IMPLEMENTASI ALGORITMA *NAÏVE BAYES* UNTUK ANALISIS SENTIMEN PADA ULASAN APLIKASI SAMSAT DIGITAL ONLINE (SIGNAL)

SKRIPSI

OLEH
MIRA PERMATA SARI
NIM. 210601110105

PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2025

IMPLEMENTASI ALGORITMA *NAÏVE BAYES* UNTUK ANALISIS SENTIMEN PADA ULASAN APLIKASI SAMSAT DIGITAL ONLINE (SIGNAL)

SKRIPSI

Diajukan Kepada
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Matematika (S.Mat)

Oleh
MIRA PERMATA SARI
NIM. 210601110105

PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2025

IMPLEMENTASI ALGORITMA *NAÏVE BAYES* UNTUK ANALISIS SENTIMEN PADA ULASAN APLIKASI SAMSAT DIGITAL ONLINE (SIGNAL)

SKRIPSI

Oleh
Mira Permata Sari
NIM. 210601110105

Telah Disetujui Untuk Diuji

Malang, 21 Februari 2025

Dosen Pembimbing I	Dosen Pembimbing II
Hisyam Fahmi, M.Kom.	Ach. Nashichuddin, M.A.
NIP. 19890727 201903 1 018	NIP. 19730705 200003 1 002

Mengetahui,
Ketua Program Studi Matematika

Dr. Elly Susanti, M.Sc.
NIP. 19741129 200012 2 005

IMPLEMENTASI ALGORITMA *NAÏVE BAYES* UNTUK ANALISIS SENTIMEN PADA ULASAN APLIKASI SAMSAT DIGITAL ONLINE (SIGNAL)

SKRIPSI

Oleh
Mira Permata Sari
NIM. 210601110105

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Matematika (S. Mat)

Tanggal 19 Mei 2025

Ketua Penguji	: Dr. Fachrur Rozi, M.Si	.....
Anggota Penguji 1	: Ria Dhea Layla Nur Karisma, M.Si	.....
Anggota Penguji 2	: Hisyam Fahmi, M.Kom	.....
Anggota Penguji 3	: Ach. Nashichuddin, M.A	.....

Mengetahui,
Ketua Program Studi Matematika

Dr. Elly Susanti, M.Sc.
NIP. 19741129 200012 2 005

PERNYATAAN KEASLIAN TULISAN

Saya bertanda tangan dibawah ini

Nama	:	Mira Permata Sari
NIM	:	210601110105
Program Studi	:	Matematika
Fakultas	:	Sains dan Teknologi
Judul Skripsi	:	Implementasi Algoritma *Naïve Bayes* untuk Analisis Sentimen pada Aplikasi Samsat Digital Online (SIGNAL)

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini merupakan hasil karya sendiri, bukan pengambilan tulisan atau pemikiran orang lain yang saya akui sebagai pemikiran saya, kecuali dengan mencantumkan sumber cuplikan pada daftar rujukan di halam terakhir. Apabila dikemudian hari terbukti skripsi ini adalah hasil jiplakan atau tiruan, maka saya bersedia menerima sanksi yang berlaku atas perbuatan tersebut.

Malang, 19 Mei 2025

Mira Permata Sari
NIM. 210601110105

MOTO

"Maka sesungguhnya bersama kesulitan ada kemudahan."

(QS. Al-Insyirah:5)

HALAMAN PERSEMBAHAN

Dengan penuh rasa syukur kepada Allah SWT, skripsi ini saya persembahkan kepada Ayahanda Sujono dan Ibunda Sulik, yang selalu menjadi sumber kekuatan, doa, dan kasih sayang dalam setiap langkah hidup penulis. Terima kasih atas segala pengorbanan, bimbingan, serta cinta tanpa syarat yang telah Ayah dan Ibu berikan. Setiap pencapaian ini tidak akan pernah terwujud tanpa doa dan dukungan kalian. Semoga Allah SWT senantiasa memberikan kesehatan, kebahagiaan, dan keberkahan dalam setiap langkah kehidupan Ayah dan Ibu. Karya ini lahir dari cinta, doa, dan dukungan tanpa batas yang Ayah dan Ibu berikan. Semoga bisa menjadi setitik kebahagiaan dan ungkapan terima kasih atas segala pengorbanan yang tak terbalaskan.

KATA PENGANTAR

Puji syukur ke hadirat Allah SWT atas rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi ini dengan baik. Proposal ini disusun sebagai salah satu syarat dalam pemenuhan mata kuliah Skripsi Matematika di Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Penyusunan skripsi ini tidak terlepas dari bantuan, dukungan, dan motivasi dari berbagai pihak. Oleh karena itu, penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu selama proses ini. Ucapan terima kasih secara khusus penulis sampaikan kepada:

1. Bapak Prof. Dr. H. M. Zainuddin, M.A., selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang, atas kepemimpinan dan arahan yang luar biasa.
2. Ibu Prof. Dr. Hj. Sri Harini, M.Si, selaku Dekan Fakultas Sains dan Teknologi, atas dukungan dan fasilitas yang diberikan selama proses studi.
3. Ibu Dr. Elly Susanti, S.Pd., M.Sc, selaku Ketua Program Studi Matematika, atas motivasi dan arahannya selama masa studi penulis.
4. Bapak Hisyam Fahmi, M.Kom, selaku Dosen Pembimbing I, yang telah memberikan bimbingan, dukungan, serta arahan yang berharga selama penyusunan proposal ini. Semoga ilmu yang Bapak berikan menjadi amal jariyah yang terus bermanfaat.
5. Bapak Ach. Nashichuddin, M.A., selaku Dosen Pembimbing II, atas kesabaran, masukan, dan arahan yang sangat berharga. Semoga segala kebaikan yang diberikan menjadi pahala dan berkah yang melimpah.
6. Seluruh dosen Program Studi Matematika, Fakultas Sains dan Teknologi, yang telah memberikan ilmu, motivasi, dan inspirasi kepada penulis selama masa studi.
7. Ayah tercinta Sujono, Ibu tersayang Sulik, Kakak Nurwahyudi, Adik Richa, serta keluarga besar, yang senantiasa memberikan doa, dukungan, dan cinta kasih tanpa henti. Kehadiran keluarga besar yang selalu mendoakan dan memberikan motivasi menjadi sumber kekuatan terbesar bagi penulis dalam

menyelesaikan skripsi ini. Semoga Allah SWT senantiasa membalas segala kebaikan dan doa yang telah diberikan.

8. Sahabat penulis Dea Irma Firnanda, yang selalu menemani dalam suka dan duka selama proses penulisan skripsi ini. Terima kasih atas kebersamaan dan motivasinya.
9. Teman-teman Komputasi, yang selalu memberikan dukungan dan semangat. Semoga kebersamaan ini terus terjalin dengan baik.
10. Seluruh mahasiswa angkatan 2021 serta teman-teman penulis atas kebersamaan, dukungan, dan motivasi selama masa studi. Kehadiran kalian menjadi bagian penting dalam perjalanan ini.

Akhirnya, penulis berharap skripsi ini dapat memberikan manfaat bagi semua pihak yang membacanya. Penulis juga menyadari bahwa skripsi ini masih memiliki kekurangan, sehingga masukan dan saran yang membangun sangat diharapkan.

Malang, 19 Mei 2025

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR SIMBOL

μ	: Rata-rata data
σ^2	: Varians
χ^2	: *Chi-Square*
E_{ij}	: *Expected Count*

DAFTAR LAMPIRAN

ABSTRAK

Sari, Mira Permata. 2025. **Implementasi Algoritma *Naïve Bayes* Untuk Analisis Sentimen Pada Ulasan Aplikasi Samsat Digital Online (SIGNAL).** Skripsi. Program Studi Matematika Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (1) Hisyam Fahmi, M.Kom. (2) Ach. Nashichuddin, M.A.

Kata kunci: Analisis Sentimen, *Naïve Bayes, Feature Selection Chi-Square, SMOTE*

Aplikasi Samsat Digital Online (SIGNAL) merupakan platform yang bertujuan untuk mempermudah masyarakat dalam melakukan pembayaran pajak kendaraan secara online. Namun, aplikasi ini menghadapi tantangan terkait dengan kualitas analisis sentimen dari ulasan pengguna yang dapat memberikan gambaran mengenai kepuasan dan keluhan mereka terhadap layanan yang disediakan. Penelitian ini bertujuan untuk mengidentifikasi tingkat akurasi dalam melakukan analisis sentimen terhadap ulasan aplikasi SIGNAL menggunakan algoritma *Naïve Bayes*, serta menganalisis pengaruh penggunaan teknik feature selection dengan uji *Chi-Square* terhadap akurasi model. Data ulasan pengguna dikumpulkan melalui *web scraping* dari Google Play Store dan diproses melalui tahap *preprocessing*, TF-IDF, serta seleksi fitur dan SMOTE. Model yang digunakan mencapai akurasi 86,19%, dengan presisi tinggi untuk sentimen positif 87,17% dan negatif 97,17%, namun masih rendah pada sentimen netral 79,41%. Rata-rata akurasi *5-Fold Cross Validation* adalah 78,40%. Hasil ini memberi wawasan bagi pengembang SIGNAL dalam meningkatkan layanan berdasarkan sentimen pengguna.

ABSTRACT

Sari, Mira Permata. 2025. **Implementation of Naïve Bayes Algorithm for Sentiment Analysis on Online Digital Samsat Application Reviews (SIGNAL).** Thesis. Mathematics Study Program, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Advisors: (1) Hisyam Fahmi, M.Kom. (2) Ach. Nashichuddin, M.A.

Keywords: Sentiment Analysis, Naïve Bayes, Feature Selection Chi-Square, SMOTE

The Samsat Digital Online (SIGNAL) application is a platform designed to simplify the process of online vehicle tax payments for the public. However, the application faces challenges related to the quality of sentiment analysis of user reviews, which can provide insights into user satisfaction and complaints about the services offered. This study aims to identify the accuracy level in sentiment analysis of SIGNAL application reviews using the Naïve Bayes algorithm, as well as analyze the impact of feature selection with Chi-Square testing on the model's accuracy. User review data was collected through web scraping from the Google Play Store and processed through preprocessing, TF-IDF, feature selection, and SMOTE. The average training accuracy of 5-Fold Cross Validation is 78,40%. The model used achieved an accuracy of 86,19% on the testing data, with high precision for positive sentiment of 87,17% and negative sentiment of 97,17%, but is still low at neutral sentiment 79,41%. These results provide insights for SIGNAL developers in improving their services based on user sentiment.

مستخلص البحث

ساري ، ميرا بيرماتا. 2025. **تنفيذ خوارزمية (نايف بايز) لتحليل المشاعر في تعليقات تطبيق سامسات الرقمي عبر الإنترنت (*SIGNAL*)**. البحث الجامعي. قسم الرياضيات، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج.
المشرف: (1) هشام فهمي، ماجستير في علم الحاسوب (2) أحمد ناصح الدين، ماجستير في الدين

الكلمات الأساسية: تحليل المشاعر، نايف بايز، اختيار السمات باختبار*Chi-Square*، *SMOTE*

تطبيق سامسات الرقمي عبر الإنترنت (*SIGNAL*) هو منصة تهدف إلى تسهيل المجتمع في دفع ضريبة المركبات عبر الإنترنت. ومع ذلك، واجه هذا التطبيق تحديات تتعلق بجودة تحليل المشاعر من مراجعات المستخدمين التي يمكن أن تعكس رضاهم وشكاواهم تجاه الخدمات المقدمة. هذف هذا البحث إلى تحديد دقة تحليل المشاعر في مراجعات تطبيق *SIGNAL* باستخدام خوارزمية نايف بايز، وكذلك تحليل تأثير استخدام تقنية اختيار السمات باختبار *Chi-Square* على دقة النموذج. تم جمع بيانات مراجعات المستخدمين من خلال *Web Scraping* من متجر *Google Play* وتمت معالجتها من خلال مراحل المعالجة المسبقة، و*TF-IDF*، واختيار السمات، و *SMOTE*. وقد حقق النموذج المستخدم دقة بلغت ٨٦,١٩٪، بدقة عالية للمشاعر الإيجابية بنسبة ٨٧,١٧٪ والسلبية بنسبة ٩٧,١٧٪، ولكنها لا تزال منخفضة للمشاعر المحايدة بنسبة ٧٩,٤١٪. أما متوسط دقة التحقق المتقاطع (*5-Fold Cross Validation*) فقد بلغ ٧٨,٤٠٪. وتوفر هذه النتائج رؤية لمطوري تطبيق *SIGNAL* لتحسين الخدمات بناءً على مشاعر المستخدمين.

BAB I
PENDAHULUAN

1.1 Latar Belakang

Analisis sentimen atau yang dikenal dengan *opinion mining* adalah suatu proses mengolah data teks secara otomatis guna mengetahui opini yang terkandung dalam kalimat, termasuk apakah opini tersebut positif, negatif atau netral (Khoirul dkk., 2023). Untuk memahami berbagai respons masyarakat terhadap suatu layanan, diperlukan adanya analisis sentimen (Rizki & Utomo, 2022). Pada analisis sentimen ini, digunakan pemrosesan bahasa alami (NLP) untuk memproses data dalam jumlah besar dari ulasan, komentar, atau postingan di media sosial. Proses ini melibatkan pemahaman serta analisis konten yang menunjukkan sentimen tertentu dan diklasifikasikan berdasarkan kecenderungan sentimen positif, negatif, atau netral (Firdaus dkk., 2024). Tujuan dari analisis sentimen ini adalah untuk mendapatkan gambaran yang lebih akurat mengenai persepsi masyarakat dan mengidentifikasi area yang memerlukan peningkatan layanan (Fersellia dkk., 2023). Hasil dari analisis sentimen diharapkan dapat memberikan wawasan yang mendalam bagi penyedia layanan untuk memperbaiki dan menyempurnakan fitur atau aspek layanan yang belum memenuhi harapan masyarakat, sehingga kualitas dan kepuasan pengguna terhadap layanan tersebut dapat terus ditingkatkan (Sriani dkk., 2024).

Klasifikasi merupakan teknik memprediksi data, membuat prediksi nilai dari suatu data yang hasilnya telah ditemukan berasal dari data yang berbeda. Tujuan klasifikasi yaitu memprediksi nilai dari suatu variabel yang tidak diketahui dari variabel lain yang telah diberikan. Penggunaan teknik ini dapat digunakan

untuk mengevaluasi data nasabah dengan memilih algoritma klasifikasi yang sesuai (Franseda dkk., 2020). Melalui adanya klasifikasi sentimen pengguna aplikasi, pengembang, dan pihak terkait dapat dengan mudah mengidentifikasi pola dan aturan yang terbentuk dari algoritma yang digunakan.

Naïve Bayes adalah metode probabilistik yang digunakan untuk tujuan klasifikasi dan pemodelan prediktif. *Naïve Bayes* dikembangkan berdasarkan Teorema Bayes dan sering diaplikasikan dalam analisis teks, terutama untuk klasifikasi seperti analisis sentimen. Algoritma ini bekerja dengan menghitung *probabilitas* dari setiap kategori atau kelas berdasarkan kemunculan fitur atau kata tertentu dalam data, dengan asumsi bahwa setiap fitur bersifat independen (Khoirul dkk., 2023). *Naïve Bayes* menggunakan pembelajaran mesin berbasis *supervised learning*, di mana model dilatih dengan data berlabel untuk mengenali pola sentimen dalam teks. Meskipun asumsi independensi fitur mungkin tidak selalu realistis, pendekatan ini sangat efektif dan efisien untuk klasifikasi teks, memberikan prediksi yang akurat meskipun dengan data berukuran besar (Rahman dkk., 2024).

Algoritma *Naïve Bayes* memiliki beberapa kelebihan, di antaranya adalah kemampuannya untuk melakukan klasifikasi secara cepat dan efisien, bahkan pada dataset berukuran besar (Reza Satria & Adinugroho, 2020). Karena algoritma ini berbasis probabilistik dan menggunakan asumsi independensi antar fitur, proses komputasinya menjadi lebih ringan dibandingkan metode lain. Selain itu, *Naïve Bayes* memiliki performa yang baik pada tugas-tugas klasifikasi teks, seperti analisis sentimen, karena mampu menangani variasi kata yang sering muncul dalam data teks. Algoritma ini juga dikenal sederhana dan mudah diimplementasikan,

serta membutuhkan waktu pelatihan yang relatif singkat. Dengan kelebihan-kelebihan ini, *Naïve Bayes* menjadi pilihan yang sesuai bagi peneliti yang memerlukan klasifikasi teks dengan efisiensi tinggi dan hasil yang akurat.

Beberapa penelitian sebelumnya yang pernah dilakukan terkait klasifikasi yaitu penerapan metode *Naïve Bayes* untuk analisis sentimen pada media sosial. Sebagai contoh, penelitian Khoirul (2023) menerapkan algoritma *Naïve Bayes* untuk mengelompokkan sentimen positif dan negatif pada ulasan pengguna media sosial. Hasil penelitian ini menunjukkan bahwa algoritma *Naïve Bayes* mampu mencapai akurasi hingga 85% menggunakan *confusion matrix*, menunjukkan efisiensinya dalam klasifikasi sentimen pada data teks yang besar. Selain itu, penelitian lain oleh Rizky (2024) Penelitian ini menunjukkan bahwa algoritma *Naïve Bayes* efektif dalam analisis sentimen pada ulasan aplikasi SIREKAP, dengan akurasi 90% hasil ini membuktikan bahwa *Naïve Bayes* dalam mengklasifikasikan sentimen pengguna lebih efisien.

Penelitian lain yang relevan dilakukan oleh Satria (2020), yang mengkombinasikan algoritma *Naïve Bayes* dan C4.5 untuk analisis sentimen pada ulasan aplikasi *mobile*. Dalam penelitian ini, *Naïve Bayes* menghasilkan akurasi sebesar 85,3% pada data tanpa normalisasi *Levenshtein Distance*, sementara penambahan C4.5 dan *Levenshtein Distance* pada pengujian mencapai akurasi tertinggi sebesar 87,1%, menunjukkan peningkatan yang signifikan dalam klasifikasi sentimen aplikasi *mobile*.

Dari beberapa literatur tersebut, dapat disimpulkan bahwa meskipun *Naïve Bayes* memiliki akurasi yang lebih rendah dibandingkan algoritma lain seperti SVM, kemudahan implementasi dan efisiensinya membuatnya menjadi pilihan

populer dalam klasifikasi sentimen. Penggunaan algoritma *Naïve Bayes* dalam penelitian ini diharapkan dapat memberikan hasil klasifikasi yang memadai dengan proses yang lebih efisien dan sederhana. Selain itu, *Naïve Bayes* juga terbukti efektif dalam menangani dataset besar dan variatif, sehingga cocok untuk analisis sentimen pada ulasan pengguna aplikasi yang jumlahnya bisa sangat banyak.

Dalam beberapa tahun terakhir, digitalisasi telah mengubah berbagai aspek kehidupan, termasuk dalam layanan publik dan administrasi. Transformasi digital ini mencakup inovasi yang memudahkan masyarakat dalam mengakses layanan pemerintah secara online, salah satunya adalah aplikasi Samsat Digital Online (SIGNAL). Aplikasi ini diperkenalkan untuk membantu masyarakat dalam melakukan pembayaran pajak kendaraan bermotor tanpa perlu datang langsung ke kantor Samsat. Dengan sistem yang lebih mudah diakses, SIGNAL diharapkan mampu meningkatkan kenyamanan dan kepatuhan masyarakat dalam melaksanakan kewajiban perpajakan kendaraan mereka (Khoirul dkk., 2023). Berdasarkan data Badan Pusat Statistik (BPS), jumlah kendaraan yang terdaftar setiap tahunnya semakin meningkat, sehingga aplikasi seperti SIGNAL menjadi solusi yang relevan untuk mengurangi antrian dan mempercepat proses administrasi (Kepolisian Negara Republik Indonesia, 2023).

Meskipun demikian, aplikasi SIGNAL tidak luput dari tantangan dan menerima berbagai ulasan dari pengguna, mulai dari keluhan teknis hingga masalah terkait layanan pelanggan. Media sosial, Google Play Store, dan forum-forum daring menjadi wadah bagi masyarakat untuk menyampaikan pendapat mereka mengenai aplikasi ini. Ulasan-ulasan tersebut mencerminkan persepsi dan kepuasan pengguna, yang dapat dievaluasi untuk memahami pengalaman mereka secara lebih

mendalam. Analisis sentimen ini memberikan wawasan yang penting bagi pemerintah dan pengembang aplikasi untuk mengidentifikasi hal yang perlu ditingkatkan dalam layanan publik berbasis digital.

Dalam mengembangkan layanan publik berbasis teknologi seperti aplikasi Samsat Digital Online (SIGNAL), prinsip keadilan, efisiensi, dan kemudahan akses harus menjadi prioritas. Aplikasi ini bertujuan mempermudah pembayaran pajak kendaraan secara online, mengurangi antrian, dan mempercepat proses administrasi. Oleh karena itu, penting memastikan layanan ini adil dan merata bagi semua pengguna tanpa memandang latar belakang sosial, ekonomi, atau teknologi. Prinsip tersebut sejalan dengan ajaran Al-Qur'an tentang keadilan dan integritas dalam pelayanan kepada masyarakat. Salah satu ayat yang relevan adalah Surah Al-Ma'idah ayat 8 (Kemenag, 2024):

يٰٓاَيُّهَا الَّذِيْنَ اٰمَنُوْا كُوْنُوْا قَوّٰمِيْنَ لِلّٰهِ شُهَدَاۤءَ بِالْقِسْطِۖ وَلَا يَجْرِمَنَّكُمْ شَنَاٰنُ قَوْمٍ عَلٰٓى اَلَّا تَعْدِلُوْاۗ اِعْدِلُوْاۗ

هُوَ اَقْرَبُ لِلتَّقْوٰىۖ وَاتَّقُوا اللّٰهَۗ اِنَّ اللّٰهَ خَبِيْرٌۢ بِمَا تَعْمَلُوْنَ

"Hai orang-orang yang beriman hendaklah kamu jadi orang-orang yang selalu menegakkan (kebenaran) karena Allah, menjadi saksi dengan adil. Dan janganlah sekali-kali kebencianmu terhadap sesuatu kaum, mendorong kamu untuk berlaku tidak adil. Berlaku adillah, karena adil itu lebih dekat kepada takwa. Dan bertakwalah kepada Allah, sesungguhnya Allah Maha Mengetahui apa yang kamu kerjakan."

Dalam ayat ini, Allah Subhaanahu wa Ta'aala memerintahkan hamba-hamba-Nya untuk berlaku adil dalam setiap urusan, tanpa dipengaruhi oleh kebencian atau emosionalitas terhadap suatu kaum atau individu (Harun, 2013). Keadilan yang dimaksud mencakup semua aspek kehidupan, baik dalam hal hubungan sosial, kesaksian, maupun pelayanan publik. Contoh penerapan keadilan dalam konteks pelayanan publik adalah memastikan bahwa semua warga negara memiliki akses yang setara dan adil terhadap layanan, seperti dalam hal penggunaan

aplikasi Samsat Digital Online (SIGNAL), di mana setiap pengguna tanpa memandang latar belakangnya harus dilayani dengan adil dan efisien.

Berdasarkan pemaparan di atas, maka penulis tertarik untuk melakukan analisis sentimen pengguna dan potensi perbaikan aplikasi SIGNAL, dengan judul penelitian "Implementasi Algoritma *Naïve Bayes* untuk Analisis Sentimen terhadap Aplikasi Samsat Digital Online (SIGNAL)." Penelitian ini bertujuan untuk menganalisis sentimen pengguna terhadap aplikasi SIGNAL menggunakan Algoritma *Naïve Bayes* serta mengidentifikasi potensi perbaikan aplikasi berdasarkan hasil analisis sentimen tersebut. Dengan melakukan klasifikasi sentimen, penelitian ini diharapkan dapat memberikan rekomendasi yang berfokus pada peningkatan kualitas layanan dan kepuasan pengguna aplikasi SIGNAL, sehingga aplikasi ini dapat lebih optimal dalam melayani masyarakat.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana tingkat akurasi metode *Naïve Bayes Gaussian* dalam melakukan analisis sentimen pada aplikasi SIGNAL?
2. Bagaimana pengaruh *feature selection* menggunakan uji *Chi-Square* terhadap akurasi analisis sentimen pada ulasan pengguna aplikasi SIGNAL?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah dapat disimpulkan tujuan penelitian antara lain:

1. Untuk mengetahui tingkat akurasi metode *Naïve Bayes* dalam melakukan analisis sentimen pada ulasan pengguna aplikasi SIGNAL.
2. Untuk menganalisis pengaruh *feature selection* menggunakan uji *Chi-Square* dalam meningkatkan akurasi analisis sentimen pada ulasan pengguna aplikasi SIGNAL.

1.4 Manfaat Penelitian

Berdasarkan penelitian ini, dapat disimpulkan beberapa manfaat sebagai berikut:

1. Memberikan informasi yang mendalam mengenai kekuatan dan kelemahan aplikasi SIGNAL berdasarkan sentimen pengguna, sehingga pengembang dapat melakukan perbaikan dan pengembangan fitur yang lebih sesuai dengan kebutuhan masyarakat.
2. Membantu pihak pemerintah dalam mengevaluasi efektivitas aplikasi layanan publik berbasis digital, khususnya dalam meningkatkan kualitas pelayanan dan kepuasan masyarakat terhadap aplikasi SIGNAL.
3. Menyediakan referensi metodologis terkait penerapan algoritma *Naïve Bayes* yang dapat dijadikan dasar bagi penelitian lebih lanjut di bidang analisis teks atau evaluasi aplikasi SIGNAL.

