

Klasifikasi Ticket Service Desk Perusahaan Asuransi Jiwa Berbasis Machine Learning

Joash Lorenzo Imbenay¹, Indra Budi²

joash.lorenzo@ui.ac.id¹, indra@cs.ui.ac.id²

^{1,2} Universitas Indonesia

Informasi Artikel

Diterima : 16 Jun 2024
Direview : 25 Jun 2024
Disetujui : 25 Jul 2024)

Kata Kunci

Service Desk, *text mining*, Recall, fitur kontekstual, balancing data.

Abstrak

Studi ini berfokus pada pengembangan model klasifikasi tiket untuk Service Desk di perusahaan asuransi guna meningkatkan efisiensi operasional. Klasifikasi tiket manual sering memakan waktu dan rentan terhadap kesalahan, sehingga penelitian ini bertujuan membandingkan kinerja berbagai algoritma klasifikasi untuk menentukan model terbaik. Metodologi melibatkan teknik penambangan teks dan pembelajaran mesin dengan empat algoritma utama: *Random Forest*, Decision Tree, Support Vector Machine (SVM), dan *Naïve Bayes*. Data berasal dari tiket Service Desk yang telah diproses melalui tahap prapemrosesan teks. Hasil penelitian menunjukkan bahwa SVM dengan fitur TF-Unibitri dan data kontekstual adalah yang terbaik untuk klasifikasi tiket Non-IT Support (Precision 97,80%, Recall 97,82%, F-Score 97,80%, Akurasi 97,81%). Untuk IT Support, SVM dengan TF-Unibi dan email memiliki performa tertinggi (Precision 97,86%, Recall 97,87%, F-Score 97,86%, Akurasi 97,86%). Implementasi model ini diharapkan meningkatkan efisiensi operasional dan kepuasan pengguna terhadap layanan IT, mempercepat penanganan tiket, mengurangi beban administratif, dan meningkatkan kepuasan pengguna.

Keywords

Service Desk, *text mining*, Recall, contextual features, data balancing.

Abstract

This study focuses on developing a ticket classification model for the Service Desk at an insurance company to enhance operational efficiency. Manual ticket classification is often time-consuming and prone to errors, so this research aims to compare the performance of various classification algorithms to determine the best model. The methodology involves text mining and machine learning techniques using four main algorithms: Random Forest, Decision Tree, Support Vector Machine (SVM), and Naïve Bayes. The data comes from Service Desk tickets processed through text preprocessing stages. Findings indicate that SVM with TF-Unibitri features and contextual data is the best for classifying Non-IT Support tickets (Precision 97.80%, Recall 97.82%, F-Score 97.80%, Accuracy 97.81%). For IT Support, SVM with TF-Unibi and email features performed the best (Precision 97.86%, Recall 97.87%, F-Score 97.86%, Accuracy 97.86%). Implementing this model is expected to improve operational efficiency and user satisfaction with IT services by speeding up ticket handling, reducing administrative workload, and enhancing user satisfaction.

A. Pendahuluan

Perusahaan asuransi telah mengembangkan sistem layanan *Service Desk* untuk memfasilitasi permintaan dukungan dan pembaruan dari semua pengguna kepada Departemen IT. Fitur utama dari sistem ini adalah sistem ticketing yang mencatat setiap permintaan pengguna. Admin *Service Desk* bertugas mengklasifikasikan jenis permintaan dan menentukan pihak yang akan menanganinya. *Service Desk* berfungsi sebagai titik kontak tunggal bagi karyawan untuk mendapatkan layanan IT, memainkan peran vital dalam memberikan dukungan TI yang komprehensif. Menurut Gupta (2020: 12), penerapan sistem seperti *Service Desk* dalam organisasi besar dapat meningkatkan efisiensi dan efektivitas dalam menangani masalah teknologi dan permintaan layanan.

