TESIS
PENINGKATAN RECOGNITION RATE KELAS MINORITAS ALGORITMA NAIVE BAYES DENGAN METODE SMOTE PADA DATA NUMERIK

Disusun Oleh:

Nama : **HIZBUL IZZI**

NIM : **22.55.1206**

Konsentrasi : **Business Intelligence**

PROGRAM STUDI S2 PJJTEKNIK INFORMATIKA

PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA

YOGYAKARTA

2025

TESIS

PENINGKATAN RECOGNITION RATE KELAS MINORITAS ALGORITMA NAIVE BAYES DENGAN METODE SMOTE PADA DATA NUMERIK

IMPROVING MINORITY CLASS RECOGNITION RATE OF THE NAIVE BAYES ALGORITHM USING THE SMOTE METHOD ON NUMERICAL DATA

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister

Disusun Oleh:

Nama : **HIZBUL IZZI**

NIM : **22.55.1206**

Konsentrasi : **Business Intelligence**

PROGRAM STUDI S2 PJJ TEKNIK INFORMATIKA

PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA

YOGYAKARTA

2025

HALAMAN PERSETUJUAN

PENINGKATAN RECOGNITION RATE KELAS MINORITAS ALGORITMA NAIVE BAYES DENGAN METODE SMOTE PADA DATA NUMERIK

IMPROVING MINORITY CLASS RECOGNITION RATE OF THE NAIVE BAYES ALGORITHM USING THE SMOTE METHOD ON NUMERICAL DATA

Yang disusun dan diajukan oleh

HIZBUL IZZI

22.55.1206

Telah disetujui oleh Dosen Pembimbing Tesis
Pada tanggal 27 Agustus 2025

Dosen Pembimbing,

Dr. Arief Setyanto, S.Si., M.T., Ph.D.
NIK. 190302036

HALAMAN PENGESAHAN

PENINGKATAN RECOGNITION RATE KELAS MINORITAS ALGORITMA NAIVE BAYES DENGAN METODE SMOTE PADA DATA NUMERIK

IMPROVING MINORITY CLASS RECOGNITION RATE OF THE NAIVE BAYES ALGORITHM USING THE SMOTE METHOD ON NUMERICAL DATA

yang disusun dan diajukan oleh

HIZBUL IZZI
22.55.1206

Telah dipertahankan di depan Dewan Penguji pada
tanggal 27 Agustus 2025

Susunan Dewan Penguji

Nama Penguji	Tanda Tangan
Arief Setyanto, S.Si., M.T., Ph.D. **NIK. 190302036**	
Dr. Andi Sunyoto, M.Kom. **NIK. 190302052**	
Alva Hendi Muhammad, S.T., M.Eng., Ph.D. **NIK. 190302493**	

Tesis ini telah diterima sebagai salah satu persyaratan untuk
memperoleh gelar Magister Komputer
Tanggal 27 Agustus 2025

DEKAN FAKULTAS ILMU KOMPUTER

Prof. Dr. Kusrini, M.Kom.
NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertanda tangan dibawah ini,

Nama mahasiswa : **HIZBUL IZZI**

NIM : **22.55.1206**

Konsentrasi : **Business Intelligence**

Menyatakan bahwa Tesis dengan judul berikut:

PENINGKATAN RECOGNITION RATE KELAS MINORITAS ALGORITMA NAIVE BAYES DENGAN METODE SMOTE PADA DATA NUMERIK

Dosen Pembimbing Utama : Arief Setyanto, S.Si., M.T., Ph.D.
Dosen Pembimbing Pendamping : Anggit Dwi Hartanto, M.Kom.

1. Karya tulis ini benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi.

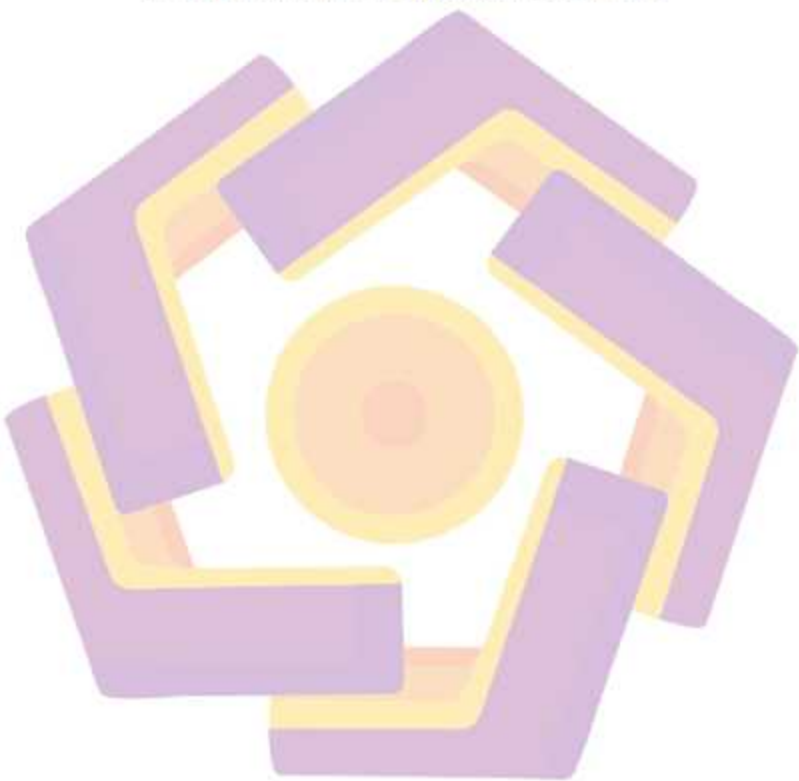
Yogyakarta, 27 Agustus 2025
Yang Menyatakan

HIZBUL IZZI

HALAMAN PERSEMBAHAN

Tesis ini saya persembahkan dengan penuh cinta dan rasa hormat kepada:

Alm ayah saya, dan Ibuku Tercinta, Semua pihak yang telah membantu dalam proses penelitian ini dan Untuk semua penuntut Ilmu

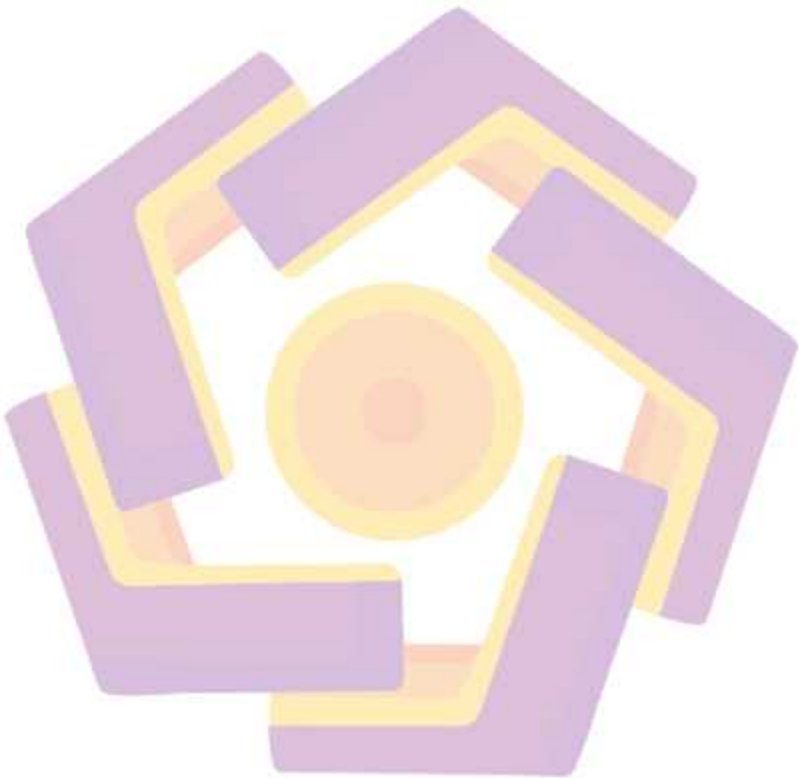


HALAMAN MOTTO

" Dengan seni hidup jadi indah

Dengan ilmu hidup jadi mudah

Dengan agama hidup jadi terarah"



KATA PENGANTAR

Puji Syukur kehadirat Allah SWT yang telah mmberikan Rahmat dan hidayah-Nya sehingga saya dapat menyelesaikan tesis yang merupakan syarat untuk menyelesaikan jenjang pendidikan S2 di Universitas AMIKOM Yogyakarta yang berjudul Peningkatan Akurasi Algoritma Naive Bayes Dengan Metode *Synthetic Minority Oversampling Technique* (Smote) Pada Data Numerik. Pada kesempatan ini, penulis ingin mengucapkan terima kasih kepada:

1. Kedua orang tua yang senantiasa memberikan doa dan dukungan sehingga penulis berhasil sampai pada tahapan terakhir dalam menempuh pendidikan S2.
2. Rektor AMIKOM Yogyakarta
3. Direktur Pasca Sarjana AMIKOM Yogyakarta
4. Dosen Pembimbing Utama
5. Dosen Pembimbing Pendamping
6. Staf akademik AMIKOM Yogyakarta
7. Rekan-rekan seperjuangan

Penulis menyadari terdapat kesalahan dalam penyusunan tesis ini, sehingga penulis berharap mendapatkan kritik dan saran yang bersifat membangun untuk perbaikan tesis ini. Penulis berharap semoga tesis ini dapat bermanfaat tidak hanya bagi penulis saja, tetapi juga bagi pembaca.

Yogyakarta,27 Agustus 2025

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

INTISARI

Pada penelitian ini akan mengklasifikasikan data numerik yaitu data loan yang diambil dari Kaggle. Data yang digunakan berjumlah 9578 dataset yang meliputi kelas data dengan peminjam dapat menyelesaikan kredit sebanyak 8045 record dan peminjaman yang tidak dapat menyelesaikan kredit sebanyak 1533 record. Dari jumlah data tersebut terdapat ketidakseimbangan kelas sehingga perlu dilakukan penyeimbangan agar mendapatkan hasil klasifikasi yang lebih akurat. Tujuan dari penelitian ini adalah meningkatkan akurasi algoritma Naïve Bayes dalam mengklasifikasikan data numerik. Penipuan dalam transaksi keuangan adalah contoh kasus data tidak seimbang, di mana jumlah transaksi yang sah jauh lebih banyak dibandingkan yang merupakan penipuan.

Optimalisasi akurasi pada kelas minoritas (penipuan) sangat penting untuk menghindari kerugian. Metode yang digunakan untuk meningkatkan akurasi algoritma yaitu *Syntetic Minority Oversampling Tecnique* (SMOTE) dengan cara meng-over sampling minoritas dataset. Selain itu juga menggunakan metoe K-Fold Cross Validation untuk mengevaluasi performa dari proses algoritma yang digunakan. Praproses data dilakukan untuk membersihkan data dari nilai-nilai yang hilang dan tidak valid serta menormalisasi data supaya semua fitur berada dalam skala yang sama dan sesuai untuk analisis klasifikasi.

Berdasarkan hasil analisis yang dilakukan, sebelum penerapan SMOTE kemampuan model mengenali kelas minoritas sebesar 16.1%, sedangkan setelah penerapan SMOTE kemampuan model mengenali kelas minoritas menjadi 48.8%. Selain itu juga sebelum penerapan SMOTE model mampu memprediksi kelas minoritas dengan benar sebanyak 10 kasus sedangkan setelah penerapan SMOTE, model mampu memprediksi kelas minoritas dengan benar sebanyak 102 kasus. Sehingga dapat disimpulkan bahwa teknik SMOTE mampu meningkatkan kemampuan model.

Kata kunci: Data Numerik, Klasifikasi, K-Fold Cross Validation, Naïve Bayes, SMOTE.

ABSTRACT

This research will classify numerical data, namely loan data taken from Kaggle. The data used amounted to 9578 datasets which included data classes with borrowers able to complete credit as many as 8045 records and loans that could not complete credit as many as 1533 records. From the amount of data there is an imbalance of classes so it is necessary to do balancing in order to get more accurate classification results. The purpose of this research is to improve the accuracy of the Naïve Bayes algorithm in classifying numerical data. Fraud in financial transactions is an example of a case of imbalanced data, where the number of legitimate transactions is much greater than those that are fraudulent.

Optimizing accuracy in minority (fraud) classes is very important to avoid losses. The method used to improve the accuracy of the algorithm is the Synthetic Minority Oversampling Technique (SMOTE) by over sampling the minority of the dataset. In addition, it also uses the K-Fold Cross Validation method to evaluate the performance of the algorithm process used. Data preprocessing is done to clean the data from missing and invalid values and normalize the data so that all features are on the same scale and suitable for classification analysis.

Based on the results of the analysis conducted, before the application of SMOTE the model's ability to recognize minority classes was 16.1%, while after the application of SMOTE the model's ability to recognize minority classes became 48.8% besides that, before the application of SMOTE the model was able to predict the minority class correctly in 10 cases while after the application of SMOTE, the model was able to predict the minority class correctly in 102 cases. So it can be concluded that the SMOTE technique is able to improve the ability of the model.

Keywords: *Classification, K-Fold Cross Validation, Naive Bayes, SMOTE*

BAB I
PENDAHULUAN

1.1 Latar Belakang

Data mining adalah analisis terhadap data untuk menemukan hubungan yang jelas serta menyimpulkannya yang belum diketahui sebelumnya dengan cara terkini dipahami dan berguna bagi pemilik data tersebut. Data mining merupakan salah satu cara yang digunakan untuk mendapatkan pengetahuan baru dengan memanfaatkan jumlah data yang sangat besar, salah satu teknik dalam data mining adalah klasifikasi. Klasifikasi adalah tugas dasar dari analisis data yang berfungsi memberikan label kelas untuk kasus yang dijelaskan oleh satu set atribut. Sehingga tujuan dari klasifikasi adalah kebenaran dalam memprediksi sebuah nilai. Beberapa algoritma yang dapat digunakan untuk klasifikasi adalah Naïve Bayes, Decision tree, Artificial Neural Network, dan lain sebagainya (Pratiwi, 2020).

Algoritma Naïve bayes adalah salah satu algoritma klasifikasi yang memiliki akurasi yang lebih tinggi dibandingkan oleh algoritma yang lain. Hasil klasifikasi sangat berpengaruh jika terjadinya data yang tidak seimbang. Ketidakseimbangan data adalah kondisi dimana jumlah sampel di satu kelas jauh lebih banyak dibandingkan dengan kelas lainnya. Untuk mengatasi masalah ketidakseimbangan ini, berbagai teknik telah dikembangkan salah satunya adalah Synthetic Minority Oversampling Technique (SMOTE) (Widia, 2023). SMOTE telah digunakan pada berbagai aplikasi, termasuk pengklasifikasi data numerik. Dalam era perkembangan teknologi yang sangat cepat, analisis data numerik menjadi sangat penting dalam berbagai bidang, seperti bisnis, medis, dan ilmu pengetahuan. Dalam beberapa aplikasi, data numerik dapat memiliki distribusi tidak seimbang, terutama ketika memiliki minoritas yang relatif kecil.

Data numerik yang digunakan pada penelitian ini adalah data loan. Data ini diperoleh dari Kaggle yakni data dari *lending club.com*. *LendingClub.com* adalah platform pinjaman *peer-to-peer* yang menghubungkan peminjam yang

membutuhkan dana dengan investor yang memiliki dana untuk diinvestasikan. Data yang digunakan dari tahun 2007–2010 yang mencangkup berbagi fitur atau variabel didalamnya. Penelitian ini menggunakan 9578 dataset yang meliputi kelas data dengan peminjam dapat menyelesaikan kredit sebanyak 8045 record dan peminjam yang tidak dapat menyelesaikan kredit sebanyak 1533 record yang dimana menunjukkan adanya ketidakseimbangan kelas. Maka, metode oversampling seperti SMOTE digunakan untuk menangani ketidakseimbangan kelas dan meningkatkan kinerja model klasifikasi pada data yang akan digunakan. Meskipun dalam beberapa kasus, metode lain seperti oversampling, undersampling, atau menggunakan algoritma lain seperti Adaboost atau Bagging dapat digunakan sebagai alternatif. Namun, SMOTE tetap menjadi salah satu metode yang paling efektif dalam mengatasi ketidakseimbangan kelas pada klasifikasi. Karena SMOTE dapat menghasilkan data sintetis yang mirip dengan data minoritas, sehingga data sintetis tersebut dapat digunakan sebagai data latih yang lebih representatif. Dengan demikian, model klasifikasi dapat lebih efektif dalam mengklasifikasi data minoritas (Nursyahfitri, Rozikin, & Adam, 2022).

Adapun beberapa penelitian sebelumnya yang terkait dengan algoritma dan metode yang digunakan pada penelitian ini antara lain. Penelitian yang dilakukan oleh (Wang, S., Dai, Y., Shen, J., & Xuan, J, 2021). Penelitian tentang perluasan dan klasifikasi data tidak seimbang berdasarkan algoritma SMOTE" berfokus pada mengatasi tantangan yang ditimbulkan oleh dataset tidak seimbang dalam pembelajaran mesin. Data yang tidak seimbang dapat menghasilkan model klasifikasi yang bias dan condong ke kelas mayoritas. Para penulis mengusulkan Teknik Over-sampling Minoritas Sintetis (SMOTE) yang ditingkatkan untuk memperluas dan menyeimbangkan data. Makalah ini merinci peningkatan algoritma dan mengevaluasi kinerjanya di berbagai dataset. Hasil eksperimen menunjukkan bahwa algoritma SMOTE yang ditingkatkan secara efektif mengatasi ketidakseimbangan kelas dan meningkatkan akurasi klasifikasi. Temuan ini menyoroti pentingnya pra-

pemrosesan data dalam mengembangkan model pembelajaran mesin yang kuat untuk data tidak seimbang.

Selanjutnya penelitian oleh (Duan, F., Zhang, S., Yan, Y., & Cai, Z, 2022). Penelitian ini berfokus pada pengembangan metode oversampling baru untuk mengatasi masalah data tidak seimbang dalam diagnosis kesalahan mekanis. Algoritma MeanRadius-SMOTE yang diusulkan menggunakan radius rata-rata untuk meningkatkan kualitas data sintetis yang dihasilkan. Penulis mengevaluasi kinerja metode ini melalui berbagai eksperimen dan membandingkannya dengan teknik oversampling yang ada. Hasil eksperimen menunjukkan bahwa MeanRadius-SMOTE secara signifikan meningkatkan akurasi diagnosis kesalahan dibandingkan dengan metode oversampling tradisional. Temuan ini menunjukkan potensi besar dari pendekatan baru ini dalam meningkatkan keandalan diagnosis kesalahan mekanis pada data tidak seimbang.

Penelitian oleh (Hairani, H., Anggrawan, A., & Priyanto, D, 2023). Penelitian ini bertujuan untuk menerapkan metode Smote-Tomeklink dan Random Forest dalam klasifikasi diabetes. Data yang digunakan adalah data diabetes yang diperoleh dari Kaggle sebanyak 768 data dengan delapan atribut input dan 1 output atribut sebagai kelas, pra-pemrosesan data digunakan untuk menyeimbangkan dataset dengan Smote-Tomeklink, klasifikasi menggunakan random forest metode, dan evaluasi kinerja berdasarkan akurasi, sensitivitas, presisi, dan skor F1. Berdasarkan pengujian yang dilakukan dengan cara membagi data menggunakan validasi silang 10 kali lipat, algoritma Random Forest dengan Smote-TomekLink mendapatkan akurasi, sensitivitas, presisi, dan skor F1 dibandingkan dengan Random Forest dengan Smote. Algoritma Random Forest dengan Smote-Tomeklink memiliki 86,4% akurasi, sensitivitas 88,2%, presisi 82,3%, dan skor F1 85,1%. Dengan demikian, penggunaan Smote-Tomeklink dapat meningkatkan kinerja metode hutan acak berdasarkan akurasi, sensitivitas, presisi, dan skor F1.

