadi di masa depan berdasarkan informasi di masa sebelumnya. Sedangkan prediksi kualitas itu sendiri merupakan suatu proses memperkirakan secara tertata dan terukur pada kualitas barang dan jasa yang akan di produksi berdasarkan informasi data sebelumnya agar kesalahan dapat diperkecil.

C. *IMBALANCED* DATA

Ketidakseimbangan data merupakan suatu keadaan dimana distribusi kelas data tidak seimbang, jumlah kelas data (*instance*) yang satu lebih banyak atau sedikit dibandingkan dengan jumlah kelas data yang lainnya Ali et al., (2015). Debray (2009) menjelaskan bahwa ketidakseimbangan kelas terutama terkait dengan kelangkaan, di mana dua situasi yang berbeda dapat terjadi. Jenis kelangkaan pertama disebut kelas langka dan menyangkut buruknya kehadiran *instance* dari kelas tertentu. Jenis kelangkaan kedua menyangkut kasus langka, yang sesuai dengan kasus yang bermakna tetapi relatif jauh dari kelas. Untuk mempermudah, kami menyebutnya *outlier* kasus langka ini. Meskipun kelas langka dan *outlier* tidak terkait langsung, studi empiris menunjukkan bahwa kelas minoritas mengandung proporsi outlier yang lebih besar daripada kelas

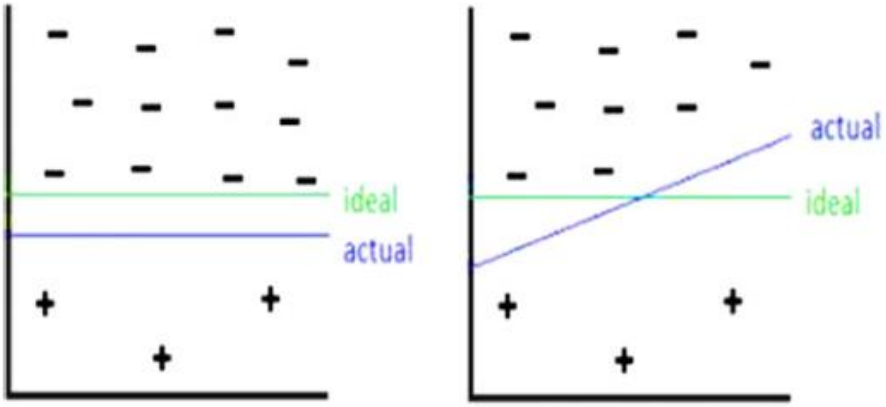
mayoritas. Karena dalam banyak situasi *outlier* tidak dapat dibedakan dari kelas yang *noise*, mengenali lebih banyak (lebih sedikit) *outlier* lebih mungkin menghasilkan *overfitting (underfitting)* kelas minoritas. Selain *noise* kelas, keberadaan *noise* fitur dapat secara signifikan mempengaruhi distribusi data minoritas. Ketika *noise* fitur terjadi dalam kelompok kelas minoritas, *instance* yang terpengaruh dapat menjadi *outlier*. Di sisi lain, ketika *noise* fitur terjadi pada *outlier, instance* yang terpengaruh dapat menghilang dalam cluster kelas minoritas. Oleh karena itu, mengenali lebih banyak (lebih sedikit) pencilan dapat kembali menghasilkan *overfitting* (kurang layak). Ketika data minoritas tersebar, *noise* fitur tidak memiliki efek ini.

D. METODE *RESAMPLING*

a. *Random Under Sampling* (RUS)

Random undersampling adalah metode non-heuristik yang bertujuan untuk menyeimbangkan distribusi kelas melalui penghapusan acak contoh kelas mayoritas. Alasan di baliknya adalah untuk mencoba menyeimbangkan dataset dalam upaya untuk mengatasi kekhasan dari algoritma *machine learning*. Kelemahan utama RUS adalah bahwa metode ini dapat membuang data yang berpotensi berguna yang bisa menjadi penting untuk proses induksi. Masalah lain dengan pendekatan ini adalah bahwa tujuan *machine learning* adalah untuk pengklasifikasi untuk memperkirakan distribusi probabilitas populasi target. Karena distribusi itu tidak diketahui, kami mencoba memperkirakan distribusi populasi menggunakan distribusi sampel. Statistik memberi tahu kita bahwa selama sampel diambil secara acak, distribusi sampel dapat digunakan untuk memperkirakan distribusi populasi dari tempat sampel itu diambil. Oleh karena itu, dengan mempelajari distribusi sampel, kita dapat belajar untuk memperkirakan distribusi target. Namun, setelah kami melakukan *undersampling* dari kelas mayoritas, sampel tidak dapat lagi dianggap acak Kotsiantis et al., (2006).