Penggunaan sistem ticketing memastikan setiap permintaan tercatat sistematis dan ditangani sesuai prioritas dan spesifikasi. Hal ini membantu dalam memastikan respons cepat dan akurat terhadap masalah TI yang dihadapi karyawan, serta memfasilitasi pelacakan dan analisis tren permintaan layanan untuk peningkatan layanan IT yang berkelanjutan. Namun, tantangan tetap ada karena tidak semua karyawan menggunakan sistem ini secara efektif. Hingga Desember 2022, tercatat sekitar 13.000 tiket telah dibuat, menunjukkan masih adanya hambatan dalam partisipasi karyawan. Data menunjukkan fluktuasi jumlah tiket yang masuk setiap bulan, dengan puncak tertinggi pada bulan Desember sebanyak 986 tiket. Analisis lebih lanjut diperlukan untuk memahami penyebab fluktuasi ini dan mengidentifikasi strategi untuk mengelola beban kerja lebih efisien.

Di perusahaan swasta, sistem *Service Desk* digunakan untuk mencatat permintaan dukungan IT dari karyawan. Namun, penggunaannya belum optimal, dengan banyak keluhan tentang keterlambatan pembuatan tiket oleh admin. Setiap bulan, jumlah tiket yang diterbitkan cukup tinggi dan fluktuatif, mencapai puncaknya pada bulan tertentu dengan volume tiket yang signifikan. Beban kerja yang tinggi pada satu admin menyebabkan penundaan respons dan penyelesaian tiket. Ketergantungan pada satu admin juga meningkatkan risiko keterlambatan saat admin cuti. Proses klasifikasi manual menambah kompleksitas dan kesalahan. Untuk meningkatkan efisiensi, perusahaan perlu mempertimbangkan penambahan staf, otomatisasi klasifikasi, dan pelatihan admin. Implementasi model klasifikasi otomatis akan mempercepat identifikasi dan prioritas tiket, mengurangi beban kerja admin, dan meningkatkan kepuasan pengguna dengan layanan IT yang lebih responsif dan terorganisir.

Penelitian ini menjelaskan pengaruh penerapan *spelling correction* menggunakan Symspell pada *incident categorization*. Shafiera (2022) menunjukkan bahwa typo pada data dapat menurunkan akurasi model, sehingga implementasi *spelling correction* diperlukan. Symspell adalah teknik *spelling correction* yang menggunakan *deletion* dan *edits distance*. Penerapan Symspell pada *preprocessing* bertujuan untuk melihat pengaruhnya terhadap nilai *Accuracy*, *Precision*, dan *Recall* model klasifikasi menggunakan SVM. Hasil penelitian menunjukkan bahwa kombinasi penerapan Symspell dan penggunaan data gabungan *subject* dan *description* tiket meningkatkan performa model dengan nilai *Accuracy* 83%, *Precision* 83.7%, dan *Recall* 83.8%, dibandingkan dengan *preprocessing* tanpa Symspell (*Accuracy* 81.9%, *Precision* 82.5%, *Recall* 83.3%).

Performa Symspell sendiri menunjukkan nilai *correction* hit sebesar 82% dan *False Positive* 0,61%.

Penelitian lainnya oleh Saputra et al. (2018) mengevaluasi penerapan *text mining* untuk *assignment complaint handling customer* terhadap divisi terkait menggunakan metode *Decision Tree* algoritma C4.5 pada PT. XL Axiata, Tbk. Kesalahan *assignment* pada keluhan pelanggan menyebabkan keluhan tidak tertangani secara maksimal, yang dapat menimbulkan dampak negatif bagi perusahaan. Penelitian ini bertujuan untuk mengoptimalkan penanganan keluhan pelanggan dengan *text mining* dan algoritma C4.5. Hasil penelitian ini divalidasi menggunakan kurva ROC untuk mengukur akurasi *Decision Tree* dalam klasifikasi pengaduan. Dengan sistem pendukung keputusan ini, pengaduan pelanggan dan kinerja pelayanan di PT. XL Axiata menjadi lebih mudah dan cepat diatasi.

Lokhacheva et al. (2020) mendesain sistem informasi untuk analisis semantik dan klasifikasi isu pada sistem *Service Desk*. Penelitian ini menggunakan beberapa model matematika dan metode analisis teks untuk klasifikasi. Sistem ini membantu dalam pengelolaan isu dengan lebih efektif melalui analisis penggunaan sistem, konstruksi skema sistem, dan class diagram. Implementasi ini bertujuan untuk meningkatkan efisiensi dalam penanganan tiket dan memberikan solusi yang lebih cepat dan akurat.