Penelitian yang dilakuan oleh (Setiawan, Erlansari, & Sari, 2023). Penelitian ini bertujuan untuk mengetahui gambaran umum dan akurasi review

pengguna aplikasi TIX ID serta perubahan signifikan akurasi apabila menggunakan Naïve Bayes dengan penambahan Feature SMOTE dan PSO. Metode yang digunakan adalah Naïve Bayes berbasis SMOTE & PSO. Hasil penelitian ini menunjukkan bahwa gambaran umum penerimaan pengguna pada variabel metode pembayaran DANA masih "Ditolak" oleh pengguna dengan nilai probabilitas penerimaan yakni P(C) untuk P(Diterima) = 0,32 < P(Ditolak) = 0,68. Sedangkan pada variabel kualitas aplikasi sudah "Diterima" oleh pengguna dengan nilai probabilitas penerimaan yakni P(C) untuk P(Diterima) = 0,585 > P(Ditolak) = 0,415. Hasil pengujian dengan akurasi tertinggi yakni pada variabel metode pembayaran DANA sebesar 93,68 % dan variabel kualitas aplikasi sebesar 96,13 % serta berdasarkan penelitian yang telah dilakukan terjadinya perubahan signifikan apabila menambahkan Feature SMOTE dan PSO yakni terjadinya peningkatan akurasi tertinggi yaitu sebesar 14.94 % untuk variabel metode pembayaran DANA dan 9.06 % % untuk variabel kualitas aplikasi.

Penelitian yang dilakukan oleh (Kurniasih & Isyara, 2023). Penelitian ini bertujuan untuk mengklasifikasikan kasus mahasiswa yang akan di drop out atau tidak menggunakan metode Naïve Bayes Gaussian dengan teknik oversampling SMOTE untuk mengatasi imbalance class. hasil dari penelitian ini yaitu Sebelum oversampling, model klasifikasi memiliki akurasi sebesar 84%, sedangkan setelah oversampling, akurasi meningkat menjadi 86%. Hal ini menunjukkan bahwa terdapat peningkatan besarnya nilai akurasi setelah penerapan oversampling menggunakan SMOTE sebesar 2%. Dapat disimpulkan bahwa penggunaan oversampling dengan SMOTE berhasil meningkatkan kemampuan model dalam mengklasifikasikan dengan benar jumlah tuple dalam data uji. Hal ini menunjukkan bahwa penggunaan SMOTE efektif dalam mengatasi masalah ketidakseimbangan kelas dan meningkatkan kinerja model dalam mengenali dan memprediksi sampel-sampel dari kelas minoritas.

Penelitian yang dilaukan oleh (Rahman & Mustikasari, 2024). Penelitian ini bertujuan untuk mengoptimalkan prediksi kelulusan mahasiswa

tepat waktu menggunakan metode Binning untuk mengelompokkan variabel ke dalam kategori diskrit dan Synthetic Minority Oversampling Technique (SMOTE) untuk mengatasi ketidakseimbangan kelas pada dataset. Hasil dari penelitian ini, pertama yaitu peningkatan performa dengan Teknik Binning dan SMOTE, implementasi teknik Binning dan Synthetic Minority Oversampling Technique (SMOTE) pada model Random Forest dan Decision Tree secara signifikan meningkatkan kinerja prediksi kelulusan mahasiswa tepat waktu. Penggunaan Binning membantu dalam menghasilkan prediksi yang lebih seimbang antara kelas mayoritas dan minoritas, sementara SMOTE berhasil mengatasi ketidakseimbangan kelas dengan menghasilkan data sintetis untuk kelas minoritas, kedua yaitu efektivitas model Random Forest dan Decision Tree di mana hasil menunjukkan bahwa model Random Forest memiliki performa yang baik dalam mengklasifikasikan kelas mayoritas. Namun, ketika digabungkan dengan teknik SMOTE, model ini mampu meningkatkan kemampuan dalam mengklasifikasikan kelas minoritas dengan tingkat Recall yang lebih baik. Model Decision Tree juga mengalami peningkatan yang signifikan dalam kinerja prediksi setelah penerapan teknik SMOTE; ketiga yaitu kombinasi optimal untuk prediksi kelulusan di mana hasil yang paling mengesankan terjadi ketika teknik Binning dan SMOTE digabungkan dengan model Random Forest dan Decision Tree. Kombinasi ini menghasilkan hasil yang sangat baik dalam hal precision, Recall, F1-score, dan akurasi keseluruhan.

Berdasarkan uraian latar belakang masalah dan penelitian terdahulu, maka peneliti akan mengklasifikasikan data numerik yaitu data loan. Dalam penelitian ini, akan menggunakan metode SMOTE untuk meningkatkan *recognition rate* pada kelas minoritas dengan algoritma Naive Bayes pada data numerik. Syntetic Minority Oversampling Technique (SMOTE) adalah metode yang digunakan untuk meningkatkan akurasi algoritma klasifikasi dengan cara meng-over sampling minoritas dalam dataset. Dengan demikian, algoritma dapat lebih baik dalam mengklasifikasikan data yang tidak seimbang. Pada penelitian ini peneliti juga menambahkan

penggunaan metode K-Fold Cross Validation. K-Fold Cross Validation berguna untuk mengevaluasi performa dari proses algoritma yang digunakan. Salah satu kelebihan yang didapatkan dari penggunaan K-Fold Cross Validation dalam pengujian adalah bisa mengetahui model dengan nilai akurasi paling baik dikarenakan sampel data dibagi secara random ke dalam K-partisi sehingga komposisi model yang paling baik dapat diketahui. Oleh karena itu, penelitian ini akan menggunakan dataset numerik yang memiliki distribusi tidak seimbang dan menguji akurasi algoritma Naive Bayes sebelum dan setelah penerapan metode SMOTE dan penggunaan K-Fold Cross Validation dalam klasifikasi pada data numerik data loan serta dapat menghasilkan model yang terbaik. Sehingga diharapkan dapat memberikan kontribusi dan manfaat.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang di atas maka rumusan masalah ini adalah:

1. Bagaimana kinerja algoritma Naïve Bayes dalam mengklasifikasikan data numerik yang tidak seimbang tanpa menggunakan metode SMOTE?
2. Bagaimana penerapaan metode SMOTE dapat menyeimbangkan distribusi kelas dalam data numerik?
3. Bagaimana peningkatan *recognition rate* dengan algoritma Naïve Bayes setelah menerapkan metode SMOTE pada data numerik?
4. Bagaimana penerapaan metode K-Fold Cross Validation dapat mengevaluasi performa dari proses algoritma?

1.3 Batasan Massalah

Agar penelitian ini tidak keluar dari pembahasan, maka pembahasan pada penelitian ini adalah sebagai berikut:

1. Menggunakan data loan yang diperoleh dari Kaggle data dari lending club.com, yang terdiri dari 9578 dataset yang meliputi kelas data dengan peminjam dapat menyelesaikan kredit sebanyak 8045 record dan

peminjam yang tidak dapat menyelesaikan kredit sebanyak 1533 record yang dimana menunjukkan adanya ketidakseimbangan kelas.

2. Proses pengolahan data hanya menggunakan algoritma Naïve Bayes dengan metode SMOTE dan K-Fold Cross Validation.
3. Menggunakan metode SMOTE untuk menangani ketidakseimbangan data.
4. Menggunaankan metode K-Fold Cross Validation untuk mengevaluasi performa dari proses algoritma.
5. Parameter yang digunakan dalam penelitian ini adalah nilai recall, precision dan F1-Score.
6. Pengolahan data menggunakan bahasa pemrograman Python.

1.4 Tujuan Penelitian

Tujuan penelitian ini adalah sebagai berikut:

1. Untuk mengetahui kinerja algoritma Naïve Bayes dalam mengklasifikasikan data numerik yang tidak seimbang tanpa menggunakan metode SMOTE.
2. Mengetahui hasil penerapan metode SMOTE dalam menyeimbangkan distribusi kelas pada data numerik.
3. Melihat meningkat atauakan tidak *recognition rate* dengan algoritma Naïve Bayes setelah menerapkan metode SMOTE pada data numeric.
4. Mengetahui hasil penerapan metode K-Fold Cross Validation dapat mengevaluasi performa dari proses algoritma.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut:

1. penelitian ini dihaarapkan dapat menambah wawasan dan ilmu pengetahuan secara teoritis maupun praktik.
2. Penelitian ini juga diharapkan dapat menjadi pengalaman dan pengetahuan serta rujukan tentang pengolahan data menggunakan algoritma Naïve Bayes dengan metode SMOTE dan K-Fold Cross Validation.

3. Memberikan pemahaman dalam mengetasi data yang tidak seimbang menggunakan metode SMOTE.

BAB II
TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Untuk dapat mengetahui keaslian dari penelitian ini, maka perlu adanya beberapa hasil kajian dari peneliti lain yang berkaitan dengan penelitian ini. Berikut ini beberapa penelitian yang sudah dilakukan sebelumnya diantaranya adalah:

Wang, S., Dai, Y., Shen, J., & Xuan, J. (2021). Penelitian tentang perluasan dan klasifikasi data tidak seimbang berdasarkan algoritma SMOTE" berfokus pada mengatasi tantangan yang ditimbulkan oleh dataset tidak seimbang dalam pembelajaran mesin. Para penulis mengusulkan Teknik Over-sampling Minoritas Sintetis (SMOTE) yang ditingkatkan untuk memperluas dan menyeimbangkan data. Hasil eksperimen menunjukkan bahwa algoritma SMOTE yang ditingkatkan secara efektif mengatasi ketidakseimbangan kelas dan meningkatkan akurasi klasifikasi. Temuan ini menyoroti pentingnya pra-pemrosesan data dalam mengembangkan model pembelajaran mesin yang kuat untuk data tidak seimbang. Dalam penelitian ini, dapat dilihat bahwa penelitian menggunakan SMOTE untuk mengatasi ketidakseimbangan data dalam model klasifikasi. Namun, mereka tidak menggabungkan SMOTE dengan Naive Bayes. Sedangkan pada penelitian yang akan dilakukan fokus pada penerapan Naive Bayes yang dikombinasikan dengan K-Fold Cross Validation pada data numerik. Hal ini memungkinkan untuk memperoleh hasil yang lebih kuat dan dapat dipercaya, karena validasi silang yang digunakan lebih mendalam dibandingkan dengan yang digunakan dalam penelitian sebelumnya.

Duan, F ., Zhang, S., Yan, Y., & Cai, Z, (2022). Penelitian ini berfokus pada pengembangan metode oversampling baru untuk mengatasi masalah data tidak seimbang dalam diagnosis kesalahan mekanis. Algoritma MeanRadius-SMOTE yang diusulkan menggunakan radius rata-rata untuk meningkatkan kualitas data sintetis yang dihasilkan. Penulis mengevaluasi kinerja metode ini melalui berbagai eksperimen dan membandingkannya dengan teknik oversampling yang ada. Hasil eksperimen menunjukkan bahwa

MeanRadius-SMOTE secara signifikan meningkatkan akurasi diagnosis kesalahan dibandingkan dengan metode oversampling tradisional. Temuan ini menunjukkan potensi besar dari pendekatan baru ini dalam meningkatkan keandalan diagnosis kesalahan mekanis pada data tidak seimbang. Penelitian ini, mengembangkan teknik SMOTE baru yang disebut MeanRadius-SMOTE untuk meningkatkan kualitas data sintetis pada ketidakseimbangan kelas. Sedangkan penelitian yang akan dilakuakan berbeda karena menggunakan SMOTE yang digabungkan dengan Naive Bayes dan menggunakan K-Fold Cross Validation. Fokus pada penerapan Naive Bayes untuk data numerik, sementara penelitian ini lebih menekankan pada pengembangan metode SMOTE yang baru.

Hairani, H., Anggrawan, A., & Priyanto, D. (2023). Penelitian ini bertujuan untuk menerapkan metode Smote-Tomeklink dan Random Forest dalam klasifikasi diabetes. Data yang digunakan adalah data diabetes yang diperoleh dari Kaggle sebanyak 768 data dengan delapan atribut input dan 1 output atribut sebagai kelas, pra-pemrosesan data digunakan untuk menyeimbangkan dataset dengan Smote-Tomeklink, klasifikasi menggunakan random forest metode, dan evaluasi kinerja berdasarkan akurasi, sensitivitas, presisi, dan skor F1. Algoritma Random Forest dengan Smote-Tomeklink memiliki 86,4% akurasi, sensitivitas 88,2%, presisi 82,3%, dan skor F1 85,1%. Dengan demikian, penggunaan Smote-Tomeklink dapat meningkatkan kinerja metode hutan acak berdasarkan akurasi, sensitivitas, presisi, dan skor F1. Perbedaan dengan penelitian yang akan dilakuakan terletak pada metode dan data yang digunakan. Penelitian ini menggunakan Smote-TomekLink dan Random Forest untuk mengklasifikasikan data diabetes. Sedangkan penelitian selanjutnya menggunakan SMOTE dengan Naive Bayes dan menggunakan K-Fold Cross Validation. Menggunakan data loan yang diperoleh dari Kaggle data dari lending club.com, yang terdiri dari 9578 dataset yang meliputi kelas data dengan peminjam dapat menyelesaikan kredit sebanyak 8045 record dan peminjam yang tidak dapat menyelesaikan kredit sebanyak 1533 record yang dimana menunjukkan adanya ketidakseimbangan kelas.

Setiawan, Erlansari, & Sari, (2023). Metode yang digunakan adalah Naïve Bayes berbasis SMOTE & PSO. pengguna pada variabel metode pembayaran DANA masih "Ditolak" oleh pengguna dengan nilai probabilitas

penerimaan yakni P(C) untuk P(Diterima) = 0,32 < P(Ditolak) = 0,68. Sedangkan pada variabel kualitas aplikasi sudah "Diterima" oleh pengguna dengan nilai probabilitas penerimaan yakni P(C) untuk P(Diterima) = 0,585 > P(Ditolak) = 0,415. Hasil pengujian dengan akurasi tertinggi yakni pada variabel metode pembayaran DANA sebesar 93,68 % dan variabel kualitas aplikasi sebesar 96,13 % serta berdasarkan penelitian yang telah dilakukan terjadinya perubahan signifikan apabila menambahkan Feature SMOTE dan PSO yakni terjadinya peningkatan akurasi tertinggi yaitu sebesar 14.94 % untuk variabel metode pembayaran DANA dan 9.06 % untuk variabel kualitas aplikasi. Sedangkan dalam penelitian yang akan dilakuakan tidak menggunakan PSO, melainkan fokus pada K-Fold Cross Validation untuk menguji model Naive Bayes.

Kurniasih & Isyara, (2023). Penelitian ini bertujuan untuk mengklasifikasikan kasus mahasiswa yang akan di drop out atau tidak menggunakan metode Naïve Bayes Gaussian dengan teknik oversampling SMOTE untuk mengatasi imbalance class. hasil dari penelitian ini yaitu Sebelum oversampling, model klasifikasi memiliki akurasi sebesar 84%, sedangkan setelah oversampling, akurasi meningkat menjadi 86%. Hal ini menunjukkan bahwa terdapat peningkatan besarnya nilai akurasi setelah penerapan oversampling menggunakan SMOTE sebesar 2%. Dapat disimpulkan bahwa penggunaan oversampling dengan SMOTE berhasil meningkatkan kemampuan model dalam mengklasifikasikan dengan benar jumlah tuple dalam data uji. Hal ini menunjukkan bahwa penggunaan SMOTE efektif dalam mengatasi masalah ketidakseimbangan kelas dan meningkatkan kinerja model dalam mengenali dan memprediksi sampel-sampel dari kelas minoritas. Sedangkan penelitian yang akan dilakuakn berbeda karena menggunakan K-Fold Cross Validation, yang memberikan hasil evaluasi yang lebih mendalam dan lebih dapat diandalkan dibandingkan dengan penelitian ini yang tidak menjelaskan penggunaan teknik validasi tersebut.

Rahman & Mustikasari, (2024). Penelitian ini bertujuan untuk mengoptimalkan prediksi kelulusan mahasiswa tepat waktu menggunakan metode Binning untuk mengelompokkan variabel ke dalam kategori diskrit dan Synthetic Minority Oversampling Technique (SMOTE) untuk mengatasi ketidakseimbangan kelas pada dataset. Hasil dari penelitian ini, pertama yaitu peningkatan performa dengan Teknik Binning dan SMOTE, implementasi

teknik Binning dan Synthetic Minority Oversampling Technique (SMOTE) pada model Random Forest dan Decision Tree secara signifikan meningkatkan kinerja prediksi kelulusan mahasiswa tepat waktu. Meskipun pada penelitian ini juga menggunakan SMOTE, namun pada penelitian yang akan dilakukan lebih fokus pada penggunaan Naive Bayes dengan K-Fold Cross Validation pada data numerik. Dengan pendekatan ini, diharapkan dapat memberikan evaluasi yang lebih kuat terhadap performa Naive Bayes, sementara penelitian sebelumnya lebih fokus pada Random Forest dan Decision Tree.

Dari penelitian di atas, dapat dilihat adanya perbedaan mendasar dengan penelitian yang akan dilakukan, terutama dalam konteks, jenis data, dan kombinasi metode yang digunakan. Penelitian sebelumnya banyak menggunakan data yang berasal dari berbagai bidang, seperti data ulasan pengguna atau data karakteristik kepribadian, yang memiliki sifat berbeda dengan data yang akan saya gunakan. Misalnya, penelitian-yang dilakukan Setiawan, Erlansari, & Sari (2023) lebih banyak menggunakan data yang memiliki atribut tekstual atau kategorikal, di mana SMOTE digunakan untuk mengatasi ketidakseimbangan kelas di dalam konteks tersebut. Namun, dalam penelitian ini, saya bermaksud untuk melihat pengaruh SMOTE pada data numerik, yang memiliki karakteristik berbeda, seperti data pinjaman atau loan data yang digunakan dalam penelitian ini.

Selain itu, penelitian sebelumnya banyak yang menggunakan pembagian statis (seperti pembagian data menjadi set pelatihan dan pengujian) dalam mengevaluasi model, tanpa memanfaatkan teknik validasi silang yang lebih kuat, seperti K-Fold Cross Validation. Hal ini dapat mempengaruhi ketepatan evaluasi kinerja model, terutama dalam mengatasi ketidakseimbangan kelas. Dalam penelitian ini, akan menggunakan K-Fold Cross Validation untuk mengevaluasi performa model Naïve Bayes yang diterapkan pada data yang telah diolah dengan SMOTE, dengan harapan mendapatkan hasil yang lebih akurat.

2.2 Landasan Teori

2.2.1 Data Mining

Data mining adalah proses ekstraksi, analisis, dan visualisasi dari data yang telah dikumpulkan untuk menemukan pola, struktur, dan informasi yang berguna dengan memilah-milah data dalam jumlah besar yang disimpan di dalam repositori, menggunakan teknologi pengenalan

pola serta teknik statistik dan matematika (Nabila, 2021). Tujuan data mining adalah untuk menemukan pola, prediksi, dan penjelasan dari data yang telah dikumpulkan, serta untuk meningkatkan efisiensi dan kualitas dalam pengambilan keputusan (Urva, 2023). Proses data mining terdiri dari serangkaian fase atau langkah yang dilakukan untuk mengekstrasi informasi atau pengetahuan berguna dari sejumlah besar data. Proses penambahan data ini bersifat iteratif, dan setiap fase dapat diulangi dan disesuaikan berdasarkan hasil evaluasi maupun baik dari pengguna dan pengambil keputusan (Haryanti, 2024).