Menurut Kaur & Gosain, (2018) RUS adalah pendekatan *undersampling* yang sangat sederhana yang secara acak menghapus *instance* dari kelas mayoritas untuk menyeimbangkan dataset sebelum menerapkan teknik klasifikasi. Konsep RUS sangat sederhana dan cepat dibandingkan dengan SMOTE. Satu-satunya batasan dengan teknik ini adalah dapat menghilangkan informasi penting yang terkandung dalam kelas mayoritas, yang mungkin tidak dapat diterima dalam beberapa kasus. Itu dapat diilustrasikan pada Gambar 2.1.



Gambar 2.1 *Classification with and without using undersampling*

Pada gambar tersebut, garis hijau menunjukkan batas keputusan ideal yang ingin kami identifikasi dan garis biru menunjukkan hasil aktual yang diperoleh. Sisi kiri gambar menunjukkan hasil klasifikasi setelah menerapkan algoritma pembelajaran mesin umum tanpa *undersampling* dan bagian kanan menunjukkan hasil setelah menerapkan *undersampling*. Dengan menerapkan *undersampling* pada kelas mayoritas, beberapa informasi kelas mayoritas yang informatif dihapus dan menyebabkan batas keputusan biru menjadi miring, menyebabkan beberapa *instance* kelas mayoritas diklasifikasi sebagai *instance* kelas minoritas secara salah.

Langkah-langkah algoritma RUS menurut Kaur & Gosain, (2018) adalah sebagai berikut:

1. Muat dataset dan identifikasi kelas minoritas dan mayoritas;
2. Hitung jumlah *instance* yang akan dihapus berdasarkan persentase *undersampling*;
3. Identifikasi *instance* acak dari kelas mayoritas dan hapus dari kelas mayoritas;
4. Ulangi langkah 3 hingga jumlah *instance* dihapus sesuai persentase yang diberikan.

b. *Random Over Sampling* (ROS)

Kotsiantis et al., (2006) berpendapat bahwa *random oversampling* adalah metode non-heuristik yang bertujuan untuk menyeimbangkan distribusi kelas melalui replikasi acak contoh kelas minoritas. Beberapa penulis seperti N. V. Chawla, L. O. Hall, K. W. Bowyer, W. P. Kegelmeyer, M. Kubat dan S. Matwin setuju bahwa pengambilan sampel secara acak dapat meningkatkan kemungkinan terjadinya *overfitting* karena membuat salinan yang tepat dari contoh kelas minoritas. Dengan cara ini, penggolong simbolis, misalnya, dapat membuat aturan yang tampaknya akurat, tetapi sebenarnya mencakup satu

contoh yang direplikasi. Selain itu, *oversampling* dapat memperkenalkan tugas komputasi tambahan jika kumpulan data sudah cukup besar tetapi tidak seimbang.

c. *Synthetic Minority Over Sampling* (SMOTE)

Menurut Gu et al., (2016) SMOTE adalah metode pengambilan sampel berlebih. Gagasan utamanya adalah untuk membuat *instance* kelas minoritas baru dengan menginterpolasi beberapa *instance* kelas minoritas asli yang bersatu. SMOTE secara acak memilih satu (atau lebih tergantung pada rasio pengambilan sampel berlebih) dari tetangga terdekat k dari *instance* kelas minoritas dan melakukan interpolasi acak dua *instance* untuk membuat *instance* sintetis baru. *Instance* sintetik dihasilkan dengan cara berikut: perbedaan antara *instance* asli dan tetangga terdekat diperoleh, perbedaan ini dikalikan dengan angka acak antara 0 dan 1, dan hasilnya ditambahkan di mana *Ge* adalah generasi saat ini, *Ge* adalah generasi maksimum, dan *rnd* (2) adalah hasil yang diperoleh setelah positif ke *instance* asli. Pada dasarnya, titik acak dipilih sepanjang segmen garis antara *instance* asli dan tetangga terdekatnya. Pendekatan ini secara efektif memaksa wilayah keputusan kelas minoritas untuk menjadi umum. Dengan demikian, masalah *over-fitting* dihindari, dan batas keputusan untuk kelas minoritas menyebar lebih jauh ke ruang kelas mayoritas.