1.5 Batasan Masalah

Penelitian ini memiliki beberapa batasan yang perlu diperhatikan untuk menjaga relevansi dan keakuratan hasil yang diperoleh, yaitu sebagai berikut:

1. Penelitian ini hanya menggunakan ulasan yang terdapat di Google Playstore sebagai sumber data, sehingga tidak mencakup ulasan atau pendapat dari platform lain seperti App Store atau media sosial.
2. Data ulasan menggunakan 5000 ulasan komentar berbahasa Indonesia yang memuat rating terhadap aplikasi SIGNAL.
3. Pelabelan data ulasan aplikasi SIGNAL dilakukan berdasarkan rating yang diberikan oleh pengguna.
4. Penelitian ini dibatasi pada data ulasan berlabel positif, negatif, dan netral. Ketiga kategori ini dipilih untuk memberikan gambaran yang lebih komprehensif tentang persepsi pengguna terhadap aplikasi SIGNAL.
5. Data ulasan yang digunakan dalam penelitian dibatasi pada periode tertentu, sehingga hasil analisis sentimen hanya mencerminkan sentimen pengguna dalam jangka waktu tersebut dan mungkin tidak sepenuhnya menggambarkan perubahan sentimen seiring waktu.

1.6 Definisi Istilah

Beberapa istilah yang digunakan dalam penelitian ini antara lain:

1. *Text Mining* : Teknik untuk mengekstrak informasi dari data teks yang tidak terstruktur, seperti ulasan pengguna.

2.	*Naïve Bayes*	:	Algoritma probabilistik yang menghitung kemungkinan kategori berdasarkan fitur dalam teks.
3.	*Feature Selection*	:	Proses memilih fitur yang paling relevan dari data untuk meningkatkan akurasi model.
4.	*Confusion Matrix*	:	Matriks untuk mengukur kinerja model klasifikasi dengan membandingkan prediksi dan hasil sebenarnya.
5.	Sentimen Analisis	:	Proses mengidentifikasi opini dalam teks sebagai positif, negatif, atau netral.
6.	TF-IDF	:	Metode pembobotan kata berdasarkan frekuensi kemunculan dalam dokumen dan seluruh korpus.
7.	*Wordcloud*	:	Representasi visual dari kata-kata yang sering muncul dalam teks, di mana ukuran kata menunjukkan frekuensinya.
8.	*Splitting* Data	:	Proses membagi data menjadi dua bagian, yaitu *training* data untuk melatih model, dan *testing* data untuk menguji performa model.
9.	*Training*	:	Proses melatih model untuk mengenali pola dalam data.
10.	*Testing*	:	Proses menguji performa model pada data baru yang belum dilihat saat *training*

BAB II
KAJIAN TEORI

2.1 Teori Pendukung

2.1.1 *Text mining*

Text mining merupakan salah satu metode yang digunakan untuk mengolah data mentah dan mengekstrak pola-pola tertentu dari kumpulan data tersebut, dengan tujuan menghasilkan pengetahuan yang lebih mendalam dan informasi yang bernilai (Kartikasari dkk., 2020). Teknik ini mengubah data yang awalnya dalam bentuk tidak terstruktur, seperti paragraf teks yang acak, menjadi data yang lebih terstruktur dalam bentuk teks yang dapat dianalisis lebih lanjut. Dengan pendekatan ini, *text mining* mampu menyederhanakan data kompleks menjadi bentuk yang lebih informatif dan bermakna bagi pengguna.

Proses *text mining* biasanya dimulai dengan pengumpulan berbagai dokumen dari sumber yang berbeda-beda, baik dari artikel, laporan, media sosial, maupun sumber digital lainnya. Setelah pengumpulan data, tahap berikutnya adalah *preprocessing*, di mana data diperiksa untuk memastikan formatnya sesuai dan karakter set yang digunakan konsisten. Proses ini penting untuk menghilangkan kesalahan format, simbol, atau karakter yang tidak relevan.

Tahap berikutnya adalah analisis teks, di mana data yang telah diproses dianalisis secara mendalam dengan tujuan mengidentifikasi pola-pola tersembunyi, topik-topik utama, atau informasi penting yang terkandung di dalamnya. Analisis ini dapat dilakukan menggunakan berbagai teknik, seperti tokenisasi, *stemming*, *lemmatization*, dan sebagainya (K. Kevin dkk., 2024). Untuk mengetahui detail pengolahannya sebagai berikut:

1. *Scraping*

   Scraping data adalah metode umum dalam text mining untuk mengumpulkan data dalam jumlah besar secara otomatis dari berbagai platform, seperti situs web atau media sosial, guna memperoleh informasi yang relevan. Dalam aplikasi praktisnya, scraping membantu mengumpulkan teks mentah untuk keperluan analisis lebih lanjut dalam skala besar dan waktu yang efisien (Gede dkk., 2024).

2. *Preprocessing*

   Proses *preprocessing* dalam *text mining* merupakan langkah penting yang meliputi pembersihan data, *case folding*, *normalized text*, *stopwords removal*, dan *stemming*. Proses ini bertujuan menyusun data menjadi lebih terstruktur sehingga memudahkan tahap analisis berikutnya. *Preprocessing* berperan dalam mengurangi *noise* dan memastikan data teks siap diolah menggunakan berbagai teknik analisis. Berikut beberapa tahapan *preprocessing* (Rizky Hanafi & Kurniawan, 2024)*:*

   a. *Case folding*

      Case folding adalah proses mengubah seluruh teks menjadi huruf kecil (*lowercase*) agar konsistensi data lebih mudah dijaga dan kata-kata yang seharusnya sama tidak dianggap berbeda hanya karena perbedaan kapitalisasi. Langkah ini membantu mengurangi variasi kata dan memudahkan analisis teks (Firdaus dkk., 2024).

   b. *Normalized text*

      Normalisasi teks melibatkan pengubahan kata-kata slang, singkatan, atau bentuk informal lainnya menjadi kata yang baku. Proses ini

penting untuk menjaga keseragaman dan memastikan setiap kata dapat diinterpretasikan dengan benar oleh algoritma analisis. Normalisasi bertujuan mengurangi *noise* dalam data dan memperbaiki kesalahan ejaan (Sriani dkk., 2024).

c. *Stopwords removal*

Stopwords removal adalah proses menghilangkan kata-kata umum seperti "dan", "atau", "juga", yang biasanya tidak memiliki makna signifikan dalam analisis teks (Salsabilla & Sulastri, 2022). Kata-kata ini sering kali muncul dalam teks, tetapi tidak memberikan kontribusi besar terhadap konteks atau sentimen yang akan dianalisis. Penghapusan *stopwords* membantu menyederhanakan data dan mengurangi kompleksitas analisis.

d. *Stemming*

Stemming adalah teknik untuk mengubah kata menjadi bentuk dasar atau akar katanya. Misalnya, kata "berjalan" diubah menjadi "jalan". *Stemming* membantu mengurangi variasi kata yang memiliki akar makna yang sama, sehingga meningkatkan efisiensi analisis teks dan mengurangi duplikasi kata dalam data (Yuni Ikhsanti dkk., 2021).

3. *Clean text* data yang hilang

Clean Text data yang hilang bertujuan untuk menangani data yang hilang setelah proses *preprocessing*, seperti penghapusan *stopwords* atau kata langka. Data yang hilang dapat mempengaruhi kualitas analisis, sehingga perlu diatasi dengan membersihkan data yang hilang. *Clean text*

membantu agar data yang diolah tidak *noise* serta dataset yang dimiliki lebih utuh dan informatif.

Hasil akhir dari keseluruhan proses *text mining* dapat berupa informasi berharga yang dapat diolah lebih lanjut, misalnya dalam bentuk ringkasan kata-kata atau kalimat dari sebuah dokumen. Informasi yang diperoleh juga dapat digunakan untuk mendukung pengambilan keputusan, menyusun laporan, hingga menggali wawasan yang lebih dalam dari data teks yang semula sulit untuk dipahami secara manual (Liawati dkk., 2024).

2.1.2 *Term Frequency-Inverse Document Frequency* (TF-IDF)

Term Frequency-Inverse Document Frequency (TF-IDF) adalah teknik statistik yang digunakan untuk menentukan tingkat kepentingan suatu kata dalam konteks tertentu. Teknik ini merupakan hasil perkalian dari dua komponen utama, yaitu *term frequency* dan *inverse document frequency*. TF-IDF berfungsi untuk mengatasi keterbatasan pencarian *boolean* pada indeks yang sering menghasilkan volume data yang besar dan tidak relevan, sehingga dapat mengurangi banyak *noise* dalam hasil pencarian (Septiani & Isabela, 2022). Penentuan frekuensi kemunculan kata menunjukkan *Term Frequency* (TF), sedangkan *Inverse Document Frequency* (IDF) menunjukkan bobot dari perkata. Perhitungan TF-IDF ditunjukkan pada persamaan berikut:

$$TF_{(W)} = \frac{Jumlah\ kemunculan\ sebuah\ kata\ dalam\ dokumen\ (d)}{Jumlah\ kata\ dalam\ satuan\ dokumen} \quad (2.1)$$

Perhitungan IDF dapat ditunjukkan dalam persamaan berikut ini:

$$IDF_{(w)} = \log_e \frac{Jumlah\ total\ dokumen}{Jumlah\ total\ dokumen\ dengan\ kata\ (w)} \quad (2.\ 2)$$

Selanjutnya, perhitungan TF-IDF dapat ditunjukan pada persamaan berikut ini:

$$TF - IDF_{(W)} = TF_{(w)}\ x\ IDF_{(w)} \quad (2.\ 3)$$

2.1.3 *Naïve Bayes*

Naïve Bayes adalah metode klasifikasi berbasis *probabilitas* yang menggunakan Teorema Bayes dengan asumsi independensi antar fitur (Syaripah dkk., 2024). Metode ini sering kali digunakan dalam klasifikasi teks dan berbagai aplikasi lain karena kesederhanaan dan keefektifannya. Inti dari *Naïve Bayes* adalah menghitung *probabilitas posterior* dari kelas tertentu berdasarkan fitur-fitur yang ada. Teorema Bayes menyatakan bahwa *probabilitas posterior* dari sebuah kelas C diberikan fitur X dapat dihitung dengan rumus berikut:

$$P(C \mid X) = \frac{P(X \mid C) \cdot P(C)}{P(X)} \quad (2.4)$$

Di mana:

$P(C \mid X)$ adalah *Probabilitas posterior* dari kelas C diberikan fitur *X*.

$P(X \mid C)$ adalah *Probabilitas likelihood* dari fitur *X* diberikan kelas C.

$P(C)$ adalah *Probabilitas prior* dari kelas C.

$P(X)$ adalah *Probabilitas* total dari fitur *X*.

Namun dalam proses klasifikasi, penyederhanaan sering dilakukan dengan mengabaikan $P(X)$ (peluang kemunculan kata/*evidence*). Hal ini karena $P(X)$ bersifat konstan untuk semua kelas C, sehingga tidak mempengaruhi perbandingan *probabilitas* relatif antar kategori. Dengan demikian, rumus posterior yang digunakan menjadi:

$$P(C \mid X) \propto P(X \mid C) \cdot P(C) \tag{2.5}$$

Dalam penelitian analisis sentimen, fitur X mewakili berbagai karakteristik atau kata-kata yang terdapat dalam teks yang akan diklasifikasikan. Fitur ini dapat berupa kata-kata individu, frekuensi kemunculan kata tertentu, atau bahkan fitur yang lebih kompleks seperti panjang kalimat atau jumlah tanda baca. Misalnya, dalam analisis sentimen pada ulasan produk, fitur-fitur X dapat mencakup kata-kata seperti "baik", "buruk", atau "mengecewakan" yang memiliki bobot atau pengaruh dalam menentukan apakah sentimen dari teks tersebut positif, negatif atau netral. Pendekatan ini memungkinkan model *Naïve Bayes* untuk menghitung *probabilitas* bahwa teks tertentu termasuk dalam kategori sentimen tertentu (misalnya, positif, negatif, atau netral), dengan memperhitungkan kontribusi dari setiap fitur X dalam teks yang diberikan.

Naïve Bayes membuat asumsi independensi antar fitur yang berarti fitur-fitur dalam data dianggap saling independen satu sama lain (Permana dkk., 2024). Oleh karena itu, *probabilitas likelihood* $P(X \mid C)$ dapat dipecah menjadi produk *probabilitas* dari masing-masing fitur:

$$P(X \mid C) = \prod_{i=1}^{n} P(Xi \mid C) \tag{2.6}$$

Di mana X_i adalah fitur ke-i dan n adalah jumlah fitur. Asumsi independensi ini menyederhanakan perhitungan dan memungkinkan metode ini untuk bekerja dengan baik meskipun fitur-fitur sering kali tidak benar-benar independen dalam data nyata. Dengan menggunakan asumsi ini, model *Naïve Bayes* dapat menghitung *probabilitas* untuk setiap kelas dan memilih kelas dengan *probabilitas* tertinggi sebagai hasil klasifikasi.

Naïve Bayes memiliki beberapa varian yang disesuaikan dengan karakteristik data yang digunakan dalam klasifikasi, sehingga fleksibel dalam berbagai kasus. *Naïve Bayes Multinomial* cocok untuk data yang memiliki distribusi multinomial, seperti kata-kata dalam teks, di mana setiap kata diperlakukan sebagai fitur yang terdistribusi secara multinomial dalam dokumen. Untuk fitur yang bersifat biner, seperti adanya atau tidak adanya kata dalam teks, *Naïve Bayes Bernoulli* menjadi pilihan yang efektif karena dapat secara sederhana mengidentifikasi kehadiran fitur tertentu dalam dokumen. Sedangkan untuk fitur yang bersifat kontinu dan mengikuti distribusi *Gaussian* atau normal, *Gaussian Naïve Bayes* sangat tepat karena dapat memanfaatkan pola distribusi tersebut untuk analisis yang lebih presisi. Dalam penelitian ini, menggunakan *Gaussian Naïve Bayes* karena data yang dianalisis memiliki fitur kontinu yang dianggap mengikuti distribusi normal. Pemilihan ini diharapkan memberikan hasil klasifikasi yang lebih akurat dan andal karena *Gaussian Naïve Bayes* mampu menyesuaikan diri dengan pola distribusi data kontinu dengan baik. Pendekatan ini telah terbukti efisien dan efektif dalam berbagai penelitian yang menggunakan *Gaussian Naïve Bayes* pada data berdistribusi normal, yang mencakup beragam aplikasi di bidang analisis sentimen dan klasifikasi teks (Septiani & Isabela, 2022).

2.1.4 *Gaussian Naïve Bayes*

Gaussian Naïve Bayes adalah varian dari algoritma *Naïve Bayes* yang dirancang khusus untuk menangani fitur kontinu dengan asumsi bahwa data mengikuti distribusi *Gaussian* (normal) (P. M. Kevin, 2022). Algoritma ini menggunakan Teorema Bayes untuk menghitung *probabilitas posterior*, yang memungkinkan klasifikasi data berdasarkan *probabilitas* kelas yang diperoleh

dari fitur-fitur yang ada. Pada *Gaussian Naïve Bayes*, *probabilitas likelihood* $P(X \mid C)$ dari fitur X yang diberikan kelas X dihitung menggunakan distribusi *Gaussian*, yang rumusnya sebagai berikut:

$$P(X|C) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(X-\mu)^2}{2\sigma^2}\right) \quad (2.\ 7)$$

Di mana:

X adalah nilai fitur

μ adalah rata-rata dari fitur *X* dalam kelas C

σ^2 adalah varians dari kelas C

π adalah konstanta *Phi* (3,14 atau $\frac{22}{7}$)

Dalam penerapan *Gaussian Naïve Bayes*, setiap fitur dalam data dianggap mengikuti distribusi *Gaussian* atau normal dalam setiap kelas yang ada (P. M. Kevin, 2022). Untuk menghitung *probabilitas* suatu data termasuk dalam kelas tertentu, *Gaussian Naïve Bayes* memerlukan dua parameter utama untuk setiap fitur, yaitu rata-rata (*mean*) dan varians. Rata-rata digunakan untuk menentukan pusat distribusi data dalam kelas tersebut, sementara varians menunjukkan tingkat penyebaran atau variasi data di sekitar rata-rata. Kedua parameter ini dihitung dari data pelatihan dan berfungsi sebagai dasar dalam perhitungan *probabilitas* berdasarkan distribusi *Gaussian*. Rumus rata-rata dan varians sebagai berikut:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \quad (2.\ 8)$$

Di mana:

μ adalah rata-rata dari fitur *X* pada kelas C

N adalah jumlah data (dokumen) yang termasuk dalam kelas C

x_i adalah nilai fitur untuk data ke-*i*

Sedangkan persamaan untuk varians sebagai berikut:

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2 \quad (2.9)$$

Di mana:

σ^2 adalah varians dari kelas C

N adalah jumlah data dalam kelas C

x_i adalah nilai fitur untuk data ke-*i* (*i* menunjukkan fitur keberapa yang sedang dihitung)

μ adalah rata-rata dari fitur dari kelas C

Varians ini menunjukkan seberapa tersebar data dalam suatu kelas di sekitar rata-ratanya. Dalam *Gaussian Naïve Bayes*, varians ini digunakan sebagai parameter σ^2 dalam distribusi *Gaussian* untuk fitur *X* dalam kelas tersebut.

2.1.5 *Feature Selection Chi-Square*

Feature selection adalah proses memilih fitur yang paling relevan untuk meningkatkan performa model dan mengurangi kompleksitas komputasi. Dalam konteks analisis teks, *feature selection* membantu mereduksi kata-kata yang tidak penting atau tidak signifikan setelah proses TF-IDF, sehingga hanya fitur yang benar-benar mendukung klasifikasi yang dipertahankan. Salah satu metode yang umum digunakan dalam *feature selection* adalah *chi-square*. Metode ini banyak digunakan karena efektif dalam memilih fitur yang memiliki hubungan kuat dengan kategori tertentu (Wati & Ernawati, 2021).

Chi-square mengukur tingkat independensi antara fitur (kata) dan kelas. Dalam analisis sentimen, *chi-square* berguna untuk mengidentifikasi kata-kata yang signifikan dalam menentukan kategori sentimen (positif, negatif atau netral).

Dengan kata lain, *chi-square* membantu memastikan bahwa fitur yang dipilih memiliki korelasi tinggi dengan kategori sentimen yang ada, sehingga model *Naïve Bayes* dapat memanfaatkan fitur-fitur tersebut untuk menghasilkan prediksi yang lebih akurat.

Feature Selection dengan *chi-square* dilakukan dengan cara pengurutan pada tiap fitur berdasarkan bobot dari nilai terbesar hingga terkecil. Perhitungan nilai pada *chi-square* kepada tiap kata yang muncul pada kelas *c* dapat dibantu dengan menggunakan tabel kontingensi. Untuk uji *chi-square* dapat dirumuskan sebagai berikut (Ernayanti dkk., 2023):

$$\chi^2 = \sum_{i=1}^{\mathrm{n}} \sum_{j=1}^{m} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \quad (2.\ 10)$$

Di mana untuk O_{ij} adalah nilai observasi atau frekuensi aktual pada baris ke-*i* (indeks baris pada tabel kontingensi) di mana merujuk pada kategori sentimen dalam analisis, yaitu sentimen positif, netral, dan negatif. Kolom ke-*j* (indeks kolom pada tabel kontingensi) merujuk pada fitur atau kata yang digunakan dalam analisis. Sedangkan E_{ij} merupakan nilai yang diharapkan pada baris ke-*i* (sentimen positif, netral, dan negatif) dan kolom ke-*j* (fitur/kata). Analisis ini didasarkan pada tabel kontingensi yang digunakan dalam uji *chi-square*. Untuk nilai derajat kebebasan dan *p-value* dihitung dengan rumus:

$$df = (r - 1) \quad (2.\ 11)$$

Di mana:

df : Derajat kebebasan

r : Jumlah baris pada tabel kontingensi

Sedangkan *p-value*

$$p - value = \mathrm{P}(\chi^2 > nilai\ Chi - Square, (r-1))$$

Keputusan yang diambil yaitu jika *p-value* lebih kecil dari tingkat signifikansi (α, misalnya 0,05), maka fitur dianggap signifikan terhadap target. Pada uji *chi-square*, terdapat tabel kontingensi yang digunakan untuk menyajikan data dalam bentuk matriks yang menggambarkan distribusi frekuensi antara dua variabel kategori. Tabel ini berfungsi sebagai dasar untuk menghitung nilai statistik *chi-square*, sehingga memungkinkan analisis hubungan atau asosiasi antara variabel-variabel tersebut.

Tabel 2.1 Tabel Kontingensi

	Kolom$_1$	**Kolom$_2$**	**…**	**Kolom$_j$**	**Jumlah**
Baris$_1$	O_{11}	O_{12}	…	O_{1j}	b_1
Baris$_2$	O_{21}	O_{22}	…	O_{2j}	b_2
…	…	…	…	…	…
Baris$_i$	O_{i1}	O_{i2}	…	O_{ij}	b_i
Jumlah	k_1	k_2	…	k_j	N

Keterangan:

O_{ij} : Frekuensi observasi (nilai aktual) pada baris ke-*i* dan kolom ke-*j*

b_i : Total frekuensi observasi pada baris ke-*i* (*i* = sentimen positif, netral, dan negatif)

k_j : Total frekuensi observasi pada kolom ke-*j* (*j* = fitur atau kata)

N : Total jumlah observasi atau frekuensi keseluruhan dalam dataset

Selain tabel kontingensi, pada uji *chi-square* juga disajikan tabel *Expected Count* yang menunjukkan nilai-nilai frekuensi yang diharapkan jika tidak terdapat hubungan atau asosiasi antara variabel-variabel yang dianalisis. Nilai ini dihitung berdasarkan distribusi frekuensi marginal dari tabel kontingensi dan digunakan

untuk membandingkan dengan frekuensi yang diamati guna menentukan signifikansi statistik. Berikut Tabel 2.2 *Expected Count*:

Tabel 2.2 Tabel Kontingensi *Expected Count*

	Kolom$_1$	**Kolom$_2$**	**…**	**Kolom$_j$**
Baris$_1$	E_{11}	E_{12}	…	E_{1j}
…	…	…	…	…
Baris$_i$	E_{i1}	E_{i2}	…	E_{ij}

Adapun rumus untuk menghitung *Expected Count:*

$$E_{ij} = \frac{b_i \times k_j}{N} \tag{2.12}$$

Keterangan:

b_i : Hasil penjumlahan baris ke i

k_j : Hasil penjumlahan kolom ke j

N : Total seluruh nilai pengamatan

Di mana baris ke i diisi oleh sentimen positif, netral, dan negatif, sedangkan kolom j berisi fitur atau kata. Dalam seleksi fitur menggunakan metode *chi-square*, keputusan didasarkan pada *p-value* dengan hipotesis sebagai berikut:

a. H_0 (Hipotesis Nol): Tidak ada hubungan antara fitur dan target, sehingga fitur tidak dipilih (terbuang).
b. H_1 (Hipotesis Alternatif): Ada hubungan yang signifikan antara fitur dan target, sehingga fitur dipilih.

Keputusan seleksi fitur didasarkan pada perbandingan antara nilai *chi-square* hitung dan nilai *chi-square* tabel sebagai berikut:

a. Jika *chi-square* hitung > *chi-square* tabel, maka H_0 ditolak dan H_1 diterima, yang berarti fitur memiliki hubungan signifikan dengan target dan dipilih.
b. Jika *chi-square* hitung ≤ *chi-square* tabel, maka H_0 gagal ditolak, yang berarti tidak ada hubungan signifikan antara fitur dan target, sehingga fitur tidak dipilih (dibuang).

Dengan pendekatan ini, hanya fitur yang memiliki hubungan signifikan dengan target yang dipertahankan dalam model analisis lebih lanjut. Hal ini memungkinkan pengurangan jumlah fitur, sehingga hanya kata-kata yang benar-benar relevan dan mewakili kategori tertentu yang tetap digunakan.

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

Synthetic Minority Over-sampling Technique (SMOTE) adalah metode yang dirancang untuk mengatasi masalah ketidakseimbangan kelas dalam dataset klasifikasi (Candra dkk., 2024). SMOTE bekerja dengan menghasilkan sampel sintetis pada kelas minoritas melalui interpolasi antara sampel yang ada, menggunakan pendekatan *k-nearest neighbors* untuk menentukan tetangga terdekat dalam ruang fitur. Prosesnya melibatkan pemilihan sampel minoritas secara acak, diikuti oleh interpolasi linier antara sampel tersebut dengan salah satu tetangganya. Penentuan tetangga terdekat dilakukan menggunakan jarak *Euclidean*, yang dirumuskan sebagai berikut:

$$d = \sqrt{\sum_{i=1}^{t}(x_i - y_i)^2} \tag{2.13}$$

Di mana d adalah jarak antara dua titik data x dan y dalam ruang fitur berdimensi *t*, dengan x_i adalah sampel minoritas dan y_i adalah tetangga terdekat dari x_i . x_i dan y_i termasuk dalam nilai fitur pada dimensi ke-i dari masing-masing titik data x dan y. x biasanya merepresentasikan sampel minoritas yang dipilih secara acak sedangkan *y* salah satu tetangga terdekat dari x yang dipilih menggunakan metode *k-nearest neighbors* (k-NN). Jarak *Euclidean* antara x dan y dihitung untuk menentukan seberapa dekat dua titik data tersebut. Setelah tetangga terdekat ditentukan, data sintetis ($x_{synthetic}$) dihasilkan melalui interpolasi linier, yang dinyatakan dengan rumus:

$$x_{synthetic} = x_i + \lambda \times (x_j - x_i) \quad (2.\ 14)$$

Di mana x_i adalah sampel minoritas asli, x_j adalah tetangga terdekat dari x_i, dan λ adalah bilangan acak antara 0 dan 1. Dengan cara ini, SMOTE mampu meningkatkan jumlah data pada kelas minoritas tanpa hanya menduplikasi data, yang sering kali menyebabkan *overfitting*. Metode ini telah terbukti meningkatkan generalisasi model dan performa klasifikasi, terutama pada data dengan distribusi kelas yang tidak seimbang. Namun, SMOTE juga memiliki keterbatasan, seperti sensitivitas terhadap noise dan distribusi data sintetis yang mungkin tidak sepenuhnya mencerminkan pola asli dataset (Yulian Pamuji, 2022).

2.1.7 *Confusion Matrix*

Evaluasi model klasifikasi didasarkan pada pengukuran terhadap kinerja dari model klasifikasi untuk menggambarkan seberapa baik sistem dalam mengklasifikasikan data. *Confusion matrix* menyajikan informasi yang

membandingkan hasil klasifikasi yang diberikan oleh sistem dengan hasil klasifikasi yang seharusnya (Surya Firmansyah dkk., 2024). Ini adalah metode yang umum digunakan untuk mengukur kinerja model klasifikasi. Setiap sel dalam *confusion matrix* memiliki angka yang menunjukkan jumlah kasus yang sebenarnya berasal dari kelas yang diamati dan dapat diprediksi.