Data Mining yang disebut juga dengan *Knowledge Discovery in Database (KDD)* adalah suatu proses secara otomatis atas pencarian data di dalam sebuah memori yang amat besar dari data untuk mengetahui pola dengan menggunakan alat seperti klasifikasi, hubungan (*association*) atau pengelompokkan (*clustering*) (Pambudi, 2023). Data mining memiliki komponen-komponen utama dalam data mining antara lain (Alfarizi, 2021) :

1. *Database dan data warehouse, World Wide Web,* atau tempat penyimpanan informasi lainnya, dapat berbentuk satu ataupun banyak *database, spreadsheet*, data *warehouse* maupun tempat penyimpanan informasi lainnya. Data *cleaning*, data selection dan data *integration* dapat dijalankan pada data tersebut.
2. *Database* dan data *warehouse server*, pada bagian ini memliki tanggung jawab untuk hal pengambilan data yang memiliki sifat relevan, didasarkan permintaan pengguna.
3. *Knowledge Based*, bagian ini merupakan *domain knowledge* yang dipergunakan untuk memberi arahan terhadap pencarian atau melakukan evaluasi pola-pola yang didapatkan. informasi tersebut meliput hirarki konsep yang dipergunakan untuk mengorganisasikan atribut atau nilai atribut kedalam level abstraksi yang berbeda. informasi tersebut juga dapat berupa kepercayaan pengguna (*user belief)*, yang dapat dipergunakan dalam menentukan pola menarik yang didapatkan.
4. *Data mining engine*, merupakan suatu bagian yang penting dalam arsitektur sistem data mining. Bagian ini terdiri atas modul-modul fungsional seperti asosiasi, analisis cluster, karakterisasi dan klasifikasi.

5. *Ghrapical user interface (GUI).* Modul ini berinteraksi dengan pengguna (*user*) dan data mining melalui komponen ini, pengguna dapat berkomunikasi dengan sistem menggunakan query.

2.2.2 Klasifikasi

Klasifikasi (*Classification*) merupakan proses untuk menemukan sekumpulan model yang menjelaskan dan membedakan kelas-kelas data, sehingga model tersebut dapat digunakan untuk memprediksi nilai suatu kelas yang belum diketahui pada sebuah objek. Untuk mendapatkan model, kita harus melakukan analisis terhadap data latih (training set) (Pambudi, 2023). Klasifikasi merupakan proses yang terdiri dari dua tahap, yaitu tahap pembelajaran dan tahap pengklasifikasian. Pada tahap pembelajaran, sebuah algoritma klasifikasi akan membangun sebuah model klasifikasi dengan cara menganalisis training data (Susana, 2022). Dalam data mining, klasifikasi digunakan untuk menemukan model dari data yang belum terklasifikasi, sehingga dapat digunakan untuk mengklasifikasi data baru.

2.2.3 Algoritma Naive Bayes

Naïve bayes adalah sebuah algoritma supervised learning berdasarkan teorema Bayes yang digunakan untuk memecahkan masalah klasifikasi dengan mengikuti pendekatan probabilistik. Naïve bayes dikemukakan oleh ilmuwan inggris Thomas Bayes, yaitu memprediksi peluang dimasa depan berdasarkan pengalaman sebelumnya sehingga dikenal sebagai Teorema Bayes (Sobri, 2023). Klasifikasi Naive Bayes diasumsikan bahwa ada atau tidak ciri tertentu dari sebuah kelas tidak ada hubungannya dengan ciri dari kelas lainnya. Naive Bayes berpotensi baik untuk mengklasifikasikan data karena kesederhanaannya. Naive Bayes dirancang untuk dipergunakan dengan asumsi bahwa antar satu kelas dengan kelas yang lain tidak saling bergantung (independen). Pada klasifikasi Naive Bayes, proses pembelajaran lebih ditekankan pada mengestimasikan probabilitas. Keuntungan dari pendekatan Naive Bayes adalah pengklasifikasian akan mendapatkan nilai error yang lebih kecil

ketika data set berjumlah besar (Weni, 2022). Klasifikasi Naive Bayes terbukti memiliki akurasi dan kecepatan yang tinggi saat dipublikasikan kedalam basis data dengan jumlah yang besar.

Persamaan dari Teorema Bayes adalah :

$$P(H)|X) = \frac{P(H|X).P(H)}{P(X)} \quad (2.1)$$

Keterangan :

X : Data dengan class yang belum diketahui

H : Hipotesis data X merupakan suatu class spesifik

$P(H)|X$: Probabilitas hipotesis H berdasar kondisi (*posteriori probability*)

$P(H)$: Probabilitas hipotesis H (prior probability)

$P(X|H)$: Probabilitas X berdasarkan kondisi pada hipotesis

$P(X)$: Probabilitas X

Kaitan antara Naive Bayes dengan klasifikasi, korelasi hipotesis, dan bukti dengan klasifkasi adalah bahwa hipotesis dalam teorema Bayes merupakan label kelas yang menjadi target pemetaan dalam klasifikasi, sedangkan bukti merupakan fitur-fitur yang menjadi masukan dalam model klasifikasi. Jika X adalah vektor masukan yang berisi fitur dan Y adalah label kelas. Naive Bayes dituliskan dengan P(Y|X). Notasi tersebut berarti probabilitas label kelas Y didapakan setelah fitur-fitur X diamati. Notasi ini disebut juga probabilitas akhir(posterior probability) untuk Y, sedangkan P(Y) disebut probabilitas akhir (prior probability) Y. Selama proses pelatihan harus dilakukan pembelajaran probalilitas akhir (P(Y|X) pada model untuk setiap kombinasi X dan Y berdasarkan informasi yang didapat dari data latih. Dengan membangun model tersebut, suatu data uji X' dapat diklasifikasikan dengan mencari nilai Y' dengan memaksimalkan nilai P(Y'|X') yang didapat (Pikir Claudia, 2021).

2.2.4 Imbalance Class

Ketidakseimbangan kelas (*imbalance class*) mengacu pada kondisi di mana jumlah sampel antara kelas-kelas yang berbeda dalam dataset tidak

seimbang atau tidak proporsional. Ini berarti ada satu atau beberapa kelas yang memiliki jumlah sampel yang jauh lebih sedikit atau jauh lebih banyak daripada kelas lainnya. Masalah ketidakseimbangan kelas dapat mempengaruhi kinerja model pembelajaran mesin. Beberapa dampak yang mungkin terjadi adalah model dapat cenderung memprediksi secara dominan kelas mayoritas karena penyebaran yang tidak seimbang. Ini mengakibatkan kinerja model yang buruk dalam mengidentifikasi sampel dari kelas minoritas. (Isyara, 2023).

2.2.5 SMOTE (*Synthetic Minority Over-sampling Technique*)

SMOTE (*Synthetic Minority Over-sampling Technique*) ialah salah satu metode yang banyak digunakan untuk mengatasi ketidakseimbangan kelas dalam dataset. Teknik ini berkerja dengan cara mengambil sampel data baru. Jumlah data sampel yang diambil menyesuaikan dengan jumlah data minoritas (Kurniadi, 2023). Jika ketidakseimbangan kelas tersebut tidak ditangani atau diabaikan maka dapat menyebabkan model memiliki bias yang sangat signifikan terhadap kelas mayoritas. Hal ini dapat mengakibatkan model yang tidak peka terhadap kasus-kasus dalam kelas minoritas yang mungkin memiliki nilai prediktif yang penting. Akurasi model dapat sangat tinggi karena dominasi kelas mayoritas, tetapi hasil prediksi pada kelas minoritas akan menjadi sangat rendah.

SMOTE (*Synthetic Minority Over-sampling Technique* bertujuan untuk meningkatkan jumlah sampel dalam kelas minoritas dengan menciptakan sampel sintetis berdasarkan data yang ada (Pulungan, 2023). Teknik ini bekerja dengan cara membuat data sintetis atau data buatan berdasarkan pengukuran kedekatan data numerik dengan jarak Euclidean, sedangkan data klasifikasinya lebih sederhana yaitu nilai modus. Berikut persamaan yang digunakan:

$$X_{syn} = X_i + (X_{knn} - X_i) \times \delta \quad (2.2)$$

Dimana, Xsyn adalah data sintesis yang akan diciptakan, X_{knn} merupakan jarak terdekat dari data yang dibuat sintesisnya, X_i adalah data dengan atribut ke-i dan δ nilai random antara 0 dan 1 (Aziziah, 2022).

2.2.6 K-Fold Cross Validation

Suatu metode dalam data mining yang dipakai untuk mendapatkan nilai akurasi paling baik ketika data dibagi menjadi data uji dan data latih disebut *cross validation*. *Cross Validation* merupakan Teknik yang dilakukan untuk dapat menilai atau dapat memvalidasi tingkat keakuratan dari sebuah model berdasarkan dataset tertentu yang mana model tersebut digunakan untuk melakukan prediksi atau klasifikasi. Dalam metode K-Fold Cross Validation, data dibagi menjadi K bagian setelah dilakukan proses pembobotan term sebelumnya. Percobaan akan dilakukan sebanyak K kali dengan menggunakan satu bagian sebagai data uji (CA, 2022).

K-fold Cross Validation dapat memberikan solusi terhadap masalah akurasi yang berbeda saat menggunakan set tes yang berbeda pada waktu evaluasi kinerja model. Dengan menggunakan K-fold Cross Validation data akan dibagi kedalam K bagian / fold dan dari setiap fold yang ada akan digunakan sebagai set pengujian (Fathoni, 2024). Pada metode K-Fold Cross Validation, dataset yang digunakan akan dibagi menjadi beberapa partisi secara random. Setelah dibagi ke dalam beberapa partisi maka data tersebut akan diolah sebanyak K kali percobaan dengan setiap K kali percobaan, data testing yang digunakan adalah data partisi ke-K dan sisa partisi yang lain digunakan sebagai data training (Nirainun, 2023).

2.3 Keaslian Penelitian

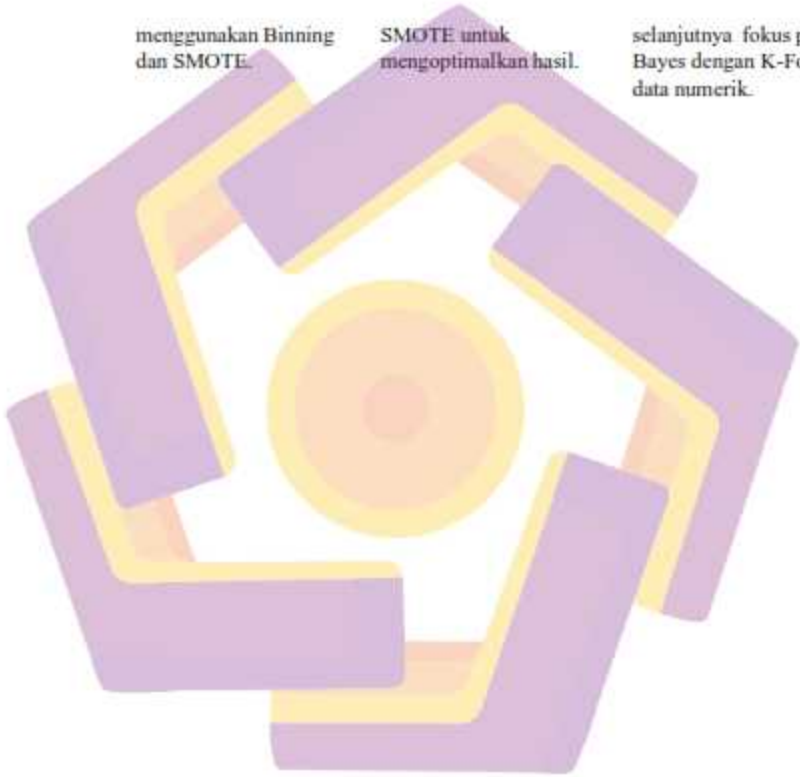
Tabel 2.1 Matriks Literatur Review dan Posisi Penelitian

Peningkatan akurasi algoritma naive bayes dengan metode Syntetic Minority Oversampling Technique (SMOTE) pada data numerik)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Saran atau Kelemahan	Perbandingan
1	Research on expansion and classification of imbalanced data based on SMOTE algorithm.	Wang, S., Dai, Y., Shen, J., & Xuan, J. (2021)	Pengembangan metode oversampling baru untuk mengatasi masalah data tidak seimbang dalam diagnosis kesalahan mekanis.	Disarankan untuk menguji metode yang diusulkan pada berbagai jenis dataset untuk memverifikasi efektivitas	Pada penelitian sebelumnya menggunakan peningkatan pada algoritma SMOTE untuk menyeimbangkan data yang tidak seimbang dan meningkatkan akurasi klasifikasi. Sedangkan pada penelitian menggunakan SMOTE dan K-Fold Cross Validation, memungkinkan untuk memperoleh hasil yang lebih kuat dan dapat dipercaya.
2	An oversampling method of unbalanced data for mechanical fault diagnosis based on MeanRadius-SMOTE	Duan, F., Zhang, S., Yan, Y., & Cai, Z. (2022).	Untuk mengembangkan metode oversampling baru yang disebut MeanRadius-SMOTE	Disarankan untuk menguji metode MeanRadius-SMOTE pada berbagai dataset dengan karakteristik yang berbeda untuk memastikan generalisasi hasil penelitian ini.	Pada penelitian sebelumnya menggunakan MeanRadius-SMOTE, yang merupakan variasi dari SMOTE, dengan tujuan utama adalah mengembangkan metode oversampling baru. Sedangkan pada penelitian menggunakan SMOTE dan K-Fold Cross Validation, untuk menjamin angka akurasi yang diperoleh stabil dan meningkatkan generalisasi.
3	Improvement performance of the random forest method on unbalanced diabetes	Hairani, H., Anggrawan, A., & Priyanto, D.	untuk menerapkan metode Smote-Tomeklink dan Random	Daoat melakuakn perbandingan dengan	Penelitian sebelumnya menggunakan metode Smote-Tomeklink dan Random Forest dalam klasifikasi diabetes. Sedangkan penelitian

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Saran atau Kelemahan	Perbandingan
				teknik oversampling lainnya untuk	
	data classification using Smote-Tomek Link	(2023)	Forest dalam klasifikasi diabetes.	memberikan pemahaman yang lebih tentang keunggulan SMOTE-Tomek Link.	selanjutnya menggunakan SMOTE dengan Naive Bayes dan menggunakan K-Fold Cross Validation.
4	Penerapan Data Mining pada Review TIX ID Menggunakan Naïve Bayes Berbasis SMOTE & PSO	Setiawan, Erlansari, & Sari (2023)	Mengetahui akurasi review pengguna aplikasi TIX ID dengan Naive Bayes berbasis SMOTE dan PSO.	Kompleksitas model meningkat, evaluasi terbatas pada akurasi, dan perlu validasi lebih kuat.	Penelitian sebelumnya menggunakan kombinasi metode (SMOTE dan PSO). Sedangkan penelitian selanjutnya menggunakan SMOTE dan K-Fold Cross Validation, untuk menguji model Naive Bayes.
5	Penggunaan Metode SMOTE pada Naïve Bayes Gaussian untuk Klasifikasi Mahasiswa Drop Out	Kurniasih & Isyara (2023)	Klasifikasi kasus mahasiswa yang akan drop out menggunakan Naive Bayes Gaussian dengan SMOTE.	Peningkatan akurasi kecil (2%), perlu metrik evaluasi tambahan, dan fokus hanya pada akurasi.	Pada penelitian sebelumnya hanya menggunakan Naive Bayes dengan SMOTE yang menunjukkan peningkatan akurasi yang relatif kecil dengan penggunaan SMOTE. Sedangkan penelitian selanjutnya menggunakan SMOTE dengan Naive Bayes dan menggunakan K-Fold Cross Validation.
6	OPTIMALISASI PREDIKSI KELULUSAN	Rahman & Mustikasari (2024)	Mengoptimalkan prediksi kelulusan mahasiswa tepat waktu	Lakukan tuning parameter lebih lanjut pada teknik Binning dan	Penelitian sebelumnya menggunakan metode Random Forest dan Decision Tree dengan Binning dan SMOTE. Sedangkan penelitian

MAHASISWA TEPAT WAKTU MENGGUNAKAN BINNING DAN SYNTHETIC MINORITY OVERSAMPLING TECHNIQUE (SMOTE)	menggunakan Binning dan SMOTE.	SMOTE untuk mengoptimalkan hasil.	selanjutnya fokus pada penggunaan Naive Bayes dengan K-Fold Cross Validation pada data numerik.



Dari penelitian sebelumnya dapat ditarik kesimpulan bahwa terdapat perbedaan secara mendasar dengan penelitian yang akan dilakukan oleh peneliti adalah berbeda dalam konteks, jenis data, dan kombinasi metode yang digunakan. Penelitian sebelumnya menggunakan berbagai jenis data, seperti data ulasan pengguna atau data karakteristik kepribadian, yang mungkin memiliki sifat berbeda dari data numerik yang akan digunakan yaitu data loan. Penelitian ini menggunakan metode SMOTE dan Fold Cross Validation yang bertujuan untuk meningkatkan akurasi model Naive Bayes. Penggunaan metode SMOTE digunakan untuk mengatasi ketidakseimbangan kelas pada klasifikasi. Sedangkan metode K-Fold Cross Validation berguna untuk mengevaluasi performa dari proses suatu algoritma. Salah satu kelebihan yang didapatkan dari penggunaan K-Fold Cross Validation dalam pengujian adalah bisa mengetahui model dengan nilai akurasi paling baik dikarenakan sampel data dibagi secara random ke dalam K-partisi sehingga komposisi model yang paling baik dapat diketahui.

Penelitian ini dilakukan dengan tujuan untuk menerapkan algoritma Naïve Bayes Classifier dan K-Fold Cross Validation dalam klasifikasi data loan serta dapat menghasilkan model yang terbaik. Oleh karena itu, dengan pendekatan yang lebih terfokus pada jenis data dan penggunaan metode yang lebih komprehensif, penelitian ini tidak hanya mengatasi kekurangan dari penelitian sebelumnya tetapi juga diharapkan memberikan peningkatan yang signifikan dalam hasil klasifikasi.

BAB III
METODE PENELITIAN

3.1 Jenis, Sifat, dan Pendekatan Penelitian

1. Jenis Penelitian

Penelitian ini merupakan penelitian terapan yang bertujuan untuk meningkatkan akurasi dari sebuah algoritma. Penelitian ini juga merupakan penelitian kuantitatif, karena penelitian yang dilakukan adalah proses perhitungan matematis untuk mendapatkan dan menemukan hasil yang diinginkan. Dalam penelitian ini dilakukan peningkatan akurasi algoritma Naïve Bayes dengan metode SMOTE untuk klasifikasi data numerik.

Sifat penelitian yang dilakukan adalah evaluasi, kaarena penelitian ini melakukan peningkatan akurasi terhadap algoritma dengan menggunakan sebuah metode yang sudah ditentukan. Pendekatan pada penelitian ini adalah pendekatan kuantitatif dimana peneliti akan melakukan penelitian sesuai dengan tahap-tahap atau alur penelitian yang telah dibuat.

3.2 Metode Pengumpulan Data

Penelitian ini menggunakan data numerik. Data numerik yang digunakan adalah data loan. Data ini digunakan karena memuat variabel yang cocok untuk algoritma Naïve Bayes. Hal ini juga sesuai dengan fungsi metode yang digunakan yaitu metode SMOTE karena pada data ini terdapat ketidakseimbangan distribusi kelas.