Paramesh dan Shreedhara (2018) meneliti sistem *Service Desk* otomatis menggunakan teknik *machine learning*. Pengelolaan tiket pada *Service Desk* dan helpdesk adalah masalah utama di industri layanan TI, terutama dalam perutean tiket ke tim yang tepat. Penelitian ini mengusulkan penggunaan *text mining* deskripsi tiket dan label history untuk membangun model klasifikasi yang dapat mengklasifikasikan tiket baru secara otomatis. Algoritma yang digunakan termasuk *Multinomial Naïve Bayes*, *Logistic Regression*, *K-Nearest Neighbor*, dan *Support Vector Machines*. Hasil penelitian menunjukkan bahwa *classifier* berbasis *Support Vector Machines* (SVM) bekerja dengan baik pada semua sampel data, meningkatkan produktivitas, pengalaman pengguna, dan mengurangi waktu resolusi tiket.

Penelitian lain oleh S. P. Paramesh et al. (2018) membahas klasifikasi tiket IT *Service Desk* menggunakan *ensemble of classifiers*. *Manual classification* dapat menghasilkan perutean tiket yang salah, menyebabkan penugasan ulang, penggunaan sumber daya yang tidak perlu, dan penundaan dalam penyelesaian tiket. Penelitian ini menggunakan teknik ansambel seperti *Bagging*, *Boosting*, dan *Voting* untuk meningkatkan akurasi sistem klasifikasi tiket dibandingkan dengan klasifikasi dasar individu. Kinerja sistem ansambel diuji menggunakan metrik kinerja yang berbeda, menunjukkan bahwa klasifikasi ansambel lebih unggul dibandingkan dengan model dasar. Keuntungan dari sistem ini termasuk UI yang disederhanakan, waktu penyelesaian yang lebih cepat, peningkatan produktivitas, kepuasan pelanggan, dan pertumbuhan bisnis. Data *Service Desk* dari infrastruktur TI perusahaan besar digunakan untuk penelitian ini, menunjukkan aplikasi praktis dari model ansambel dalam meningkatkan pengelolaan tiket.

Penelitian-penelitian ini menunjukkan bahwa penerapan teknik *text mining*, *machine learning*, dan *spelling correction* dapat meningkatkan efisiensi dan akurasi dalam pengelolaan tiket pada *Service Desk*. Implementasi Symspell dalam *preprocessing* meningkatkan performa model klasifikasi dengan mengurangi

kesalahan akibat typo. Penggunaan algoritma C4.5, SVM, dan teknik ansambel dalam klasifikasi tiket menunjukkan peningkatan kinerja dalam penanganan keluhan pelanggan dan pengelolaan tiket IT. Dengan mengotomatisasi proses ini, perusahaan dapat mengurangi waktu respons, meningkatkan kepuasan pengguna, dan meningkatkan efisiensi operasional secara keseluruhan.