Confusion matrix biasanya disajikan dalam bentuk tabel, seperti yang ditunjukkan pada Tabel 2.3 dengan menggunakan informasi dari *confusion matrix*, tingkat akurasi hasil klasifikasi dapat dihitung untuk kinerja model (Wati & Ernawati, 2021).

Tabel 2.3 *Confusion Matrix*

Confusion Matrix		**Hasil Prediksi**		
		1	**2**	**3**
Data Aktual	**1**	*True Positive* (TP)	*False Negatif* (FN)	*False Positif* (FP)
	2	*False Neutral* (FNt)	*True Neutral* (TNt)	*False Neutral* (FNt)
	3	*False Negative* (FN)	*False Negatif* (FN)	*True Negatif* (TN)

1. *True Positive* (TP) : Kondisi di mana data yang sebenarnya bernilai positif juga diprediksi sebagai positif.
2. *False Positive* (FP) : Kondisi di mana data yang sebenarnya bernilai negatif diprediksi sebagai positif.
3. *True Neutral* (TNt) : Kondisi di mana data yang sebenarnya bernilai netral juga diprediksi sebagai netral.
4. *False Neutral* (FNt) : Kondisi di mana data yang sebenarnya tidak bernilai netral (misalnya positif atau negatif) diprediksi sebagai netral.

5. *True Negative* (TN) : Kondisi di mana data yang sebenarnya bernilai negatif juga diprediksi sebagai negatif.
6. *False Negative* (FN) : Kondisi di mana data yang sebenarnya bernilai positif diprediksi sebagai negatif.

Akurasi pada *confusion matrix* adalah rasio antara jumlah prediksi benar (TP + TN) dengan jumlah total prediksi (TP + TN + FP + FN). Akurasi mengukur seberapa sering model yang dibuat prediksi yang sesuai dengan kenyataan.

$$\text{Akurasi} = \frac{TP+TN+TNt}{TP+FP+FN+FNt+TN+TNt} \times 100\% \quad (2.15)$$

Recall pada *confusion matrix* merupakan rasio antara jumlah prediksi positif yang benar (TP) dan jumlah total kasus positif (TP + FN+TNt). *Recall* mengukur seberapa baik model yang dapat menemukan semua kasus positif yang ada atau daya ingat. *Recall* bisa digunakan untuk menilai kinerja model klasifikasi untuk kelas positif tertentu, terutama jika ingin meminimalkan kesalahan jenis FN *(false negative)*.

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (2.16)$$

Precision atau presisi pada *confusion matrix* merupakan rasio antara jumlah prediksi positif yang benar (TP) dan jumlah total prediksi positif (TP + FP). *Precision* mengukur seberapa akurat model ketika membuat prediksi positif. Presisi dapat digunakan untuk menilai kinerja model klasifikasi untuk kelas positif tertentu, terutama jika ingin meminimalkan kesalahan jenis FP *(false positive).*

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (2.17)$$

F1-Score pada *confusion matrix* merupakan rata-rata harmonis dari precision dan *recall*. *F1-Score* mengukur seberapa baik model dapat menyeimbangkan antara presisi dan *recall*. *F1-Score* dapat digunakan untuk

menilai kinerja model klasifikasi untuk kelas positif tertentu, terutama jika ingin memaksimalkan kedua matriks tersebut.

$$F1\text{-}Score = 2 \times \frac{\text{presisi x } recall}{\text{presisi}+recall} \quad (2.18)$$

Confusion matrix ini diterapkan pada tahap *testing* untuk mengevaluasi kinerja model terhadap data yang belum pernah dilihat sebelumnya. Penggunaan *confusion matrix* pada tahap ini memberikan gambaran tentang sejauh mana model mampu menggeneralisasi atau mengklasifikasikan data baru dengan baik. Dengan membandingkan hasil prediksi model dengan hasil aktual, kita dapat menilai akurasi serta efektivitas model dalam mengenali pola pada data yang belum dikenal.

2.1.8 *Cross Validation*

Cross Validation merupakan metode statistik yang digunakan untuk mengevaluasi kinerja model pembelajaran mesin pada dataset. Teknik ini membagi dataset menjadi beberapa bagian (*fold*), melatih model pada beberapa bagian, dan mengujinya pada bagian yang tidak digunakan dalam pelatihan. Proses ini dilakukan secara berulang, dan hasil evaluasi rata-rata digunakan sebagai estimasi akurasi model yang lebih stabil dibandingkan metode evaluasi tradisional. *Cross Validation* memiliki peran penting dalam memastikan model yang dihasilkan dapat bekerja dengan baik pada data yang belum pernah dilihat sebelumnya, sehingga membantu para peneliti dan praktisi untuk menghindari *overfitting* (Widodo dkk., 2022).

Salah satu jenis *Cross Validation* yang sering digunakan adalah *Stratified K-Fold Cross Validation*. Dalam metode ini, data dibagi menjadi *k* bagian yang

seimbang berdasarkan distribusi kelas. Model dilatih pada *k-1* bagian dan diuji pada bagian yang tersisa, kemudian proses ini diulang sebanyak *k* kali sehingga setiap bagian data digunakan sebagai data uji. Teknik ini sangat bermanfaat pada kasus data yang tidak seimbang, seperti pada penelitian deteksi kanker serviks. Penelitian menunjukkan bahwa metode ini mampu meningkatkan akurasi model dengan distribusi kelas yang tetap seimbang di setiap *fold*, menjadikannya pilihan yang efektif untuk berbagai kebutuhan penelitian (Sucipto dkk., 2024).

2.1.9 *Natural Language Processing*

Natural Language Processing (NLP) adalah bidang ilmu komputer yang memungkinkan komputer untuk memahami, menganalisis, dan menafsirkan bahasa manusia. NLP memfasilitasi hubungan antara kecerdasan buatan dan pemahaman bahasa melalui pemrosesan teks dan ucapan (Young dkk., 2017). Dengan teknik NLP, komputer dapat menangkap konteks, maksud, dan sentimen dari teks atau ucapan manusia, memungkinkan aplikasi seperti analisis sentimen, pengenalan suara, dan penerjemahan otomatis (Huda, 2019). NLP umumnya menggunakan kombinasi metode linguistik dan model statistik, termasuk *machine learning*, *deep learning*, serta aturan berbasis bahasa.

Salah satu aplikasi NLP yang luas adalah analisis sentimen, yang berfungsi untuk mendeteksi perasaan atau opini dari teks. Analisis ini sering diterapkan di bidang pemasaran, layanan pelanggan, dan riset sosial untuk memahami sikap dan emosi pengguna terhadap produk atau topik tertentu. Dalam hal ini, model analisis sentimen mengklasifikasikan teks menjadi kategori emosional seperti positif, negatif, atau netral (Purnamasari dkk., 2023). Biasanya, model ini diimplementasikan dengan metode *machine learning* atau *deep learning*

yang memungkinkan komputer untuk mempelajari pola dari data pelatihan dan mengklasifikasikan teks secara otomatis berdasarkan sentimen yang terkandung di dalamnya.

2.1.10 Analisis Sentimen

Analisis sentimen adalah cabang penting dari *Natural Language Processing* (NLP) yang bertujuan untuk mengidentifikasi dan mengklasifikasikan opini atau emosi dalam teks terhadap topik tertentu (Samsir dkk., 2021). Proses ini memungkinkan sistem untuk mengenali apakah sentimen dalam suatu teks bersifat positif, negatif, atau netral. Dalam penerapannya, analisis sentimen digunakan untuk mengetahui pandangan publik terhadap berbagai aspek seperti layanan, produk, berita, dan isu-isu sosial (Purnamasari dkk., 2023). Dengan analisis sentimen, perusahaan atau lembaga bisa mendapatkan wawasan mengenai persepsi masyarakat yang dapat membantu dalam pengambilan keputusan strategis.

Dalam konteks sosial dan politik, analisis sentimen memungkinkan organisasi untuk mempelajari pendapat publik terkait isu-isu tertentu berdasarkan data dari media sosial atau artikel berita. Pendekatan ini dapat membantu dalam memahami pola opini masyarakat dan respons terhadap isu-isu sensitif atau peristiwa terkini, yang sangat berguna untuk strategi komunikasi publik (Liawati dkk., 2023). Misalnya dalam penelitian sosial, analisis sentimen dapat digunakan untuk mengukur reaksi masyarakat terhadap kebijakan baru atau respons terhadap kampanye publik.

Dari segi teknis, analisis sentimen dilakukan dengan memanfaatkan teknik NLP dan *machine learning* untuk mengenali pola dalam teks. Awalnya, model

statistik sederhana seperti *Naïve Bayes* atau *Support Vector Machines* (SVM) digunakan, namun perkembangan *deep learning* telah meningkatkan akurasi analisis sentimen. Model berbasis *neural networks*, seperti *Convolutional Neural Networks* (CNN) dan *Recurrent Neural Networks* (RNN), serta *transformers* seperti *BERT* dan *RoBERTa*, kini lebih umum digunakan karena kemampuannya dalam menangkap konteks dan makna kata yang lebih kompleks (Purnamasari dkk., 2023). Selain memberikan hasil lebih akurat, model ini juga memungkinkan analisis sentimen dalam aplikasi *real-time*, seperti pemantauan opini publik di media sosial atau sistem otomatis yang merespons *feedback* pengguna secara langsung.

2.1.11 Aplikasi Samsat Digital Online (SIGNAL)

Aplikasi Samsat Digital Online (SIGNAL) adalah platform yang disediakan oleh pemerintah Indonesia untuk memudahkan masyarakat dalam mengakses layanan Samsat secara online. Layanan yang disediakan melalui aplikasi ini meliputi pembayaran pajak kendaraan bermotor, pengecekan status kendaraan, dan layanan administrasi lainnya (Kacung dkk., 2024). Dengan aplikasi ini, diharapkan masyarakat dapat lebih mudah dan cepat dalam mengurus keperluan administrasi kendaraan tanpa harus datang langsung ke kantor Samsat.

2.2 Keadilan dalam Pelayanan Publik Berdasarkan Perspektif Islam

Keadilan adalah nilai inti dalam ajaran Islam yang mencakup segala aspek kehidupan, baik dalam hubungan pribadi, sosial, hingga kepemimpinan. Secara umum, keadilan dalam islam dapat didefinisikan sebagai memberikan hak kepada setiap orang sesuai dengan apa yang menjadi bagiannya, serta menempatkan

sesuatu pada tempat yang semestinya. Keadilan adalah landasan penting dalam membangun masyarakat yang harmonis dan damai, serta merupakan perintah langsung dari Allah Subhanahu wa Ta'ala.

Menurut para ulama, keadilan adalah sikap yang konsisten dalam menegakkan kebenaran tanpa memihak, tidak membedakan status sosial, ekonomi, maupun latar belakang seseorang. Dalam Al-Qur'an, Allah memerintahkan umat-Nya untuk selalu bersikap adil, karena keadilan adalah salah satu bentuk dari ketakwaan kepada Allah. Salah satu ayat yang menjelaskan pentingnya keadilan adalah firman Allah dalam Surah Al-Ma'idah ayat 8 (Kemenag, 2024):

يَٰٓأَيُّهَا ٱلَّذِينَ ءَامَنُواْ كُونُواْ قَوَّٰمِينَ لِلَّهِ شُهَدَآءَ بِٱلْقِسْطِ ۖ وَلَا يَجْرِمَنَّكُمْ شَنَـَٔانُ قَوْمٍ عَلَىٰٓ أَلَّا تَعْدِلُواْ ۚ ٱعْدِلُواْ هُوَ أَقْرَبُ لِلتَّقْوَىٰ ۖ وَٱتَّقُواْ ٱللَّهَ ۚ إِنَّ ٱللَّهَ خَبِيرٌۢ بِمَا تَعْمَلُونَ

"Hai orang-orang yang beriman hendaklah kamu jadi orang-orang yang selalu menegakkan (kebenaran) karena Allah, menjadi saksi dengan adil. Dan janganlah sekali-kali kebencianmu terhadap sesuatu kaum, mendorong kamu untuk berlaku tidak adil. Berlaku adillah, karena adil itu lebih dekat kepada takwa. Dan bertakwalah kepada Allah, sesungguhnya Allah Maha Mengetahui apa yang kamu kerjakan."

Ayat ini memiliki makna yang mendalam, menurut Tafsir Ibnu Katsir bahwa Allah menyeru umat Islam untuk menegakkan keadilan dalam segala keadaan, bahkan ketika menghadapi kebencian atau emosi terhadap kelompok tertentu. Ini menunjukkan bahwa keadilan tidak boleh dipengaruhi oleh perasaan subjektif atau prasangka. Keadilan adalah bentuk ketaatan yang mendekatkan seseorang kepada Allah, karena adil itu lebih dekat kepada takwa. Dalam hal ini, tidak hanya dalam hubungan pribadi, tetapi juga dalam hubungan masyarakat dan pelayanan publik, keadilan harus menjadi pijakan utama (Ibnu Katsir, 1997).

Keadilan juga ditonjolkan dalam hadis Nabi Muhammad shallallahu 'alaihi wa sallam, di mana beliau bersabda:

"Sesungguhnya orang-orang yang adil di sisi Allah berada di atas mimbar-mimbar dari cahaya, yaitu orang-orang yang berlaku adil dalam keputusan mereka, dalam keluarga mereka, dan dalam apa yang mereka pimpin." (HR. Muslim No. 1827).

Hadis ini menegaskan bahwa orang yang bersikap adil akan mendapatkan tempat istimewa di sisi Allah. Orang-orang yang berlaku adil dalam semua urusan mereka, baik dalam keputusan-keputusan penting yang mereka buat, dalam keluarga, maupun dalam peran kepemimpinan yang mereka emban, akan diberi ganjaran tinggi. Ini menunjukkan bahwa keadilan adalah esensi yang harus dipegang oleh pemimpin dan siapa pun yang memiliki tanggung jawab publik, termasuk dalam konteks pelayanan modern seperti pengelolaan aplikasi digital untuk publik.

Untuk mencapai keadilan dalam masyarakat, Islam sangat menekankan pentingnya penegakan hukum yang adil. Hukum harus diterapkan secara konsisten tanpa memandang status sosial, jabatan, atau kedudukan seseorang. Setiap individu harus mendapatkan perlakuan yang setara di hadapan hukum, tanpa diskriminasi (Al-Qaradhawi, 2010). Selain itu, keadilan berarti memberikan hak kepada siapa pun yang berhak menerimanya, baik dalam urusan ekonomi, sosial, maupun keluarga. Setiap orang harus memperoleh hak atas kepemilikan, pendidikan, dan perlakuan yang setara dalam kehidupan sehari-hari (An-Nawawi, 2007).

Islam juga melarang segala bentuk diskriminasi, termasuk berdasarkan ras, warna kulit, agama, atau jenis kelamin. Masyarakat yang adil harus menjamin bahwa setiap orang memiliki kesempatan yang sama untuk berkembang, serta mendapatkan akses yang setara ke pendidikan, kesehatan, dan pekerjaan. Keadilan sosial juga mencakup menjaga keseimbangan antara golongan kaya dan miskin, dengan memberikan perhatian khusus kepada mereka yang kurang beruntung.

Islam menekankan pentingnya memastikan bahwa hak-hak kelompok yang lemah tetap terpenuhi (Ahmad, 2023).

Selain itu, Islam mengingatkan bahwa setiap pemimpin atau orang yang diberi tanggung jawab dalam masyarakat harus menghindari penyalahgunaan wewenang. Pemimpin yang adil harus membuat keputusan yang mempertimbangkan kepentingan semua pihak dan berpegang pada prinsip keadilan untuk kemaslahatan bersama (Charis F dkk., 2020). Dengan penerapan prinsip-prinsip ini, masyarakat yang adil dan seimbang dapat tercapai.

Dengan demikian, keadilan dalam Islam bukan hanya menjadi dasar bagi individu, tetapi juga fondasi utama dalam membangun masyarakat yang adil dan seimbang. Penerapan prinsip-prinsip keadilan akan menciptakan kehidupan yang harmonis, di mana setiap orang mendapatkan haknya dan menjalani kehidupan yang bermartabat.

2.3 Kajian Topik dengan Teori Pendukung

Penelitian ini dilakukan untuk mengembangkan ilmu pengetahuan dalam bidang analisis data, khususnya dalam analisis sentimen menggunakan algoritma *Naïve Bayes* pada aplikasi Samsat Digital Online (SIGNAL). Pada era digital seperti ini, analisis sentimen menjadi sangat penting untuk memahami opini publik dan meningkatkan kualitas layanan publik. Algoritma *Naïve Bayes* dipilih karena kemampuannya dalam menghasilkan prediksi yang akurat serta menyajikan hasil yang mudah dipahami, sehingga dapat memberikan wawasan yang bermanfaat bagi pengambilan keputusan.

Naïve Bayes adalah algoritma klasifikasi berbasis probabilistik yang bekerja dengan menghitung *probabilitas* suatu kategori berdasarkan kemunculan

fitur atau kata-kata dalam teks. Pendekatan ini memanfaatkan prinsip Teorema Bayes untuk mengestimasi kemungkinan suatu data termasuk dalam kelas tertentu (misalnya, sentimen positif, negatif, atau netral) berdasarkan fitur-fitur yang ada dalam teks. Algoritma ini dikenal sederhana dan efisien, terutama dalam pemrosesan data teks yang besar, sehingga sangat sesuai untuk analisis sentimen.

Pemilihan analisis sentimen sebagai objek penelitian dipertimbangkan karena keputusan ini merupakan salah satu yang krusial dalam meningkatkan layanan publik. Layanan publik seperti Samsat Digital memiliki karakteristik, yakni pengguna yang menjadi fokus penting dalam menentukan keputusan perbaikan layanan. Melalui analisis sentimen, masukan dari pengguna dapat diidentifikasi dan digunakan untuk memperbaiki aspek-aspek layanan yang masih perlu ditingkatkan.

Pada penelitian ini, algoritma *Naïve Bayes* diterapkan pada dataset ulasan pengguna aplikasi SIGNAL untuk mengklasifikasikan sentimen menjadi kategori positif, negatif, atau netral. Proses analisis dimulai dengan tahap seleksi data untuk memilih atribut yang relevan, diikuti oleh tahap *preprocessing* yang meliputi pembersihan, normalisasi, dan transformasi data agar sesuai dengan format yang dibutuhkan oleh algoritma *Naïve Bayes*. Setelah data siap, model *Naïve Bayes* dilatih untuk mengenali pola dalam ulasan yang mengindikasikan sentimen tertentu.

Evaluasi kinerja algoritma *Naïve Bayes* dilakukan dengan menggunakan metrik evaluasi seperti akurasi*, presisi*, *recall*, dan *F1-Score*. Hasil evaluasi ini akan memberikan gambaran yang jelas tentang sejauh mana algoritma mampu mengklasifikasikan sentimen dengan tepat. Melalui penelitian ini, diharapkan dapat diperoleh wawasan yang lebih baik tentang efektivitas algoritma *Naïve Bayes*

dalam analisis sentimen pada aplikasi SIGNAL serta rekomendasi yang dapat diambil berdasarkan hasil analisis sentimen pengguna.

BAB III
METODE PENELITIAN

3.1 Jenis Penelitian

Penelitian ini menggunakan metode kuantitatif dengan pendekatan deskriptif serta studi literatur. Metode kuantitatif melibatkan penggunaan populasi atau sampel tertentu, di mana data dikumpulkan dan dianalisis secara statistik. Sementara itu, pendekatan deskriptif berfungsi untuk menganalisis hasil penelitian. Studi literatur dimulai dengan pencarian dan pengumpulan referensi dari buku, jurnal ilmiah, tesis, dan disertasi yang memiliki sumber kredibel dan relevan dengan tujuan penelitian.

Penelitian ini berfokus terhadap analisa sentimen pengguna terhadap aplikasi SIGNAL. Analisa sentimen berdasarkan komentar pengguna yang nantinya dilakukan *scraping* untuk dilakukan penelitian. Jenis penelitian yang dilakukan adalah kuantitatif karena menggunakan data numerik yang dihasilkan dari analisis sentimen terhadap aplikasi Samsat Digital Online (SIGNAL) dengan algoritma *Naïve Bayes*. Penelitian ini akan dilakukan beberapa tahapan yaitu *preprocessing, labeling*, split data, TF-IDF, *Feature Selection*, *Modeling*, dan selanjutnya akan dilakukan evaluasi.

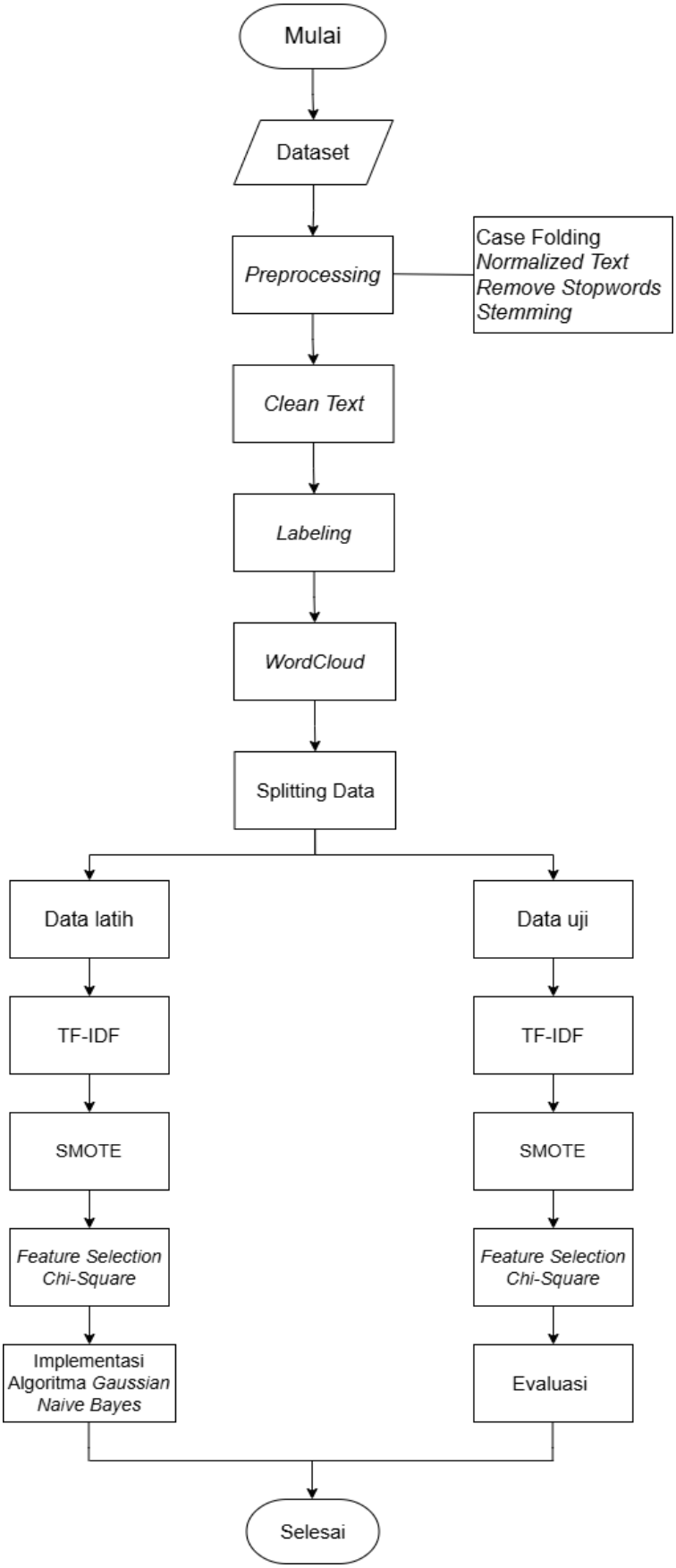
3.2 Data dan Sumber Data

Penelitian ini menggunakan data berupa ulasan. Data ulasan diperoleh dari ulasan Aplikasi SIGNAL pada Google Playstore. Data diambil dengan metode *scraping*. Dalam penelitian ini, teks ulasan pengguna berperan sebagai variabel independen (variabel X), sedangkan kategori sentimen seperti positif, negatif, dan

netral berperan sebagai variabel dependen (variabel Y). Variabel X berupa data teks akan melalui tahapan *preprocessing* dan representasi fitur seperti TF-IDF, sementara variabel Y merupakan label yang menunjukkan kecenderungan opini pengguna terhadap aplikasi SIGNAL berdasarkan ulasan mereka.

3.3 Tahap Penelitian

Tahap penelitian merupakan serangkaian langkah atau tahapan yang dijalankan secara sistematis dalam menyusun tugas akhir. Prosedur ini berfungsi sebagai panduan dalam melakukan survei agar hasil yang diperoleh sesuai dengan tujuan yang diharapkan dan tidak menyimpang. Tahapan digambarkan pada *flowchart* tersebut menggambarkan tahapan penelitian yang dimulai dari pengumpulan dataset, dilanjutkan dengan proses *preprocessing* yang mencakup *case folding*, normalisasi teks, penghapusan *stopwords*, dan *stemming*. Setelah itu, dilakukan pembersihan teks dan pelabelan data sebelum dianalisis dengan visualisasi *WordCloud*. Data kemudian dibagi menjadi data latih dan data uji, yang masing-masing melalui proses TF-IDF, penerapan teknik SMOTE, dan seleksi fitur menggunakan metode *Chi-Square*. Data latih digunakan untuk implementasi algoritma *Gaussian Naïve Bayes*, sedangkan data uji digunakan untuk evaluasi performa model. Semua tahapan ini bertujuan untuk menghasilkan klasifikasi teks yang akurat dan representatif terhadap data yang dianalisis. Gambar 3.1 merupakan *flowchart* alur kerja penelitian.