Penelitian ini menggunakan data loan yang merupakan data yang diperoleh dari Kaggle yakni data dari lendingclub.com. LendingClub.com adalah platform pinjaman *peer-to-peer* yang menghubungkan peminjam yang membutuhkan dana dengan investor yang memiliki dana untuk diinvestasikan. Data yang diperoleh untuk penelitian ini berjumlah 9578 dataset yang meliputi kelas data dengan peminjam dapat menyelesaikan kredit sebanyak 8045 record dan peminjam yang tidak dapat menyelesaikan kredit sebanyak 1533 record yang dimana menunjukkan adanya

ketidakseimbangan kelas. Data yang digunakan dari tahun 2007–2010 yang terdiri dari 14 variabel dimana terdapat 12 variabel yang akan digunakan, satu diantaranya merupakan variabel target. Variabel target merupakan variabel yang nilainya akan dimodelkan dan diprediksi oleh variabel lainnya. Variabel-variabel tersebut diantaranya sebagai berikut:

1. credit.policy (Kebijakan Kredit)

   Variabel ini menggambarkan kelayakan kredit dari nasabah, dengan nilai 1 menunjukkan bahwa peminjam memenuhi kriteria kredit yang ditetapkan oleh LendingClub, sedangkan nilai 0 menunjukkan sebaliknya. Variabel ini merupakan indikator penting mengenai kelayakan nasabah.
2. Purpose (Tujuan Pinjaman)

   Variabel ini menunjukkan tujuan pinjaman, dengan kategori seperti *credit_card, debt_consolidatin, educational, major_purchase, small_business, dan all_other.* Tujuan pinjaman dapat memengaruhi risiko dan kecenderungan peminjam untuk membayar kembali.
3. *Int.rate/interest rate* (Tingkat Bunga)

   Variabel ini menggambarkan tingkat bunga pinjaman yang diperoleh oleh nasabah. Peminjam yang dinilai lebih berisiko akan gagal bayar biasanya dikenakan bunga yang lebih tiggi.
4. Installment (Angsuran)

   Variabel ini menggambarkan jumlah angsuran mingguan, bulanan atau tahunan yang harus dibayar oleh nasabah jika pinjamannya disetujui.
5. Log.annual.inc

   Variabel ini merefresentasikan pendapatan tahunan yang telah di log. kan dengan tujuan agar data pendapatan tahunan menjadi lebih merata. Menggunakan logaritma membantu dalam menormalkan data pendapatan yang mungkin sangat bervariasi.

6. DTI (*Debt to Income*)

   Variabel ini menunjukkan proporsi pendapatan bulanna yang dialokasiakan untuk membayar hutnag. Variabel ini digunakan untuk mengukur kemampuan seseorang dalam memenuhi kewajiban

kreditnya dan sering dijadikan salah satu indikator kelayakan kredit.

7. FICO

   FICO score adalah angka (biasanya antara 300–850) yang mencerminkan seberapa besar kemungkinan seseorang mampu membayar pinjaman tepat waktu. Semakin tinggi skor FICO, semakin rendah risiko seseorang gagal bayar, dan semakin besar kemungkinan disetujui untuk pinjaman atau kartu kredit.

8. Days.With.Cr.Line

   Variabel *Days.With.Cr.Line* menunjukkan berapa lama peminjam telah memiliki akun kredit. Riwayat kredit yang panjang umumnya dihubungkan dengan tingkat risiko gagal bayar yang lebih rendah karena mencerminkan kestabilan dan pengalaman dalam mengelola kredit.

9. Revol.bal (*revolving balance*)

   Variabel revol.bal merepresentasikan jumlah pinjaman aktif (terutama kartu kredit atau sejenisnya) yang dimiliki oleh peminjam. Nilai ini digunakan untuk menilai beban hutang yang sedang ditanggung oleh peminjam pada saat pengajuan pinjaman

10. Revol.uti

    Persentase dari batas maksimum (limit) kredit yang sudah digunakan oleh nasabah.

11. Inq.last.6mths

    Variabel ini menunjukkan frekuensi permintaan atau pengajuan kredit yang dilakukan oleh peminjam dalam 6 bulan terakhir. Nilai ini digunakan untuk menilai potensi risiko, karena terlalu banyak pengajuan kredit dalam waktu singkat dapat mengindikasikan ketidakstabilan finansial.

12. Delinq.2yrs

    Variabel ini menggambarkan jumlah kejadian keterlambatan pembayaran lebih dari 30 hari yang dilakukan oleh peminjam dalam dua tahun terakhir. Nilai ini menjadi indikator penting dalam menilai risiko peminjam, karena riwayat keterlambatan sering dikaitkan dengan kemungkinan gagal bayar di masa mendatang.

13. Pub.rec/*Public records*

    Variabel ini menunjukkan jumlah catatan publik negatif terkait masalah keuangan peminjam, seperti kebangkrutan atau keputusan pengadilan terkait utang. Variabel ini menjadi indikator penting karena keberadaan catatan publik menunjukkan risiko gagal bayar yang

signifikan.

14. Not fully paid

    Digunakan sebagai variabel target dalam pengklasifiasian. Misalnya, model dilatih untuk memprediksi kemungkinan peminjam berakhir dengan status *"fully paid"*(lunas) atau *"not fully paid"* (belum lunas) berdasarkan variabel yang ada dalam dataset.

3.3 Metode Analisis Data

Metode analisis data pada penelitian ini adalah kuantitatif karena sesuai dengan penelitian yang dilakukan. Penelitian ini menggunakan beberapa teknik dan alat untuk meningkatkan akurasi algoritma *Naive Bayes* dengan metode *Synthetic Minority Oversampling Technique* (SMOTE) dan *K-Fold Cross Validation*. Sofware yang digunakan adalah Python. Python adalah bahasa pemrograman tingkat tinggi yang bersifat *interpreted*, *general-purpose*, dan *open-source*, yang pertama kali dikembangkan oleh Guido van Rossum dan dirilis secara resmi pada tahun 1991.

Dalam konteks penelitian ilmiah, Python banyak digunakan karena sifatnya yang fleksibel dan kemampuannya dalam menangani berbagai macam kebutuhan komputasi. Python menyediakan berbagai pustaka (*libraries*) dan kerangka kerja (*frameworks*) yang mendukung aktivitas penelitian seperti analisis data, visualisasi, pembelajaran mesin (*machine learning*), pemrosesan bahasa alami (*natural language processing*), dan pengolahan citra digital. Beberapa libraries python yang digunakan antara lain adalah:

a. *import pandas as pd*
b. *import train_test_split*
c. *import LabelEncoder*
d. *import GaussianNB*
e. *import classification_report, confusion_matrix*
f. *import numpy as np*
g. *import matplotlib.pyplot as plt*
h. *import seaborn as sns*

Adapun metode analisis data yang dilakukan dalam penelitian ini adalah:

1) Pengumpulan Data

   Pada tahap ini mengumpulkan data loan yang berjumlah 9578 dataset yang meliputi kelas data dengan peminjam dapat menyelesaikan kredit sebanyak 8045 record dan peminjam yang tidak dapat menyelesaikan

kredit sebanyak 1533 record yang dimana menunjukkan adanya ketidakseimbangan kelas. Data yang digunakan dari tahun 2007–2010. Data diperoleh dari Kaggle yakni data dari lendingclub.com.

2) *Preprosessing* Data

Preprosessing data untuk membersihkan dan mempersiapkan data agar siap digunakan dalam model machine learning. Proses ini mencakup pengubahan tipe data yang tidak sesuai dengan kebutuhan analisis dan normalisasi data.

3) *Synthetic Minority Oversampling Technique* (SMOTE)

Menerapkan *Synthetic Minority Oversampling Technique* (SMOTE) pada data latih untuk menyeimbangkan jumlah sampel antara kelas mayoritas dan minoritas. SMOTE menghasilkan sampel sintetis dari kelas minoritas dengan melakukan interpolasi antara sampel-sampel minoritas yang ada. Ini dilakukan untuk mengatasi masalah ketidakseimbangan kelas yang dapat mempengaruhi performa model.

4) Klasifikasi Model

Algoritma yang digunakan dalam penelitian ini yaitu Naive Bayes. Naive Bayes adalah algoritma klasifikasi yang didasarkan pada Teorema Bayes dan asumsi independensi antar fitur. Teorema Bayes sendiri adalah rumus matematika yang digunakan untuk menghitung probabilitas suatu kejadian berdasarkan informasi atau pengetahuan yang ada sebelumnya.

5) *K-Fold Cross Validation*

K-Fold Cross Validation adalah teknik validasi yang digunakan untuk mengevaluasi performa model dengan membagi data menjadi k subset atau "folds" dan menjalankan iterasi pelatihan dan pengujian sebanyak k kali.

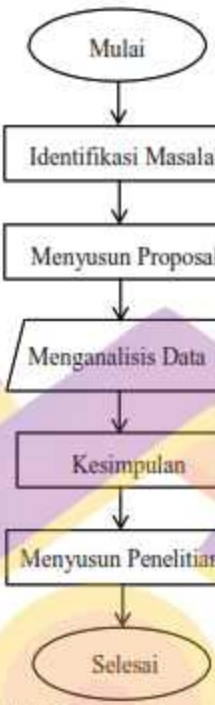
6) Evaluasi

Evaluasi dilakuakn dengan tujuan mengukur kinerja model klasifikasi yang telah dilakuakan. Tahapan ini untuk mendapatkan hasil akurassi yang terbaik dari algoritma yang digunakan yaitu algoritma Naïve Bayes.

3.4 Alur Penelitian

Alur penelitian merupakan penjelasan secara lengkap dan terperinci tentang langkah-langkah yang dilakukan dalam melakukan penelitian. Alur penelitian ini disajikan dalam bentuk diagram alir, berikut adalah alur penyusunan penelitian dan proses analisis penelitian yang dilakukan

3.4.1. Diagram alir penyusunan penelitian



Gambar 3.4.1.1. Diagram alir penyusunan penelitian

Pada Gamabar 3.4.1.1. dijelaskan mengenai langkah-langkah yang dilakukan dalam melakukan proses penyusunan penelitian ini :

1) Mulai

Tahap awal memulai proses penelitian, yang dimulai dari rasa ingin tahu atau permasalahan teknis yang ditemukan, seperti rendahnya akurasi klasifikasi akibat ketidakseimbangan data

2) Identifikasi Masalah

Mengidentifikasi bahwa data numerik yang digunakan memiliki distribusi kelas yang tidak seimbang (misalnya, jumlah data kelas minoritas jauh lebih sedikit), sehingga memengaruhi performa algoritma klasifikasi seperti Naive Bayes.

3) Menyusun Proposal

Menyusun proposal penelitian yang detail mengenai bagaimana rencana penelitian yang berisi latar belakang, tujuan, metode serta tinjauan pustaka.

4) Mengumpulkan Data

Mengumpulkan data yang relevan yakni yang memiliki fitur numerik dan kelas target tidak seimbang

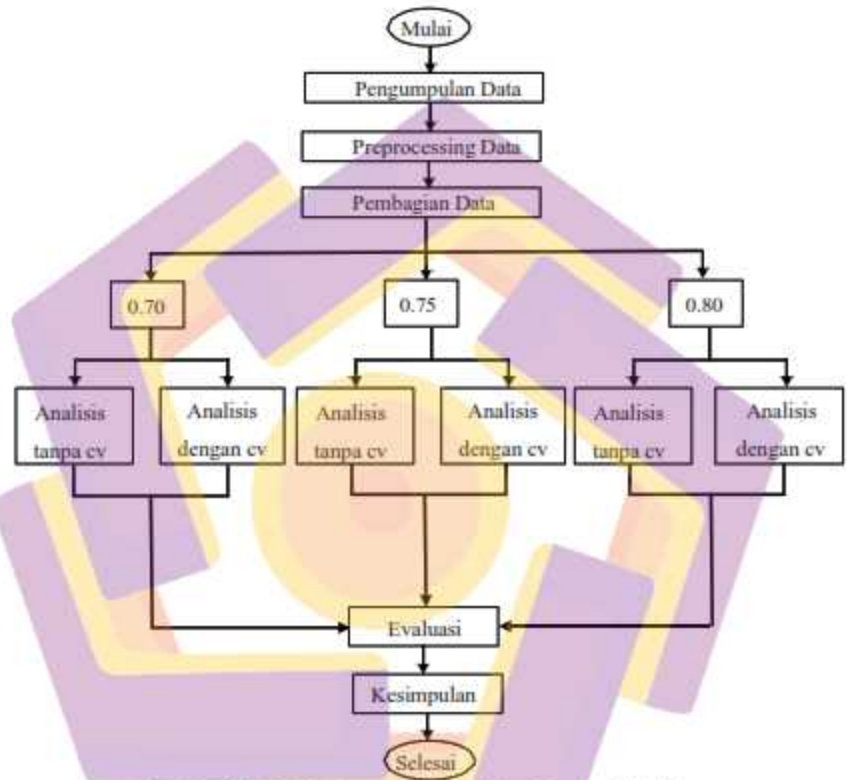
5) Menganalisis Data

Menganalisis data untuk menemukan hasil dari penelitian yang dilakukan

6) Kesimpulan

Menarik kesimpulan akhir berdasarkan temuan yang diperoleh pada saat proses analisis data.

3.4.2. Diagram alir proses analisis data tanpa SMOTE



Gambar 3.4.2.1. Daigram Alir Penelitian Tanpa SMOTE

Pada Gamabar 3.4.2.1. dijelaskan tentang langkah-langkah yang dilakukan dalam melakukan peelitian ini :

1) Mulai

Pada tahapan ini merupakan tahapan awal untuk melakukan proses analisis data.

2) Pengumpulan Data

Tahapan ini merupakan tahapan untuk mengumpulkan dataset yang akan digunakan dalam penelitian. Dalam penelitian ini menggunakan 9578 dataset yang meliputi kelas data dengan peminjam dapat menyelesaikan kredit sebanyak 8045 record dan peminjam yang tidak dapat menyelesaikan kredit sebanyak 1533 record yang dimana menunjukkan adanya

ketidakseimbangan kelas. Data yang digunakan merupakan data dari tahun 2007–2010 yang diperoleh dari Kaggle yakni data dari lendingclub.com. LendingClub.com adalah platform pinjaman *peer-to-peer* yang menghubungkan peminjam yang membutuhkan dana dengan investor yang memiliki dana untuk diinvestasikan.

3) Praprocessing Data

Tahap ini mencakup pengubahan tipe data. Pengubahan dilakukan apabila terdapat variabel yang tidak sesuai dengan kebutuhan dalam tahapan analisis klasifikasi untuk model yang digunakan.

4) Pembagian Dataset

Pada tahap ini data dibagi menjadi data latih (*training*) dan data uji (*testing*). Proporsi pembagian ini adalah 70%, 75% dan 80% untuk data latih (training) dan begitu juga untuk data untuk data uji(*testing*) sebesar 30%, 25% dan 20%. Pembagian data dilakukan secara acak untuk memastikan representasi yang adil dari setiap kelas dalam data *training* dan data *testing*.

5) Klasifikasi

Pada tahap ini melakukan proses klasifikasi dengan menggunakan algoritma Naive Bayes. Model Naive Bayes dilatih menggunakan data latih yang belum dan telah dioversample dengan SMOTE.

6) K-Fold Cross Validation

Pada tahapan ini yang dilaukan adalah melakukan *K-Fold Cross Validation* untuk mengevaluasi model lebih lanjut. Untuk memastikan model yang dilatih tidak overfitting dan memberikan gambaran yang lebih akurat tentang performa model.

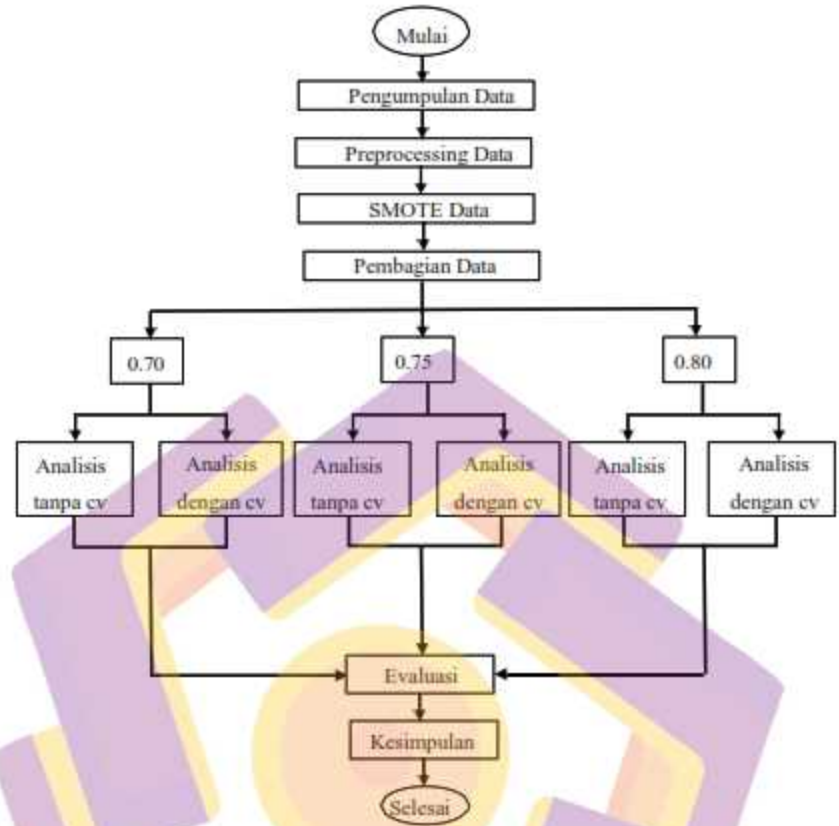
7) Evaluasi

Evaluasi dilakuakn dengan tujuan mengukur kinerja model klasifikasi yang telah dilakuakan. Data yang digunakan untuk melakukan evaluasi kinerja model yang dihasilkan yakni dengan menggunakan data *testing*. Evaluasi dilakukan dengan cara membandingkan akurasi model sebelum dan sesudah penerapan SMOTE.

8) Kesimpulan

Menarik kesimpulan berdasarkan hasil analisis. Menginterpretasikan hasil evaluasi dan efektivitas penerapan SMOTE dan K-Fold Cross Validation dalam meningkatkan akurasi dan performa model Naive Bayes

3.4.3. Diagram alir proses analisis menggunakan metode SMOTE



Gambar 3.4.3.1. Daigram Alir Penelitian Dengan SMOTE

Pada Gamabar 3.4.3.1. dijelaskan tentang langkah-langkah yang dilakukan dalam melakukan peelitian ini :

1) Perumusan masalah dilakukan untuk menentukan masalah yang dihadapi dalam penelitian, dan selanjutnya dilakukan perancangan penelitian.
2) Pengumpulan Data

   Mengumpulkan dataset yang akan digunakan dalam penelitian. Penelitian ini menggunakan 9578 dataset yang meliputi kelas data dengan peminjam dapat menyelesaikan kredit sebanyak 8045 record dan peminjam yang tidak dapat menyelesaikan kredit sebanyak 1533 record yang dimana menunjukkan adanya ketidakseimbangan kelas. Data yang digunakan dari tahun 2007–2010. Diperoleh dari Kaggle yakni data dari lendingclub.com. LendingClub.com adalah platfrm pinjaman peer-to-peer yang menghubungkan peminjam yang membutuhkan dana dengan investor yang memiliki dana untuk diinvestasikan.
3) Praprocessing Data

   Tahap ini mencakup pengubahan tipe data. Pengubahan dilakukan apabila

terdapat variabel yang tidak sesuai dengan kebutuhan dalam tahapan analisis klasifikasi untuk model yang digunakan dan normalisasi data.

4) Penerapan SMOTE

Menerapkan SMOTE pada data latih untuk menyeimbangkan jumlah sampel antara kelas mayoritas dan minoritas sebelum dilakukan klasifikasi.

5) Pembagian Dataset

Pada tahap ini data dibagi menjadi data latih dan data uji. Proporsi pembagian ini adalah 70% untuk data latih dan 30% untuk data uji. Pembagian data dilakukan secara acak untuk memastikan representasi yang adil dari setiap kelas dalam data latih dan data uji.

6) Klasifikasi

Pada tahap ini Algoritma yang digunakan dalam pengklasifikasian yaitu Naive Bayes. Model Naive Bayes dilatih menggunakan data latih yang belum dan telah dioversample dengan SMOTE.

7) K-Fold Cross Validation

Melakukan K-Fold Cross Validation untuk mengevaluasi model lebih lanjut. Untuk memastikan model yang dilatih tidak overfitting dan memberikan gambaran yang lebih akurat tentang performa model pada data yang tidak terlihat.