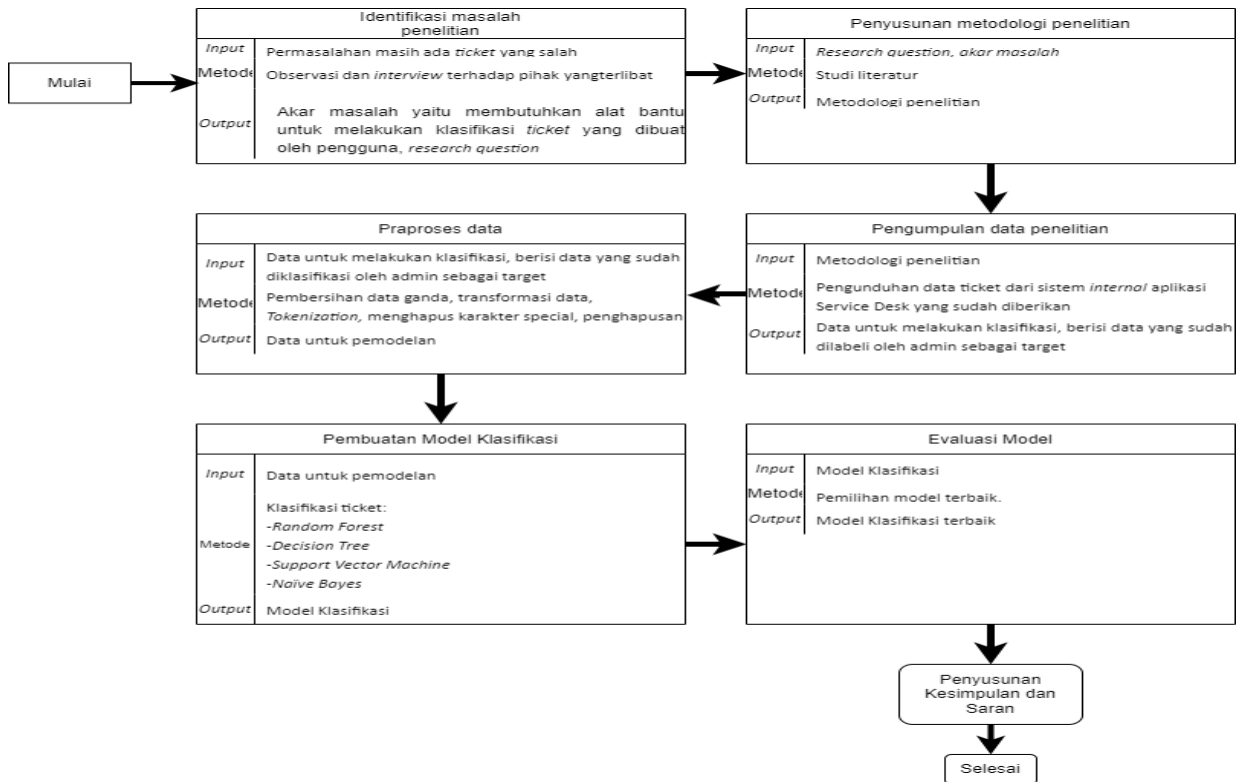
Dari uraian diatas, maka penelitian ini mencoba mengimplementasikan *machine learning*, dengan berfokus pada penentuan model klasifikasi terbaik. Ruang lingkup penelitian ini meliputi data *Service Desk* dari portal perusahaan swasta, berupa teks tahun 2022 dalam Bahasa Indonesia. Informasi yang diekstrak mencakup jenis permintaan, tingkat prioritas, dan personel IT yang ditugaskan, dengan klasifikasi antara tiket *Non-IT Support* dan *IT Support*.

Tujuan penulisan adalah membuat model klasifikasi permintaan user pada *Service Desk*, memungkinkan admin secara otomatis mengklasifikasikan setiap permintaan, sehingga admin dapat fokus pada tugas lainnya. Manfaat penulisan untuk perusahaan adalah meningkatkan efisiensi operasional dengan otomatisasi klasifikasi, mengurangi waktu tunggu pengguna, dan meningkatkan kepuasan layanan. Sistem otomatis juga mengurangi kesalahan manusia dalam klasifikasi. Secara akademis, penelitian ini memberikan wawasan tentang aplikasi *text mining* dan *machine learning*, mendukung teori dengan bukti empiris, dan membuka jalan untuk studi lebih lanjut dalam evaluasi model klasifikasi, mendorong inovasi dalam pengembangan alat-alat pembelajaran mesin.

Berdasarkan rumusan masalah yang telah dijabarkan tersebut maka dapat dirumuskan sebuah pertanyaan penelitian yaitu Bagaimana perbandingan kinerja model klasifikasi algoritma yang berbeda dalam menangani permintaan user pada portal *Service Desk* Perusahaan Asuransi untuk menentukan algoritma dengan kinerja terbaik?