Menurut Siringoringo (2018) metode *Synthetic Minority Over-sampling Technique* (SMOTE) merupakan metode yang populer diterapkan dalam rangka menangani ketidak seimbangan kelas. Teknik ini mensintesis sampel baru dari kelas minoritas untuk menyeimbangkan dataset dengan cara sampling ulang sampel kelas minoritas. Sedangkan menurut Kaur & Gosain, (2018) SMOTE adalah pendekatan *oversampling* yang secara sintetis menghasilkan *instance* dengan memilih secara acak *instance* dari kelas minoritas dan menggunakan metode interpolasi untuk menghasilkan *instance* antara titik yang dipilih dan *instance* yang berdekatan. Dalam proses ini, setiap *instance* kelas minoritas dipertimbangkan dan *instance* kelas minoritas baru dihasilkan di sepanjang segmen garis yang bergabung dengan k-tetangga terdekatnya. Jumlah *instance* sintetis dihasilkan berdasarkan persentase *oversampling* yang diperlukan.

Kaur & Gosain, (2018) menjelaskan langkah-langkah algoritma SMOTE adalah sebagai berikut:

1. Muat dataset dan identifikasi kelas minoritas dan mayoritas;
2. Hitung jumlah *instance* yang akan dihasilkan berdasarkan persentase *oversampling*;
3. Identifikasi contoh acak dari kelas minoritas dan temukan tetangga terdekatnya;

4. Pilih salah satu tetangga terdekat dan temukan perbedaan antara *instance* acak dan tetangga terdekat terpilihnya;
5. Lipat gandakan selisih dengan angka acak yang dihasilkan antara 0 dan 1;
6. Tambahkan perbedaan ini ke *instance* acak yang dipilih;
7. Ulangi proses dari 3 hingga 6 hingga jumlah *instance* dihasilkan sesuai persentase yang diberikan.

E. ALGORITMA PENGKLASIFIKASI

a. Algoritma *Naïve Bayes*

Menurut Debray (2009) algoritma *Naive Bayes* adalah algoritma pembelajaran berdasarkan aturan *Bayes*, yang mengasumsikan fitur $X_1 \ldots X_m$ semuanya tergantung kondisi satu sama lain, mengingat Y. Nilai dari asumsi ini adalah bahwa secara dramatis menyederhanakan representasi P (xjy), dan masalah memperkirakannya dari *training* data. Untuk menangani fitur yang berkelanjutan, disarankan dua pendekatan. Pendekatan pertama melibatkan estimasi kepadatan oleh distribusi *Gaussian*. Pendekatan kedua melibatkan estimasi kepadatan oleh kernel non parametrik. Pembelajaran *Naive Bayes* ternyata bekerja dengan sangat baik dalam algoritma pembelajaran tujuan umum; versi yang ditingkatkan adalah salah satu algoritma pembelajaran tujuan umum yang paling efektif. *Naive Bayes* menskala dengan baik untuk masalah yang sangat besar: dengan fitur n *Boolean*, hanya ada 2n+1 parameter untuk diperkirakan, dan tidak ada pencarian yang diperlukan untuk menemukan hipotesis *Naive Bayes* kemungkinan-maksimum. Akhirnya, pembelajaran *Naïve Bayes* tidak memiliki kesulitan dengan data berisik dan dapat memberikan prediksi probabilistik saat yang tepat.

Algoritma *Naive Bayes* didasarkan pada asumsi penyederhanaan bahwa nilai atribut secara kondisional saling bebas jika diberikan nilai *output*. Dengan kata lain, diberikan nilai *output*, probabilitas mengamati secara bersama adalah produk dari probabilitas individu. Dalam metode *Naive Bayes* data *string* yang bersifat konstan dibedakan dengan data numerik yang bersifat kontinyu, perbedaan ini akan terlihat pada saat menentukan nilai probabilitas setiap kriteria baik itu kriteria dengan nilai data *string* maupun kriteria dengan nilai data numerik Patil & Sherekar, (2013).

b. Algoritma *Random Forest*

Menurut Breiman (2001) metode *random forest* merupakan pengembangan dari metode CART, yaitu dengan menerapkan metode *bootstrap aggregating (bagging)* dan *random feature selection. Random forest* adalah pengklasifikasi yang terdiri dari kumpulan pengklasifikasi pohon terstruktur

dimana masing-masing pohon melemparkan unit suara untuk kelas paling populer di *input* x.