8) Evaluasi

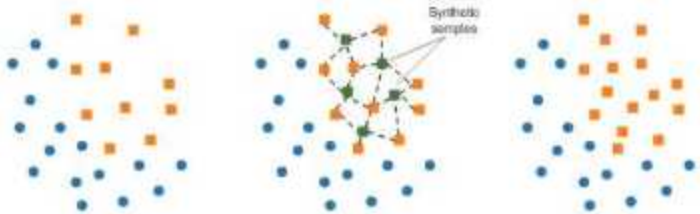
Evaluasi dilakuakn dengan tujuan mengukur kinerja model klasifikasi yang telah dilakuakan. Menggunakan data uji untuk memprediksi dan mengevaluasi performa model. Hitung dan bandingkan akurasi model sebelum dan sesudah penerapan SMOTE.

9) Kesimpulan

Menarik kesimpulan berdasarkan hasil analisis. Menginterpretasikan hasil evaluasi dan efektivitas penerapan SMOTE dan K-Fold Cross Validation dalam meningkatkan akurasi dan performa model Naive Bayes.

Perbedaan antara diagram alir 3.4.2.1. dengan diagram alir 3.4.3.1 yiatu pada proses penerapan teknik SMOTE. Pada diagram alir 3.4.2.1. analisis yang dilakukan yakni dengan membagi terlebih dahulu dataset yang akan dimiliki menjadi data *training* dan *testing* baru kemudian dilakukan proses SMOTE. Sedangkan pada diagram alir 3.4.3.1. proses yang dilakukan terlebih dahulu yakni dengan menyeimbangkan dataset terlebih dahulu dengan teknik SMOTE baru kemudian melakukam pembagian data untuk lanjut ke proses analisis klasifikasi.

3.4.4. Proses analisis SMOTE



Gambar 3.4.4.1. Proses Analisis SMOTE

Pada gambar 3.4.4.1 terlihat bahwa terdapat data yang tidak seimbang (*imbalance data*) yang kemudian di *generating* atau data yang minoritas ditambahkan dengan data sintetiknya dengan menggunakan metode SMOTE sehingga menghasilkan data yang seimbnag.

BAB IV
HASIL DAN PEMBAHASAN

Analisa dan pembahasan pada bab ini mencakup Pengantar yang berisi deskripsi singkat dan menjelaskan kembali latar belakang masalah, kemudian deskripsi data sebelum mendapatkan perlakuan metode SMOTE dan setelah mendapatkan perlakuan, kemudian hasil eksperimen sebelum SMOTE dan setelah SMOTE, kemudian perbandingan hasil dan pembahasan temuan.

4.1 Pengumpula Data

Data – data yang digunakan dalam penelitian ini merupakan data yang diperoleh website *kaggle* yakni *loan_data* yang merupakan data pinjaman dari salah satu bank pembiayaan di Amerika Serikat yakni Lending.com. Data yang digunakan terdiri dari 12 variabel dan salah satu diantaranya variabel target yakni variabel *not.fully.paid* yang dimana variabel yang menunjukkan apakah nasabah dapat membayar pinjamannya secara full ataukah tidak. Berikut penjelasan terkait data yang digunakan

Tabel 4.1.1 Statistik Deskriptif Dataset Loan

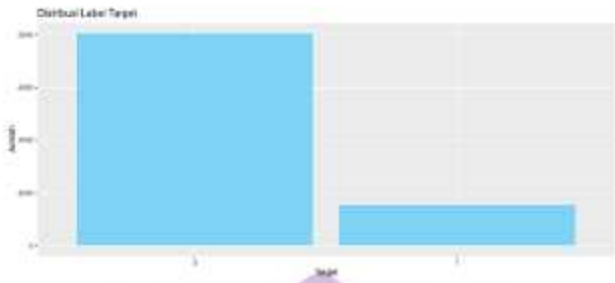
	vars	n	mean	med	min	max
int.rate	1	9578	0.12	0.12	0.06	0.22
Installment	2	9578	319.09	268.95	15.67	940.14
Log.annual.inc	3	9578	10.93	10.93	7.55	14.53
Dti	4	9578	12.61	12.66	0.00	29.96
fico	5	9578	710.85	707.00	612.00	827.00
days.with.cr.line	6	9578	4560.77	4139.96	178.96	17639.96
revol.bal	7	9578	16913.96	8596.00	0.00	1207359.00
revol.util	8	9578	46.80	46.30	0.00	119.00
inq.last.6mths	9	9578	1.58	1.00	0.00	33.00
delinq.2yrs	10	9578	0.16	0.00	0.00	13.00
pub.rec	11	9578	0.06	0.00	0.00	5.00
not.fully.paid	12	9578	0.16	-	0.00	1.00

Berikut penjelasan dari deskripsi statistik pada dataset loan:

- Tingkat Bunga (*int.rate*): Rata-rata tingkat bunga dari pinjaman adalah sekitar 12% dengan bunga terendah 6% dan tertinggi 22%. Artinya, sebagian besar pinjaman memiliki bunga di sekitar 12%.
- Angsuran Bulanan (*installment*): Rata-rata angsuran perbulan yakni sebesar 319 dengan angsuran terendah sebesar 15 dan tertinggi sebesar 940. Sehingga jumlah yang harus dibayarkan oleh nasabah setiap bulannya cukup bervariasi.
- Pendapatan Tahunan (*log.annual.inc*): Rata-rata pendapatan dari peminjam sekitar 10.93, yang berkisar antara sekitar 7.55 hingga 14 juta rupiah per tahun.
- Rasio Utang terhadap Pendapatan (DTI): Rata-rata rasio utang terhadap pendapatan adalah 12,6%, yang berarti sebagian besar orang memiliki utang sekitar 12,6% dari pendapatan mereka. Rasio ini bervariasi, mulai dari 0% hingga sekitar 30%.
- Jumlah Pencarian Kredit dalam 6 Bulan Terakhir (*inq.last.6mths*): Rata-rata orang melakukan sekitar 1,5 kali pencarian kredit dalam 6 bulan terakhir. Namun, ada beberapa orang yang melakukan banyak pencarian, hingga 33 kali.
- Delinkuensi dalam 2 Tahun Terakhir (delinq.2yrs): Rata-rata delinkuensi (telat bayar) dalam dua tahun terakhir sangat rendah, yaitu hanya 0,16. Sebagian besar peminjam tidak terlambat dalam melakukan pembayaran selama dua tahun terakhir.
- Pinjaman yang Belum Sepenuhnya Dibayar (not.fully.paid): Sekitar 16% pinjaman dalam dataset ini belum dibayar sepenuhnya, yang berarti sebagian kecil peminjam belum menyelesaikan pembayaran pinjaman mereka.

Secara keseluruhan, dataset ini memberikan gambaran tentang kondisi keuangan peminjam, termasuk berapa banyak yang mereka pinjam, seberapa besar angsuran yang harus mereka bayar, serta profil kredit mereka. Variabel yang digunakan sebagai variabel target yakni variabel *not.fully.paid* yang dimana merupakan pinjaman yang diberikan kepada individu atau entitas belum sepenuhnya dilunasi. Ada sejumlah pembayaran yang masih tertunda atau sisa saldo utang yang belum dilunasi. Berikut perbandingan antara status telah

dilunasi dan belum



Gambar 4.1.1. Histogram variabel not.fully.paid

Label not.fully.paid menunjukkan apakah pinjaman belum dibayar sepenuhnya atau tidak. Nilai 0 berarti pinjaman telah dibayar sepenuhnya, sementara nilai 1 berarti pinjaman belum dibayar sepenuhnya. Berdasarkan deskripsi, nilai rata-rata untuk variabel ini adalah 0.16, yang menunjukkan bahwa sekitar 16% pinjaman dalam dataset ini belum dibayar sepenuhnya.

Jika kita melihat histogram dari variabel ini, maka terdapat distribusi yang tidak seimbang (*imbalanced*), yaitu:

- Mayoritas data bernilai 0, yang berarti sebagian besar pinjaman sudah dibayar sepenuhnya.
- Minoritas data bernilai 1, yang berarti hanya sekitar 16% pinjaman yang belum sepenuhnya dibayar.

Distribusi yang tidak seimbang ini sering kali menjadi masalah dalam analisis data dan pemodelan, terutama jika kita menggunakan teknik seperti klasifikasi. Jika terdapat kelas yang tidak seimbang maka model akan cenderung untuk memprediksi kelas mayoritas (pinjaman yang sudah dibayar). Hal ini akan mempengaruhi akurasi dari model, karena model akan memiliki akurasi yang tinggi dengan memprediksi kelas mayoritas, namun tidak baik untuk akurasi kelas minoritas.

4.2 Praprosessing Data

Pada tahapan praprosessing data, tahapan yang dilakukan yakni mengubah tipe variabel target yang awalnya numerik menjadi factor. Pengubahan ini dilakukan karena model yang akan dilakukan untuk analisis klasifikasi memerlukan tipe data untuk variabel target yang bertipe factor. Selain dari variabel target, semua variabel yang digunakan bertipe numerik

Tabel 4.2.1 Tipe Data Asli

Variabel	Tipe
int.rate	Numerik
Installment	Numerik
Log.annual.inc	Numerik
Dti	Numerik
Fico	Numerik
days.with.cr.line	Numerik
revol.bal	Numerik
revol.util	Numerik
inq.last.6mths	Numerik
delinq.2yrs	Numerik
pub.rec	Numerik
not.fully.paid	Biner

Dataset ini terdiri dari 9.578 baris dan 12 kolom, yang berarti ada 9.578 data peminjam dan 12 variabel terkait informasi keuangan dan status pinjaman yang semua variabelnya bertipe biner, sehingga untuk variabel target yakni variabel *not.fully.paid* harus dirubah menjadi tipe faktor.

Tabel 4.2.2. Tipe Data Setelah Dirubah

Variabel	Tipe
int.rate	Numerik
Installment	Numerik
Log.annual.inc	Numerik
Dti	Numerik
Fico	Numerik
days.with.cr.line	Numerik
revol.bal	Numerik
revol.util	Numerik
inq.last.6mths	Numerik
delinq.2yrs	Numerik
pub.rec	Numerik
not.fully.paid	Factor

Pada tabel 4.2.2, variabel target yakni variabel *not.fully.paid* telah dirubah menjadi tipe faktor sehingga telah sesuai dengan ketentuan dari yang dibutuhkan dalam proses analisis. Setelah tipe data variabel target yakni variabel "*not.fully.paid*" diubah dari biner menjadi factor, maka sekarang dari 12 variabel yang digunakan terdapat satu variabel yang bertipe factor. Syntax

atau kode yang digunakan untuk mengubah tipe data dari numerik menjadi faktor yakni

Tabel 4.2.3 Syntax untuk mengubah tipe data

```
# Encode target if categorical

# Convert columns with comma decimals to float
for col in X.columns:
   if X[col].dtype == 'object':
      try:
         X[col] = X[col].str.replace(',', '.',
regex=False).astype(float)
      except ValueError:
         pass # Keep the column as is if conversion fails
```

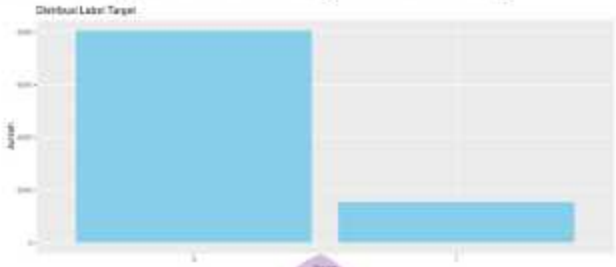
Hubungan antara variabel independen dengan variabel objek

No	Variabel	Info Gain	Penjelasan
1.	Int.rate	0.5418	Sangat berpengaruh
2.	Inq.last.6mths	0.0897	Berpengaruh
3.	Fico	0.0713	Lumayan berpengaruh
4.	Days.with.cr.line	0.0297	Lumayan berpengaruh
5.	DTI	0.0154	Agak berpengaruh
6.	Installment	0.0141	Agak berpengaruh
7.	Revol.bal	0.0114	Pengaruh kecil
8.	Revol.ulti	0.0089	Pengaruh Kecil
9.	Log.annual.inc	0.0028	Hampir tidak berpengaruh
10.	Pub.rec	0.0027	Hampir tidak berpengaruh
11.	Deling.2yrs	0.0000	Tidak memberikan informasi

Setelah menemukan hubungan antar variabel, langkah selanjutnya yaitu melakukan normalisasi data. Teknik normalisasi yang digunakan adalah salah satu bentuk min-max normalization yakni teknik yang umum digunakan untuk mengubah skala data numerik agar berada dalam rentang [0, 1].

```
> normalize <- function(x) {
+   return((x - min(x)) / (max(x) - min(x)))
+ }
> X_norm <- as.data.frame(lapply(X, function(col) {
+   if(is.numeric(col)) normalize(col) else col
+ }))
> loan_norm <- data.frame(X_norm, not.fully.paid = y)
```

4.3 Analisis Naïve Bayes Pada Data Yang Tidak Seimbang



Gambar 4.3.1. Histogram data yang tidak seimbang

Berdasarkan histogram, nilai 0 mendominasi dengan jumlah peminjam yang berhasil melunasi sebanyak 8045, sementara hanya sedikit peminjam yang belum membayar pinjaman mereka (nilai 1) yakni sebanyak 1533. Dalam kasus ini, kategori 0 (sudah dibayar sepenuhnya) memiliki batang histogram yang jauh lebih tinggi daripada kategori 1 (belum dibayar sepenuhnya). Penjelasan dari syntax diatas yakni:

- set.seed: Fungsi ini mengatur seed, sehingga setiap kali menjalankan kode yang menghasilkan angka yang acak, hasilnya akan sama.
- 123: merupakan nilai *seed* yang spesifik. Angka yang digunakan bisa dengan angka lain, tetapi menggunakan angka yang sama akan menghasilkan urutan angka acak yang sama setiap kali.
- "trainindex": merupakan kode yang digunakan untuk membagi data menjadi data training dan testing.
- "trainData": merupakan data training
- "testData": merupakan data testing

4.3.1. Membagi Data Dengan Perbandingan 0.67 dan 0.33

```
# 3. Split dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33,
random_state=42)
```

a. Analisis Tanpa penerapan K-Fold Cross Validation

- *Confusion matrix*

	0	1
0	2255	395
1	338	173

- *Classification report*

	precision	recall	F-1 score	Support
0	0.87	0.85	0.86	2650
1	0.30	0.34	0.32	511
Accuracy			0.77	3161
Macro avg	0.59	0.59	0.59	3161
Weighted avg	0.78	0.77	0.77	3161

b. Analisis Dengan penerapan K-Fold Cross Validation

- *Confusion matrix*

Tabel 4.3.1.3 Output model naive bayes

	0	1
0	6880	1165
1	1053	480

- *Classification report*

	precision	recall	F-1 score	Support
0	0.87	0.86	0.86	8045
1	0.29	0.31	0.30	1533
Accuracy			0.77	9578
Macro avg	0.58	0.58	0.58	9578
Weighted avg	0.77	0.77	0.77	9578

4.3.2. Membagi Data Dengan Perbandingan 0.75 dan 0.25

Tabel 4.3.2.1. Syntax pembagian data training dan testing 0.75

```
# 3. Split dataset
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.25, random_state=42)
```

Berikut ini adalah jumlah data training dan testing dengan perbandingan 0.75 data traing dan 0.25 data testing

Tabel 4.3.2.2. Hasil pembagian dataset dengan 0.75

Pembagian	Nilai	
	0	1
Training	6034	1150
Testing	2011	383
Total	8049	1533

Dari dataset training, kita dapat melihat bahwa 6034 peminjam sudah membayar pinjaman mereka sepenuhnya, sedangkan hanya 1150 peminjam yang belum membayar pinjaman mereka sepenuhnya. Kategori 0 memiliki jumlah yang lebih banyak dibandingkan dengan kategori 1. Dalam dataset testing, distribusi data menunjukkan ketidakseimbangan. Dalam distribusi data terdapat 2011 peminjam sudah membayar pinjaman mereka sepenuhnya sedangkan 383 peminjam belum membayar pinjaman mereka sepenuhnya.

a. Analisis Klasifikasi Tanpa Cross Validation

- *Confusion matrix*

	0	1
0	1696	316
1	246	137

- *Classification report*

	precision	recall	F-1 score	Support
0	0.87	0.84	0.86	2012
1	0.30	0.36	0.33	383
Accuracy			0.77	2395
Macro avg	0.59	0.60	0.59	2395
Weighted avg	0.78	0.77	0.77	2395

b. Analisis Dengan K-Fold Cross Validation

- *Confusion matrix*

Tabel 4.3.1.3 Output model naive bayes

	0	1
0	6904	1141
1	1051	482

- *Classification report*

	precision	recall	F-1 score	Support
0	0.87	0.86	0.86	8045
1	0.30	0.31	0.30	1533
Accuracy			0.77	9578
Macro avg	0.58	0.59	0.58	9578
Weighted avg	0.78	0.77	0.77	9578

4.3.3. Membagi data dengan perbandingan 0.80 dan 0.20

Tabel 4.3.3.1. Syntax pembagian data training dan testing untuk 0.80

```
# 3. Split dataset
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.20, random_state=42)
```

Berikut ini adalah jumlah data training dan testing dengan perbandingan 0.80 data traing dan 0.20 data testing

Tabel 4.3.3.2. Hasil Pembagian Data Training dan Testing untuk 0.80

Pembagian	Nilai	
	0	1
Training	6436	1227
Testing	1609	306
Total	8049	1533

Dari dataset training, kita dapat melihat bahwa 6436 peminjam sudah membayar pinjaman mereka sepenuhnya, sedangkan hanya 1227 peminjam yang belum membayar pinjaman mereka sepenuhnya. Kategori 0 memiliki jumlah yang lebih banyak dibandingkan dengan kategori 1. Dalam dataset testing, distribusi data menunjukkan adanya ketidakseimbangan. Dalam distribusi data tersebut terdapat 1609 peminjam sudah membayar pinjaman mereka sepenuhnya sedangkan 306 peminjam belum membayar pinjaman mereka sepenuhnya

a. Analisis Tanpa K-Fold Cross Validation

- *Confusion matrix*

	0	1
0	1344	267
1	187	118

- *Classification report*

	precision	recall	F-1 score	Support
0	0.88	0.83	0.86	1611
1	0.31	0.39	0.34	305
Accuracy			0.76	1916
Macro avg	0.59	0.61	0.60	1916
Weighted avg	0.79	0.76	0.77	1916

b. Analisis dengan K-Fold Cross Validation

Tabel 4.3.3.4. Output model naive bayes tanpa cv

- *Confusion matrix*

Tabel 4.3.1.3 Output model naive bayes

	0	1
0	6898	1147
1	1057	476

- *Classification report*

	precision	recall	F-1 score	Support
0	0.87	0.86	0.86	8045
1	0.29	0.31	0.30	1533
Accuracy			0.77	9578
Macro avg	0.58	0.58	0.58	9578
Weighted avg	0.77	0.77	0.77	9578

4.3.4. Evaluasi

Tabel 4.3.4.1. Metrik Evaluasi Model Naive Bayes Tanpa SMOTE

Pembagian	Kondisi	Akurasi							
		TP	TN	FP	FN	ACC	Recall	F1-score	Presisi
0.67	Sebelum cv	2255	173	395	338	0.77	0.34	0.32	0.30
	Setelah cv	6880	480	1165	1053	0.77	0.31	0.30	0.29
0.75	Sebelum cv	1696	137	316	246	0.77	0.36	0.33	0.30
	Setelah cv	6904	482	1141	1051	0.77	0.31	0.30	0.30
0.80	Sebelum cv	1344	118	267	187	0.76	0.39	0.34	0.31
	Setelah cv	6898	476	1157	1057	0.77	0.31	0.30	0.30

Ket:

- Sebelum cv : Sebelum penerapan *k-fold cross validation*
- Setelah cv : Setelah penerapan *k-fold cross validation*
- TP : True Positif yakni model memprediksi kelas positif dengan benar
- TN : True Negative yakni model memprediksi negatif dan benar negatif
- FP : False Negative yakni model memprediksi positif padahal negatif
- FN : False Negative yakni model memprediksi negatif padahal positif
- ACC : Akurasi yakni prediksi yang benar dari seluruh prediksi yang dilakukan
- F1-Score :
- Presisi : Prediksi positif yang benar-benar positif

Penjelasan dari metrik evaluasi pada tabel 4.3.4.1:

a. Pembagian 0.67

- **Sebelum Penerapan K-Fold Cross Validation**

a) True Positives (TP): 2255 (Model memprediksi 0 dan benar, yaitu pinjaman yang sudah dibayar sepenuhnya)

b) True Negatives (TN): 173 (Model memprediksi 1 dan benar, yaitu pinjaman yang belum dibayar sepenuhnya)

c) False Positives (FP): 395 (Model memprediksi 0, tetapi seharusnya 1

d) False Negatives (FN): 338 (Model memprediksi 1, tetapi seharusnya 0

e) Acc (Akurasi) = 0.77 atau akurasi sebesar 77% menunjukkan bahwa model berhasil memprediksi dengan benar sekitar 77% dari seluruh sampel yang diuji

f) Recall = 0.34 untuk mengukur kemampuan model untuk mengenali kelas 0 (pinjaman yang sudah dibayar sepenuhnya).

g) F1-Score = 0.34 ukuran gabungan dari precision dan recall

h) Precision = 0.30 menunjukkan bahwa 30% dari prediksi 0 oleh model adalah benar-benar 0, yaitu pinjaman yang sudah dibayar sepenuhnya.