B. Metode Penelitian

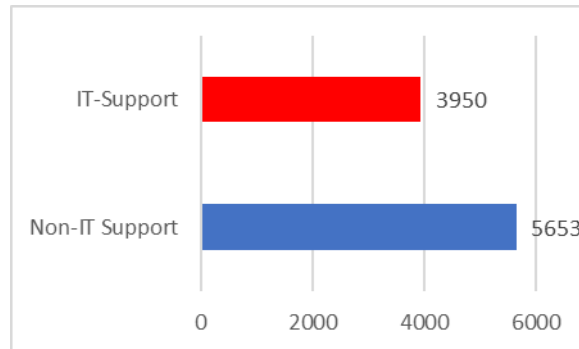
Tahapan Penelitian



Gambar 1. Tahapan Penelitian

Penelitian ini mengidentifikasi masalah pada aplikasi *Service Desk* perusahaan swasta melalui analisis langsung dan wawancara dengan staf, menunjukkan kebutuhan akan solusi otomatis untuk mengatasi keterlambatan dan kesalahan dalam pengelolaan tiket. Studi literatur mengeksplorasi topik seperti *data mining*, *text mining*, dan algoritme klasifikasi (*Random Forest*, *Decision Tree*, *Naïve Bayes*, dan *SVM*), yang kemudian membentuk kerangka teoretis penelitian. Metodologi penelitian dikembangkan untuk mencakup pemilihan algoritme, pengumpulan data, dan langkah-langkah *preprocessing* data. Data diambil dari database *Service Desk*, mencakup tiket layanan dari 2020 hingga 2021, kemudian diproses melalui *tokenization*, penghapusan *stopwords*, dan ekstraksi fitur. Empat algoritme (*SVM*, *Naïve Bayes*, *Decision Tree*, dan *Random Forest*) digunakan untuk mengembangkan model klasifikasi tiket. Model dibandingkan berdasarkan akurasi, presisi, *Recall*, dan *F1-Score* untuk menentukan performa terbaik. Kesimpulan diambil dari hasil penelitian, dengan saran untuk penelitian masa depan guna meningkatkan metodologi dan cakupan analisis. Penelitian ini diharapkan dapat memberikan solusi efisien untuk mengelola tiket layanan, meningkatkan responsivitas, dan kepuasan pengguna.

Dataset



Gambar 2. Grafik Jumlah Data Kelompok *IT Support* dan *Non-IT Support*

Penelitian ini menggunakan dataset dari *Service Desk* perusahaan asuransi swasta, terdiri dari 9.603 tiket: *IT Support* (3.950 tiket) dan *Non-IT Support* (5.653 tiket). **Gambar 2** menunjukkan ketidakseimbangan data ini. Untuk mengatasi ketidakseimbangan, digunakan metode *undersampling* pada kelas *Non-IT Support*, menghapus data secara acak hingga jumlahnya seimbang dengan *IT Support*. Tujuan *undersampling* adalah mengurangi bias model dan meningkatkan akurasi prediksi. Penelitian menekankan metrik *Recall* untuk memastikan semua permintaan *IT Support* terprediksi dengan benar, meskipun ada risiko beberapa *Non-IT Support* terprediksi sebagai *IT Support*, guna memastikan tidak ada permintaan *IT Support* yang terlewatkan.

Pre-processing

Penelitian ini menjelaskan proses pengolahan data menggunakan *library pandas* dan *NLTK* untuk meningkatkan kualitas model klasifikasi tiket pada *Service Desk* SMiLe. Langkah pertama adalah pemilihan fitur, di mana data pada kolom 'Label' diubah menjadi deskripsi yang lebih jelas dengan mengganti nilai '1' menjadi '*Non-IT Support*' dan '2' menjadi '*IT Support*'. Analisis frekuensi kemunculan label menggunakan metode *value_counts()* menunjukkan distribusi tiket yang kemudian divisualisasikan menggunakan *bar chart horizontal* untuk memperjelas perbedaan antara dua kategori tiket tersebut (Bishop, 2006: 67-68).

Langkah berikutnya adalah transformasi data menjadi huruf kecil menggunakan fungsi *str.lower()* dari *pandas*, yang bertujuan untuk menyeragamkan teks dalam kolom '*Text*'. Ini penting karena perbedaan kapitalisasi dapat mempengaruhi hasil analisis teks (Manning et al., 2008: 12). Setelah itu, dilakukan tokenisasi teks menggunakan *library NLTK*. Proses ini memecah teks menjadi unit-unit kecil yang disebut token untuk analisis lebih lanjut. Tokenisasi membantu memberikan struktur yang lebih baik pada data teks, memudahkan analisis seperti *stemming* dan *lemmatization* (Bird et al., 2009: 56).