Gambar 3.1 *Flowchart* Penelitian

1. Pengumpulan data

   Pengumpulan data ulasan aplikasi SIGNAL ini dilakukan dengan teknik *web scraping* pada Google Play Store, dengan fokus pada tanggapan pengguna terkait kepuasan dan pengalaman mereka dalam menggunakan aplikasi. Data yang dikumpulkan mencakup ulasan mengenai kualitas layanan, keluhan teknis, serta saran perbaikan untuk aplikasi tersebut. Setiap ulasan dikumpulkan hingga mencapai total 5.000 data komentar. Data ini diharapkan dapat memberikan gambaran yang komprehensif mengenai persepsi pengguna dan menjadi dasar analisis lebih lanjut.

2. *Pre-processing Data*

   Ada beberapa tahapan yang dilalui pada proses *preprocessing* diantaranya:

   a. *Case Folding*

      Case Folding digunakan untuk menghapus beberapa karakter yang bukan termasuk dari huruf abjad sehingga kata-kata yang mengandung *emoticon*, karakter angka, *hashtag* (#), *mention* (@), dan URL dan *link* yang ada pada komentar akan secara otomatis terhapus dari pengambilan data ketika *web scrapping* komentar atau ulasan dari aplikasi SIGNAL di Google Playstore yang akan diteliti.

   b. *Normalized Text*

      Normalized Text harus mencakup serangkaian aturan dan prosedur yang mencakup berbagai singkatan dan variasi kata yang mungkin dijumpai dalam teks analisis sentimen. Proses ini bertujuan untuk menyederhanakan variasi kata ke bentuk standar guna meningkatkan konsistensi hasil analisis. Kamus diambil dari sumber github pada link

berikut ini raw.githubusercontent.com/ksnugroho/klasifikasi-spam sms/master/data/key_norm.csv.

c. *Remove Stopwords*

Setelah proses normalisasi, data akan melalui pengapusan *stopwords* untuk menghilangkan kata yang tidak memiliki bobot pada analisis sentimen.

d. *Stemming*

Kemudian dilakukan proses mengembalikan kata tersebut jadi kata dasar tanpa imbuhan. Tujuannya untuk meningkatkan kesamaan kata demi meningkatkan akurasi prediksi.

3. *Clean Text* data yang hilang

Setelah *preprocessing*, beberapa data dapat hilang akibat penghapusan *stopwords*, normalisasi, atau pembersihan karakter yang tidak relevan. Untuk memastikan kualitas data tetap optimal, diperlukan langkah pembersihan lebih lanjut guna mengidentifikasi dan menangani teks yang kosong atau tidak lagi bermakna. Dengan proses ini, analisis sentimen dapat dilakukan dengan lebih akurat tanpa terganggu oleh hilangnya data yang tidak terkontrol.

4. *Labeling*

Setelah selesai menerapkan *preprocessing* nantinya akan dilakukan *labeling*. Pada tahap ini menggunakan *score* atau rating aplikasi pada dataset sebagai penentuan sentimen. Penentuan sentimen memiliki 3 indikator yaitu positif, netral, dan negatif. *Score* dengan nilai 1 dan 2 digolongkan negatif,

score dengan nilai 3 digolongkan netral, dan *score* dengan nilai 4 dan 5 digolongkan positif.

5. *Wordcloud*

   Setelah selesai menerapkan menerapkan *labeling* akan dilakukan visualisasi. Dalam mevisualisasikan data menggunakan *wordcloud* sebagai bentuk visualnya. *Wordcloud* memvisualisasikan huruf berdasarkan frekuensi kemunculan huruf pada dataset komentar. Pada *wordcloud* masing-masing dilakukan filter sentimen. Filter *wordcloud* sentimen positif, filter *wordcloud* dengan sentimen negatif, dan filter *wordcloud* dengan sentimen netral.

6. *Splitting* Data

   Data akan dipisahkan menjadi dua bagian yaitu data *training* dan data *testing*. Di mana data *training* itu lebih besar dibandingkan dengan data *testing*. Data *training* digunakan untuk melatih masing masing model dan data *testing* digunakan untuk menguji model. Tahapan *Splitting* Data:

   a. Data yang telah melalui tahap *preprocessing* digunakan sebagai langkah awal dalam proses pembagian data.
   b. Proses pemisahan data (data *splitting*) dilakukan secara acak pada data yang sudah dipreproses.
   c. Data ini kemudian dibagi menjadi data pelatihan (*training*) dan data pengujian (*testing*).

7. Pemodelan dengan algoritma *Naïve Bayes*

   Pada metode *Naïve Bayes*, dilakukan proses TF-IDF pada setiap dokumen dengan mengasumsikan bahwa setiap kata dalam dokumen memiliki

pengaruh yang independen terhadap klasifikasi. Setelah data diubah ke dalam bentuk numerik menggunakan TF-IDF, dilakukan proses seleksi fitur (*feature selection*) untuk memilih fitur-fitur yang paling relevan dalam prediksi kelas. Proses implementasi algoritma ini dimulai dengan membangun model dari data latih yang sudah diberi label kelasnya. Model kemudian akan belajar dari data latih ini untuk memprediksi kelas pada data uji. Tahapan dalam pemodelan menggunakan algoritma *Naïve Bayes* sebagai berikut:

a. Data latih digunakan untuk proses klasifikasi dengan algoritma *Naïve Bayes*.
b. Dilakukan proses TF-IDF untuk menghitung bobot setiap kata berdasarkan frekuensi kemunculannya dalam dokumen tertentu sehingga kata yang relevan dapat lebih diutamakan.
c. Setelah data diubah menjadi bentuk numerik dengan TF-IDF, dilakukan tahap seleksi fitur (*feature selection*). *Feature selection* ini bertujuan untuk memilih fitur atau kata yang paling berpengaruh dan relevan dalam menentukan kelas. Di mana proses ini akan membantu mengurangi kompleksitas data dan meningkatkan akurasi model.
d. Setelah tahap *feature selection*, kemudian dilakukan SMOTE karena data ulasan yang tidak imbang.
e. Data yang telah melewati tahap SMOTE kemudian diklasifikasikan menggunakan algoritma *Naïve Bayes*.

8. Evaluasi

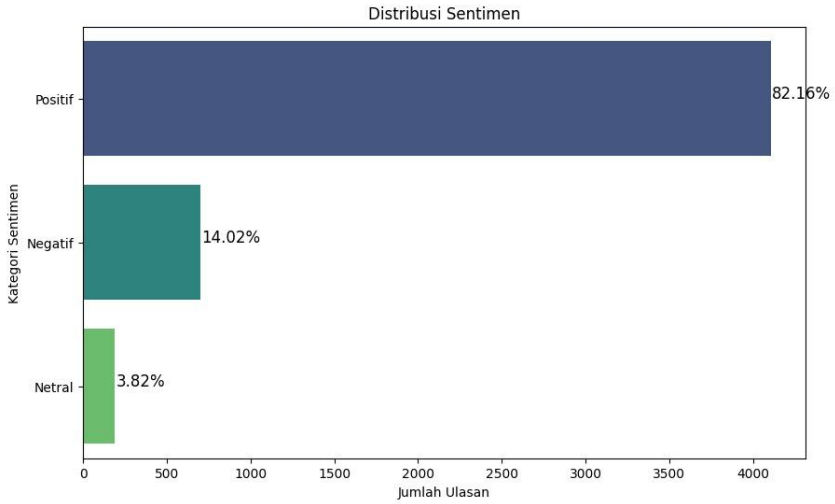
Tahap terakhir pada penelitian akan dilakukan evaluasi. Evaluasi dilakukan pada tiap model algoritma yang nanti memberikan hasil berupa *classification report* dan *confusion matrix*. Evaluasi ini penting untuk mengukur seberapa baik model dapat menggeneralisasi pola dari data latih ke data uji. Adapun tahapan evaluasi seperti berikut:

a. Data uji (*testing*) digunakan sebagai pengujian proses evaluasi.
b. Hasil model yang digunakan akan divalidasi dengan data uji (*testing*).
c. Hasil prediksi dari model kemudian di evaluasi menggunakan *confusion matrix*. *Confusion matrix* ini memberikan gambaran mengenai performa model.
d. Hasil evaluasi dengan *confusion Matrix*.

BAB IV
HASIL DAN PEMBAHASAN

4.1 Pengumpulan Data

Dataset yang digunakan dalam penelitian ini merupakan hasil *scraping* sebanyak 5000 ulasan dari platform Google Play Store terkait aplikasi Samsat Digital Online (SIGNAL). Proses pengumpulan data dilakukan menggunakan metode *web scraping* dengan bahasa pemrograman Python di lingkungan Google Collaboratory. Data yang diambil mencakup rentang waktu tertentu sesuai dengan fokus penelitian. Berikut Visualisasi perbandingan jumlah ulasan berdasarkan kategori sentimen dapat dilihat pada Gambar 4.1.



Gambar 4.1 Grafik Distribusi Sentimen

Pada Gambar 4.1, terlihat bahwa data dengan sentimen positif mendominasi sebesar 82,16%, diikuti oleh data dengan sentimen negatif sebesar 14,02%, dan data dengan sentimen netral sebesar 3,82% dari jumlah keseluruhan data ulasan yang digunakan. Tabel 4.1 menyajikan sampel data ulasan yang telah dikategorikan berdasarkan sentimen.

Tabel 4.1 Sampel Data Komentar

No	Komentar	Label
1.	Mantap bayar pajak kendaraan sekarang dimudahkan...Semoga aplikasinya bisa ditingkatkan layanannya...	Positif
2.	Layanan pengaduannya tidak memberikan solusi dengan baik. Pembayaran sudah dilakukan dgn ada bukti pembayaran tp status blm melakukan pembayaran dn jdi kedaluarsa	Negatif
3.	Bayar pajak aja ribet bgt mending gausah sekalian, tinggal bayar aja gagal trus udh di coba semuanya ga bisa, mau ngadu ke cs malah di jawab bot hadehhh emng msh bobrok sytem pemerintah	Negatif
4.	cepet, sat set tanpa ribet. Aman dan terpercaya	Positif
5.	Semoga bisa bukan nama pemilik.	Netral

4.2 *Preprocessing*

Preprocessing merupakan langkah untuk mengubah dokumen yang tidak terstruktur menjadi lebih terstruktur dengan cara menghapus atribut yang tidak relevan, sehingga data menjadi lebih sistematis dan mengurangi *noise*. Tahap preprocessing sangat penting karena data yang digunakan berasal dari ulasan pengguna SIGNAL, yang tidak memiliki batasan dalam penggunaan kata dalam kalimat, sehingga dapat memengaruhi hasil klasifikasi. Melalui *preprocessing*, data yang awalnya tidak terstruktur diubah menjadi data terstruktur. Proses ini menggunakan bantuan *library* Sastrawi, yang sering digunakan dalam *Natural Language Processing* (NLP) untuk teks berbahasa Indonesia.

1. *Case Folding*

   Case folding adalah proses menyederhanakan teks dengan mengubah semua huruf menjadi kecil, menghapus elemen tidak relevan seperti URL, angka, tanda baca, dan spasi berlebih, serta mengganti tanda baca tertentu dengan spasi. Proses ini memastikan teks lebih bersih untuk analisis lebih

lanjut, seperti terlihat pada Tabel 4.2 yang membandingkan hasil sebelum dan sesudah case folding pada ulasan aplikasi SIGNAL.

Tabel 4.2 Contoh Hasil *Case Folding* Data Komentar SIGNAL

Sebelum	Sesudah
Mantap bayar pajak kendaraan sekarang dimudahkan...Semoga aplikasinya bisa ditingkatkan layanannya...	mantap bayar pajak kendaraan sekarang dimudahkan semoga aplikasinya bisa ditingkatkan layanannya
Layanan pengaduannya tidak memberikan solusi dengan baik. Pembayaran sudah dilakukan dgn ada bukti pembayaran tp status blm melakukan pembayaran dn jdi kedaluarsa	layanan pengaduannya tidak memberikan solusi dengan baik pembayaran sudah dilakukan dgn ada bukti pembayaran tp status blm melakukan pembayaran dn jdi kedaluarsa
Bayar pajak aja ribet bgt mending gausah sekalian, tinggal bayar aja gagal trus udh di coba semuanya ga bisa, mau ngadu ke cs malah di jawab bot hadehhh emng msh bobrok sytem pemerintah	bayar pajak aja ribet bgt mending gausah sekalian tinggal bayar aja gagal trus udh di coba semuanya ga bisa mau ngadu ke cs malah di jawab bot hadehhh emng msh bobrok sytem pemerintah
cepet, sat set tanpa ribet. Aman dan terpercaya	cepet sat set tanpa ribet aman dan terpercaya
Semoga bisa bukan nama pemilik.	semoga bisa bukan nama pemilik

2. *Normalized Text*

Setelah *case folding*, dilakukan normalisasi teks untuk mengubah kata tidak baku atau slang menjadi sesuai bahasa Indonesia standar, seperti "bgt" menjadi "banget" dan "gk" menjadi "tidak". Proses ini penting untuk meningkatkan akurasi analisis, mengingat data ulasan sering menggunakan bahasa informal. Kamus normalisasi diambil dari sumber GitHub raw.githubusercontent.com/ksnugroho/klasifikasi-spam

sms/master/data/key_norm.csv. Hasil normalisasi ditampilkan pada Tabel 4.3 untuk ulasan aplikasi SIGNAL.

Tabel 4.3 Contoh Hasil *Normalized Text* Data Komentar SIGNAL

Sebelum	Sesudah
mantap bayar pajak kendaraan sekarang dimudahkan semoga aplikasinya bisa ditingkatkan layanannya	mantap bayar pajak kendaraan sekarang dimudahkan semoga aplikasi bisa ditingkatkan layanannya
layanan pengaduannya tidak memberikan solusi dengan baik pembayaran sudah dilakukan dgn ada bukti pembayaran tp status blm melakukan pembayaran dn jdi kedaluarsa	layanan pengaduannya tidak memberikan solusi dengan baik pembayaran sudah dilakukan dengan ada bukti pembayaran tapi status belum melakukan pembayaran dan jadi kedaluwarsa
bayar pajak aja ribet bgt mending gausah sekalian tinggal bayar aja gagal trus udh di coba semuanya ga bisa mau ngadu ke cs malah di jawab bot hadehhh emng msh bobrok sytem pemerintah	bayar pajak saja ribet banget mending tidak usah sekalian tinggal bayar saja gagal terus sudah di coba semuanya tidak bisa mau ngadu ke cs malah di jawab bot hadehhh memang masih bobrok sistem pemerintah
cepet sat set tanpa ribet aman dan terpercaya	cepat sat set tanpa ribet aman dan terpercaya
semoga bisa bukan nama pemilik	semoga bisa bukan nama pemilik

3. *Remove Stopwords*

Penghapusan *stopwords* dilakukan untuk menghilangkan kata-kata umum seperti "dan", "yang", dan "di" yang tidak signifikan dalam analisis sentimen, sehingga teks lebih fokus pada informasi utama. Dalam penelitian ini, daftar stopwords diambil dari *Natural Language Toolkit* (NLTK), yang menyediakan kumpulan kata umum dalam Bahasa

Indonesia yang sering diabaikan dalam pemrosesan teks. Hasilnya ditampilkan pada Tabel 4.4 untuk ulasan aplikasi SIGNAL.

Tabel 4.4 Contoh Hasil *Stopwords* Data Komentar SIGNAL

Sebelum	Sesudah
mantap bayar pajak kendaraan sekarang dimudahkan semoga aplikasi bisa ditingkatkan layanannya	mantap bayar pajak kendaraan sekarang dimudahkan aplikasi tingkatkan layanan
layanan pengaduannya tidak memberikan solusi dengan baik pembayaran sudah dilakukan dengan ada bukti pembayaran tapi status belum melakukan pembayaran dan jadi kedaluwarsa	layanan adu tidak memberikan solusi baik pembayaran dilakukan bukti pembayaran tapi status belum pembayaran jadi kedaluwarsa
bayar pajak saja ribet banget mending tidak usah sekalian tinggal bayar saja gagal terus sudah di coba semuanya tidak bisa mau ngadu ke cs malah di jawab bot hadehhh memang masih bobrok sistem pemerintah	bayar pajak ribet tidak sekalian tinggal bayar gagal terus coba semuanya tidak bisa ngadu cs jawab bot sistem pemerintah
cepet sat set tanpa ribet aman dan terpercaya	cepat tanpa ribet aman terpercaya
semoga bisa bukan nama pemilik	semoga bukan nama milik

4. *Stemming*

Stemming adalah proses mengubah kata berimbuhan menjadi kata dasar menggunakan *library Sastrawi*, seperti "dilakukan" menjadi "laku" dan "berjalan" menjadi "jalan". Proses ini meningkatkan akurasi analisis dengan menyatukan kata bermakna sama. Hasil stemming ditampilkan pada Tabel 4.5 untuk ulasan aplikasi SIGNAL.

Tabel 4.5 Contoh Hasil *Stemming* Data Komentar SIGNAL

Sebelum	Sesudah
mantap bayar pajak kendaraan sekarang dimudahkan aplikasi tingkatkan layanan	mantap bayar pajak kendaraan sekarang mudah aplikasi tingkat layan
layanan adu tidak beri solusi baik bayar laku bukti bayar tapi status belum bayar jadi kedaluwarsa	layanan adu tidak beri solusi baik bayar laku bukti bayar tapi status belum bayar jadi kadaluwarsa
bayar pajak ribet tidak sekalian tinggal bayar gagal terus coba semuanya tidak bisa ngadu cs jawab bot sistem pemerintah	bayar pajak ribet tidak sekalian tinggal bayar gagal terus coba semua tidak bisa adu cs jawab bot sistem pemerintah
cepat tanpa ribet aman terpercaya	cepat tanpa ribet aman percaya
semoga bukan nama milik	semoga bukan nama milik

5. *Clean Text* data yang hilang

Clean text data yang hilang terjadi ketika tidak ada kata tersisa setelah *preprocessing*, biasanya karena semua kata adalah *stopwords*, teks hanya berisi elemen tidak relevan seperti angka atau simbol, atau teks terlalu pendek. Identifikasi kondisi ini penting untuk memastikan data relevan tetap terjaga dalam analisis. Oleh karena itu, perlu dilakukan evaluasi terhadap hasil *preprocessing* untuk menghindari kehilangan informasi penting yang dapat mempengaruhi akurasi model dan interpretasi hasil analisis secara keseluruhan. Pada Gambar 4.2 menampilkan hasil *Clean Text* data yang hilang.

	content	score	clean_text
0	aplikasi sangat membantu.... proses nya cepat....	5	aplikasi bantu proses nya cepat terima kasih
1	Alhamdulillah dikirim ke rumah dengan cepat, g...	5	alhamdulillah kirim rumah cepat fc ktp stnk bp...
2	Enak kalau diluar daerah,dari pembuatan STNK kita	5	enak luar daerah buat stnk
3	Sangat membantu	5	bantu
4	mantap bayar pajak jadi mudah	5	mantap bayar pajak mudah
...	...	...	...
4995	Saya sangat terbantu dengan adanya sistem paja...	5	bantu sistem pajak online maju polri
4996	Mempermudah	5	mudah
4997	Terimakasih sangat membantu,,,	5	terimakasih bantu
4998	mantaaaap	5	mantaaaap
4999	aplikasinya sering eror harus diperbaiki lagi ...	4	aplikasi eror baik live chat nya susah banget ...

4964 rows × 3 columns

Gambar 4.2 Hasil *Clean Text* Data Hilang

4.3 *Labeling*

Proses *labeling* ditentukan berdasarkan skor rating dari aplikasi SIGNAL untuk mengklasifikasikan sentimen yang terkandung dalam data. Rating diberikan dalam skala dari 1 hingga 5, dengan masing-masing angka mewakili kategori sentimen tertentu. Rating 4 dan 5 dianggap sebagai sentimen positif, yang menunjukkan bahwa umpan balik atau data tersebut memiliki konotasi baik. Sebaliknya, rating 1 dan 2 dianggap sebagai sentimen negatif, yang menunjukkan bahwa umpan balik tersebut lebih cenderung merujuk pada pengalaman atau opini yang tidak memuaskan. Sementara itu, rating 3 diperlakukan sebagai netral, yang berarti bahwa umpan balik tersebut tidak menunjukkan kecenderungan positif atau negatif yang jelas. Dengan demikian, hal ini mempermudah pengelompokan data berdasarkan sentimen, sehingga analisis dapat dilakukan lebih efektif dan menghasilkan wawasan yang lebih jelas. Berikut Gambar 4.3 menunjukkan hasil proses *labeling*.

score	clean_text	sentimen
5	aplikasi bantu proses nya cepat terima kasih	positif
5	alhamdulillah kirim rumah cepat fc ktp stnk bp...	positif
5	enak luar daerah buat stnk	positif
5	bantu	positif
5	mantap bayar pajak mudah	positif
...	...	...
5	bantu sistem pajak online maju polri	positif
5	mudah	positif
5	terimakasih bantu	positif
5	mantaaaap	positif
4	aplikasi eror baik live chat nya susah banget ...	positif

Gambar 4.3 Hasil *Labeling*

4.4 *Wordcloud*

Wordcloud adalah visualisasi yang menunjukkan frekuensi kemunculan kata dalam teks, di mana kata yang lebih sering muncul akan ditampilkan dengan ukuran lebih besar. Dalam penelitian ini, *wordcloud* digunakan untuk menganalisis ulasan aplikasi SIGNAL berdasarkan sentimen. Pada sentimen positif, kata seperti "mudah" dan "bantu" mendominasi, mencerminkan kepuasan pengguna. Sentimen negatif didominasi oleh kata "gagal" dan "susah", menunjukkan keluhan utama. Sementara itu, sentimen netral menampilkan kata "kirim" dan "aplikasi", yang lebih bersifat deskriptif tanpa penilaian emosional. *Wordcloud* membantu mengidentifikasi tema utama setiap kategori sentimen untuk pengembangan

aplikasi lebih lanjut. Gambar 4.4 menunjukkan Visualisasi dari *Wordcloud* untuk sentiment positif, netral, dan negatif.

Gambar 4.4 *Wordcloud* Positif, Netral, dan Negatif

4.5 *Splitting* Data

Pada tahap ini, data ulasan aplikasi Samsat Digital Online (SIGNAL) yang telah melalui proses preprocessing dibagi menjadi dua bagian utama, yaitu data latih (*training*) dan data uji (*testing*). Pembagian data dilakukan dengan rasio 80:20, di mana 80% dari total data digunakan untuk melatih model, sedangkan 20% sisanya digunakan untuk menguji performa model. Jumlah total ulasan dalam dataset ini setelah dilakukan *preprocessing* adalah 4964 ulasan, yang terdiri atas 4.075 ulasan positif, 701 ulasan negatif, dan 188 ulasan netral. Setelah dilakukan pembagian data, sebanyak 3.971 ulasan digunakan sebagai data latih untuk melatih model, sementara 993 ulasan digunakan sebagai data uji untuk mengevaluasi performa model.

4.6 Pemodelan dengan Algoritma *Naïve Bayes*

4.6.1 TF-IDF

Dalam penelitian ini, algoritma *machine learning* membutuhkan input data dalam bentuk numerik untuk memungkinkan proses pelatihan dan pembuatan model. Oleh karena itu, metode TF-IDF akan diterapkan untuk mengubah data

menjadi format numerik. Proses ini melibatkan perhitungan bobot untuk setiap kata berdasarkan frekuensi kemunculannya dalam satu dokumen (TF) serta frekuensi kemunculannya dalam seluruh kumpulan dokumen (IDF), yang dihitung menggunakan Persamaan (2.1) dan (2.2). Selanjutnya, perhitungan berdasarkan Persamaan (2.3) dilakukan untuk memperoleh nilai TF dan IDF. Sampel data yang akan digunakan untuk proses TF-IDF dapat dilihat pada Tabel 4.6.

Tabel 4.6 Kalimat Sampel TF-IDF

D	Kalimat
D1	aplikasi cepat bagus minim eror tidak ribet
D2	proses aplikasi cepat tidak ada keluhan eror
D3	aplikasi tidak eror tidak ribet lancar cepat bayar pajak
D4	aplikasi bagus tidak karena lemot ribet eror register motor
D5	sudah ajukan keluhan aplikasi tapi ribet proses bayar eror
D6	keluhan muncul pajak aplikasi tidak cepat sangat ribet bayar
D7	aplikasi pajak bagus serta proses keluhan lama sekali eror
D8	tidak sampai tempat stnk proses bayar pajak aplikasi cepat tidak ribet
D9	proses sistem cepat bagus bayar ribet error minim keluhan
D10	simpel aplikasi praktis bayar pajak bagus cepat
D11	aplikasi pajak ribet banyak keluhan kurang bagus gagal bayar
D12	tinggal bayar pajak online proses pengisian tidak ribet cepat error login tidak ribet
D13	aplikasi cepat mudah bagus bayar pajak kendaraan
D14	kirim lambat bayar agak ribet eror mudah aplikasi pajak
D15	aplikasi keluhan ribet proses bayar online lemot tidak bagus
D16	bayar pajak cepat bagus proses eror aplikasi lama ribet
D17	cepat proses tidak ribet tidak eror bayar pajak bagus samsat
D18	dapat keluhan bayar pajak proses cepat sering eror
D19	cepat proses bayar kirim tidak ada keluhan eror
D20	keluhan aplikasi sedikit ribet registrasi proses bayar pajak lumayan eror
D21	keluhan eror aplikasi pajak buruk
D22	bayar pajak mudah aplikasi sering eror aneh
D23	bayar pajak tapi mudah eror registrasi aplikasi

Setelah dokumen sampel diperoleh, setiap dokumen akan diuraikan menjadi *term* atau kata.