- **Setelah Penerapan K-Fold Cross Validation**

a) True Positives (TP): 6880 (Model memprediksi 0 dan benar, yaitu pinjaman yang sudah dibayar sepenuhnya)

b) True Negatives (TN): 480 (Model memprediksi 1 dan benar, yaitu pinjaman yang belum dibayar sepenuhnya)

c) False Positives (FP): 1165 (Model memprediksi 0, tetapi seharusnya 1

d) False Negatives (FN): 1053 (Model memprediksi 1, tetapi seharusnya 0

e) Acc (Akurasi) = 0.77 atau akurasi sebesar 77% menunjukkan bahwa model berhasil memprediksi dengan benar sekitar 77% dari seluruh sampel yang diuji

f) Recall = 0.31 untuk mengukur kemampuan model untuk mengenali kelas 0 (pinjaman yang sudah dibayar sepenuhnya).

g) F1-Score = 0.30 ukuran gabungan dari precision dan recall

h) Precision = 0.29 menunjukkan bahwa 29% dari prediksi 0 oleh model adalah benar-benar 0, yaitu pinjaman yang sudah dibayar sepenuhnya.

b. Pembagian 0.75

- **Sebelum Penerapan K-Fold Cross Validation**

a) True Positives (TP): 1696 (Model memprediksi 0 dan benar, yaitu pinjaman yang sudah dibayar sepenuhnya)

b) True Negatives (TN): 137 (Model memprediksi 1 dan benar, yaitu pinjaman yang belum dibayar sepenuhnya)

c) False Positives (FP): 316 (Model memprediksi 0, tetapi seharusnya 1

d) False Negatives (FN): 246 (Model memprediksi 1, tetapi seharusnya 0

e) Acc (Akurasi) = 0.77 atau akurasi sebesar 77% menunjukkan bahwa model berhasil memprediksi dengan benar sekitar 77% dari seluruh sampel yang diuji

f) Recall = 0.36 untuk mengukur kemampuan model untuk mengenali kelas 0 (pinjaman yang sudah dibayar sepenuhnya).

g) F1-Score = 0.33 ukuran gabungan dari precision dan recall

h) Precision = 0.30 menunjukkan bahwa 30% dari prediksi 0 oleh model adalah benar-benar 0, yaitu pinjaman yang sudah dibayar sepenuhnya.

- **Setelah Penerapan K-Fold Cross Validation**

a) True Positives (TP): 6904 (Model memprediksi 0 dan benar, yaitu pinjaman yang sudah dibayar sepenuhnya)

b) True Negatives (TN): 482 (Model memprediksi 1 dan benar, yaitu pinjaman yang belum dibayar sepenuhnya)

c) False Positives (FP): 1141 (Model memprediksi 0, tetapi seharusnya 1

d) False Negatives (FN): 1051 (Model memprediksi 1, tetapi seharusnya 0

e) Acc (Akurasi) = 0.77 atau akurasi sebesar 77% menunjukkan bahwa model berhasil memprediksi dengan benar sekitar 84% dari seluruh sampel yang diuji

f) Recall = 0.31 untuk mengukur kemampuan model untuk mengenali kelas 0 (pinjaman yang sudah dibayar sepenuhnya).

g) F1-Score = 0.30 ukuran gabungan dari precision dan recall

h) Presisi = 0.30 menunjukkan bahwa 30% dari prediksi 0 oleh model adalah benar-benar 0, yaitu pinjaman yang sudah dibayar sepenuhnya.

c. Pembagian 0.80

- **Sebelum Penerapan K-Fold Cross Validation**

a) True Positives (TP): 1344 (Model memprediksi 0 dan benar, yaitu pinjaman yang sudah dibayar sepenuhnya)

b) True Negatives (TN): 118 (Model memprediksi 1 dan benar, yaitu pinjaman yang belum dibayar sepenuhnya)

c) False Positives (FP): 267 (Model memprediksi 0, tetapi seharusnya

1

d) False Negatives (FN): 187 (Model memprediksi 1, tetapi seharusnya 0

e) Acc (Akurasi) = 0.76 atau akurasi sebesar 76% menunjukkan bahwa model berhasil memprediksi dengan benar sekitar 76% dari seluruh sampel yang diuji

f) Recall = 0.39 untuk mengukur kemampuan model untuk mengenali kelas 0 (pinjaman yang sudah dibayar sepenuhnya).

g) F1-Score = 0.34 ukuran gabungan dari precision dan recall

h) Presisi = 0.31 menunjukkan bahwa 31% dari prediksi 0 oleh model adalah benar-benar 0, yaitu pinjaman yang sudah dibayar sepenuhnya.

- **Setelah penearan K-Fold Cross Validation**

a) True Positives (TP): 6898 (Model memprediksi 0 dan benar, yaitu pinjaman yang sudah dibayar sepenuhnya)

b) True Negatives (TN): 476 (Model memprediksi 1 dan benar, yaitu pinjaman yang belum dibayar sepenuhnya)

c) False Positives (FP): 1157 (Model memprediksi 0, tetapi seharusnya 1

d) False Negatives (FN): 1057 (Model memprediksi 1, tetapi seharusnya 0

e) Acc (Akurasi) = 0.77 atau akurasi sebesar 77% menunjukkan bahwa model berhasil memprediksi dengan benar sekitar 77% dari seluruh sampel yang diuji

f) Recall = 0.31 untuk mengukur kemampuan model untuk mengenali kelas 0 (pinjaman yang sudah dibayar sepenuhnya).

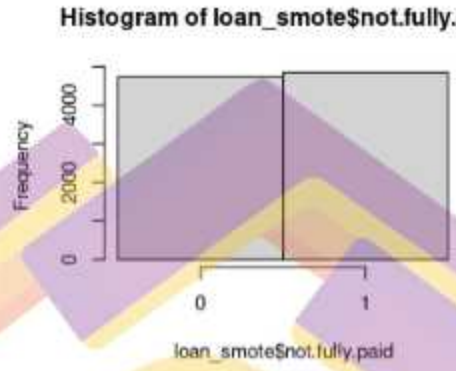
g) F1-Score = 0.30 ukuran gabungan dari precision dan recall

h) Presisi = 0.30 menunjukkan bahwa 30% dari prediksi 0 oleh model adalah benar-benar 0, yaitu pinjaman yang sudah dibayar sepenuhnya.

4.4.Analisis Klasifikasi Naïve Bayes Dengan Penerapan Teknik SMOTE

Jumlah data yang digunakan yakni sebanyak 9578. Terdapat tiga komposisi perbandingan data training dan testing yang digunakan yakni sebesar 0.7:0.3, 0.75:0.25, dan 0.80:0.20. Perbandingan variabel yang berkode 0 dengan 1 yakni sebesar 0.84 : 0.16 . Berdasarkan ketidakseimbangan data

yang ada, maka akan dilakukan pengolahan data untuk mengatasi masalah ketidakseimbangan kelas dalam dataset. Metode ini bekerja dengan cara menghasilkan data sintetik dari kelas minoritas sehingga menghasilkan data yang seimbang antara kelas mayoritas dengan kelas minoritas. Setelah melalui proses penyeimbangan dengan menggunakan teknik SMOTE, berikut distribusi data seimbang yang dapat dilihat pada histogram gambar



Gambar 4.4.1. Histogam Data Setelah Penerapan SMOTE

Berdasarkan histogram, kategori 0 dan 1 terlihat telah seimbang dengan persentase peminjam kategori lunas (0) sebesar 0.49% dan kategori belum lunas membayar (1) sebesar 0.51%. Persentase tersebut menunjukkan bahwa baik kelas 0 maupun kelas 1 tidak ada yang mnejadi kelas dengan sebutan kelas mayoritas, maka hasil analisis klasifikasi yang diperoleh pun akan berbeda. teknik SMOTE yang digunakan yakni *package* ROSE (*Random Over Sampling Examples)* yang dimana teknik tersebut digunakan untuk meresampling data dengan menghasilkan sampel sintetis berdasarkan distribusi kelas dan dapat membantu meningkatkan performa model.

4.4.1. Dengan perbandingan 0.67 dan 0.33

a. Analisis Tanpa K-Fold Cross Validation

- *Confusion matrix*

	0	1
0	1786	915
1	1130	1479

- *Classification report*

	precision	recall	F-1 score	Support
0	0.61	0.66	0.64	2701

1	0.62	0.57	0.59	2609
Accuracy			0.61	5130
Macro avg	0.62	0.61	0.61	5130
Weighted avg	0.62	0.61	0.61	5130

b. Analisis Dengan K-Fold Cross Validation

Pada penelitian ini, metode yang digunakan untuk mengoptimalkan model naïve bayes yang dihasilkan yakni dengan menggunakan metode *K-Fold Cross Validation*. Fungsi yang digunakan merupakan fungsi "*trainControl*" dari paket "caret". Kemudian untuk metode dalam *K-Fold Cross Validation* itu sendiri menggunakan metode "cv" atau *cross validation*. Hasil analisis denga SMOTE dan menggunakan *cross validation*.

- *Confusion matrix*

Tabel 4.3.1.3 Output model naive bayes

	0	1
0	5435	2610
1	699	834

- *Classification report*

	precision	recall	F-1 score	Support
0	0.89	0.68	0.77	8045
1	0.24	0.54	0.34	1533
Accuracy			0.65	9578
Macro avg	0.56	0.61	0.55	9578
Weighted avg	0.78	0.65	0.70	9578

6.4.2. Dengan perbandingan 0.75 dan 0.25

a. Analisis Klasifikasi Tanpa Cross Validation

- *Confusion matrix*

Tabel 4.3.1.3 Output model naive bayes

	0	1
0	1328	691
1	883	1121

- *Classification report*

	precision	recall	F-1 score	Support
0	0.60	0.66	0.63	2019
1	0.62	0.56	0.59	2004
Accuracy			0.61	4023
Macro avg	0.61	0.61	0.61	4023
Weighted avg	0.61	0.61	0.61	4023

b. Analisis Dengan K-Fold Cross Validation

- *Confusion matrix*

Tabel 4.3.1.3 Output model naive bayes

	0	1
0	5532	2512
1	718	815

- *Classification report*

	precision	recall	F-1 score	Support
0	0.88	0.69	0.77	8045
1	0.24	0.53	0.33	1533
Accuracy			0.66	9578
Macro avg	0.56	0.61	0.55	9578
Weighted avg	0.78	0.66	0.70	9578

6.4.3. Dengan perbandingan 0.80 dan 0.20

a. Analisis Tanpa K-Fold Cross Validation

- *Confusion matrix*

Tabel 4.3.1.3 Output model naive bayes

	0	1
0	1062	552
1	687	917

- *Classification report*

	precision	recall	F-1 score	Support
0	0.61	0.66	0.63	1614
1	0.62	0.57	0.60	1604
Accuracy			0.61	3218
Macro avg	0.62	0.61	0.61	3218
Weighted avg	0.62	0.61	0.61	3218

b. Analisis Dengan K- Fold Cross Validation

- *Confusion matrix*

Tabel 4.3.1.3 Output model naive bayes

	0	1
0	5419	2626
1	695	838

- *Classification report*

	precision	recall	F-1 score	Support
0	0.89	0.67	0.76	8045
1	0.24	0.55	0.33	1533
Accuracy			0.65	9578
Macro avg	0.56	0.61	0.55	9578
Weighted avg	0.78	0.65	0.70	9578

6.4.4. Evaluasi

Tabel 4.4.4.1. Metrik Evaluasi pada model setelah penerapan smote

Pembagian	Kondisi	Akurasi							
		TP	TN	FP	FN	ACC	Recall	F1-score	Presisi
0.67	Sebelum cv	1178	1479	915	1130	0.61	0.61	0.61	0.62
	Setelah cv	5435	834	2610	699	0.65	0.54	0.34	0.24
0.75	Sebelum cv	1328	1121	883	691	0.61	0.56	0.59	0.62
	Setelah cv	5533	815	2512	718	0.66	0.33	0.53	0.24
0.80	Sebelum cv	1344	118	267	187	0.76	0.76	0.77	0.79
	Setelah cv	6898	476	1147	1057	0.77	0.31	0.30	0.29

Ket:

- Sebelum cv : Sebelum penerapan *k-fold cross validation*
- Setelah cv : Setelah penerapan *k-fold cross validation*
- TP : True Positif yakni model memprediksi kelas positif dengan benar
- TN : True Negative yakni model memprediksi negatif dan benar negatif
- FP : False Negative yakni model memprediksi positif padahal negatif
- FN : False Negative yakni model memprediksi negatif padahal positif

- ACC : Akurasi yakni prediksi yang benar dari seluruh prediksi yang dilakukan
- Recall : proporsi data positif yang benar-benar diprediksi positif oleh model
- F1-Score = ukuran gabungan dari presicion dan recall
- Presisi : Prediksi positif yang benar-benar positif

Penjelasan dari tabel 4.4.4.1 yakni tabel metrik evaluasi pada model setelah penerapan smote yakni sebagai berikut:

a. Pembagian 0.67

- **Sebelum Penerapan K-Fold Cross Validation**
  a) True Positives (TP): 1178 (Model memprediksi 0 dan benar, yaitu pinjaman yang sudah dibayar sepenuhnya)
  b) True Negatives (TN): 1479 (Model memprediksi 1 dan benar, yaitu pinjaman yang belum dibayar sepenuhnya)
  c) False Positives (FP): 915 (Model memprediksi 0, tetapi seharusnya 1
  d) False Negatives (FN): 1130 (Model memprediksi 1, tetapi seharusnya 0
  e) Acc (Akurasi) = 0.61 atau akurasi sebesar 61% menunjukkan bahwa model berhasil memprediksi dengan benar sekitar 61% dari seluruh sampel yang diuji
  f) Recall = 0.61 untuk mengukur kemampuan model untuk mengenali kelas 0 (pinjaman yang sudah dibayar sepenuhnya).
  g) F1-Score = 0.61 ukuran gabungan dari precision dan recall
  h) Presisi = 0.62 menunjukkan bahwa 62% dari prediksi 0 oleh model adalah benar-benar 0, yaitu pinjaman yang sudah dibayar sepenuhnya.

- **Setelah Penerapan K-Fold Cross Validation**
  a) True Positives (TP): 5435 (Model memprediksi 0 dan benar, yaitu pinjaman yang sudah dibayar sepenuhnya)
  b) True Negatives (TN): 834 (Model memprediksi 1 dan benar, yaitu pinjaman yang belum dibayar sepenuhnya)
  c) False Positives (FP): 2610 (Model memprediksi 0, tetapi seharusnya 1
  d) False Negatives (FN): 699 (Model memprediksi 1, tetapi seharusnya 0

e) Acc (Akurasi) = 0.65 atau akurasi sebesar 65% menunjukkan bahwa model berhasil memprediksi dengan benar sekitar 65% dari seluruh sampel yang diuji

f) Recall = 0.54 untuk mengukur kemampuan model untuk mengenali kelas 0 (pinjaman yang sudah dibayar sepenuhnya).

g) F1-Score = 0.34 ukuran gabungan dari precision dan recall

h) Presisi = 0.24 menunjukkan bahwa 24% dari prediksi 0 oleh model adalah benar-benar 0, yaitu pinjaman yang sudah dibayar sepenuhnya.

b. Pembagian 0.75

- **Sebelum peneraoa K-Fold Cross Validation**

a) True Positives (TP): 1328 (Model memprediksi 0 dan benar, yaitu pinjaman yang sudah dibayar sepenuhnya)

b) True Negatives (TN): 1121 (Model memprediksi 1 dan benar, yaitu pinjaman yang belum dibayar sepenuhnya)

c) False Positives (FP): 883 (Model memprediksi 0, tetapi seharusnya 1

d) False Negatives (FN): 691 (Model memprediksi 1, tetapi seharusnya 0

e) Acc (Akurasi) = 0.61 atau akurasi sebesar 61% menunjukkan bahwa model berhasil memprediksi dengan benar sekitar 61% dari seluruh sampel yang diuji

f) Recall = 0.56 untuk mengukur kemampuan model untuk mengenali kelas 0 (pinjaman yang sudah dibayar sepenuhnya).

g) F1-Score = 0.59 ukuran gabungan dari precision dan recall

h) Presisi = 0.62 menunjukkan bahwa 62% dari prediksi 0 oleh model adalah benar-benar 0, yaitu pinjaman yang sudah dibayar sepenuhnya.

- **Setelah penerana K-Fold Cross Validation**

a) True Positives (TP): 5533 (Model memprediksi 0 dan benar, yaitu pinjaman yang sudah dibayar sepenuhnya)

b) True Negatives (TN): 815 (Model memprediksi 1 dan benar, yaitu pinjaman yang belum dibayar sepenuhnya)

c) False Positives (FP): 2512 (Model memprediksi 0, tetapi seharusnya 1

d) False Negatives (FN): 718 (Model memprediksi 1, tetapi

seharusnya 0

e) Acc (Akurasi) = 0.66 atau akurasi sebesar 66% menunjukkan bahwa model berhasil memprediksi dengan benar sekitar 66% dari seluruh sampel yang diuji
f) Recall = 0.33 untuk mengukur kemampuan model untuk mengenali kelas 0 (pinjaman yang sudah dibayar sepenuhnya).
g) F1-Score = 0.53 ukuran gabungan dari precision dan recall
h) Presisi = 0.24 menunjukkan bahwa 24% dari prediksi 0 oleh model adalah benar-benar 0, yaitu pinjaman yang sudah dibayar sepenuhnya.

d. Pembagian 0.80

- **Sebelum penerana K-Fold Cross Validation**

a) True Positives (TP): 1344 (Model memprediksi 0 dan benar, yaitu pinjaman yang sudah dibayar sepenuhnya)
b) True Negatives (TN): 118 (Model memprediksi 1 dan benar, yaitu pinjaman yang belum dibayar sepenuhnya)
c) False Positives (FP): 267 (Model memprediksi 0, tetapi seharusnya 1
d) False Negatives (FN): 187 (Model memprediksi 1, tetapi seharusnya 0
e) Acc (Akurasi) = 0.76 atau akurasi sebesar 76% menunjukkan bahwa model berhasil memprediksi dengan benar sekitar 76% dari seluruh sampel yang diuji
f) Recall = 0.76 untuk mengukur kemampuan model untuk mengenali kelas 0 (pinjaman yang sudah dibayar sepenuhnya).
g) F1-Score = 0.77 ukuran gabungan dari precision dan recall
h) Presisi = 0.79 menunjukkan bahwa 79% dari prediksi 0 oleh model adalah benar-benar 0, yaitu pinjaman yang sudah dibayar sepenuhnya.