Dengan kata lain *random forest* terdiri dari sekumpulan *decision tree* yang dimana digunakan untuk mengklasifikasi data ke suatu kelas. Pada gugus data yang terdiri atas n amatan dan p peubah penjelas, *random forest* dilakukan dengan cara Breiman (2001) dan Cutler et al., (2011):

1. Lakukan penarikan contoh acak berukuran n dengan pemulihan pada gugus data. Tahapan ini merupakan tahapan *bootstrap*.
2. Dengan menggunakan contoh *bootstrap*, pohon dibangun sampai mencapai ukuran maksimum (tanpa pemangkasan). Pada setiap simpul, pemilihan pemilah dilakukan dengan memilih m peubah penjelas secara acak, dimana m << p. Pemilah terbaik dipilih dari m peubah penjelas tersebut. Tahapan ini adalah tahapan *random feature selection*.
3. Ulangi langkah 1 dan 2 sebanyak k kali, sehingga terbentuk sebuah hutan yang terdiri atas k pohon.

Respon suatu amatan diprediksi dengan menggabungkan (*aggregating*) hasil prediksi k pohon. Pada masalah klasifikasi dilakukan berdasarkan *majority vote* (suara terbanyak). *Error* klasifikasi *random forest* diduga melalui *error* OOB yang diperoleh dengan cara Breiman (2001 dan Cutler et al., (2011):

1. Lakukan prediksi terhadap setiap data *out of bag* (OOB) pada pohon yang bersesuaian. Data OOB adalah data yang tidak termuat dalam contoh *bootstrap.*
2. Secara rata-rata, setiap amatan gugus data asli akan menjadi data OOB sebanyak sekitar 36% dari banyak pohon. Oleh karena itu, pada langkah 1, masing-masing amatan gugus data asli mengalami prediksi sebanyak sekitar sepertiga kali dari banyaknya pohon. Jika a adalah sebuah amatan dari gugus data asli, maka hasil prediksi *random forest* terhadap a adalah gabungan dari hasil prediksi setiap kali a menjadi data OOB.
3. *Error* OOB dihitung dari proporsi misklasifikasi hasil prediksi *random forest* dari seluruh amatan gugus data asli.

c. Algoritma C 4.5

Algoritma C 4.5 yaitu sebuah algoritma yang digunakan untuk membangun *decision tree* (pengambilan keputusan). Algoritma C.45 adalah salah satu algoritma induksi pohon keputusan yaitu ID3 (*Iterative Dichotomiser 3*). ID3 dikembangkan oleh J. Ross Quinlan. Dalam prosedur algoritma ID3, *input* berupa sampel *training*, label *training* dan atribut. Algoritma C 4.5 yaitu sebuah algoritma yang digunakan untuk membangun *decision tree* (pengambilan keputusan). Algoritma C 4.5 merupakan pengembangan dari ID3. Beberapa pengembangan yang dilakukan pada C 4.5 adalah sebagai antara lain

bisa mengatasi *missing value*, bisa mengatasi continyu data, dan pemangkasan menurut Azwanti (2018).

Untuk memilih atribut akar, didasarkan pada nilai *gain* tertinggi dari atribut-atribut yang ada. Untuk menghitung *gain* digunakan rumus seperti yang tertera dalam persamaan berikut.

$$Gain\ (S,A) = Entropy\ (S) - \sum_{i=1}^{n} \frac{|Si|}{|S|} * Entropy\ (Si) \ldots\ldots\ldots\ldots(1)$$

Dimana :
S : himpunan kasus
A : atribut
N : jumlah partisi atribut A
|Si| : jumlah kasus pada partisi ke-i
|S| : jumlah kasus dalam S

Sementara itu, perhitungan nilai entropi dapat dilihat pada persamaan 2 berikut.

$$Entropy\ (S) = \sum_{i=1}^{n} - pi * \ log_2\ pi \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Dimana :
S : himpunan kasus
A : fitur
N : jumlah partisi S
pi : proporsi dari Si terhadap S

Secara umum algoritma C 4.5 untuk membangun pohon keputusan adalah sebagai berikut :

1) Pilih atribut sebagai akar.
2) Buat cabang untuk tiap-tiap nilai.
3) Bagi kasus dalam cabang.
4) Ulangi proses untuk setiap cabang sampai semua kasus pada cabang memiliki kelas yang sama.

F. PENGUKURAN PERFORMA

a. Confusion Matrix

Confusion matrix merupakan alat yang paling populer dalam mengevaluasi performa klasifikasi. Pada tabel berikut ditampilkan *confusion matrix* untuk kelas biner, yaitu dataset dengan dua jenis kelas saja.