Tabel 4.7 Menghitung Frekuensi Kemunculan Kata pada Dokumen

Dokumen	Frekuensi								
	aplikasi	cepat	bagus	ribet	proses	keluhan	error	bayar	pajak
D1	1	1	1	1	0	0	1	0	0
D2	1	1	0	0	1	1	1	0	0
D3	1	1	0	1	0	0	1	1	1
D4	1	0	1	1	0	0	1	0	0
D5	1	0	0	1	1	1	1	1	0
D6	1	1	0	1	0	1	0	1	1
D7	1	0	1	0	1	1	1	0	1
D8	1	1	0	1	1	0	0	1	1
D9	0	1	1	1	1	1	1	1	0
D10	1	1	1	0	0	0	0	1	1
D11	1	0	1	1	0	1	0	1	1
D12	0	1	0	1	1	0	1	1	1
D13	1	1	1	0	0	0	0	1	1
D14	1	0	0	1	0	0	1	1	1
D15	1	0	1	1	1	1	0	1	0
D16	1	1	1	1	1	0	1	1	1
D17	0	1	1	1	1	0	1	1	1
D18	0	1	0	0	1	1	1	1	1
D19	0	1	0	0	1	1	1	1	0
D20	0	0	0	1	1	1	1	1	1
D21	1	0	0	0	0	1	1	0	1
D22	1	0	0	0	0	0	1	1	1
D23	1	0	0	0	0	0	1	1	1

Tabel 4.7 menunjukkan representasi frekuensi kemunculan kata dalam dokumen yang dianalisis. Setiap baris dalam tabel merepresentasikan sebuah kata fitur tertentu, sedangkan setiap kolom (D1-D23) mewakili dokumen yang dianalisis. Nilai 1 menandakan bahwa kata tersebut muncul dalam dokumen tertentu, sedangkan nilai 0 menunjukkan bahwa kata tersebut tidak ditemukan dalam dokumen tersebut.

Tabel 4.8 Perhitungan TF

Dokumen	TF							
	aplikasi	**cepat**	**bagus**	**ribet**	**proses**	**keluhan**	**error**	**bayar**
D1	0,143	0,143	0,143	0,143	0	0	0	0
D2	0,143	0,143	0	0	0,143	0,143	0	0
D3	0,111	0,111	0	0	0	0	0,111	0,111
D4	0,111	0,111	0,111	0	0	0	0,111	0
D5	0,111	0,111	0	0,111	0,111	0,111	0	0,091
D6	0,111	0,111	0	0,111	0	0	0	0,111
D7	0,111	0,111	0,111	0	0	0,111	0	0
D8	0,091	0,091	0	0	0,091	0	0	0,091
D9	0	0,111	0	0,111	0,111	0	0,111	0,111
D10	0,143	0,143	0,143	0	0	0	0	0,143
D11	0,100	0	0,100	0,100	0	0,100	0	0,100
D12	0	0,083	0	0,083	0,083	0	0,083	0,083
D13	0,143	0,143	0,143	0	0	0	0	0,143
D14	0,111	0	0	0,111	0	0	0,111	0,111
D15	0,111	0	0,111	0,111	0,111	0,111	0	0,111
D16	0,111	0,111	0,111	0,111	0,111	0	0,111	0,111
D17	0	0,100	0,100	0,100	0,100	0	0,100	0,100
D18	0	0,125	0	0	0,125	0,125	0,125	0,125
D19	0	0,125	0	0	0,125	0,125	0,125	0,125
D20	0	0	0	0,111	0,111	0,111	0,111	0,111
D21	0,200	0	0	0	0	0,200	0,200	0
D22	0,143	0	0	0	0	0	0,143	0,143
D23	0,143	0	0	0	0	0	0,143	0,143

Tabel 4.8 perhitungan TF memberikan gambaran mengenai distribusi kata dalam dokumen. Kata dengan nilai TF tinggi seperti 0,200 cenderung merepresentasikan topik utama dalam teks yang dianalisis, sedangkan kata dengan TF rendah seperti 0,083 yang hanya muncul dalam dokumen tertentu, dapat menjadi indikator penting dalam analisis sentimen atau klasifikasi teks.

Tabel 4.9 Perhitungan IDF

Df	IDF (log D/df)
17	0,131
13	0,248
10	0,362
14	0,216
12	0,283
11	0,320
17	0,131
18	0,106
16	0,158

Tabel 4.9 menunjukkan nilai *Inverse Document Frequency* (IDF) yang mengukur seberapa penting suatu kata dalam kumpulan dokumen. Untuk kata yang lebih umum memiliki IDF rendah, sementara kata yang lebih jarang muncul memiliki IDF tinggi, sehingga lebih bermakna dalam analisis teks. Langkah selanjutnya menghitung TF-IDF dengan mengalikan hasil TF dan IDF.

Tabel 4.10 Hasil Perhitungan TF-IDF

Dokumen	TF-IDF								
	aplikasi	cepat	bagus	ribet	proses	keluhan	error	bayar	pajak
D1	0,019	0,035	0,052	0,031	0	0	0	0	0
D2	0,019	0,035	0	0	0,040	0,046	0	0	0
D3	0,015	0,028	0	0	0	0	0,015	0,012	0,018
D4	0,015	0,028	0,040	0	0	0	0,015	0	0
D5	0,015	0,028	0	0,024	0,031	0,036	0	0,010	0
D6	0.015	0,028	0	0,024	0	0	0	0,012	0,018
D7	0,015	0,028	0,040	0	0	0,036	0	0	0,018
D8	0,012	0,023	0	0	0,026	0	0	0,010	0
D9	0	0,028	0	0,024	0,031	0	0,015	0,012	0
D10	0,029	0,035	0,052	0	0	0	0	0,015	0,023
D11	0,013	0	0,036	0,022	0	0,032	0	0,011	0,016
D12	0	0,021	0	0,018	0,024	0	0,011	0,009	0,013
D13	0,019	0,035	0,052	0	0	0	0	0.015	0,023
D14	0,015	0	0	0,024	0	0	0,015	0,012	0.018
D15	0,015	0	0,040	0,024	0,031	0,036	0	0,012	0
D16	0,015	0,028	0,040	0,024	0,031	0	0,015	0,012	0,018
D17	0	0,025	0,036	0.022	0,028	0	0,013	0,011	0,016
D18	0	0,031	0	0	0,035	0,040	0,016	0,013	0,02
D19	0	0,031	0	0	0,035	0,040	0,016	0,013	0
D20	0	0	0	0,024	0,031	0,036	0,015	0,012	0,018
D21	0,026	0	0	0	0	0,064	0.026	0	0,032
D22	0,019	0	0	0	0	0	0,019	0,015	0,023
D23	0,019	0	0	0	0	0	0,019	0,015	0,023

Tabel 4.10 TF-IDF menunjukkan nilai bobot setiap kata dalam dokumen berdasarkan tingkat kepentingannya. Kata yang sering muncul di banyak dokumen, seperti "aplikasi", memiliki bobot lebih rendah karena dianggap umum, sedangkan kata yang lebih jarang muncul tetapi spesifik dalam dokumen tertentu, seperti "keluhan" atau "bagus", memiliki bobot lebih tinggi. Nilai TF-IDF = 0 menunjukkan bahwa kata tersebut tidak muncul dalam dokumen tertentu atau

memiliki signifikansi yang sangat rendah. Hasil perhitungan TF-IDF ini kemudian digunakan pada tahap selanjutnya dalam penerapan algoritma *Naïve Bayes.*

4.6.2 SMOTE

Setelah dilakukan tahap seleksi fitur menggunakan *chi-square,* ditemukan bahwa distribusi data antar kelas tidak seimbang. Dalam dataset ini, kelas positif memiliki 2.605 sampel, sedangkan kelas negatif hanya 449 dan kelas netral hanya 122. Ketidakseimbangan ini dapat berdampak pada kinerja model pembelajaran mesin, di mana model cenderung lebih mudah mengenali pola dari kelas mayoritas (positif) dan kesulitan dalam mengklasifikasikan data dari kelas minoritas (negatif dan netral). Hal ini dapat menyebabkan model menjadi bias terhadap kelas positif, sehingga akurasi prediksi untuk kelas negatif dan netral menjadi lebih rendah. Oleh karena itu, untuk menangani ketidakseimbangan ini, diterapkan metode SMOTE guna menambah sampel sintetis pada kelas minoritas agar distribusi data menjadi lebih seimbang.

Proses SMOTE diawali dengan memilih sampel dari kelas minoritas, baik dari kelas negatif maupun netral. Berikut contoh tahapan pengambilan sampel sintetis:

1. Memilih Sampel dari Kelas Minoritas

   SMOTE bekerja dengan memilih beberapa sampel dari kelas minoritas (dalam hal ini kelas negatif dan netral) sebagai titik awal untuk membuat sampel baru. Misalnya, dalam kelas negatif terdapat dua sampel dengan representasi fitur numerik (0,02 dan 0,05) dan (0,04 dan 0,06).

2. Menentukan Tetangga Terdekat

Setelah memilih satu sampel dari kelas minoritas, SMOTE mencari beberapa sampel lain dalam kelas yang sama yang memiliki kemiripan fitur. Pemilihan tetangga terdekat ini biasanya menggunakan metode *K-Nearest Neighbors* (KNN) dengan nilai *k* yang telah ditentukan. Jarak *Euclidean* digunakan untuk menghitung kedekatan antara dua titik. Berikut contoh perhitungan jarak dengan sampel di atas dengan rumus pada Persamaan 2.13:

$$d = \sqrt{(0{,}04 - 0{,}02)^2 + (0{,}06 - 0{,}05)^2}$$

$$d = \sqrt{(0{,}02)^2 + (0{,}01)^2} = \sqrt{0{,}0005} \approx 0{,}022$$

Jadi, sampel (0,02 dan 0,05) dan (0,04 dan 0,06) memiliki jarak *Euclidean* sekitar 0,022.

3. Membuat Data Sintetis Baru

Setelah menemukan tetangga terdekat, SMOTE membuat sampel baru dengan cara mengambil titik di antara dua sampel yang sudah ada dalam ruang fitur. Data sintetis ini dibuat dengan interpolasi linier, yaitu menempatkan titik baru di antara kedua titik yang dipilih. Jika dua sampel awal adalah (0,02 dan 0,05) dan (0,04 dan 0,06), maka SMOTE dapat membuat sampel sintetis baru di antara dua sampel yang terdekat menggunakan interpolasi linier**.** Berikut contoh perhitungan menggunakan rumus pada Persamaan 2.14:

Di mana untuk nilai:

$$x_i = 0{,}02 \text{ dan } 0{,}05$$

$$x_j = 0{,}04 \text{ dan } 0{,}06$$

$$\lambda = 0{,}5$$

Untuk $x_i = 0{,}02$ dan $x_j = 0{,}04$ sebagai berikut:

$$x_{synthetic} = (0{,}02) + 0{,}5 \times (0{,}04 - 002)$$

$$x_{synthetic} = 0{,}02 + 0{,}01 = 0{,}03$$

Untuk $x_i = 0{,}05$ dan $x_j = 0{,}06$ sebagai berikut:

$$x_{synthetic} = (0{,}05) + 0{,}5 \times (0{,}06 - 005)$$

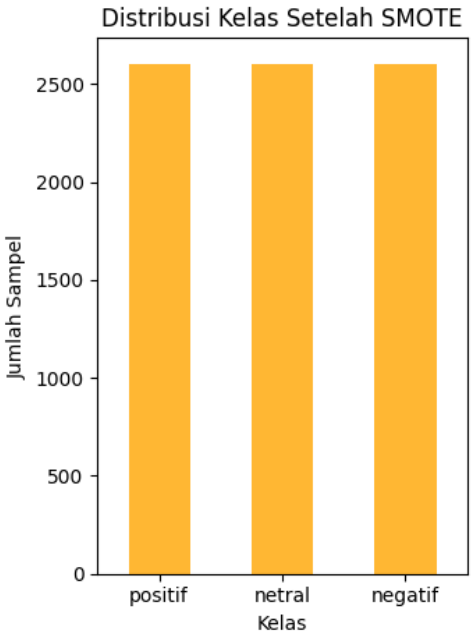
$$x_{synthetic} = 0{,}05 + 0{,}005 = 0{,}055$$

Jadi, titik sintetis yang dihasilkan adalah (0,03 dan 0,055) dengan $\lambda = 0{,}5$.

4. Menambahkan Data Sintetis ke Dataset

   Sampel sintetis yang telah dibuat kemudian ditambahkan ke dataset untuk meningkatkan jumlah data pada kelas minoritas. Proses ini diulang hingga kelas minoritas memiliki jumlah data yang lebih mendekati kelas mayoritas, sehingga distribusi data menjadi lebih seimbang. Setelah menerapkan SMOTE, jumlah sampel dalam kelas negatif dan netral akan meningkat, memungkinkan model pembelajaran mesin untuk lebih mudah mengenali pola dalam kelas tersebut.

Selain itu, metode SMOTE juga membantu mengurangi bias terhadap kelas mayoritas dan meningkatkan kemampuan model dalam mengklasifikasikan data dari kelas negatif dan netral secara lebih efektif. Berikut Gambar 4.5 hasil *oversampling* menggunakan SMOTE untuk data keseluruhan.



Gambar 4.5 Hasil *Oversampling* menggunakan SMOTE

Gambar 4.5 menunjukkan hasil setelah proses *oversampling* menggunakan metode SMOTE, terlihat bahwa distribusi jumlah data pada semua kelas telah seimbang. Data sebelum dilakukannya SMOTE, kelas positif memiliki 2.605 data, kelas negatif 449 data, dan kelas netral 122 data. Setelah SMOTE, setiap kelas menjadi memiliki jumlah data yang sama, yaitu 2.605 data, sehingga total keseluruhan data mencapai 7.815 data. Proses ini menghasilkan data sintetis baru untuk kelas minoritas (negatif dan netral) guna mencapai keseimbangan antar kelas.

4.6.3 *Feature Selection Chi - Square*

Metode *chi-square* digunakan untuk mengevaluasi hubungan antara masing-masing fitur dengan variabel target. Teknik ini bekerja dengan membandingkan distribusi frekuensi yang diamati (*observed*) dengan frekuensi yang diharapkan (*expected*) jika tidak ada hubungan antara fitur dan target. Hasil dari metode *chi-square* menghasilkan skor numerik untuk setiap fitur, di mana skor yang lebih tinggi menunjukkan hubungan yang lebih signifikan.

Pada penelitian ini, dilakukan seleksi terhadap fitur awal dengan menghitung nilai *chi-square* untuk masing-masing fitur. Selanjutnya, fitur-fitur dengan nilai pada χ^2 hitung > χ^2 tabel $\alpha = 0{,}05$ dianggap memiliki hubungan yang signifikan dengan kelas target, dan dipilih untuk dimasukkan ke dalam model analisis lanjutan. Sebagai contoh perhitungan manual *chi-square,* langkah awal menentukan kontigensi seperti pada Tabel 4.11.

Tabel 4.11 Kontingensi

	Chi-Square									
Sentimen	**aplikasi**	**cepat**	**bagus**	**ribet**	**proses**	**keluhan**	**error**	**bayar**	**pajak**	**Total**
Positif	5	8	3	5	6	1	5	3	5	41
Netral	5	2	4	6	3	8	6	7	3	44
Negatif	8	2	3	6	5	2	7	4	3	40
Total	18	12	10	17	14	11	18	14	11	125

Setelah Tabel 4.11 kontingensi ini disusun, langkah selanjutnya adalah menghitung nilai harapan (*E*) untuk setiap sel dalam tabel. Nilai harapan ini digunakan untuk mengevaluasi sejauh mana distribusi frekuensi yang diamati (*O*) pada tabel kontingensi sesuai dengan distribusi yang diharapkan berdasarkan asumsi independensi antara kategori sentimen dan kata-kata kunci. Perhitungan dengan menggunakan rumus pada Persamaan 2.12, sehingga perhitungannya seperti dibawah ini:

$$E_{Positif,aplikasi} = \frac{41 \times 18}{125} = 5{,}904$$

$$E_{Positif,cepat} = \frac{41 \times 12}{125} = 3{,}936$$

$$E_{Positif,bagus} = \frac{41 \times 10}{125} = 3{,}280$$

$$E_{Positif,ribet} = \frac{41 \times 17}{125} = 5{,}576$$

$$E_{Positif,proses} = \frac{41 \times 14}{125} = 4{,}592$$

$$E_{Positif,keluhan} = \frac{41 \times 11}{125} = 3{,}608$$

$$E_{Positif,error} = \frac{41 \times 18}{125} = 5{,}904$$

$$E_{Positif,bayar} = \frac{41 \times 14}{125} = 4{,}592$$

$$E_{Positif,pajak} = \frac{41 \times 11}{125} = 3{,}608$$

Kemudian menghitung untuk baris kedua:

$$E_{Netral,aplikasi} = \frac{44 \times 18}{125} = 6{,}336$$

$$E_{Netral,cepat} = \frac{44 \times 12}{125} = 4{,}224$$

$$E_{Netral,bagus} = \frac{44 \times 10}{125} = 3{,}520$$

$$E_{Netral,ribet} = \frac{44 \times 17}{125} = 5{,}984$$

$$E_{Netral,proses} = \frac{44 \times 14}{125} = 4{,}928$$

$$E_{Netral,keluhan} = \frac{44 \times 11}{125} = 3{,}872$$

$$E_{Netral,error} = \frac{44 \times 18}{125} = 6{,}336$$

$$E_{Netral,bayar} = \frac{44 \times 14}{125} = 4{,}928$$

$$E_{Netral,pajak} = \frac{44 \times 11}{125} = 3{,}872$$

Perhitungan baris ketiga:

$$E_{Negatif,aplikasi} = \frac{40 \times 18}{125} = 5{,}760$$

$$E_{Negatif,cepat} = \frac{40 \times 12}{125} = 3{,}840$$

$$E_{Negatif,bagus} = \frac{40 \times 10}{125} = 3{,}200$$

$$E_{Negatif,ribet} = \frac{40 \times 17}{125} = 5{,}440$$

$$E_{Negatif,proses} = \frac{40 \times 14}{125} = 4{,}480$$

$$E_{Negatif,keluhan} = \frac{40 \times 11}{125} = 3{,}520$$

$$E_{Negatif,error} = \frac{40 \times 18}{125} = 5{,}760$$

$$E_{Negatif,bayar} = \frac{40 \times 14}{125} = 4{,}480$$

$$E_{Negatif,pajak} = \frac{40 \times 11}{125} = 3{,}520$$

Dari perhitungan di atas, nilai *Expected Count* (*E*) dalam tabel ini menunjukkan perkiraan jumlah kemunculan setiap kata dalam kategori sentimen positif, netral, dan negatif berdasarkan distribusi total kata. Sehingga Tabel 4.12 kontingensi *Expected Count* sebagai berikut.

Tabel 4.12 *Expected Count*

	Nilai Harapan								
Sentimen	**aplikasi**	**cepat**	**bagus**	**ribet**	**proses**	**keluhan**	**error**	**bayar**	**pajak**
Positif	5,904	3,936	3,280	5,576	4,592	3,608	5,904	4,592	3,608
Netral	6,336	4,224	3,520	5,984	4,928	3,872	6,336	4,928	3,872
Negatif	5,760	3,840	3,200	5,440	4,480	3,520	5,760	4,480	3,520

Setelah menghitung nilai *Expected Count* yang ditunjukkan pada Tabel 4.12, langkah selanjutnya yaitu menghitung selisih kuadrat antara nilai yang diobservasi (*O*) dan nilai yang diharapkan (*E*), lalu membaginya dengan nilai yang diharapkan (*E*). Berdasarkan nilai *Expected Count* maka didapatkan perhitungan masing-masing *term* yaitu:

$$(positif, aplikasi) = \frac{(5 - 5{,}904)^2}{5{,}904} = 0{,}138$$

$$(positif, cepat) = \frac{(8 - 3{,}936)^2}{3{,}936} = 4{,}196$$

$$(positif, bagus) = \frac{(3 - 3{,}280)^2}{3{,}280} = 0{,}024$$

$$(positif, ribet) = \frac{(5 - 5{,}576)^2}{5{,}576} = 0{,}060$$

$$(positif, proses) = \frac{(6 - 4{,}592)^2}{4{,}592} = 0{,}432$$

$$(positif, keluhan) = \frac{(1-3{,}608)^2}{3{,}608} = 1{,}885$$

$$(positif, error) = \frac{(5-5{,}904)^2}{5{,}904} = 0{,}138$$

$$(positif, bayar) = \frac{(3-4{,}592)^2}{4{,}592} = 0{,}552$$

$$(positif, pajak) = \frac{(5-3{,}608)^2}{3{,}608} = 0{,}537$$

Perhitungan Kelas Netral:

$$(netral, aplikasi) = \frac{(5-6{,}336)^2}{6{,}336} = 0{,}282$$

$$(netral, cepat) = \frac{(2-4{,}224)^2}{4{,}224} = 1{,}171$$

$$(netral, bagus) = \frac{(4-3{,}520)^2}{3{,}520} = 0{,}065$$

$$(netral, ribet) = \frac{(6-5{,}984)^2}{5{,}984} = 0{,}001$$

$$(netral, proses) = \frac{(3-4{,}928)^2}{4{,}928} = 0{,}754$$

$$(netral, keluhan) = \frac{(8-3{,}872)^2}{3{,}872} = 4{,}401$$

$$(netral, error) = \frac{(6-6{,}336)^2}{6{,}336} = 0{,}018$$

$$(netral, bayar) = \frac{(7-4{,}928)^2}{4{,}928} = 0{,}871$$

$$(netral, pajak) = \frac{(3-3{,}872)^2}{3{,}872} = 0{,}196$$

Perhitungan Kelas Negatif:

$$(negatif, aplikasi) = \frac{(8-5{,}760)^2}{5{,}760} = 0{,}871$$

$$(negatif, cepat) = \frac{(2-3{,}840)^2}{3{,}840} = 0{,}882$$

$$(negatif, bagus) = \frac{(3-3{,}200)^2}{3{,}200} = 0{,}013$$

$$(negatif, tidak) = \frac{(6-5{,}440)^2}{5{,}440} = 0{,}058$$

$$(negatif, proses) = \frac{(5-4{,}480)^2}{4{,}480} = 0{,}060$$

$$(negatif, keluhan) = \frac{(2-3{,}520)^2}{3{,}520} = 0{,}656$$

$$(negatif, error) = \frac{(7-5{,}760)^2}{5{,}760} = 0{,}267$$

$$(negatif, bayar) = \frac{(4-4{,}480)^2}{4{,}480} = 0{,}051$$

$$(negatif, pajak) = \frac{(3-3{,}520)^2}{3{,}520} = 0{,}077$$

Setelah melakukan perhitungan untuk setiap kategori, langkah berikutnya adalah menjumlahkan seluruh hasilnya untuk memperoleh total *chi-square*, sebagai berikut:

$$\chi^2 \text{ (aplikasi)} = 0{,}138 + 0{,}282 + 0{,}871 = 1{,}291$$

$$\chi^2 \text{ (cepat)} = 4{,}196 + 1{,}171 + 0{,}882 = 6{,}249$$

$$\chi^2 \text{ (bagus)} = 0{,}024 + 0{,}065 + 0{,}013 = 0{,}102$$

$$\chi^2 \text{ (ribet)} = 0{,}060 + 0{,}001 + 0{,}058 = 0{,}119$$

$$\chi^2 \text{ (proses)} = 0{,}432 + 0{,}754 + 0{,}060 = 1{,}246$$

$$\chi^2 \text{ (keluhan)} = 1{,}885 + 4{,}401 + 0{,}656 = 6{,}942$$

$$\chi^2 \text{ (error)} = 0{,}138 + 0{,}018 + 0{,}267 = 0{,}423$$

$$\chi^2 \text{ (bayar)} = 0{,}552 + 0{,}871 + 0{,}051 = 1{,}475$$

$$\chi^2 \text{ (pajak)} = 0{,}537 + 0{,}196 + 0{,}077 = 0{,}810$$

Setelah menghitung nilai *chi-square* untuk setiap fitur, langkah selanjutnya adalah menentukan derajat kebebasan (df), yang digunakan untuk memahami distribusi *chi-square* dan mengukur apakah nilai yang ditemukan signifikan. Derajat kebebasan dihitung berdasarkan jumlah kategori dalam variabel target.

$$df = (r - 1)$$

$$df = (3 - 1) = 2$$

Dengan df = 2, artinya terdapat dua nilai yang bebas untuk bervariasi dalam pengujian ini. Ini menunjukkan bahwa saat menguji hubungan antara fitur dan kelas target, kita memiliki dua parameter yang dapat berubah secara independen dalam distribusi *chi-square*.

Dalam pengujian *chi-square*, keputusan didasarkan pada perbandingan antara nilai *chi-square* hitung (χ^2 hitung) dan nilai chi-square tabel (χ^2 tabel) pada taraf signifikansi tertentu. Apabila χ^2 hitung > χ^2 tabel, maka H_0 ditolak, yang berarti terdapat hubungan yang signifikan antara fitur yang diuji dengan kelas target. Sebaliknya, jika χ^2 hitung ≤ χ^2 tabel, maka H_0 gagal ditolak, yang berarti tidak ada cukup bukti untuk menyatakan adanya hubungan yang signifikan.

Berdasarkan tabel *chi-square* dengan nilai α = 0,05 dan nilai df = 2, didapatkan nilai *chi-square* tabel sebesar 5,991. Selanjutnya, dengan nilai *chi-square* yang telah dihitung dilakukan pengujian signifikansi untuk setiap fitur. Pengambilan keputusan dalam uji *chi-square* ini tidak hanya didasarkan pada nilai *p-value*, tetapi juga dibandingkan dengan nilai *chi-square* tabel pada α = 0,05, yaitu 5,991. Jika nilai *chi-square* hitung > 5,991, maka H_0 ditolak, yang berarti terdapat hubungan yang signifikan antara fitur/kata dengan kelas target. Sebaliknya, jika chi-square hitung ≤ 5,991, maka H_0 gagal ditolak. Langkah ini bertujuan untuk memastikan bahwa hanya fitur-fitur yang relevan dan signifikan akan dipertahankan dalam proses seleksi. Keputusan akhir untuk masing-masing fitur disajikan pada Tabel 4.13 berikut:

Tabel 4.13 Keputusan Uji *Feature Selection Chi-Square*

Fitur/Kata	*Chi-Square*	*p-value*	Keputusan
aplikasi	0,291	0,524	Gagal Tolak H_0 karena 0,291 < 5,991
cepat	6,249	0,044	Tolak H_0 karena 6,249 > 5,991
bagus	0,102	0,950	Gagal Tolak H_0 karena 0,102 < 5,991
ribet	0,119	0,944	Gagal Tolak H_0 karena 0,119 < 5,991
proses	1,246	0,537	Gagal Tolak H_0 karena 1,246 < 5,991
keluhan	6,942	0,031	Tolak H_0 karena 6,942 > 5,991
error	0,423	0,810	Gagal Tolak H_0 karena 0,423 < 5,991
bayar	1,475	0,477	Gagal Tolak H_0 karena 1,475 < 5,991
pajak	0,810	0,668	Gagal Tolak H_0 karena 0,810 < 5,991

Dari hasil uji *chi-square* pada Tabel 4.13, dapat disimpulkan bahwa hanya dua fitur/kata yang memiliki hubungan yang signifikan dengan kelas target, yaitu "cepat" dan "keluhan", karena nilai *chi-square* keduanya lebih besar dari nilai *chi-square* tabel (5,991). Sementara itu, fitur seperti "aplikasi", "bagus", "ribet", "proses", "error", "bayar", dan "pajak" memiliki nilai *chi-square* yang jauh di bawah ambang batas 5,991. Ini menunjukkan bahwa keberadaan kata-kata tersebut tidak memberikan kontribusi signifikan secara statistik terhadap pembeda antar kelas target. Oleh karena itu, fitur-fitur ini dapat dipertimbangkan untuk dieliminasi dalam tahap seleksi fitur guna menyederhanakan model dan mengurangi *noise*.