Proses pembersihan data teks dengan menghapus karakter spesial menggunakan *Python* dan *NLTK* adalah langkah berikutnya. Menghapus karakter spesial seperti tanda baca, simbol, dan URL membantu mengurangi noise dalam data, meningkatkan akurasi analisis teks (Jurafsky & Martin, 2009: 25). Penghapusan *stopwords* dilakukan dengan menggunakan daftar *stopwords* bahasa Indonesia dari *NLTK* dan daftar tambahan dari file teks. Menghapus *stopwords*

membantu meningkatkan kualitas dan akurasi model dengan fokus pada kata-kata yang lebih bermakna dan relevan (Ramasubramanian & Ramya, 2013).

Langkah normalisasi data dilakukan untuk mengubah kata-kata tidak baku atau slang menjadi bentuk baku. Normalisasi ini memastikan konsistensi dan meningkatkan akurasi analisis teks (Han et al., 2011). Terakhir, proses *stemming* bertujuan untuk mengubah kata berimbuhan menjadi bentuk dasar menggunakan *library* Sastrawi. *Stemming* membantu menyederhanakan teks dan meningkatkan akurasi serta efisiensi dalam analisis teks (Adriani et al., 2007; Abdul Jabbar et al., 2023).

Ekstraksi Fitur

Penelitian ini menggunakan *library pandas* dan *scikit-learn* untuk pemrosesan teks dan ekstraksi fitur. Langkah pertama adalah mengonversi daftar token yang telah dinormalisasi kembali menjadi string menggunakan fungsi `join` pada kolom `text_normalized` dalam *DataFrame df*. Selanjutnya, teknik vektorisasi seperti *CountVectorizer* dan *TfidfVectorizer* digunakan untuk mengubah teks menjadi representasi vektor. *CountVectorizer* menghitung frekuensi kata (*Term Frequency*), sedangkan *TfidfVectorizer* memberi bobot berdasarkan frekuensi terbalik dokumen (TF-IDF), yang lebih efektif dalam menganalisis teks. Variasi vektorisasi dilakukan menggunakan unigram, bigram, dan trigram untuk mengeksplorasi berbagai representasi fitur teks (Pedregosa et al., 2011).

Selain itu, penelitian ini juga menggunakan *regular expression* (Regex) untuk mengidentifikasi kata kunci tertentu dalam teks. Regex membantu mendeteksi dan memfilter postingan dengan kata kunci terkait, seperti "spaj," "data," dan "polis" untuk Non-IT, serta "email," "akses," dan "Service Desk" untuk IT. Implementasi ini meningkatkan akurasi dalam klasifikasi tiket, memungkinkan sistem untuk mengelompokkan tiket secara otomatis ke dalam kategori yang tepat, sehingga memperbaiki efisiensi dan efektivitas penanganan tiket (Rahman et al., 2020; Purbaya et al., 2023). Metode ini, bertujuan untuk meningkatkan kualitas model klasifikasi tiket *Service Desk*, memastikan teks yang diproses lebih akurat dan informatif.

Sampling

Penelitian ini menghadapi masalah ketidakseimbangan dataset dan menggunakan metode *undersampling* untuk mengatasinya. Dengan melakukan penghapusan data secara acak dari kelas terbesar, teknik ini bertujuan menyeimbangkan jumlah data antara kelas yang berbeda, sehingga meningkatkan akurasi model. Meskipun *undersampling* dapat mengurangi bias yang disebabkan oleh ketidakseimbangan data, teknik ini juga memiliki risiko menghilangkan informasi penting dari kelas mayoritas (Rahman et al., 2020; Purbaya et al., 2023; Islam et al., 2021). Penelitian ini mengevaluasi efek keseimbangan data terhadap hasil analisis untuk meningkatkan akurasi dan keandalan model.