- **Setelah penerana K-Fold Cross Validation**

a) True Positives (TP): 6898 (Model memprediksi 0 dan benar, yaitu pinjaman yang sudah dibayar sepenuhnya)
b) True Negatives (TN): 476 (Model memprediksi 1 dan benar, yaitu pinjaman yang belum dibayar sepenuhnya)
c) False Positives (FP): 1147 (Model memprediksi 0, tetapi

seharusnya 1

d) False Negatives (FN): 1057 (Model memprediksi 1, tetapi seharusnya 0
e) Acc (Akurasi) = 0.77 atau akurasi sebesar 77% menunjukkan bahwa model berhasil memprediksi dengan benar sekitar 77% dari seluruh sampel yang diuji
f) Recall = 0.31 untuk mengukur kemampuan model untuk mengenali kelas 0 (pinjaman yang sudah dibayar sepenuhnya).
g) F1-Score = 0.30 ukuran gabungan dari precision dan recall
h) Presisi = 0.29 menunjukkan bahwa 29% dari prediksi 0 oleh model adalah benar-benar 0, yaitu pinjaman yang sudah dibayar sepenuhnya.

4.5. Hasil Analisis

Data yang digunakan merupakan data yang tidak seimbang antara kelas 0 dengan kelas 1, dengan jumlah kelas 0 sebanyak 8045 dan kelas 1 sebanyak 1533. Hal ini dapat menurunkan performa kinerja model dalam melakukan prediksi maupun klasifikasi. Sehingga perlunya penyeimbangan data agar hasil kinerja model menjadi lebih baik.

Teknik penyeimbangan data atau *balancing data* yang digunakan yaitu dengan menggunakan salah satu *library* yang ada di pemrograman Python. Algoritma yang digunakan yakni algoritma naive bayes yang merupakan salah satu algoritma klasifikasi untuk dataset yang berukuran besar. Algoritma naive bayes didasarkan pada teorema bayes dengan asumsi bahwa setiap fitur bersifat independen satu sama lain sehingga setiap fitur memberikan kontribusi terhadap hasil prediksi secara terpisah tanpa memperhatikan hubungan antar fitur.

Dataset yang digunakan terbagi menjadi data training dan data testing, yang dimana ada 3 perbandingan untuk data training yang digunakan untuk melatih model kalsifikasi dan 3 perbandingan data testing yang digunakan untuk mengevaluasi performa dari model yang terbentuk dengan proporsi data training dan testing sebesar 63% dan 33%, 75% dan 25%, dan 80% dan 20%. Setelah model dilatih dengan data training, data testing digunakan untuk menguji performa model yang kemudian model akan membuat prediksi terhadap data testing. Performa model diukur menggunakan metrik evaluasi

seperti akurasi, presisi, *recall* dan F1-Score untuk melihat performa model yang dihasilkan. Sebaran dataset yang digunakan juga terlihat sangat jelas bahwa data memang tidak seimbang karena penyebaran datanya yang tidak merata antara kelas pinjaman sudah lunas dibayarkan dan belum lunas dibayarkan. Namun setelah penerapan teknik SMOTE terlihat bahwa data dengan kelas lunas dan belum sudah mulai menyebar dengan rata.

Dalam kasus data yang tidak seimbang, akurasi dapat menjadi metrik yang kurang tepat digunakan, dikarenakan model akan selalu memprediksi data dengan kelas yang mayoritas akan selalu memiliki akurasi lebih tinggi sehingga pada penelitian ini fokus yang dijadikan acuan untuk melihat performa model yang dihasilkan adalah dengan menggunakan metrik *recall*, presicion dan F1-Score. Berdasarkan model yang dihasilkan dengan algoritma naïve bayes dengan menggunakan teknik *k-fold cross validation* pada data yang belum seimbang, diperoleh nilai *recall, presicion* dan F1-Scorenya lebih tinggi dibandingkan sebelum penerapan teknik SMOTE.

Pada penelitian oleh (Chieka & Kurniasih, 2023) menunjukkan bahwa setelah penerapan SMOTE akurasi model naive bayes menurun dari 82% menjadi 69%. Tetapi pada recall naik dari 56% jadi 66%. Artinya peningkatan recall penting karena mampu meminimalkan resiko kesalahan dalam mendeteksi pelanggan berpotensi churn. Artinya SMOTE berhasil menyeimbangkan datanya. Begitupun dengan penelitian ini, nilai akurasinya menurun setelah penerapan SMOTE namun meningkat pada nilai recall, presicion dan F1-Score. Artinya, meskipun akurasinya turun namun performa dari modelnya menjadi lebih baik.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian yang dilakukan, maka dapat bahwa:

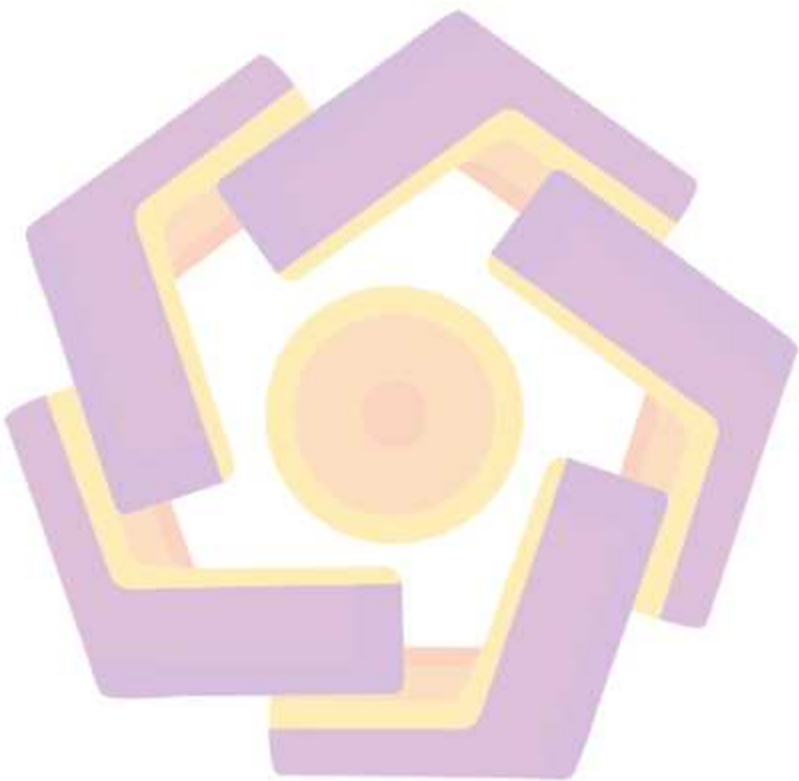
1. Algoritma naïve bayes mampu mengklsifikasikan data yang tidak seimbang, namun hasil yang diperoleh kurang tepat dikarenakana hasil klasifikasi akan lebih merujuk kepada kelas yang mayoritas sehingga ketika akan dilakukan prediksi maka model tersebut tidak mampu memprediksi kelas minoritas dengan benar
2. Metode SMOTE merupakan salah satu teknik untuk menyeimbangkan data atau kelas dalam data yang tidak seimbang, dalam penelitian ini kelas 0 dan kelas 1 memiliki proporsi sebersar 0.839% untuk kelas 0 dan 0.161% untuk kelas tidak 1, namun setelah diterapkan metode SMOTE, proporsi kelas 0 dan 1 menjadi seimbang baik pada 0.63, 0.75 dan 0.80
3. Baik dari nilai recall, presicion dan F1-Score meningkat setelah data yang tidak seimbang diseimbangkan dengan teknik SMOTE, hal itu berarti bahwa dalam data numerik yang digunakan, teknik SMOTE mampu meningkatkan performa model Naive Bayes.
4. Metode K-Fold Cross Validation merupakan metode untuk yang efektif untuk meningkatkan performa algoritma dengan cara membagi dataset menjadi k subset (fold) untuk meningkatkan penilaian kerja model. Hal tersebut terlihat jelas dikarenakan hasil prediksinya jauh lebih seimbang dibandingkan dengan sebelum penerapan *k-fold cross validation*.

5.2 Saran

Beberapa saran untuk pengembangan penelitian ini:

1. Menggunakan teknik selain SMOTE-ROSE dalam menyeimbangkan data, agar mengetahui teknik mana yang lebih baik pada data loan
2. Menggunakan algoritma lainnya selain dari algoritma naïve bayes, seperti random fores, svm atau decision tree, lalu komparasikan hasil

akurasi yang diperoleh untuk menentukan algortima mana yang lebih baik pada data pneumonia tersebut.



DAFTAR PUSTAKA

PUSTAKA BUKU

Pratiwi, D. A., Awangga, R. M., & Setyawan, M. Y. H. (2020). Seleksi Calon Kelulusan Tepat Waktu Mahasiswa Teknik Informatika Menggunakan Metode Naive Bayes (Vol. 1). Kreatif.

PUSTAKA MAJALAH/JURNAL ILMIAH/PROSIDING

Alfarizi, A. D., & Andri, A. (2021). Pemanfaatan data mining dalam memprediksi produksi pada PT Pupuk Sriwidjaja Palembang menggunakan algoritma regresi liniear berganda. *Jurnal Nasional Ilmu Komputer*, *2*(1), 51-63.

Azizah, H., Rintyarna, B. S., & Cahyanto, T. A. (2022). Sentimen Analisis Untuk Mengukur Kepercayaan Masyarakat Terhadap Pengadaan Vaksin Covid-19 Berbasis Bernoulli Naive Bayes. *BIOS: Jurnal Teknologi Informasi dan Rekayasa Komputer*, *3*(1), 23-29.

CA, N. A., Citra, D. H., Purnama, W., Nisa, C., & Kurnia, A. R. (2022). The Implementation of Naïve Bayes Algorithm for Sentiment Analysis of Shopee Reviews on Google Play Store Implementasi Algoritma Naive Bayes untuk Analisis Sentimen Ulasan Shopee pada Google Play Store. Journal Homepage: https://journal. irpi. or. id/index. php/malcom, 2(1), 47-54.

Duan, F., Zhang, S., Yan, Y., & Cai, Z. (2022). An oversampling method of unbalanced data for mechanical fault diagnosis based on MeanRadius-SMOTE. Sensors, 22(14), 5166.

Fathoni, F. M., Putra, C. A., & Nurlaili, A. L. (2024). Klasifikasi Penyakit Daun Anggur menggunakan metode k-nearest neighbor Berdasarkan Gray level co-occurrence matrix. Biner: Jurnal Ilmiah Informatika dan Komputer, 3(1), 8-15.

Hairani, H., Anggrawan, A., & Priyanto, D. (2023). Improvement performance of the random forest method on unbalanced diabetes data classification

using Smote-Tomek Link. JOIV: international journal on informatics visualization, 7(1), 258-264.

Haryanti, M. F., Fauzi, A., Jelita, A. A., Setiyowati, A., Octarina, A., Edina, E. P., ... & Fitriana, S. (2024). Pengaruh Data Mining, Strategi Perusahaan, Terhadap Laporan Kinerja Perusahaan. Jurnal Portofolio: Jurnal Manajemen dan Bisnis, 3(1), 71-90.

Kurniadi, D., Nuraeni, F., & Firmansyah, M. (2023). Klasifikasi Masyarakat Penerima Bantuan Langsung Tunai Dana Desa Menggunakan Naïve Bayes dan SMOTE. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, *10*(2).

Isyara, K., & Kurniasih, A. (2023, December). Penggunaan Metode SMOTE pada Naïve Bayes Gaussian untuk Klasifikasi Mahasiswa Drop Out. In *Prosiding Seminar Nasional Mahasiswa Bidang Ilmu Komputer dan Aplikasinya* (Vol. 4, No. 2, pp. 616-623).

Nabila, Z., Isnain, A. R., Permata, P., & Abidin, Z. (2021). Analisis data mining untuk clustering kasus covid-19 di Provinsi Lampung dengan algoritma k-means. *Jurnal Teknologi Dan Sistem Informasi*, *2*(2), 100-108.

Nursyahfitri, R., Rozikin, C., & Adam, R.I. (2022). Penerapan Metode SMOTE dalam Klasifikasi Daerah Rawan Banjir di Karawang Menggunakan Algoritma Naive Bayes. Jurnal Sistem dan Teknologi Informasi (JustIN).

Pikir Claudia, S. G. (2021). *Analisis Sentimen Kuliah Online selama pandemi Covid-19 Menggunakan Algoritma Naive Bayes* (Doctoral dissertation, sistem informasi).

Pulungan, M. P., Purnomo, A., & Kurniasih, A. (2023). Penerapan SMOTE untuk Mengatasi Imbalance Class dalam Klasifikasi Kepribadian MBTI Menggunakan Naive Bayes Classifier. *Jurnal Teknologi Informasi dan Ilmu Komputer*, *10*(7), 1493-1502.

Pambudi, L. (2023). Penerapan Data Mining Untuk Menganalisis Kepuasan Peserta Program Indonesia Bisa Baca Quran Menggunakan Algoritma Decision

Tree (C4. 5) Berbasis Web. *Jurnal Teknorama (Informatika dan Teknologi El Rahma)*, *1*(1), 14-20.

Urva, G., Albanna, I., Sungkar, M. S., Gunawan, I. M. A. O., Adhicandra, I., Ramadhan, S., ... & Junaidi, S. (2023). *PENERAPAN DATA MINING DI BERBAGAI BIDANG: Konsep, Metode, dan Studi Kasus*. PT. Sonpedia Publishing Indonesia.

Setiawan, D. F., Erlansari, A., & Sari, J. P. (2022). Penerapan Data Mining pada Review TIX ID Menggunakan Naïve Bayes Berbasis SMOTE & PSO. Jurnal Eksplora Informatika, 12(1), 37-45.

Sulistiyowati, N., & Jajuli, M. (2020). Integrasi naive bayes dengan teknik sampling SMOTE untuk menangani data tidak seimbang. Nuansa Informatika, 14(1), 34-37.

Susana, H. (2022). Penerapan Model Klasifikasi Metode Naive Bayes Terhadap Penggunaan Akses Internet. *Jurnal Riset Sistem Informasi Dan Teknologi Informasi (JURSISTEKNI)*, *4*(1), 1-8.

Sobri, A., Satrianansyah, S., & Noverendi, B. A. (2023). Implementasi Sistem Pakar Diagnosis Penyakit Pada Ibu Hamil Menggunakan Metode Naïve Bayes. *Journal of Information System Research (JOSH)*, *4*(4), 1245-1252.

Wang, S., Dai, Y., Shen, J., & Xuan, J. (2021). Research on expansion and classification of imbalanced data based on SMOTE algorithm. Scientific reports, 11(1), 24039.

Weni, I., Arsa, D., & Fahreza, A. (2022). *Analisis Sentimen Opini Publik Terhadap Pariwisata Di Masa Pandemi Covid-19 Menggunakan Metode Naive Bayes* (Doctoral dissertation, Sistem Informasi).

Widya Amalia Putri, R. I. S. W. A. N. D. H. A. (2023). Evaluasi Performa Synthetic Minority Oversampling Technique (Smote) Untuk Mengatasi Klasifikasi Data Tidak Seimbang Pada Metode K-Nearest Neighbor (Knn) Dan Support Vector Machine (Svm).

LAMPIRAN

Lampiran 1 data yang digunakan

policy	purpose	int.rate	installment	log.annual.inc	dti	fico	days.with.cr.line	revol.bal	revol.util	inq.last.6mths	delinq.2yrs	pub.rec	not.fully.paid
1	debt_consolidation	0.1189	829.1	11.35041	19.48	737	5639.958	28854	52.1	0	0	0	0
1	credit_card	0.1071	228.22	11.08214	14.29	707	2760	33623	76.7	0	0	0	0
1	debt_consolidation	0.1357	366.86	10.37349	11.63	682	4710	3511	25.6	1	0	0	0
1	debt_consolidation	0.1008	162.34	11.35041	8.1	712	2699.958	33667	73.2	1	0	0	0
1	credit_card	0.1426	102.92	11.29973	14.97	667	4066	4740	39.5	0	1	0	0
1	credit_card	0.0788	125.13	11.90497	16.98	727	6120.042	50807	51	0	0	0	0
1	debt_consolidation	0.1496	194.02	10.71442	4	667	3180.042	3839	76.8	0	0	1	1
1	all_other	0.1114	131.22	11.0021	11.08	722	5116	24220	68.6	0	0	0	1
1	home_improvement	0.1134	87.19	11.40756	17.25	682	3989	69909	51.1	1	0	0	0
1	debt_consolidation	0.1221	84.12	10.20359	10	707	2730.042	5630	23	1	0	0	0
1	debt_consolidation	0.1347	360.43	10.43412	22.09	677	6713.042	13846	71	2	0	1	0
1	debt_consolidation	0.1324	253.58	11.83501	9.16	662	4298	5122	18.2	2	1	0	0
1	debt_consolidation	0.0859	316.11	10.93311	15.49	767	6519.958	6068	16.7	0	0	0	0
1	small_business	0.0714	92.82	11.51293	6.5	747	4384	3021	4.8	0	1	0	0
1	debt_consolidation	0.0863	209.54	9.487972	9.73	727	1559.958	6282	44.6	0	0	0	0
1	major_purchase	0.1103	327.53	10.73892	13.04	702	8159.958	5394	53.4	1	0	0	0
1	all_other	0.1317	77.69	10.52277	2.26	672	3895.958	2211	88.4	0	0	0	0
1	credit_card	0.0894	476.58	11.60824	7.07	797	6510.958	7586	52.7	1	0	0	0
1	debt_consolidation	0.1039	584.12	10.49127	3.8	712	2760	8311	59.8	0	0	0	0
1	major_purchase	0.1513	173.65	11.0021	2.74	667	1126.958	591	84.4	3	0	0	0
1	all_other	0.08	188.02	11.22524	16.08	772	4888.958	29797	23.2	1	0	0	0

1	all_other	0.0863	474.42	10.81978	2.59	797	11951	5656	27.6	0	0	0	0
1	credit_card	0.1355	339.6	11.51293	7.94	662	1939.958	21162	57.7	0	0	0	0
1	credit_card	0.0788	484.85	11.73607	7.05	782	5640.042	16931	34.6	1	0	0	0
1	debt_consolidation	0.1229	320.19	11.26446	8.8	672	3760.958	4822	58.1	0	0	1	0
1	all_other	0.0901	159.03	12.42922	10	712	1551.958	14354	36.6	0	2	0	0
1	all_other	0.0743	155.38	11.08214	0.28	802	4649.958	1576	5.7	1	0	0	0
1	debt_consolidation	0.1375	255.43	9.998798	14.29	662	1318.958	4175	51.5	0	1	0	0
1	all_other	0.0743	155.38	12.20607	0.28	772	4516.958	3164	13.7	0	0	0	0
1	all_other	0.0743	155.38	12.20607	3.72	812	6778.958	85607	0.7	0	0	0	0
...	...	...	...	...	...	...	...	...	...	....	....	..	....
...	...	...	...	...	...	...	...	...	...	....	....	..	....
...	...	...	...	...	...	...	...	...	...	....	....	..	....
0	debt_consolidation	0.1311	192.35	11.0021	5.72	692	10236	2077	1.1	4	3	0	1
0	debt_consolidation	0.1025	323.85	11.17044	13.98	732	3750	6639	43.6	4	0	0	0
0	all_other	0.1348	468.16	10.97288	15.16	702	3180.042	14466	50.2	5	0	0	0
0	debt_consolidation	0.1311	236.22	10.85707	13.11	687	3270	8076	36.1	4	0	0	0
0	debt_consolidation	0.1348	271.4	11.19821	12.44	682	4230.042	11706	34.9	4	0	0	0
0	all_other	0.0788	78.21	10.81978	20.38	732	7410.042	16150	32.6	4	0	0	0
0	debt_consolidation	0.157	262.59	10.66896	21.35	662	4260.042	9659	84.7	4	0	0	0
0	educational	0.1607	147.82	9.862666	16.19	667	1260.042	4445	53.5	4	0	0	1
0	home_improvement	0.1607	87.99	10.77896	14.2	667	4080	1530	36.4	7	0	0	1
0	home_improvement	0.2164	729.7	11.87757	8.63	667	8280.042	55442	66.9	9	0	1	1
0	all_other	0.1459	137.86	10.08581	1.15	732	1230.042	972	11.3	5	0	0	0
0	home_improvement	0.1348	508.87	11.73607	16.85	707	7440.042	206877	92.5	1	0	0	1
0	debt_consolidation	0.1311	337.45	10.69194	23.62	702	3780.042	6255	56.9	5	2	0	0
0	debt_consolidation	0.1385	545.67	11.77529	10.8	697	4110	197716	74.9	4	0	0	0