Demikian, berdasarkan hasil perhitungan nilai *chi-square* pada berbagai fitur, kita dapat menyimpulkan bahwa fitur "cepat" dan "keluhan" memiliki ketergantungan yang signifikan dengan kelas (positif, negatif, dan netral), sementara fitur lainnya seperti "aplikasi", "bagus", "ribet", "proses", "eror", "bayar", dan "pajak" tidak menunjukkan ketergantungan yang signifikan dengan kelas. Oleh karena itu, fitur "cepat" dan "keluhan" terpilih dalam seleksi fitur, sementara fitur lainnya terbuang. Seleksi fitur menggunakan *chi-square*

membantu dalam mengurangi jumlah fitur yang tidak relevan dengan target, sehingga memperbaiki efisiensi model.

4.6.4 Implementasi Algoritma *Gaussian Naïve Bayes*

Setelah melalui tahapan *preprocessing*, perhitungan TF-IDF, dan SMOTE, data latih telah dipersiapkan untuk membangun model klasifikasi yang bertujuan untuk menganalisis sentimen pada ulasan aplikasi SIGNAL. Data uji kemudian digunakan untuk mengukur kemampuan model dalam mengklasifikasikan sentimen ulasan berdasarkan fitur-fitur yang telah diekstraksi. Dalam implementasi ini, algoritma *Gaussian Naive Bayes* diaplikasikan untuk menghitung *probabilitas* sentimen, dengan asumsi bahwa setiap fitur bersifat independen dan mengikuti distribusi normal.

Pada tahap pengujian, algoritma menghitung *probabilitas* untuk setiap kelas sentimen (positif, netral, atau negatif) berdasarkan fitur-fitur pada data uji. Proses ini menggunakan rumus-rumus probabilistik untuk menghitung *probabilitas posterior*, yang kemudian digunakan untuk menentukan kelas dengan *probabilitas* tertinggi. Sentimen dari ulasan diklasifikasikan sesuai dengan hasil perhitungan ini. Hasil klasifikasi selanjutnya dievaluasi menggunakan metrik akurasi untuk mengukur sejauh mana model dapat memprediksi sentimen dengan benar. Evaluasi ini memberikan gambaran tingkat keberhasilan algoritma dalam menganalisis sentimen pada ulasan aplikasi SIGNAL.

Proses perhitungan *Gaussian Naïve Bayes* menggunakan nilai TF-IDF dari fitur “cepat” dan “keluhan” sebagai representasi fitur teks ulasan. Algoritma ini pertama-tama menghitung *probabilitas prior* untuk setiap kelas sentimen (positif, netral, dan negatif), serta rata-rata (*mean*) dan varians untuk setiap fitur dalam

masing-masing kelas. Pada tahap klasifikasi, algoritma menghitung *likelihood* berdasarkan distribusi *Gaussian*, di mana nilai *X* yang dipilih untuk fitur "cepat" adalah 0,125, dan untuk fitur "keluhan" adalah 0,143. Nilai ini berasal dari data *testing* yang berada di luar nilai TF-IDF yang telah dihitung sebelumnya. Nilai *X* tersebut digunakan dalam perhitungan *likelihood* untuk menentukan kemungkinan fitur tersebut muncul dalam masing-masing kelas sentimen. Setelah itu, nilai *likelihood* dikalikan dengan *probabilitas prior* untuk memperoleh *probabilitas posterior*. Data uji kemudian diklasifikasikan ke dalam kelas dengan *probabilitas posterior* tertinggi, yang menentukan sentimen akhir dari ulasan yang dianalisis. Berikut perhitungan *Naïve Bayes*:

1. *Probabilitas prior*

   Nilai *probabilitas prior* diperoleh berdasarkan jumlah masing-masing kelas pada sampel data sebanyak 23 ulasan, bukan dari keseluruhan dataset yang digunakan dalam pelatihan model. Oleh karena itu, angka 8 untuk kelas positif, 8 untuk kelas netral, dan 7 untuk kelas negatif menggambarkan distribusi sementara atau representatif dari data sampel, yang digunakan untuk menjelaskan perhitungan probabilitas dasar.

   P (Positif) $= \frac{Jumlah\ Kelas\ Positif}{Total\ Jumlah\ Data} = \frac{8}{23} = 0{,}348$

   P (Negatif) $= \frac{Jumlah\ Kelas\ Negatif}{Total\ Jumlah\ Data} = \frac{7}{23} = 0{,}304$

   P (Netral) $= \frac{Jumlah\ Kelas\ Netral}{Total\ Jumlah\ Data} = \frac{8}{23} = 0{,}348$

   Nilai *probabilitas prior* di atas diperoleh berdasarkan jumlah masing-masing kelas pada sampel data sebanyak 23 ulasan, bukan dari keseluruhan dataset yang digunakan dalam pelatihan model. Oleh karena itu, angka 8

untuk kelas positif, 8 untuk kelas netral, dan 7 untuk kelas negatif menggambarkan distribusi sementara atau representatif dari data sampel, yang digunakan untuk menjelaskan perhitungan probabilitas dasar.

Probabilitas prior mencerminkan distribusi awal kelas sebelum mempertimbangkan fitur tambahan, dengan sentimen positif dan netral masing-masing 0,348, serta negatif 0,304. Ini mengindikasikan bahwa dataset tidak terlalu bias terhadap satu kelas tertentu, sehingga model klasifikasi dapat bekerja lebih adil dalam mengenali pola sentimen.

2. Varians dan Rata-rata

Kelas Positif

Fitur "cepat"

$$\mu = \frac{0{,}035 + 0{,}035 + 0{,}028 + 0{,}035 + 0{,}021 + 0{,}035 + 0{,}025 + 0{,}031}{8} = 0{,}03$$

$$\sigma^2 = \frac{(0{,}035 - 0{,}03)^2 + (0{,}035 - 0{,}03)^2 + (0{,}028 - 0{,}03)^2 + (0{,}035 - 0{,}03)^2 +}{8}$$

$$\frac{(0{,}021 - 0{,}03)^2 + (0{,}035 - 0{,}03)^2 + (0{,}035 - 0{,}03)^2 + (0{,}025 - 0{,}03)^2 +}{8}$$

$$\frac{(0{,}031 - 0{,}03)^2}{8} = 1{,}221 \times 10^{-4}$$

Fitur "keluhan"

$$\mu = \frac{0 + 0{,}046 + 0 + 0 + 0 + 0 + 0 + 0{,}040}{8} = 0{,}01$$

$$\sigma^2 = \frac{(0 - 0{,}01)^2 + (0{,}046 - 0{,}01)^2 + (0 - 0{,}01)^2 + (0 - 0{,}01)^2 + (0 - 0{,}01)^2 +}{8}$$

$$\frac{(0 - 0{,}01)^2 + (0 - 0{,}01)^2 + (0{,}046 - 0{,}01)^2}{8} = 4{,}115 \times 10^{-4}$$

Kelas Negatif

Fitur "cepat"

$$\mu = \frac{0{,}028 + 0{,}028 + 0{,}028 + 0 + 0 + 0{,}031 + 0}{7} = 0{,}016$$

$$\sigma^2 = \frac{(0{,}028 - 0{,}016)^2 + (0{,}028 - 0{,}016)^2 + (0{,}028 - 0{,}016)^2 + (0 - 0{,}016)^2 +}{7}$$

$$\frac{(0 - 0{,}016)^2 + (0{,}031 - 0{,}016)^2 + (0 - 0{,}016)^2}{7} = 2{,}035 \times 10^{-4}$$

Fitur “keluhan”

$$\mu = \frac{0 + 0{,}036 + 0 + 0{,}032 + 0{,}036 + 0{,}040 + 0{,}064}{7} = 0{,}029$$

$$\sigma^2 = \frac{(0 - 0{,}029)^2 + (0{,}036 - 0{,}029)^2 + (0 - 0{,}029)^2 + (0{,}032 - 0{,}029)^2 +}{7}$$

$$\frac{(0{,}036 - 0{,}029)^2 + (0{,}040 - 0{,}022)^2 + (0{,}064 - 0{,}022)^2}{7} = 5{,}54 \times 10^{-4}$$

Kelas Netral

Fitur “cepat”

$$\mu = \frac{0{,}028 + 0{,}028 + 0{,}028 + 0 + 0{,}028 + 0 + 0 + 0}{8} = 0{,}014$$

$$\sigma^2 = \frac{(0{,}028 - 0{,}014)^2 + (0{,}028 - 0{,}014)^2 + (0 - 0{,}014)^2 + (0{,}028 - 0{,}014)^2}{8}$$

$$\frac{+(0 - 0{,}014)^2 + (0 - 0{,}014)^2 + (0 - 0{,}014)^2}{8} = 1{,}715 \times 10^{-4}$$

Fitur “keluhan”

$$\mu = \frac{0{,}036 + 0 + 0 + 0 + 0 + 0{,}036 + 0 + 0}{8} = 0{,}009$$

$$\sigma^2 = \frac{(0{,}036 - 0{,}009)^2 + (0 - 0{,}009)^2 + (0 - 0{,}009)^2 + (0 - 0{,}009)^2 +}{8}$$

$$\frac{(0{,}036 - 0{,}009)^2 + (0 - 0{,}009)^2 + (0 - 0{,}009)^2}{8} = 2{,}33 \times 10^{-4}$$

Hasil perhitungan rata-rata (μ) dan *varians* (σ^2) untuk fitur "cepat" dan "keluhan" pada masing-masing kelas sentimen menunjukkan bagaimana kedua fitur ini terdistribusi dalam dataset. Secara umum, fitur "cepat" memiliki rata-rata kemunculan yang lebih tinggi pada kelas positif (0,03)

dibandingkan dengan kelas negatif (0,016) dan netral (0,014), dengan varians yang relatif kecil, mengindikasikan bahwa fitur ini lebih sering muncul dalam ulasan positif dengan distribusi yang cukup stabil. Sebaliknya, fitur "keluhan" lebih dominan pada kelas negatif dengan rata-rata 0,029, dibandingkan dengan kelas positif (0,01) dan netral (0,009), serta memiliki varians yang lebih tinggi ($5{,}54 \times 10^{-4}$), yang menunjukkan bahwa penyebarannya lebih bervariasi dalam ulasan negatif.

3. *Probabilitas likelihood*

Kelas Positif

Fitur “cepat”:

$$P(0{,}125|C) = \frac{1}{\sqrt{2\pi(1{,}221\times10^{-4})}} exp(-\frac{(0{,}125-0{,}03)^2}{2(1{,}221\times10^{-4})}) = 3{,}21 \times 10^{-15}$$

Fitur “keluhan”:

$$P(0{,}143|C) = \frac{1}{\sqrt{2\pi(4{,}115\times10^{-4})}} exp(-\frac{(0{,}143-0{,}01)^2}{2(4{,}115\times10^{-4})}) = 9{,}11 \times 10^{-9}$$

Kelas Negatif

Fitur “cepat”:

$$P(0{,}125|C) = \frac{1}{\sqrt{2\pi(2{,}035\times10^{-4})}} \exp\left(-\frac{(0{,}125-0{,}016)^2}{2(2{,}035\times10^{-4})}\right) = 5{,}87 \times 10^{-12}$$

Fitur “keluhan”:

$$P(0{,}143|C) = \frac{1}{\sqrt{2\pi(5{,}54\times10^{-4})}} exp(-\frac{(0{,}143-0{,}029)^2}{2(5{,}54\times10^{-4})}) = 1{,}36 \times 10^{-4}$$

Kelas Netral

Fitur “cepat”:

$$P(0{,}125|C) = \frac{1}{\sqrt{2\pi(1{,}715\times10^{-4})}} \exp\left(-\frac{(0{,}125-0{,}014)^2}{2(1{,}715\times10^{-4})}\right) = 7{,}64 \times 10^{-15}$$

Fitur “keluhan”:

$$P(0{,}143|C) = \frac{1}{\sqrt{2\pi(2{,}33 \times 10^{-4})}} exp\left(-\frac{(0{,}143 - 0{,}009)^2}{2(2{,}33 \times 10^{-4})}\right) = 4{,}82 \times 10^{-16}$$

Sehingga nilai *likelihood* dari masing-masing sentimen sebagai berikut:

$P(X|C = Positif) = (3{,}21 \times 10^{-15}) \times (9{,}11 \times 10^{-9}) = 2{,}92 \times 10^{-23}$

$P(X|C = Negatif) = (5{,}87 \times 10^{-12}) \times (1{,}36 \times 10^{-4}) = 7{,}99 \times 10^{-16}$

$P(X|C = Netral) = (7{,}64 \times 10^{-15}) \times (4{,}82 \times 10^{-16}) = 3{,}68 \times 10^{-30}$

Hasil perhitungan *likelihood* menunjukkan seberapa besar kemungkinan fitur yang diamati (*X*) muncul dalam masing-masing kelas sentimen. Dari nilai yang diperoleh, *probabilitas P(X |C = Negatif)* = 7,99 × 10^{-16} lebih besar dibandingkan *P(X |C = Positif)* = 2,92 × 10^{-23} dan *P(X |C = Netral)* = 3,68 × 10^{-30}. Ini menunjukkan bahwa fitur yang diamati lebih sering muncul dalam ulasan dengan sentimen negatif, sementara kemungkinan fitur tersebut muncul dalam ulasan positif atau netral jauh lebih kecil. Dengan kata lain, berdasarkan *likelihood*, fitur ini memiliki kecenderungan lebih kuat untuk diklasifikasikan sebagai sentimen negatif, karena kemunculannya dalam kelas negatif lebih tinggi dibandingkan dua kelas lainnya. Namun, *likelihood* saja belum cukup untuk menentukan kelas akhir. Langkah selanjutnya adalah menghitung *probabilitas posterior*, yang menggabungkan informasi dari *likelihood* dengan *probabilitas prior* untuk menentukan kategori sentimen yang paling mungkin sesuai dengan data yang diamati.

4. *Probabilitas posterior*

   Kelas Positif

$$(2{,}92 \times 10^{-23}).\,0{,}348 = 1{,}02 \times 10^{-23}$$

Kelas Negatif

$$(7{,}99 \times 10^{-16}).\,0{,}304 = 2{,}43 \times 10^{-16}$$

Kelas Netral

$$(3{,}68 \times 10^{-30}).\,0{,}348 = 1{,}28 \times 10^{-30}$$

Berdasarkan hasil perhitungan sampel tersebut nilai *probabilitas* terbesar terdapat pada kelas negatif, maka fitur yang dianalisis masuk ke dalam sentimen negatif. Ini menunjukkan bahwa berdasarkan perhitungan *probabilitas*, fitur tersebut lebih cenderung dikategorikan sebagai bagian dari sentimen negatif dibandingkan dengan sentimen positif atau netral.

4.7 Evaluasi

4.7.1 *Cross Validation*

Model klasifikasi analisis sentimen dengan algoritma *Gaussian Naïve Bayes* menunjukkan performa yang cukup baik. Namun, sebelum diterapkan pada data berskala besar, model ini perlu dievaluasi lebih lanjut untuk memperoleh gambaran performa yang lebih akurat. Evaluasi dilakukan menggunakan metode *5-Fold Cross Validation*, yang membagi data menjadi lima bagian, di mana setiap bagian secara bergantian digunakan sebagai data uji, sementara empat bagian lainnya digunakan untuk pelatihan. Proses ini dilakukan sebanyak lima kali guna menghasilkan nilai akurasi yang lebih representatif terhadap keseluruhan data.

Dataset sebelum seleksi fitur berjumlah 10,389 fitur, setelah dilakukan seleksi fitur berjumlah 1000 fitur. Selanjutnya untuk dataset yang digunakan dalam evaluasi ini berjumlah berjumlah 7,815 data. Dataset tersebut kemudian

dibagi menjadi dua bagian yaitu sebanyak 80% dari dataset, yaitu 6,252 data, digunakan sebagai data *training*, sementara 20%, yaitu 1,563 data, digunakan sebagai data *testing*. Proses evaluasi dilakukan dengan membangun model klasifikasi Gaussian Naïve Bayes menggunakan data training, kemudian menguji performanya setelah model terbentuk. Hasil klasifikasi dan akurasinya disajikan pada Tabel 4.14.

Tabel 4.14 Hasil *Cross Validation* tanpa *Chi-Square*

K	Presisi	*Recall*	*F1-Score*	Akurasi
1	75%	68%	71%	68%
2	74%	70%	71%	70%
3	76%	69%	71%	69%
4	73%	65%	68%	65%
5	75%	66%	70%	66%
Rata-rata	74,60%	67,60%	71,00%	67,60%

Berdasarkan Tabel 4.14, diperoleh rata-rata presisi sebesar 74,60%**,** *recall* sebesar 67,60%**,** *F1-Score* sebesar 71,00%**,** dan akurasi sebesar 67,60%**.** Sedangkan hasil evaluasi *cross validation* menggunakan *chi-square* didapatkan hasil klasifikasi dengan akurasi yang ditampilkan pada Tabel 4.15.

Tabel 4.15 Hasil *Cross Validation* dengan *Chi-Square*

K	Presisi	*Recall*	*F1-Score*	Akurasi
1	84%	79%	78%	79%
2	85%	80%	79%	80%
3	84%	79%	78%	79%
4	83%	76%	75%	76%
5	84%	78%	77%	78%
Rata-rata	84,00%	78,40%	77,40%	78,40%

Berdasarkan Tabel 4.15, diperoleh rata-rata presisi sebesar 84,00%**,** *recall* sebesar 78,40%**,** *F1-Score* sebesar 77,40%**,** dan akurasi sebesar 78,40%**.** Hasil ini menunjukkan bahwa model memiliki performa yang cukup baik dan akurat dalam

mengklasifikasikan data, dengan nilai evaluasi yang konsisten di setiap metrik yang digunakan.

4.7.2 *Confusion Matrix*

Berikut merupakan hasil *confusion matrix* yang diperoleh dari model klasifikasi menggunakan algoritma *Naïve Bayes* menggunakan data testing. Hasil tersebut ditampilkan pada Tabel 4.16 *confusion matrix* dengan *chi-square*.

Tabel 4.16 *Confusion Matrix* tanpa *Chi-Square*

Label		**Hasil Prediksi**		
		Positif	**Netral**	**Negatif**
Data Aktual	**Positif**	220	120	40
	Netral	60	330	30
	Negatif	90	70	200

Berdasarkan Tabel 4.16 confusion matrix, hasil evaluasi menunjukkan bahwa model berhasil memprediksi 220 data sentimen positif dengan benar. Namun, terdapat 120 data sentimen positif yang salah diklasifikasikan sebagai netral, dan 40 data sebagai negatif. Untuk sentimen netral, sebanyak 330 data diprediksi dengan benar, sementara 60 data diklasifikasikan sebagai positif dan 30 sebagai negatif. Sedangkan pada sentimen negatif, model berhasil memprediksi 200 data dengan benar, namun terjadi kesalahan klasifikasi terhadap 90 data yang diprediksi sebagai positif dan 70 data sebagai netral.

Tabel 4.17 *Confusion Matrix* dengan *Chi-Square*

Label		**Hasil Prediksi**		
		Positif	**Netral**	**Negatif**
Data Aktual	**Positif**	412	34	9
	Netral	16	455	0
	Negatif	45	84	309

Berdasarkan Tabel 4.17 *confusion matrix* dengan *chi-square*, model berhasil memprediksi dengan benar 412 data sentimen positif, 455 data netral, dan 309 data negatif. Kesalahan prediksi terjadi pada 34 data positif yang diklasifikasikan sebagai netral dan 9 sebagai negatif, 16 data netral diklasifikasikan sebagai positif, serta 45 data negatif diprediksi sebagai positif dan 84 sebagai netral. Hasil ini kemudian menjadi dasar perhitungan metrik evaluasi seperti akurasi**,** presisi**,** *recall***,** dan *F1-Score***.**

Perhitungan manual tanpa *chi-square*

1. Nilai Akurasi

$$\text{Akurasi} = \frac{220+330+200}{220+330+120+40+60+30+90+70} \times 100\%$$

$$= \frac{750}{1170} \times 100\% = 72{,}00\%$$

2. Nilai Presisi

$$\text{Presisi Kelas Positif} = \frac{220}{220+210} \times 100\% = \frac{220}{430} = 51{,}16\%$$

$$\text{Presisi Kelas Netral} = \frac{330}{330+130} \times 100\% = \frac{330}{460} = 71{,}74\%$$

$$\text{Presisi Kelas Negatif} = \frac{200}{200+70} \times 100\% = \frac{200}{270} = 74{,}07\%$$

3. Nilai *Recall*

$$\textit{Recall}\text{ Kelas Positif} = \frac{220}{220+160} \times 100\% = \frac{220}{380} = 57{,}89\%$$

$$\textit{Recall}\text{ Kelas Netral} = \frac{330}{330+90} \times 100\% = \frac{330}{420} = 78{,}57\%$$

$$\textit{Recall}\text{ Kelas Negatif} = \frac{200}{200+110} \times 100\% = \frac{200}{310} = 64{,}52\%$$

4. Nilai *F1-Score*

$$\textit{F1-Score}\text{ Kelas Positif} = 2 \times \frac{51{,}16 \times 57{,}89}{51{,}16 + 57{,}89} \times 100\% = 54{,}28\%$$

$$\textit{F1-Score}\text{ Kelas Netral} = 2 \times \frac{71{,}74 \times 78{,}57}{71{,}74 + 78{,}57} \times 100\% = 75{,}88\%$$

$$\textit{F1-Score}\text{ Kelas Negatif} = 2 \times \frac{74{,}07 \times 64{,}52}{74{,}07 + 64{,}52} \times 100\% = 69{,}87\%$$

Perhitungan Manual dengan *chi-square*

1. Nilai Akurasi

   $$\text{Akurasi} = \frac{412+455+309}{412+34+9+16+455+0+45+84+309} \times 100\% = \frac{1176}{1364} = 86{,}19\%$$

2. Nilai Presisi

   $$\text{Presisi Kelas Positif} = \frac{412}{412+61} \times 100\% = \frac{412}{473} = 87{,}17\%$$

   $$\text{Presisi Kelas Netral} = \frac{455}{455+118} \times 100\% = \frac{455}{573} = 79{,}41\%$$

   $$\text{Presisi Kelas Negatif} = \frac{309}{309+9} \times 100\% = \frac{309}{318} = 97{,}17\%$$

3. Nilai *Recall*

   $$\textit{Recall}\ \text{Kelas Positif} = \frac{412}{412+43} \times 100\% = \frac{412}{455} = 90{,}55\%$$

   $$\textit{Recall}\ \text{Kelas Netral} = \frac{455}{455+16} \times 100\% = \frac{455}{471} = 96{,}60\%$$

   $$\textit{Recall}\ \text{Kelas Negatif} = \frac{309}{309+129} \times 100\% = \frac{309}{438} = 70{,}55\%$$

4. Nilai *F1-Score*

   $$\textit{F1-Score}\ \text{Kelas Positif} = 2 \times \frac{87{,}17 \times 90{,}55}{87{,}17 + 90{,}55} \times 100\% = 88{,}83\%$$

   $$\textit{F1-Score}\ \text{Kelas Netral} = 2 \times \frac{79{,}41 \times 96{,}60}{79{,}41 + 96{,}60} \times 100\% = 87{,}19\%$$

   $$\textit{F1-Score}\ \text{Kelas Negatif} = 2 \times \frac{97{,}17 \times 70{,}55}{97{,}17 + 70{,}55} \times 100\% = 81{,}90\%$$

Tabel 4.18 Hasil Perbandingan Evaluasi

Hasil Evaluasi		***Naïve Bayes* tanpa *Chi-Square***	***Naïve Bayes* dengan *Chi-Square***
Akurasi		72,00%	86,19%
Presisi	Positif	51,16%	87,17%
	Netral	71,74%	79,41%
	Negatif	74,07%	97,17%
Recall	Positif	57,89%	90,55%
	Netral	78,57%	96,60%
	Negatif	64,52%	70,55%
F1-Score	Positif	54,28%	88,83%
	Netral	75,88%	87,19%
	Negatif	69,87%	81,90%

Berdasarkan hasil perbandingan evaluasi pada Tabel 4.18, penerapan *chi-square* pada algoritma *Naïve Bayes* memberikan peningkatan yang signifikan pada performa model. Akurasi model meningkat dari 72,00% (tanpa *chi-square*) menjadi 86,19% (dengan *chi-square*), menunjukkan bahwa kombinasi kedua teknik ini berhasil memperbaiki kemampuan prediksi secara keseluruhan. Pada kelas positif, presisi meningkat dari 51,16% menjadi 87,17%, *recall* dari 57,89% menjadi 90,55%, dan *F1-Score* dari 54,28% menjadi 88,83%, yang menandakan peningkatan performa yang stabil dan signifikan.