Tabel 2.1 *Confusion Matrix*

Class	***Predictive Positive***	***Predictive Negative***
Actual Positive	*True Positive (TP)*	*False Negative (FN)*
Actual Negative	*False Positive (FP)*	*True Negative (TN)*

Sumber : (Tahir et al., 2019)

True Positive (TP) dan *True Negative* (TN) merupakan jumlah kelas positif dan negatif yang diklasifikasikan dengan tepat, *False Positive* (FP) dan *False Negative* (FN) merupakan jumlah kelas positif dan negatif yang tidak diklasifikasikan dengan tepat. Berdasarkan *confusion matrix* tersebut dapat ditentukan kriteria performa seperti *Accuracy, Precision, Sensitivity, Specificity, F-Measure,* dan *G-Mean* menurut Siringoringo (2018).

b. Accuracy (Acc)

Menurut Tahir et al., (2019) akurasi (*accuracy*) merupakan kriteria yang paling umum untuk mengukur kinerja klasifikasi, tetapi jika bekerja pada kelas tidak seimbang, kriteria ini kurang tepat karena kelas minoritas akan memiliki sumbangsih yang kecil pada kriteria *accuracy* Tahir et al., (2019).

$$\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} \quad \ldots\ldots\ldots\ldots (3)$$

Dimana :
TP : *True Positive*
TN : *True Negative*
FP : *False Positive*
FN : *False Negative*

c. Precision (Pr)

Precision merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif Tahir et al., (2019).

$$Pr = \frac{TP}{TP+FP} \quad \ldots\ldots\ldots\ldots(4)$$

Dimana :
TP : *True Positive*
FP : *False Positive*

d. Sensitivity (Sn)

Sensitivity adalah rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif menurut Tahir et al., (2019).

$$Sn = \frac{TP}{TP+FN} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

Dimana :
TP : *True Positive*
FN : *False Negative*

e. Specificity (Sp)

Specificity adalah kebenaran memprediksi negatif dibandingkan dengan keseluruhan data negatif menurut Tahir et al., (2019).

$$Sp = \frac{TN}{TN+FP} \quad \ldots\ldots\ldots\ldots\ldots\ldots(6)$$

Dimana :
TN : *True Negative*
FP : *False Positive*

f. F-Measure

F-Measure digunakan untuk mengukur klasifikasi kelas minoritas pada kelas tidak seimbang menurut Tahir et al., (2019).

$$FM = 2 \text{ x } \frac{Pr * Sn}{Pr + Sn} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (7)$$

Dimana :
Pr : *Precision*
Sn : *Sensitivity*

g. *G-Mean*

Sedangkan *G-Mean* menurut Tahir et al., 2019) digunakan untuk mengukur performa keseluruhan (*overall classification performance*).

$$\text{GM} = \sqrt{Sp * Sn} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (8)$$

Dimana :

Sp : *Specificity*

Sn : *Sensitivity*

G. PENELITIAN TERKAIT

Penelitian yang membahas penanganan ketidakseimbangan data, telah banyak dilakukan. Beberapa penelitian terkait erat dengan penelitian ini. Penelitian itu diantaranya :

No.	Penulis	Judul	Sumber	Deskripsi Penelitian
1.	Rimbun Siringoringo	Klasifikasi Data Tidak Seimbang Menggunakan Algoritma SMOTE dan *K-Nearest Neighbor*	Jurnal ISD Vol.3 No.1 Januari - Juni 2018, pISSN : 2477-863X, eISSN : 2528-5114	Klasifikasi data yang tidak seimbang merupakan masalah yang krusial pada bidang *machine learning* dan data mining. Ketidakseimbangan data memberikan dampak yang buruk pada hasil klasifikasi dimana kelas minoritas sering disalah klasifikasikan sebagai kelas mayoritas. *k-Nearest Neighbor* merupakan salah satu metode klasifikasi yang sangat populer dan sederhana tetapi, tidak dilengkapi dengan kemampuan untuk bekerja pada dataset tidak seimbang. Pada penelitian ini, *Synthetic Minority Over-sampling Technique* (SMOTE) diterapkan untuk menyelesaikan masalah ketidak seimbangan kelas pada dataset *Credit Card Fraud*. Dengan menerapkan skema