0	small_business	0.1533	870.71	11.84223	16.16	707	4230.042	56909	49.8	5	0	0	0
0	home_improvement	0.1311	674.9	12.29225	9.94	717	5730.042	39576	27.7	5	0	0	1
0	debt_consolidation	0.1385	136.42	11.0021	18.18	677	3423.042	15301	85	4	0	0	0
0	credit_card	0.1025	466.35	12.20607	13.97	722	6120.042	338935	78.3	2	0	0	0
0	debt_consolidation	0.1533	696.57	11.8056	17.21	682	2790.042	38578	86.9	4	0	0	0
0	credit_card	0.1273	688.11	11.31447	21.13	732	5881	35227	54.3	5	0	0	0
0	all_other	0.1867	547.36	11.40756	15.76	667	10050.04	13255	88.4	7	0	0	0
0	all_other	0.0788	115.74	10.9991	10.17	722	4410	11586	61.6	4	0	0	0
0	debt_consolidation	0.1348	508.87	10.93311	17.76	717	3870.042	8760	28.2	6	0	0	0
0	debt_consolidation	0.1099	556.5	11.22524	17.84	727	6840.042	18753	29	4	0	0	1
0	all_other	0.1385	511.56	12.32386	12.33	687	6420.042	385489	51.2	4	0	0	0
0	all_other	0.1459	396.35	10.30895	21.04	697	3390	26117	78.4	6	0	0	1
0	all_other	0.2164	551.08	11.0021	24.06	662	1800	16441	49.8	9	0	0	1
0	all_other	0.1311	101.24	10.9682	8.23	687	2790.042	1514	13.8	5	0	0	0
0	all_other	0.1979	37.06	10.64542	22.17	667	5916	28854	59.8	6	0	1	0
0	home_improvement	0.1426	823.34	12.42922	3.62	722	3239.958	33575	83.9	5	0	0	1
0	all_other	0.1671	113.63	10.64542	28.06	672	3210.042	25759	63.8	5	0	0	1
0	all_other	0.1568	161.01	11.22524	8	677	7230	6909	29.2	4	0	1	1
0	debt_consolidation	0.1565	69.98	10.11047	7.02	662	8190.042	2999	39.5	6	0	0	1
0	all_other	0.1461	344.76	12.18075	10.39	672	10474	215372	82.1	2	0	0	1
0	all_other	0.1253	257.7	11.14186	0.21	722	4380	184	1.1	5	0	0	1
0	debt_consolidation	0.1071	97.81	10.59663	13.09	687	3450.042	10036	82.9	8	0	0	1
0	home_improvement	0.16	351.58	10.81978	19.18	692	1800	0	3.2	5	0	0	1
0	debt_consolidation	0.1392	853.43	11.26446	16.28	732	4740	37879	57	6	0	0	1

Lampiran 2 syntax

1). syntax 67% : 33%

```
from google.colab import files
import pandas as pd

# Upload file
uploaded = files.upload()

# Read the Excel file (replace with the actual filename after upload)
df = pd.read_csv("dsnosmote.csv", delimiter=';')

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report, confusion_matrix

# Replace 'TargetColumn' with your actual target column name
X = df.drop("not.fully.paid", axis=1)
y = df["not.fully.paid"]

if y.dtype == "object":
    le = LabelEncoder()
    y = le.fit_transform(y)

# Encode target if categorical

# Convert columns with comma decimals to float
for col in X.columns:
    if X[col].dtype == 'object':
        try:
            X[col] = X[col].str.replace(',', '.', regex=False).astype(float)
        except ValueError:
            pass # Keep the column as is if conversion fails


# 3. Split dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=42)
```

```
# 4. Choose model
model = GaussianNB()

# 5. Train model
model.fit(X_train, y_train)

# 6. Evaluate
y_pred = model.predict(X_test)
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))

# 7. Predict new data (example)
# Replace with your actual new data with 19 features
new_data = X_test # Using X_test as an example
print("Prediction:", model.predict(new_data))
```

DENGAN SMOTE TANPA VALIDASI

```
# ==========================================
# 1. Install imbalanced-learn in Colab
# ==========================================
!pip install imbalanced-learn

# ==========================================
# 2. Import Libraries
# ==========================================
from google.colab import files
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report, confusion_matrix
from imblearn.over_sampling import SMOTE

# ==========================================
# 5. Apply SMOTE
```

```
# ==========================================
sm = SMOTE(random_state=42)
X_resampled, y_resampled = sm.fit_resample(X, y)

print("\nAfter SMOTE Class Distribution:")
print(pd.Series(y_resampled).value_counts())

# ==========================================
# 6. Train/Test Split
# ==========================================
Xr_train, Xr_test, yr_train, yr_test = train_test_split(
    X_resampled, y_resampled, test_size=0.33, random_state=42
)

# ==========================================
# 7. Train Naive Bayes
# ==========================================
model = GaussianNB()
model.fit(Xr_train, yr_train)

# ==========================================
# 8. Evaluate
# ==========================================
yr_pred = model.predict(Xr_test)

print("\nConfusion Matrix:")
print(confusion_matrix(yr_test, yr_pred))

print("\nClassification Report:")
print(classification_report(yr_test, yr_pred))
```

CROSS VALIDATION

```
from sklearn.model_selection import StratifiedKFold, cross_val_score
from sklearn.preprocessing import LabelEncoder
from sklearn.naive_bayes import GaussianNB
from imblearn.pipeline import Pipeline
from imblearn.over_sampling import SMOTE
cv = StratifiedKFold(n_splits=3, shuffle=True, random_state=42)
```

```
# ==========================================
# 2. Build Pipeline (SMOTE + Naive Bayes)
# ==========================================
pipeline = Pipeline([
    ("smote", SMOTE(random_state=42)),
    ("nb", GaussianNB())
])

# Evaluate using accuracy, recall, f1
from sklearn.model_selection import cross_validate

scoring = ["accuracy", "precision_weighted", "recall_weighted", "f1_weighted"]

scores = cross_validate(pipeline, X, y, cv=cv, scoring=scoring, return_train_score=False)

# ==========================================
# 4. Show Results
# ==========================================
import numpy as np

print("Average Results (5-Fold CV):")
for metric in scoring:
    print(f"{metric}: {np.mean(scores['test_'+metric]):.4f}")
```

DENGAN SMOTE DAN CROSS VALIDASI

```
# ==========================================
# 2. Build Pipeline (SMOTE + Naive Bayes)
# ==========================================
pipeline = Pipeline([
    ("smote", SMOTE(random_state=42)),
    ("nb", GaussianNB())
])

# ==========================================
# 3. K-Fold Cross Validation (Stratified)
# ==========================================
cv = StratifiedKFold(n_splits=3, shuffle=True, random_state=42)
```

```
# Get predictions across all folds
from sklearn.model_selection import cross_val_predict
y_pred = cross_val_predict(pipeline, X, y, cv=cv)

# ========================================
# 4. Per-Class Metrics
# ========================================
print("Confusion Matrix:")
print(confusion_matrix(y, y_pred))

print("\nClassification Report (Per Class):")
print(classification_report(y, y_pred, digits=4))
```

TANPA SMOTE DENGAN CROSS VALIDASI

```
# ========================================
# 4. Pipeline without SMOTE
# ========================================
pipeline_no_smote = Pipeline([
    ("nb", GaussianNB())
])

y_pred_no_smote = cross_val_predict(pipeline_no_smote, X, y, cv=cv)

print("\n========== WITHOUT SMOTE ==========")
print("Confusion Matrix:")
print(confusion_matrix(y, y_pred_no_smote))
print("\nClassification Report:")
print(classification_report(y, y_pred_no_smote, digits=4))
```

VISUALISASI

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import StratifiedKFold, cross_val_predict
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.naive_bayes import GaussianNB
```

```
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler

# Example pipeline (scaler + Naive Bayes)
pipeline = Pipeline([
    ('scaler', StandardScaler()),
    ('classifier', GaussianNB())
])

# Perform Stratified K-Fold Cross Validation
skf = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)

# Get cross-validated predictions
y_pred = cross_val_predict(pipeline, X, y, cv=skf)

# Confusion Matrix
cm = confusion_matrix(y, y_pred)
plt.figure(figsize=(6,5))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=np.unique(y),
yticklabels=np.unique(y))
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix (Cross-Validated Predictions)")
plt.show()

# Classification Report (Precision, Recall, F1 per class)
print("Classification Report:")
print(classification_report(y, y_pred))
```

2). syntax 75% : 25%

```
from google.colab import files
import pandas as pd

# Upload file
uploaded = files.upload()

# Read the Excel file (replace with the actual filename after upload)
df = pd.read_csv("dsnosmote.csv", delimiter=';')

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report, confusion_matrix

# Replace 'TargetColumn' with your actual target column name
X = df.drop("not.fully.paid", axis=1)
y = df["not.fully.paid"]

if y.dtype == "object":
    le = LabelEncoder()
    y = le.fit_transform(y)

# Encode target if categorical

# Convert columns with comma decimals to float
for col in X.columns:
    if X[col].dtype == 'object':
        try:
            X[col] = X[col].str.replace(',', '.', regex=False).astype(float)
        except ValueError:
            pass # Keep the column as is if conversion fails


# 3. Split dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)
```

```
# 4. Choose model
model = GaussianNB()

# 5. Train model
model.fit(X_train, y_train)

# 6. Evaluate
y_pred = model.predict(X_test)
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))

# 7. Predict new data (example)
# Replace with your actual new data with 19 features
new_data = X_test # Using X_test as an example
print("Prediction:", model.predict(new_data))
```

DENGAN SMOTE TANPA VALIDASI

```
# ==========================================
# 1. Install imbalanced-learn in Colab
# ==========================================
!pip install imbalanced-learn

# ==========================================
# 2. Import Libraries
# ==========================================
from google.colab import files
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report, confusion_matrix
from imblearn.over_sampling import SMOTE

# ==========================================
# 5. Apply SMOTE
# ==========================================
```

```
sm = SMOTE(random_state=42)
X_resampled, y_resampled = sm.fit_resample(X, y)

print("\nAfter SMOTE Class Distribution:")
print(pd.Series(y_resampled).value_counts())

# ==========================================
# 6. Train/Test Split
# ==========================================
Xr_train, Xr_test, yr_train, yr_test = train_test_split(
    X_resampled, y_resampled, test_size=0.25, random_state=42
)

# ==========================================
# 7. Train Naive Bayes
# ==========================================
model = GaussianNB()
model.fit(Xr_train, yr_train)

# ==========================================
# 8. Evaluate
# ==========================================
yr_pred = model.predict(Xr_test)

print("\nConfusion Matrix:")
print(confusion_matrix(yr_test, yr_pred))

print("\nClassification Report:")
print(classification_report(yr_test, yr_pred))
```

CROSS VALIDATION

```
from sklearn.model_selection import StratifiedKFold, cross_val_score
from sklearn.preprocessing import LabelEncoder
from sklearn.naive_bayes import GaussianNB
from imblearn.pipeline import Pipeline
from imblearn.over_sampling import SMOTE
cv = StratifiedKFold(n_splits=4, shuffle=True, random_state=42)
```

```
# ==========================================
# 2. Build Pipeline (SMOTE + Naive Bayes)
# ==========================================
pipeline = Pipeline([
    ("smote", SMOTE(random_state=42)),
    ("nb", GaussianNB())
])

# Evaluate using accuracy, recall, f1
from sklearn.model_selection import cross_validate

scoring = ["accuracy", "precision_weighted", "recall_weighted", "f1_weighted"]

scores = cross_validate(pipeline, X, y, cv=cv, scoring=scoring, return_train_score=False)

# ==========================================
# 4. Show Results
# ==========================================
import numpy as np

print("Average Results (5-Fold CV):")
for metric in scoring:
    print(f"{metric}: {np.mean(scores['test_'+metric]):.4f}")
```

DENGAN SMOTE DAN CROSS VALIDASI

```
# ==========================================
# 2. Build Pipeline (SMOTE + Naive Bayes)
# ==========================================
pipeline = Pipeline([
    ("smote", SMOTE(random_state=42)),
    ("nb", GaussianNB())
])

# ==========================================
# 3. K-Fold Cross Validation (Stratified)
# ==========================================
cv = StratifiedKFold(n_splits=4, shuffle=True, random_state=42)
```

```
# Get predictions across all folds
from sklearn.model_selection import cross_val_predict
y_pred = cross_val_predict(pipeline, X, y, cv=cv)

# ========================================
# 4. Per-Class Metrics
# ========================================
print("Confusion Matrix:")
print(confusion_matrix(y, y_pred))

print("\nClassification Report (Per Class):")
print(classification_report(y, y_pred, digits=4))
```

TANPA SMOTE DENGAN CROSS VALIDASI

```
# ========================================
# 4. Pipeline without SMOTE
# ========================================
pipeline_no_smote = Pipeline([
    ("nb", GaussianNB())
])

y_pred_no_smote = cross_val_predict(pipeline_no_smote, X, y, cv=cv)

print("\n========== WITHOUT SMOTE ==========")
print("Confusion Matrix:")
print(confusion_matrix(y, y_pred_no_smote))
print("\nClassification Report:")
print(classification_report(y, y_pred_no_smote, digits=4))
```

VISUALISASI

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import StratifiedKFold, cross_val_predict
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.naive_bayes import GaussianNB
from sklearn.pipeline import Pipeline
```

```
from sklearn.preprocessing import StandardScaler

# Example pipeline (scaler + Naive Bayes)
pipeline = Pipeline([
    ('scaler', StandardScaler()),
    ('classifier', GaussianNB())
])

# Perform Stratified K-Fold Cross Validation
skf = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)

# Get cross-validated predictions
y_pred = cross_val_predict(pipeline, X, y, cv=skf)

# Confusion Matrix
cm = confusion_matrix(y, y_pred)
plt.figure(figsize=(6,5))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=np.unique(y),
yticklabels=np.unique(y))
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix (Cross-Validated Predictions)")
plt.show()

# Classification Report (Precision, Recall, F1 per class)
print("Classification Report:")
print(classification_report(y, y_pred))
```

1). syntax 80% : 20%

```
from google.colab import files
import pandas as pd

# Upload file
uploaded = files.upload()

# Read the Excel file (replace with the actual filename after upload)
df = pd.read_csv("dsnosmote.csv", delimiter=';')

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report, confusion_matrix

# Replace 'TargetColumn' with your actual target column name
X = df.drop("not.fully.paid", axis=1)
y = df["not.fully.paid"]

if y.dtype == "object":
    le = LabelEncoder()
    y = le.fit_transform(y)

# Encode target if categorical

# Convert columns with comma decimals to float
for col in X.columns:
    if X[col].dtype == 'object':
        try:
            X[col] = X[col].str.replace(',', '.', regex=False).astype(float)
        except ValueError:
            pass # Keep the column as is if conversion fails


# 3. Split dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
# 4. Choose model
model = GaussianNB()

# 5. Train model
model.fit(X_train, y_train)

# 6. Evaluate
y_pred = model.predict(X_test)
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))

# 7. Predict new data (example)
# Replace with your actual new data with 19 features
new_data = X_test # Using X_test as an example
print("Prediction:", model.predict(new_data))
```

DENGAN SMOTE TANPA VALIDASI

```
# ==========================================
# 1. Install imbalanced-learn in Colab
# ==========================================
!pip install imbalanced-learn

# ==========================================
# 2. Import Libraries
# ==========================================
from google.colab import files
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report, confusion_matrix
from imblearn.over_sampling import SMOTE

# ==========================================
# 5. Apply SMOTE
# ==========================================
```

```
sm = SMOTE(random_state=42)
X_resampled, y_resampled = sm.fit_resample(X, y)

print("\nAfter SMOTE Class Distribution:")
print(pd.Series(y_resampled).value_counts())

# ==========================================
# 6. Train/Test Split
# ==========================================
Xr_train, Xr_test, yr_train, yr_test = train_test_split(
    X_resampled, y_resampled, test_size=0.2, random_state=42
)

# ==========================================
# 7. Train Naive Bayes
# ==========================================
model = GaussianNB()
model.fit(Xr_train, yr_train)

# ==========================================
# 8. Evaluate
# ==========================================
yr_pred = model.predict(Xr_test)

print("\nConfusion Matrix:")
print(confusion_matrix(yr_test, yr_pred))

print("\nClassification Report:")
print(classification_report(yr_test, yr_pred))
```

CROSS VALIDATION

```
from sklearn.model_selection import StratifiedKFold, cross_val_score
from sklearn.preprocessing import LabelEncoder
from sklearn.naive_bayes import GaussianNB
from imblearn.pipeline import Pipeline
from imblearn.over_sampling import SMOTE
cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
```

```
# ==========================================
# 2. Build Pipeline (SMOTE + Naive Bayes)
# ==========================================
pipeline = Pipeline([
    ("smote", SMOTE(random_state=42)),
    ("nb", GaussianNB())
])

# Evaluate using accuracy, recall, f1
from sklearn.model_selection import cross_validate

scoring = ["accuracy", "precision_weighted", "recall_weighted", "f1_weighted"]

scores = cross_validate(pipeline, X, y, cv=cv, scoring=scoring, return_train_score=False)

# ==========================================
# 4. Show Results
# ==========================================
import numpy as np

print("Average Results (5-Fold CV):")
for metric in scoring:
    print(f"{metric}: {np.mean(scores['test_'+metric]):.4f}")
```

DENGAN SMOTE DAN CROSS VALIDASI

```
# ==========================================
# 2. Build Pipeline (SMOTE + Naive Bayes)
# ==========================================
pipeline = Pipeline([
    ("smote", SMOTE(random_state=42)),
    ("nb", GaussianNB())
])

# ==========================================
# 3. K-Fold Cross Validation (Stratified)
# ==========================================
cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
```

```
# Get predictions across all folds
from sklearn.model_selection import cross_val_predict
y_pred = cross_val_predict(pipeline, X, y, cv=cv)

# ========================================
# 4. Per-Class Metrics
# ========================================
print("Confusion Matrix:")
print(confusion_matrix(y, y_pred))

print("\nClassification Report (Per Class):")
print(classification_report(y, y_pred, digits=4))
```

TANPA SMOTE DENGAN CROSS VALIDASI

```
# ========================================
# 4. Pipeline without SMOTE
# ========================================
pipeline_no_smote = Pipeline([
    ("nb", GaussianNB())
])

y_pred_no_smote = cross_val_predict(pipeline_no_smote, X, y, cv=cv)

print("\n========== WITHOUT SMOTE ==========")
print("Confusion Matrix:")
print(confusion_matrix(y, y_pred_no_smote))
print("\nClassification Report:")
print(classification_report(y, y_pred_no_smote, digits=4))
```

VISUALISASI

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import StratifiedKFold, cross_val_predict
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.naive_bayes import GaussianNB
from sklearn.pipeline import Pipeline
```

```
from sklearn.preprocessing import StandardScaler

# Example pipeline (scaler + Naive Bayes)
pipeline = Pipeline([
    ('scaler', StandardScaler()),
    ('classifier', GaussianNB())
])

# Perform Stratified K-Fold Cross Validation
skf = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)

# Get cross-validated predictions
y_pred = cross_val_predict(pipeline, X, y, cv=skf)

# Confusion Matrix
cm = confusion_matrix(y, y_pred)
plt.figure(figsize=(6,5))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=np.unique(y),
yticklabels=np.unique(y))
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix (Cross-Validated Predictions)")
plt.show()

# Classification Report (Precision, Recall, F1 per class)
print("Classification Report:")
print(classification_report(y, y_pred))
```