Untuk kelas netral, penerapan *chi-square* juga memberikan peningkatan yang sangat jelas. Presisi meningkat dari 71,74% menjadi 79,41%, *recall* dari 78,57% menjadi 96,60%, dan *F1-Score* dari 75,88% menjadi 87,19%. Hasil ini menunjukkan bahwa model dengan *chi-square* jauh lebih baik dalam mengenali sentimen netral dibandingkan sebelumnya. Pada kelas negatif, performa juga mengalami peningkatan konsisten dengan presisi naik dari 74,07% menjadi 97,17%, *recall* dari 64,52% menjadi 70,55%, dan *F1-Score* dari 69,87% menjadi 81,90%. Secara keseluruhan, seleksi fitur dengan *chi-square* terbukti mampu meningkatkan akurasi dan ketepatan klasifikasi model *Naïve Bayes* untuk ketiga kelas sentimen

4.8 Pembahasan

Berdasarkan hasil penelitian, penggunaan *Gaussian Naïve Bayes* dengan *chi-square* membantu meningkatkan akurasi. Akurasi yang diperoleh setelah penerapan *chi-square* mencapai 86,19%. Selain itu, penerapan *chi-square* sebagai teknik seleksi fitur turut berkontribusi dalam meningkatkan performa model, menunjukkan bahwa kombinasi *chi-square* efektif dalam memperbaiki kemampuan

prediksi secara keseluruhan dibandingkan tanpa *chi-square* yang hanya mendapatkan akurasi sebesar 72,00%.

Penerapan *Gaussian Naïve Bayes* yang dibantu dengan *chi-square* juga menunjukkan peningkatan pada metrik evaluasi lainnya, khususnya untuk kelas positif dan negatif. Pada kelas positif, presisi mencapai 87,17%, *recall* 90,55%, dan *F1-Score* 88,83%. Sementara itu, pada kelas negatif, presisi sebesar 97,17%, *recall* 70,55%, dan *F1-Score* 81,90%. Untuk kelas netral, model juga menunjukkan performa yang baik dengan presisi 79,41%, *recall* 96,60%, dan *F1-Score* 87,19%. Secara keseluruhan, *Gaussian Naïve Bayes* mampu mengklasifikasikan data dengan baik, ditunjukkan oleh akurasi dan metrik evaluasi yang tinggi di semua kelas sentimen setelah dilakukan seleksi fitur dan penyeimbangan data.

4.9 Analisis Sentimen Aplikasi SIGNAL dalam Perspektif Islam

Pelayanan publik yang adil merupakan salah satu prinsip utama yang ditekankan dalam Al-Qur'an, sebagaimana tertuang dalam Surah Al-Maidah ayat 8. Ayat ini menekankan pentingnya menegakkan keadilan tanpa keberpihakan, baik terhadap orang kaya maupun miskin, teman maupun kerabat, sehingga pelayanan yang diberikan harus berdasarkan prinsip kebenaran. Dalam konteks pelayanan publik, ayat ini memberikan landasan untuk bertindak obyektif, transparan, dan tidak dipengaruhi oleh hawa nafsu atau kepentingan pribadi. Aspek-aspek penting dari ayat ini meliputi universalitas keadilan, tanggung jawab moral, serta pentingnya obyektivitas dalam keputusan, sehingga setiap individu mendapatkan hak mereka tanpa diskriminasi atau perlakuan yang tidak adil.

Dalam era modern, aplikasi SIGNAL menjadi anugerah dalam pelayanan publik karena memberikan kemudahan kepada masyarakat untuk mengakses

layanan secara cepat, efisien, dan transparan. Melalui aplikasi ini, masyarakat dapat menyampaikan keluhan, memberikan masukan, dan menerima solusi dengan lebih efektif dibandingkan metode konvensional. SIGNAL membantu menciptakan ekosistem pelayanan yang inklusif dan memungkinkan setiap individu mendapatkan haknya secara merata. Keberadaan aplikasi ini juga mendukung transparansi dan akuntabilitas penyedia layanan karena seluruh data dan interaksi dapat dicatat dan dianalisis dengan sistematis.

Untuk mendukung pelayanan publik yang lebih adil, algoritma *Gaussian Naïve Bayes* dapat diterapkan dalam analisis sentimen terhadap ulasan dan masukan pengguna aplikasi SIGNAL. Algoritma ini bekerja dengan mengklasifikasikan sentimen masyarakat menjadi kategori positif, netral, atau negatif berdasarkan data yang diperoleh. Proses ini memungkinkan penyedia layanan untuk mengidentifikasi aspek-aspek yang menjadi keluhan atau kebutuhan utama masyarakat, seperti responsivitas layanan, kualitas aplikasi, atau kecepatan penyelesaian masalah. Dengan *Gaussian Naïve Bayes*, analisis data dilakukan secara obyektif tanpa memihak kelompok tertentu atau faktor subjektif seperti lokasi, gender, atau usia pengguna, sehingga pengambilan keputusan dapat didasarkan pada fakta dan data aktual.

Penerapan algoritma *Gaussian Naïve Bayes* dalam analisis sentimen ini sejalan dengan prinsip keadilan yang diajarkan dalam Surah Al-Maidah ayat 8. Keputusan berbasis data membantu penyedia layanan bertindak tanpa diskriminasi dan memberikan solusi yang relevan bagi masyarakat. Dengan demikian, aplikasi SIGNAL tidak hanya menjadi alat untuk meningkatkan kualitas pelayanan publik tetapi juga menjadi sarana untuk mewujudkan keadilan sebagaimana yang

diajarkan dalam Islam. Integrasi teknologi modern ini memungkinkan keadilan ditegakkan secara lebih efektif, transparan, dan sesuai dengan nilai-nilai universal yang terkandung dalam ajaran Al-Qur'an.

BAB V
KESIMPULAN

5.1 Kesimpulan

Berdasarkan rumusan masalah dan pembahasan pada bab sebelumnya maka diperoleh kesimpulan sebagai berikut:

1. Tingkat akurasi yang diperoleh dari analisis sentimen pada data ulasan aplikasi SIGNAL menggunakan metode *Gaussian Naïve Bayes* dengan SMOTE mencapai 86,19%. Hal ini menunjukkan bahwa metode *Gaussian Naïve Bayes* mampu mengklasifikasikan data teks dengan cukup baik, terutama setelah penerapan teknik *oversampling* seperti SMOTE untuk menangani ketidakseimbangan data.
2. Pengaruh penggunaan teknik seleksi fitur *chi-square* terhadap analisis sentimen juga sangat signifikan dalam meningkatkan akurasi model. Dengan penerapan *chi-square*, fitur-fitur yang lebih relevan dapat dipilih, sehingga performa klasifikasi secara keseluruhan menjadi lebih baik. Hal ini terbukti dari peningkatan akurasi hingga 86,19%, presisi untuk kelas positif sebesar 87,17%, *recall* untuk kelas positif mencapai 90,55%, dan *F1-score* untuk kelas positif mencapai 88,83%. Pada kelas negatif, presisi mencapai 97,17%, *recall* 70,55%, dan *F1-Score* 81,90%. Namun, performa pada kelas netral masih rendah, dengan presisi 7,41%, *recall* 96,60%, dan *F1-Score* 87,19%.

5.2 Saran

Adapun beberapa hal yang dapat dijadikan acuan untuk penyempurnaan penelitian selanjutnya maka penulis memberikan saran berikut:

1. Penelitian selanjutnya diharapkan dapat mengatasi kelemahan model pada kategori sentimen netral, misalnya dengan mengeksplorasi teknik praproses data atau algoritma klasifikasi yang lebih kompleks, seperti *deep learning*.
2. Penggunaan metode lain seperti *Multinomial Naïve Bayes* atau kombinasi beberapa algoritma dapat dipertimbangkan untuk memperoleh hasil yang lebih optimal dalam klasifikasi sentimen.
3. Disarankan untuk menggunakan dataset yang lebih besar dan beragam untuk meningkatkan generalisasi model, sehingga performa klasifikasi pada data baru dapat lebih andal.
4. Penelitian dapat diperluas dengan mengintegrasikan metode lain untuk menangani ketidakseimbangan data, seperti ADASYN atau *ensemble learning*, untuk membandingkan hasilnya dengan SMOTE.
5. Analisis lebih mendalam pada kelas sentimen netral diperlukan untuk memahami faktor-faktor yang mempengaruhi rendahnya performa model pada kelas ini, serta mencari solusi yang lebih efektif.
6. Penelitian selanjutnya juga dapat mengembangkan analisis berbasis aspek (*aspect-based sentiment analysis*) untuk menggali opini pengguna terhadap fitur atau layanan tertentu secara lebih spesifik, sehingga hasil klasifikasi menjadi lebih informatif dan aplikatif.

DAFTAR PUSTAKA

Ahmad, M. (2023). Human Rights In Islam: *Compatible And Incompatible Aspects*. *International Journal Of Psychosocial Rehabilitation*, *24*, 2020. Https://Doi.Org/10.53555/V24i8/400248

Al-Qaradhawi, Y. (2010). *Fiqh Al-Zakah. Beirut: Dar Al-Fikr*. Https://Archive.Org/Details/Terjemah-Fiqhul-Zakat-Qardhawi.

An-Nawawi, I. (2007). *Keadilan Dalam Islam: Perspektif Fikih Dan Sosial*. Https://Hidayatullah.Com/.

Candra, C., Chandra, K. W., & Irsyad, H. (2024). Efektifitas Smote Dalam Mengatasi Imbalanced Class Algoritma *K-Nearest Neighbors* Pada Analisis Sentimen Terhadap Starlink. *Jurnal Ilmu Komputer Dan Informatika*, *4*(1), 31–42. Https://Doi.Org/10.54082/Jiki.132

Charis F, M., Ammar A, M., Wijokongko, D., & Alhafizd, M. (2020). Kategori Kepemimpinan Dalam Islam. *Jurnal Edukasi Nonformal*, *1*, 171–189.

Ernayanti, T., Mustafid, M., Rusgiyono, A., & Hakim, A. R. (2023). Penggunaan Seleksi Fitur Chi-Square Dan Algoritma Multinomial *Naïve Bayes* Untuk Analisis Sentimen Pelangggan Tokopedia. *Jurnal Gaussian*, *11*(4), 562–571. Https://Doi.Org/10.14710/J.Gauss.11.4.562-571

Fersellia, F., Utami, E., & Yaqin, A. (2023). *Sentiment Analysis Of Shopee Food Application User Satisfaction Using The C4.5 Decision Tree Method*. *Sinkron*, *8*(3), 1554–1563. Https://Doi.Org/10.33395/Sinkron.V8i3.12531

Firdaus, M. R., Rahaningsih, N., & Dana, R. D. (2024). Analisis Sentimen Aplikasi Shopee Di Goole Play Store Menggunakan Klasifikasi Algoritma *Naïve Bayes*. *Jurnal Informatika Dan Rekayasa Perangkat Lunak*, *6*, 228–237.

Franseda, A., Kurniawan, W., Anggraeni, S., & Gata, W. (2020). Integrasi Metode Decision Tree Dan Smote Untuk Klasifikasi Data Kecelakaan Lalu Lintas. *Jurnal Sistem Dan Teknologi Informasi (Justin)*, *8*(3), 282. Https://Doi.Org/10.26418/Justin.V8i3.40982

Gede, I., Sudipa, I., & Darmawiguna, M. (2024). *Buku Ajar Data Mining*. Https://Www.Researchgate.Net/Publication/377415198

Harun, N. (2013). Makna Keadilan Dalam Perspektif Hukum Islam Dan Perundang-Undangan. *Jurnal Al-Syir'ah*, *11*.

Hr. Muslim No. 1827. (T.T.). *Sahih Muslim*. Https://Sunnah.Com/Muslim/33.

Huda, I. (2019). *Implementasi Natural Language Processing (Nlp) Untuk Aplikasi Pencarian Lokasi*.

Ibnu Katsir. (1997). *Tafsir Ibnu Katsir 7*. Https://Www.Alquran-Sunnah.Com/Download/18-E-Book/7-Tafsir-Dan-Hadits/165-Tafsir-Ibnu-Katsir-7.Html?Start=20.

Kacung, S., Pamungkas, C., Bagyana, P., & Cahyono, D. (2024). *Analisis Sentimen Terhadap Layanan Samsat Digital Nasional (Signal) Menggunakan Metode Svm* (Vol. 7, Nomor 1).

Kartikasari, T. S., Setiawan, H., & Lucky Tirma Irawan, P. (2020). Implementasi Text Mining Untuk Analisis Opini Publik Terhadap Calon Presiden. *Jurnal Simantec*, *7*(1), 39–47. Https://Doi.Org/10.21107/Simantec.V7i1.6528

Kemenag. (2024). *Al-Quran Surah Al-Maidah Ayat 8*. Https://Quran.Kemenag.Go.Id/Quran/Per-Ayat/Surah/5?From=8&To=120.

Kepolisian Negara Republik Indonesia. (2023). *Perkembangan Jumlah Kendaraan Bermotor Menurut Jenis (Unit)*. Badan Pusat Statistik (Bps) . Https://Www.Bps.Go.Id/Id/Statisticstable/3/Vjj3ngrga3dkrk5mtlu1bvnfotvvbmqyvurstvfumdkjmw==/Jumlah-Kendaraan-Bermotor-Menurut-Provinsi-Dan-Jenis-Kendaraan--Unit---2023.Html?Year=2023

Kevin, K., Enjeli, M., & Wijaya, A. (2024). Analisis Sentimen Pengunaaan Aplikasi Kinemaster Menggunakan Metode Naive Bayes. *Jurnal Ilmiah Computer Science*, *2*(2). Https://Doi.Org/10.58602/Jics.V2i2.24

Kevin, P. M. (2022). *Probabilistic Machine Learning*.

Khoirul, M., Hayati, U., & Nurdiawan, O. (2023). Analisis Sentimen Aplikasi Brimo Pada Ulasan Pengguna Di Google Play Menggunakan Algoritma Naive Bayes. Dalam *Jurnal Mahasiswa Teknik Informatika* (Vol. 7, Nomor 1).

Liawati, A., Narasati, R., Solihudin, D., Lukman Rohmat, C., & Eka Permana, S. (2023). Analisis Sentimen Komentar Politik Di Media Sosial X Dengan Pendekataan Deep Learning. Dalam *Jurnal Mahasiswa Teknik Informatika* (Vol. 7, Nomor 6).

Liawati, A., Narasati, R., Solihudin, D., Lukman Rohmat, C., & Eka Permana, S. (2024). Analisis Sentimen Komentar Politik Di Media Sosial X Dengan Pendekataan Deep Learning. *Jati (Jurnal Mahasiswa Teknik Informatika)*, *7*(6). Https://Doi.Org/10.36040/Jati.V7i6.8248

Permana, H., Herry Chrisnanto, Y., & Ashaury, H. (2024). Analisis Sentimen Terhadap Bakal Calon Presiden 2024 Dengan Algoritma Multinomial *Naïve Bayes* Dan Oversampling Smote. *Jati (Jurnal Mahasiswa Teknik Informatika)*, *7*(5). Https://Doi.Org/10.36040/Jati.V7i5.7309

Purnamasari, D., Bayu, A., Desy, A., Fanka, W. A. P., Reza, A., Safrila, M., Yanda, O. N., & Hidayati, U. (2023). *Pengantar Metode Analisis Sentimen*.

Rahman, I. F., Hasanah, A. N., & Heryana, N. (2024). Analisis Sentimen Ulasan Pengguna Aplikasi Samsat Digiital Nasional (Signal) Dengan Menggunakan Metode *Naïve Bayes* Classifier. *Jurnal Informatika Dan Teknik Elektro Terapan*, *12*(2). Https://Doi.Org/10.23960/Jitet.V12i2.4073

Reza Satria, A., & Adinugroho, S. (2020). *Analisis Sentimen Ulasan Aplikasi Mobile Menggunakan Algoritma Gabungan Naïve Bayes Dan C4.5 Berbasis Normalisasi Kata Levenshtein Distance* (Vol. 4, Nomor 11). Http://J-Ptiik.Ub.Ac.Id

Rizky Hanafi, M., & Kurniawan, R. (2024). *Sentiment Analysis On Sirekap App Reviews On Google Play Using Naive Bayes Algorithm*. *Malcom: Indonesian Journal Of Machine Learning And Computer Science*, *4*, 1578–1586. Https://Doi.Org/10.57152/Malcom.V4i4.1693

Salsabilla, T., & Sulastri, S. (2022). Implementasi Algoritma C4.5 Untuk Klasifikasi Produk Laris Sepeda Motor Honda Pada Cv Cendana Motor Cepiring. *Rabit : Jurnal Teknologi Dan Sistem Informasi Univrab*, *7*(2), 164–171. Https://Doi.Org/10.36341/Rabit.V7i2.2489

Samsir, Ambiyar, Verawardina, U., Edi, F., & Watrianthos, R. (2021). Sistem Informasi Umkm Bengkel Berbasis Web Menggunakan Metode Scrum. *Jurnal Media Informatika Budidarma*, *5*(1), 149. Https://Doi.Org/10.30865/Mib.V5i1.2604

Septiani, D., & Isabela, I. (2022). Analisis Term Frequency Inverse Document Frequency (Tf-Idf) Dalam Temu Kembali Informasi Pada Dokumen Teks. *Sintesia: Jurnal Sistem Dan Teknologi Informasi Indonesia*, *1*, 81.

Sriani, Suhardi, & Saufa Yardha, L. (2024). Analisis Sentimen Pengguna Aplikasi Mobile Jkn Menggunakan Algoritma *Naïve Bayes*. Dalam *Journal Of Science And Social Research* (Nomor 2). Http://Jurnal.Goretanpena.Com/Index.Php/Jssr

Sucipto, S. C., Sulandari, W., & Susanti, Y. (2024). *Implementation Of Cross-Validation On Hang Seng Index Forecasting Using Holts's Exponential Smoothing And Auto-Arima Method*. *Jurnal Ilmu Matematika Dan Terapan*, *19*(1), 13–24.

Surya Firmansyah, A., Aziz, A., & Ahsan, Moh. (2024). Optimasi K-Nearest Neighbor Menggunakan Algoritma Smote Untuk Mengatasi Imbalance Class Pada Klasifikasi Analisis Sentimen. *Jati (Jurnal Mahasiswa Teknik Informatika)*, *7*(6). Https://Doi.Org/10.36040/Jati.V7i6.7257

Syaripah, I., Martanto, M., & Suprapti, T. (2024). Analisis Sentimen Pengguna Terhadap Aplikasi Binance Pada Ulasan Google Play Store Menggunakan Algoritma *Naïve Bayes*. *Jati (Jurnal Mahasiswa Teknik Informatika)*, *7*(6). Https://Doi.Org/10.36040/Jati.V7i6.8289

Wati, R., & Ernawati, S. (2021). Analisis Sentimen Persepsi Publik Mengenai Ppkm Pada Twitter Berbasis Svm Menggunakan Python. *Jurnal Teknik Informatika Unika Santo Thomas*. Https://Doi.Org/10.54367/Jtiust.V6i2.1465

Widodo, S., Brawijaya, H., & Samudi, S. (2022). *Stratified K-Fold Cross Validation Optimization On Machine Learning For Prediction*. *Sinkron*, *7*(4), 2407–2414. Https://Doi.Org/10.33395/Sinkron.V7i4.11792

Young, T., Hazarika, D., Poria, S., & Cambria, E. (2017). *Recent Trends In Deep Learning Based Natural Language Processing*. Http://Arxiv.Org/Abs/1708.02709

Yulian Pamuji, F. (2022). Pengujian Metode Smote Untuk Penanganan Data Tidak Seimbang Pada Dataset Binary. *Seminar Nasional Sistem Informasi*, *2022*.

Yuni Ikhsanti, A., Fauziah, Y., & Perwira, R. I. (2021). *Computing And Information Processing Letters Implementation Of The C4.5 Decision Tree Learning Algorithm For Sentiment Analysis In E-Commerce Application Reviews On Google Play Store*. *1*(1), 25–30. Https://Doi.Org/10.31315/Cip.Vxix.Xx

LAMPIRAN

Lampiran 1 Data Ulasan Aplikasi SIGNAL

https://bit.ly/DataUlasanSIGNAL

Contoh Data:

No	*Content*	*Score*	*At*
1	aplikasi sangat membantu..., proses nya cepat...terima kasih	5	6/8/2024 11:33
2	Alhamdulillah dikirim ke rumah dengan cepat, gak perlu fc ktp , stnk dan bpkb, yg penting pajak masuk kas negara	5	6/8/2024 11:21
3	Enak kalau diluar daerah,dari pembuatan STNK kita	5	6/8/2024 11:21
4	Sangat membantu	5	6/8/2024 11:15
5	mantap bayar pajak jadi mudah	5	6/8/2024 11:13
…	…	…	…
4997	Saya sangat terbantu dengan adanya sistem pajak online, maju terus untuk POLRI 👍👍👍👍👮‍♂️ID	5	3/28/2024 8:14
4998	Mempermudah	5	3/28/2024 7:52
4999	Terimakasih sangat membantu,,,	5	3/28/2024 7:41
5000	Mantaaaap	5	3/28/2024 7:28
5001	Daftar sulit banget Padahal udah bener semua gajelas apknya	1	3/28/2024 7:19

Lampiran 2 *Source Code Scrapping* Data

```
pip install google-play-scraper
from google_play_scraper import search, reviews_all
import pandas as pd
# Kata kunci untuk mencari aplikasi
query = 'samsat digital nasional'
# Mencari aplikasi berdasarkan kata kunci
result = search(query, lang='en', country='ID')
if not result:
    print("Aplikasi tidak ditemukan.")
else:
    # Mengambil ID aplikasi pertama yang ditemukan
    app_id = result[0]['appId']
```

```
# Mengambil data ulasan
reviews = reviews_all(
    app_id,
    lang='id',  # bahasa ulasan
    country='id'  # negara
)
# Memeriksa jumlah ulasan yang diambil
if len(reviews) > 5000:
    reviews = reviews[:5000]
elif len(reviews) < 5000:
    print(f'Only {len(reviews)} reviews available')
# Mengonversi data ulasan ke DataFrame
df = pd.DataFrame(reviews)
# Menampilkan beberapa baris pertama dari DataFrame
print(df.head())
# Menyimpan data ulasan ke file CSV
df.to_csv('ulasan_SIGNAL5K.csv', index=False)
```

Lampiran 3 Hasil *Preprocessing*

https://bit.ly/Preprocessing

Contoh Data:

No	*Content*	*Score*
1	aplikasi bantu proses nya cepat terima kasih	5
2	alhamdulillah kirim rumah cepat fc ktp stnk bpkb pajak masuk kas negara	5
3	enak luar daerah buat stnk	5
4	Bantu	5
5	mantap bayar pajak mudah	5
…	…	…
4997	bantu sistem pajak online maju polri	5
4998	Mudah	5
4999	terimakasih bantu	5
5000	mantap bayar pajak mudah	5
5001	daftar sulit sudah benar tidak jelas aplikasi	1