				evaluasi *10-cross fold validation* diperoleh hasil bahwa SMOTE meningkatkan rata-rata *G-Mean* dari 53,4% ke 81,0% dan rata-rata *F-Measure* dari 38,7 ke 81,8%.
2.	Prabhjot Kaur and Anjana Gosain	*Comparing the Behavior of Oversampling and Undersampling Approach of Class Imbalance Learning by Combining Class Imbalance Problem with Noise*	*Springer Nature Singapore Pte Ltd. 2018 A.K. Saini et al. (eds.), ICT Based Innovations, Advances in Intelligent Systems and Computing 653.*	*Class imbalance learning is a recent topic, which helps us to detect the classes from unbalanced datasets. In various real scenarios, where we need to find the exceptional cases like credit card problem, brain tumor detection, etc., the traditional classification algorithms fail because they are designed in such a way that either their results are overwhelmed by the bigger class and or they ignore the smaller class as a noise and avoid it. In recent studies, it has been found that class imbalance problem itself is not a problem but there are certain other data distribution complexities, which when combined with the class imbalance problem degrade the performance of classifier. One of the major issues is noise in the data, which is a part of every real data in one form or another. This paper compares the oversampling and undersampling approaches of class imbalance learning in noisy environment and tries to find out which is the better approach in such case.*

3.	Bekir Karlık, Abdulkerim M. Yibre, and Barış Koçer	*Comprising Feature Selection and Classifier Methods with SMOTE for Prediction of Male Infertility*	*International Journal of Fuzzy Systems And Advanced Applications, Volume 3, 2016*	*The aim of this study is on prediction of male fertility problems by using different classifier algorithms and applying feature selection methods. Used data in this study is gathered from UCI data repository. The ratio of between normal and abnormal (alter) samples is more than 7. This indicates the dataset is originally imbalanced in order to balance the examples of the dataset SMOTE technique which is applied for a better accuracy and representative result of classifiers. Feature selection and classification methods are comprised for prediction of male fertility. In this study, MLP, Naïve Bayes, Random Forest, KNN, and SVM classifiers were used. Comparison results show that Naive Bayes classifier has better classification accuracy as 90.65% than the others.*
4.	Aries Saifudin dan Romi Satria Wahono	Pendekatan Level Data untuk Menangani Ketidakseimbangan Kelas pada Prediksi Cacat Software	*Journal of Software Engineering, Vol. 1, No. 2, December 2015*	Pada penelitian ini diusulkan pendekatan level data dengan *resampling*, yaitu *random oversampling (ROS)*, dan *random undersampling (RUS)*, dan mensintesis menggunakan algoritma FSMOTE. Pengklasifikasi yang digunakan adalah *Naïve Bayes* (NB). Hasil penelitian menunjukkan bahwa model FSMOTE+NB merupakan model pendekatan level data terbaik pada prediksi

				cacat *software* karena nilai sensitivitas dan *G-Mean* model FSMOTE+NB meningkat secara signifikan, sedangkan model ROS+NB dan RUS+NB tidak meningkat secara signifikan.

H. POSISI PENELITIAN

Keterkaitan penelitian ini dengan penelitian-penelitian sebelumnya dijelaskan dalam tabel 2.1 berikut ini :

Tabel 2.1
Posisi Penelitian

Penulis	Kasus	Teknik Preprocessing	Algoritma Pengklasifikasi
Aries Saifudin & Romi Satria Wahono (2015)	Prediksi Cacat Software	*1) Random Oversampling;* *2) Random Undersampling,* 3) FSMOTE	1) *Naïve Bayes*
Bekir Karlık, Abdulkerim M. Yibre, and Barış Koçer (2016)	*Prediction of Male Infertility*	*1) Synthetic Minority Oversampling Technique*	1) *MLP* *2) Naïve Bayes* *3) Random Forest* *4) KNN* *5) SVM*
Prabhjot Kaur and Anjana Gosain (2016)	*Comparing the Behavior of Oversampling and Undersampling Approach of Class Imbalance*	*1) Random Undersampling;* *2) Synthetic Minority Oversampling Technique*	*1) Cost Sensitive*
Rimbun Siringoringo (2018)	Penipuan Kartu Kredit	*1) Synthetic Minority Oversampling Technique.*	1) *KNN*
Usulan (2020)	Semikonduktor	*1) Random Oversampling;* *2) Random Undersampling;* *3) Synthetic Minority Oversampling Technique*; dan 4) SMOTE+RUS	*1) Naïve Bayes* *2) C 4.5* *3) Random Forest*