Lampiran 4 *Source Code* Analisis Sentimen Metode *Gaussian Naïve Bayes*

```
import pandas as pd
df=pd.read_csv(r'C:\Users\INTEL\Downloads\SKRIPSI\ulasan_SIGNAL5K.c
  sv')
data = df[['content', 'score']]
data
import pandas as pd
import matplotlib.pyplot as plt
```

```
import seaborn as sns
# Load data
df = pd.read_csv(r'C:\Users\INTEL\Downloads\SKRIPSI\ulasan_SIGNAL5K.csv')
# Tambahkan kolom Sentimen
df['Sentimen'] = df['score'].apply(lambda x: 'Positif' if x > 3 else ('Negatif' if x < 3 else 'Netral'))
# Statistik Deskriptif
print("Statistik Deskriptif Data:")
print(df.describe())
# Distribusi Sentimen
sentimen_counts = df['Sentimen'].value_counts()
print("\nDistribusi Sentimen:")
print(sentimen_counts)
# Visualisasi Distribusi Sentimen
plt.figure(figsize=(8, 6))
sns.countplot(x='Sentimen', data=df, palette='viridis')
plt.title("Distribusi Sentimen")
plt.xlabel("Kategori Sentimen")
plt.ylabel("Jumlah Ulasan")
plt.show()
# Panjang Teks Tiap Sentimen
df['Panjang Teks'] = df['content'].astype(str).apply(len)
plt.figure(figsize=(8, 6))
sns.boxplot(x='Sentimen', y='Panjang Teks', data=df, palette='viridis')
plt.title("Distribusi Panjang Teks Berdasarkan Sentimen")
plt.xlabel("Kategori Sentimen")
plt.ylabel("Panjang Teks")
plt.show()
# Frekuensi Kata yang Paling Sering Muncul (Word Frequency)
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer(stop_words='english', max_features=20)
X = vectorizer.fit_transform(df['content'].dropna())
word_counts = pd.DataFrame(X.toarray(), columns=vectorizer.get_feature_names_out())
# Visualisasi Word Frequency
word_freq = word_counts.sum().sort_values(ascending=False)
plt.figure(figsize=(10, 6))
word_freq.plot(kind='bar', color='skyblue')
plt.title("20 Kata yang Paling Sering Muncul")
plt.xlabel("Kata")
plt.ylabel("Frekuensi")
plt.xticks(rotation=45)
plt.show()
import numpy as np
from tqdm.notebook import tqdm
```

```
# Untuk mempermudah, simpan setiap objek agar dapat digunakan untuk
  pemodelan maupun deployment. Gunakan library Pickle
import pickle
import re
%matplotlib inline
!pip -q install sastrawi
import re
# Buat fungsi untuk langkah case folding
def casefolding(text):
  text = re.sub(r'\n', ' ', text)                  # Menghilangkan Enter
  text = text.lower()                             # Mengubah huruf menjadi huruf kecil
  text = text.replace(":", " ")                    # Hapus :
  text = re.sub(r'https?://\S+|www\.\S+', ' ', text) # Menghapus URL
  text = re.sub(r'[-+]?[0-9]+', ' ', text)          # Menghapus angka
  text = re.sub(r'[^\w\s*]',' ', text)              # Menghapus karakter tanda baca
  text = text.strip()                             # Menghapus whitespace di awal dan d
  akhir
  text = text.replace("https", " ")                    # Hapus :
  return text
data["casefolding"] = data["content"].apply(lambda x: casefolding(x))
data.head()
key_norm = pd.read_csv("key_norm.csv")
def text_normalize(text):
    text = ' '.join([key_norm[key_norm['singkat'] == word]['hasil'].values[0] if
  (key_norm['singkat'] == word).any() else word for word in text.split()])
    text = str.lower(text)
    return text
data['textnormalize'] = data['casefolding'].apply(text_normalize)
data.head()
# download Stopwords bahasa indonesia
import nltk
nltk.download('stopwords')
from nltk.tokenize import sent_tokenize, word_tokenize
from nltk.corpus import stopwords
stopwords_ind = stopwords.words('indonesian')
stopwords_ind
def remove_stop_words(text):
    clean_words = []
    text = text.split()
    for word in text:
        if word not in stopwords_ind:
            clean_words.append(word)
    return " ".join(clean_words)
data['stopwordremoval'] = data['textnormalize'].apply(remove_stop_words)
data.head()
from tqdm import tqdm
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
# Set up the stemming factory
```

```
factory = StemmerFactory()
stemmer = factory.create_stemmer()
# Buat fungsi untuk langkah stemming bahasa Indonesia
def stemming(text):
    return stemmer.stem(text)
# Tambahkan support untuk tqdm pada pandas
tqdm.pandas()
# Terapkan stemming dengan tqdm untuk melihat progress
data['stemming'] = data['stopwordremoval'].progress_apply(stemming)
# Lihat hasilnya
data.head()
data["clean_text"] = data['stemming']
data.drop(["casefolding","textnormalize","stopwordremoval","stemming"],axi
  s=1,inplace=True)
data.head()
data.to_excel('data_preprocessing.xlsx', index=False)
import pandas as pd
import warnings
warnings.filterwarnings("ignore")
data = pd.read_excel('data_preprocessing.xlsx')
data = data.dropna()
data
#hapus clean text
data = data[data["clean_text"]!=""]
data
#labeling
def labeling(x):
  if x == 1 or x == 2:
    return "negatif"
  elif x == 3:
    return "netral"
  elif x == 4 or x == 5:
    return "positif"
# Menerapkan fungsi labeling ke kolom 'rating' untuk membuat kolom baru
  'sentimen'
data['sentimen'] = data['score'].apply(labeling)
# Menampilkan dataframe yang sudah difilter dan kolom 'sentimen' yang baru
data
# Menyimpan data yang sudah dilabeling ke dalam file CSV
file_path = "hasil_labeling.csv"
data.to_csv(file_path, index=False)
# Konfirmasi file telah disimpan
print(f"File hasil labeling telah disimpan sebagai {file_path}.")
data['sentimen'].value_counts()
!pip -q install wordcloud
from wordcloud import WordCloud
import matplotlib.pyplot as plt
# Memisahkan data berdasarkan sentimen
```

```
data_positif = data[data['sentimen'] == 'positif']
data_negatif = data[data['sentimen'] == 'negatif']
data_netral = data[data['sentimen'] == 'netral']
# Menggabungkan teks untuk masing-masing sentimen
text_positif = ' '.join(data_positif["clean_text"].values.tolist())
text_negatif = ' '.join(data_negatif["clean_text"].values.tolist())
text_netral = ' '.join(data_netral["clean_text"].values.tolist())
# Membuat wordcloud untuk sentimen positif, negatif, dan netral
wordcloud_positif = WordCloud(width=800, height=800,
                        background_color='white',
                        stopwords=None,
                        min_font_size=10).generate(text_positif)
wordcloud_negatif = WordCloud(width=800, height=800,
                        background_color='white',
                        stopwords=None,
                        min_font_size=10).generate(text_negatif)
wordcloud_netral = WordCloud(width=800, height=800,
                       background_color='white',
                       stopwords=None,
                       min_font_size=10).generate(text_netral)
# Menampilkan wordcloud
plt.figure(figsize=(24, 8))
# Wordcloud untuk sentimen positif
plt.subplot(1, 3, 1)
plt.imshow(wordcloud_positif, interpolation='bilinear')
plt.title('Sentimen Positif')
plt.axis('off')
# Wordcloud untuk sentimen negatif
plt.subplot(1, 3, 2)
plt.imshow(wordcloud_negatif, interpolation='bilinear')
plt.title('Sentimen Negatif')
plt.axis('off')
# Wordcloud untuk sentimen netral
plt.subplot(1, 3, 3)
plt.imshow(wordcloud_netral, interpolation='bilinear')
plt.title('Sentimen Netral')
plt.axis('off')
plt.show()
#memisahkan menjadi 2 bagian yaitu 20% data uji 80% data latih
from sklearn.model_selection import train_test_split
X_raw = data["clean_text"]
y_raw = data["sentimen"]
X_train, X_test, y_train, y_test = train_test_split(X_raw.values, y_raw.values,
  test_size=0.2, random_state=42)
#tahap merubah data menjadi bilangan vektor
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer(ngram_range=(1,2))
vectorizer.fit(X_train)
```

```
X_train_TFIDF = vectorizer.transform(X_train).toarray()
X_test_TFIDF = vectorizer.transform(X_test).toarray()
X = vectorizer.transform(data["clean_text"]).toarray()
kolom = vectorizer.get_feature_names_out()
train_tf_idf = pd.DataFrame(X_train_TFIDF, columns=kolom)
test_tf_idf = pd.DataFrame(X_test_TFIDF, columns=kolom)
train_tf_idf.head()
# Simpan DataFrame hasil TF-IDF ke CSV
train_tf_idf.to_csv("train_tf_idf_output.csv", index=False)
print("Hasil TF-IDF berhasil disimpan ke file train_tf_idf_output.csv")
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
chi2_features = SelectKBest(chi2, k=1000)
X_kbest_features = chi2_features.fit_transform(train_tf_idf, y_train)
print('Banyaknya fitur awal:', train_tf_idf.shape[1])
print('banyaknya fitur setelah di seleksi:', X_kbest_features.shape[1])
# Pastikan hasil transformasi Chi-Square tersimpan dalam variabel
X_kbest_features = chi2_features.fit_transform(train_tf_idf, y_train)
# Import library yang dibutuhkan
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report, confusion_matrix,
  accuracy_score
# Split data menjadi data latih dan data uji
X_train, X_test, y_train, y_test = train_test_split(
    tfidf_mat_unigram,  # Gunakan matriks TF-IDF
    data['sentimen'],   # Target label
    test_size=0.2,      # Rasio data uji 20%
    random_state=42     # Seed untuk replikasi
)
# Buat instance GaussianNB
gnb = GaussianNB()
# Latih model menggunakan data latih
gnb.fit(X_train, y_train)
# Prediksi data uji
y_pred = gnb.predict(X_test)
# Evaluasi model
print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
print("\nAccuracy Score:")
print(f"{accuracy_score(y_test, y_pred) * 100:.2f}%")
from sklearn.model_selection import StratifiedKFold, cross_validate
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import make_scorer, accuracy_score, precision_score,
  recall_score, f1_score
import numpy as np
```

```
# Data yang digunakan
# Pastikan tfidf_mat_unigram dan data['sentimen'] sudah didefinisikan sebelumnya
X = tfidf_mat_unigram
y = data['sentimen']
# Buat instance GaussianNB
gnb = GaussianNB()
# Definisikan scoring metrics untuk cross-validation
scoring_metrics = {
    'accuracy': make_scorer(accuracy_score),
    'precision': make_scorer(precision_score, average='weighted'),
    'recall': make_scorer(recall_score, average='weighted'),
    'f1': make_scorer(f1_score, average='weighted')
}
# Buat objek StratifiedKFold
skf = StratifiedKFold(n_splits=5, shuffle=True, random_state=22)
# Lakukan cross-validation dengan StratifiedKFold
cv_results = cross_validate(
    gnb,
    X,
    y,
    cv=skf,
    scoring=scoring_metrics,
    return_train_score=True
)
# Tampilkan hasil untuk setiap fold
print("Cross-Validation Results per Fold:")
for i in range(5):  # Karena n_splits=5
    print(f'Fold {i + 1}:')
    print(f"  Accuracy: {cv_results['test_accuracy'][i] * 100:.2f}%")
    print(f"  Precision: {cv_results['test_precision'][i] * 100:.2f}%")
    print(f"  Recall: {cv_results['test_recall'][i] * 100:.2f}%")
    print(f"  F1-Score: {cv_results['test_f1'][i] * 100:.2f}%")
    print("-" * 30)
# Tampilkan rata-rata hasil cross-validation
print("\nAverage Cross-Validation Results:")
print(f"Mean Accuracy: {np.mean(cv_results['test_accuracy']) * 100:.2f}%")
print(f"Mean Precision: {np.mean(cv_results['test_precision']) * 100:.2f}%")
print(f"Mean Recall: {np.mean(cv_results['test_recall']) * 100:.2f}%")
print(f"Mean F1-Score: {np.mean(cv_results['test_f1']) * 100:.2f}%")
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
# Data dari cross-validation
acc = cv_results['test_accuracy']
prec = cv_results['test_precision']
rec = cv_results['test_recall']
f1score = cv_results['test_f1']
```

```
# Ubah data menjadi DataFrame untuk mempermudah plotting
data = {
    'Fold': ['Fold 1', 'Fold 2', 'Fold 3', 'Fold 4', 'Fold 5'],
    'Accuracy': acc,
    'Precision': prec,
    'Recall': rec,
    'F1-Score': f1score
}
df = pd.DataFrame(data)
# Konversi ke persentase (%)
df[['Accuracy', 'Precision', 'Recall', 'F1-Score']] *= 100
# Membuat figure dan subplot
fig, axs = plt.subplots(2, 2, figsize=(12, 10))
# Fungsi untuk menambahkan nilai persentase di atas setiap bar
def add_percentage_labels(ax, feature):
    for p in ax.patches:
        ax.annotate(
            f"{p.get_height():.2f}%",
            (p.get_x() + p.get_width() / 2., p.get_height()),
            ha='center', va='bottom', fontsize=10
        )
# Subplot 1: Accuracy
sns.barplot(
    x='Fold', y='Accuracy', data=df, ax=axs[0, 0], palette='viridis'
)
add_percentage_labels(axs[0, 0], 'Accuracy')
axs[0, 0].set_title('Accuracy per Fold')
axs[0, 0].set_ylabel('Percentage (%)')
# Subplot 2: Precision
sns.barplot(
    x='Fold', y='Precision', data=df, ax=axs[0, 1], palette='plasma'
)
add_percentage_labels(axs[0, 1], 'Precision')
axs[0, 1].set_title('Precision per Fold')
axs[0, 1].set_ylabel('Percentage (%)')
# Subplot 3: Recall
sns.barplot(
    x='Fold', y='Recall', data=df, ax=axs[1, 0], palette='inferno'
)
add_percentage_labels(axs[1, 0], 'Recall')
axs[1, 0].set_title('Recall per Fold')
axs[1, 0].set_ylabel('Percentage (%)')
# Subplot 4: F1-Score
sns.barplot(
    x='Fold', y='F1-Score', data=df, ax=axs[1, 1], palette='magma'
)
add_percentage_labels(axs[1, 1], 'F1-Score')
axs[1, 1].set_title('F1-Score per Fold')
```

```
axs[1, 1].set_ylabel('Percentage (%)')
# Atur tata letak agar lebih rapi
plt.tight_layout()
plt.show()
# Bagi data menjadi training dan testing
X_train, X_test, y_train, y_test = train_test_split(X_kbest_features, y_train, test_size=0.2, random_state=42, stratify=y_train)
# Terapkan SMOTE pada data training
smote = SMOTE(random_state=42)
X_train_balanced, y_train_balanced = smote.fit_resample(X_train, y_train)
# Periksa distribusi kelas setelah SMOTE
import pandas as pd
print("Distribusi kelas sebelum SMOTE:")
print(pd.Series(y_train).value_counts())
print("\nDistribusi kelas setelah SMOTE:")
print(pd.Series(y_train_balanced).value_counts())
# Simpan hasil SMOTE ke file (opsional)
X_train_balanced_df = pd.DataFrame(X_train_balanced)
y_train_balanced_df = pd.DataFrame(y_train_balanced, columns=["label"])
balanced_data = pd.concat([X_train_balanced_df, y_train_balanced_df], axis=1)
balanced_data.to_csv("balanced_data_after_chi_square.csv", index=False)
print("Data hasil SMOTE setelah Chi-Square berhasil disimpan ke balanced_data_after_chi_square.csv")
import matplotlib.pyplot as plt
import pandas as pd
# Data sebelum SMOTE
class_counts_before = pd.Series(y_train).value_counts()
# Data setelah SMOTE
class_counts_after = pd.Series(y_train_balanced).value_counts()
# Plot distribusi kelas setelah SMOTE
plt.subplot(1, 2, 2)
class_counts_after.plot(kind='bar', color='orange', alpha=0.8)
plt.title('Distribusi Kelas Setelah SMOTE')
plt.xlabel('Kelas')
plt.ylabel('Jumlah Sampel')
plt.xticks(rotation=0)
# Tampilkan grafik
plt.tight_layout()
plt.show()
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
# Pastikan data setelah SMOTE sudah dalam variabel X_train_balanced dan y_train_balanced
```

```
# Inisialisasi model Gaussian Naive Bayes
nb_model = GaussianNB()
# Latih model dengan data training yang sudah seimbang
nb_model.fit(X_train_balanced, y_train_balanced)
# Prediksi label untuk data testing
y_pred = nb_model.predict(X_test)
# Evaluasi hasil prediksi
print("\nLaporan Klasifikasi:")
print(classification_report(y_test, y_pred, target_names=['negatif', 'netral',
  'positif']))
# Hitung akurasi
accuracy = accuracy_score(y_test, y_pred)
print(f"Akurasi Model: {accuracy * 100:.2f}%")
# Buat confusion matrix
kolom = ['negatif', 'netral', 'positif']
confm = confusion_matrix(y_test, y_pred)
df_cm = pd.DataFrame(confm, index=kolom, columns=kolom)
# Visualisasi confusion matrix sebagai heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(df_cm, cmap='Blues', annot=True, fmt=".0f")
plt.title('Confusion Matrix')
plt.xlabel('Prediksi')
plt.ylabel('Sebenarnya')
plt.show()
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
# Pastikan data setelah SMOTE sudah dalam variabel X_train_balanced dan
  y_train_balanced
# dan data testing dalam X_test dan y_test
# Inisialisasi model Gaussian Naive Bayes
nb_model = GaussianNB()
# Latih model dengan data training yang sudah seimbang
nb_model.fit(X_train_balanced, y_train_balanced)
# Prediksi label untuk data testing
y_pred = nb_model.predict(X_test)
# Hitung akurasi
accuracy = accuracy_score(y_test, y_pred)
# Buat diagram batang tunggal
plt.figure(figsize=(4, 8))
plt.bar(['Naive Bayes'], [accuracy], color='green', alpha=0.8)
plt.ylim(0, 1)  # Skala 0-1 untuk akurasi
plt.title('Akurasi Model Naive Bayes Gaussian', fontsize=14)
plt.ylabel('Akurasi', fontsize=12)
from sklearn.model_selection import cross_validate
import numpy as np
# Definisikan metrik yang ingin dievaluasi
```

```
scoring_metrics = ['accuracy', 'precision_weighted', 'recall_weighted', 'f1_weighted']
# Lakukan cross-validation dengan evaluasi metrik lebih lanjut
cv_results = cross_validate(nb_model, X_train_balanced, y_train_balanced, cv=kfold, scoring=scoring_metrics)
# Menampilkan hasil metrik per lipatan (fold)
for i in range(5):
    print(f"\nLipatan {i+1}:")
    print(f"  Akurasi: {cv_results['test_accuracy'][i]:.2f}")
    print(f"  Presisi: {cv_results['test_precision_weighted'][i]:.2f}")
    print(f"  Recall: {cv_results['test_recall_weighted'][i]:.2f}")
    print(f"  F1-Score: {cv_results['test_f1_weighted'][i]:.2f}")
# Menampilkan rata-rata hasil dari semua lipatan
print("\nRata-rata hasil dari semua lipatan:")
print(f"  Akurasi Rata-rata: {np.mean(cv_results['test_accuracy'])*100:.2f}%")
print(f"  Presisi Rata-rata: {np.mean(cv_results['test_precision_weighted'])*100:.2f}%")
print(f"  Recall Rata-rata: {np.mean(cv_results['test_recall_weighted'])*100:.2f}%")
print(f"  F1-Score Rata-rata: {np.mean(cv_results['test_f1_weighted'])*100:.2f}%")
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
# Simulasi hasil cross-validation untuk 5 fold
# Misalnya: [fold1, fold2, fold3, fold4, fold5]
acc = cv_results['test_accuracy']
prec = cv_results['test_precision_weighted']
rec = cv_results['test_recall_weighted']
f1score = cv_results['test_f1_weighted']
# Convert hasil menjadi DataFrame untuk visualisasi
acc_df = pd.DataFrame(acc, columns=['value'])
prec_df = pd.DataFrame(prec, columns=['value'])
rec_df = pd.DataFrame(rec, columns=['value'])
f1score_df = pd.DataFrame(f1score, columns=['value'])
# Define the number of unique values (bars) for each plot
num_bars = 5
# Function to generate a color palette from a colormap
def get_colors_from_cmap(num_colors, cmap_name):
    cmap = plt.get_cmap(cmap_name)
    return [cmap(i / num_colors) for i in range(num_colors)]
# Generate color palettes from a colormap
colors_01 = get_colors_from_cmap(num_bars, 'viridis')
colors_02 = get_colors_from_cmap(num_bars, 'plasma')
colors_03 = get_colors_from_cmap(num_bars, 'inferno')
colors_04 = get_colors_from_cmap(num_bars, 'magma')
# X axis values (common for all subplots)
```

```
x = range(num_bars)
# Create a figure and a set of subplots (5 subplots for 5 fold results)
fig, axs = plt.subplots(3, 2, figsize=(15, 12))  # 3x2 grid to hold 5 subplots
# Subplot 1 - Accuracy
sns.barplot(x=x, y=acc_df['value'], ax=axs[0, 0], palette=colors_01)
axs[0, 0].set_title('Accuracy per Fold')
axs[0, 0].set_xticklabels([f'Fold {i+1}' for i in range(num_bars)])
# Menambahkan nilai di atas setiap bar
for i, v in enumerate(acc_df['value']):
    axs[0, 0].text(i, v + 0.01, f'{v:.2f}', ha='center')
# Subplot 2 - Precision
sns.barplot(x=x, y=prec_df['value'], ax=axs[0, 1], palette=colors_02)
axs[0, 1].set_title('Precision per Fold')
axs[0, 1].set_xticklabels([f'Fold {i+1}' for i in range(num_bars)])
# Menambahkan nilai di atas setiap bar
for i, v in enumerate(prec_df['value']):
    axs[0, 1].text(i, v + 0.01, f'{v:.2f}', ha='center')
# Subplot 3 - Recall
sns.barplot(x=x, y=rec_df['value'], ax=axs[1, 0], palette=colors_03)
axs[1, 0].set_title('Recall per Fold')
axs[1, 0].set_xticklabels([f'Fold {i+1}' for i in range(num_bars)])
# Menambahkan nilai di atas setiap bar
for i, v in enumerate(rec_df['value']):
    axs[1, 0].text(i, v + 0.01, f'{v:.2f}', ha='center')
# Subplot 4 - F1-Score
sns.barplot(x=x, y=f1score_df['value'], ax=axs[1, 1], palette=colors_04)
axs[1, 1].set_title('F1-Score per Fold')
axs[1, 1].set_xticklabels([f'Fold {i+1}' for i in range(num_bars)])
# Menambahkan nilai di atas setiap bar
for i, v in enumerate(f1score_df['value']):
    axs[1, 1].text(i, v + 0.01, f'{v:.2f}', ha='center')
# Remove the empty subplot in the 3rd row, 2nd column (axs[2, 1])
fig.delaxes(axs[2, 1])
# Adjust layout for better display
plt.tight_layout()
# Show the plot
plt.show()
# Tambahkan nilai akurasi di atas batang
for i, v in enumerate([accuracy]):
    plt.text(i, v + 0.02, f"{v:.2f}", ha='center', fontsize=12)
plt.show()
# Prediksi sentimen untuk data uji
y_pred = nb_model.predict(X_test)
# Menambahkan hasil prediksi sebagai kolom baru ke dalam DataFrame asli
data_test = pd.DataFrame(X_test, columns=["clean_text"])
data_test['sentimen_asli'] = y_test
data_test['sentimen_prediksi'] = y_pred
# Melihat 5 data teratas untuk verifikasi
```

```
data_test
```

Lampiran 5 *Source code* menentukan *p-value*

```
from scipy.stats import chi2

# Given chi-square values for each feature
chi_square_values = [0.291, 6.249, 0.102, 0.119, 1.246, 6.942, 0.423, 1.475, 0.810]

p_values = [chi2.sf(x, df=2) for x in chi_square_values]

p_values
```

Lampiran 6 Tabel *Chi-Square*

dk	Taraf Signifikansi					
	50%	30%	20%	10%	5%	1%
1	0.455	1.074	1.642	2.706	3.481	6.635
2	0.139	2.408	3.219	3.605	5.591	9.210
3	2.366	3.665	4.642	6.251	7.815	11.341
4	3.357	4.878	5.989	7.779	9.488	13.277
5	4.351	6.064	7.289	9.236	11.070	15.086
6	5.348	7.231	8.558	10.645	12.592	16.812
7	6.346	8.383	9.803	12.017	14.017	18.475
8	7.344	9.524	11.030	13.362	15.507	20.090
9	8.343	10.656	12.242	14.684	16.919	21.666
10	9.342	11.781	13.442	15.987	18.307	23.209
11	10.341	12.899	14.631	17.275	19.675	24.725
12	11.340	14.011	15.812	18.549	21.026	26.217
13	12.340	15.19	16.985	19.812	22.368	27.688
14	13.332	16.222	18.151	21.064	23.685	29.141
15	14.339	17.322	19.311	22.307	24.996	30.578
16	15.338	18.418	20.465	23.542	26.296	32.000
17	16.337	19.511	21.615	24.785	27.587	33.409
18	17.338	20.601	22.760	26.028	28.869	34.805
19	18.338	21.689	23.900	27.271	30.144	36.191
20	19.337	22.775	25.038	28.514	31.410	37.566
21	20.337	23.858	26.171	29.615	32.671	38.932
22	21.337	24.939	27.301	30.813	33.924	40.289
23	22.337	26.018	28.429	32.007	35.172	41.638
24	23.337	27.096	29.553	33.194	35.415	42.980
25	24.337	28.172	30.675	34.382	37.652	44.314
26	25.336	29.246	31.795	35.563	38.885	45.642
27	26.336	30.319	32.912	36.741	40.113	46.963

28	27.336	31.391	34.027	37.916	41.337	48.278
29	28.336	32.461	35.139	39.087	42.557	49.588
30	29.336	33.530	36.250	40.256	43.775	50.892

RIWAYAT HIDUP

Mira Permata Sari, lahir di Malang pada 18 Agustus 2002. Penulis merupakan putri kedua dari tiga bersaudara dari pasangan Bapak Sujono dan Ibu Sulik. Pendidikan formal penulis dimulai di TK Dharma Wanita Persatuan, yang diselesaikan pada tahun 2009. Selanjutnya, penulis melanjutkan pendidikan di SDN Karangwidoro 01 dan lulus pada tahun 2015. Pendidikan tingkat menengah pertama ditempuh di SMPN 13 Malang dan lulus pada tahun 2018. Penulis kemudian menyelesaikan pendidikan menengah atas di SMA Laboratorium UM Malang pada tahun 2021. Pada tahun yang sama, penulis melanjutkan pendidikan tinggi di Universitas Islam Negeri Maulana Malik Ibrahim Malang, dengan memilih Program Studi Matematika di Fakultas Sains dan Teknologi. Selama menempuh pendidikan di universitas, penulis aktif berpartisipasi dalam berbagai kegiatan akademik dan non-akademik. Penulis pernah menjadi anggota kepanitiaan berbagai acara kampus, yang memberikan pengalaman dalam mengelola dan menyukseskan kegiatan mahasiswa. Selain itu, penulis juga pernah menjadi anggota DEMA Fakultas Saintek pada departemen luar negeri selama 1 periode. Penulis juga pernah menjadi asisten praktikum, yang memperkuat kemampuan penulis dalam bidang pengajaran dan pendampingan akademik. Penulis juga aktif mengikuti berbagai seminar dan pelatihan, baik di dalam maupun di luar kampus, untuk meningkatkan wawasan dan keterampilan di bidang akademik dan profesional.

KEMENTERIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Gajayana No.50 Dinoyo Malang Telp. / Fax. (0341)558933

BUKTI KONSULTASI SKRIPSI

Nama	:	Mira Permata Sari
NIM	:	210601110105
Fakultas/Jurusan	:	Sains dan Teknologi / Matematika
Judul Skripsi	:	Implementasi Algoritma *Naïve Bayes* untuk Analisis Sentimen pada Ulasan Aplikasi Samsat Digital Online (SIGNAL)
Pembimbing I	:	Hisyam Fahmi, M.Kom
Pembimbing II	:	Ach. Nashichuddin, M.A.

No	Tanggal	Hal	Tanda Tangan
1.	1 Oktober 2024	Konsultasi Topik dan Data	1.
2.	7 Oktober 2024	Konsultasi Bab I, II, dan III	2.
3.	8 Oktober 2024	ACC Bab I, II, dan III	3.
4.	9 Oktober 2024	Konsultasi Kajian Agama Bab I dan II	4.
5.	10 Oktober 2024	ACC Kajian Agama Bab I dan II	5.
6.	30 Oktober 2024	Konsultasi Revisi Seminar Proposal	6.
7.	22 November 2024	Konsultasi Bab IV dan V	7.
8.	28 November 2024	Konsultasi Bab IV dan V	8.
9.	11 Desember 2024	Konsultasi Bab IV dan V	9.
10.	13 Desember 2024	Konsultasi Bab IV dan V	10.
11.	17 Januari 2025	ACC Bab IV dan V	11.
12.	18 Januari 2025	Konsultasi Kajian Agama Bab IV	12.
13.	19 Januari 2025	ACC Kajian Agama Bab IV	13.
14.	20 Januari 2025	ACC Seminar Hasil	14.
15.	5 Maret 2025	Konsultasi Revisi Seminar Hasil	15.

KEMENTERIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Gajayana No.50 Dinoyo Malang Telp. / Fax. (0341)558933

16.	18 Maret 2025	Konsultasi Revisi Seminar Hasil	16.
17.	21 April 2025	ACC Sidang Skripsi	17.
18.	19 Mei 2025	ACC Keseluruhan	18.

Malang, 19 Mei 2025
Mengetahui,
Ketua Program Studi Matematika

Dr. Elly Susanti, M.Sc.
NIP. 19741129 200012 2 005