Pemodelan Klasifikasi Data

Penelitian ini menggunakan algoritme *Naïve Bayes*, Support Vector Machine (SVM), *Decision Tree*, dan *Random Forest* dari *library sklearn* untuk mengevaluasi kinerja teknik klasifikasi teks. *Naïve Bayes* digunakan karena kesederhanaannya dalam menangani teks, SVM untuk menemukan hyperplane optimal, *Decision Tree*

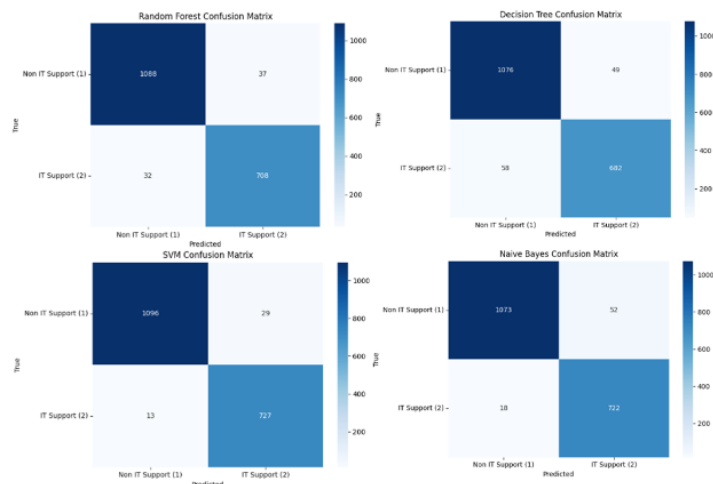
untuk interpretabilitas dengan visualisasi keputusan, dan *Random Forest* untuk meningkatkan akurasi melalui *ensemble method*. Setiap algoritme diterapkan pada data yang telah di-vektorisasi, dilatih, dan diuji. Evaluasi kinerja dilakukan menggunakan metrik akurasi, presisi, *Recall*, dan *F1-Score*. Hasilnya menunjukkan kombinasi parameter yang tepat dapat meningkatkan efisiensi dan akurasi model, memastikan klasifikasi teks yang lebih andal.

C. Hasil dan Pembahasan

Penelitian ini mengevaluasi empat algoritme utama: *Random Forest*, *Decision Tree*, SVM, dan *Naïve Bayes* melalui tiga skenario analisis. Skenario pertama menguji algoritme menggunakan *Term Frequency* (TF) dan *Term Frequency-Inverse Document Frequency* (TF-IDF) dengan kombinasi unigram, bigram, dan trigram, membandingkan hasil berdasarkan nilai *Recall* untuk menemukan konfigurasi terbaik (Rahman et al., 2020; Purbaya et al., 2023).

Tabel 1. Hasil dari analisis ekstraksi fitur

Kombinasi Algoritme dan Fitur	Precision	Recall	F-Score	Accuracy
<i>Random Forest</i> - TF-IDF Unigram	96.08%	96.19%	96.14%	96.30%
<i>Decision Tree</i> TF-IDF Unigri	94.09%	93.90%	93.99%	94.26%
SVM TF-Unibitri	97.49%	97.83%	97.65%	97.74%
<i>Naïve Bayes</i> TF-Unibitri	95.81%	96.47%	96.10%	96.24%



Gambar 3. Hasil *Confusion matrix* TF dan TF-IDF

Tabel 1 dan **gambar 3** di atas memberikan wawasan mendalam tentang kinerja empat model klasifikasi teks: *Random Forest*, *Decision Tree*, Support Vector Machine (SVM), dan *Naïve Bayes* dalam skenario pertama. Model SVM dengan kombinasi fitur TF-Unibitri menunjukkan performa terbaik dengan *Precision* 97.49%, *Recall* 97.83%, *F-Score* 97.65%, dan Akurasi 97.74%, menunjukkan keakuratannya dalam prediksi dan konsistensinya dalam mengidentifikasi kelas yang benar. *Confusion matrix* menunjukkan bahwa SVM memiliki jumlah prediksi benar tertinggi untuk kedua kelas, menandakan keandalan dan akurasi yang

superior dalam menangani teks. *Random Forest* dan *Naïve Bayes* juga menunjukkan hasil yang baik, namun dengan beberapa kesalahan klasifikasi terutama dalam kelas *IT Support*. *Decision Tree*, meskipun memiliki interpretabilitas tinggi, menunjukkan akurasi yang lebih rendah dibandingkan model lainnya. Keseluruhan analisis menegaskan bahwa SVM dengan kombinasi fitur *TF-Unibitri* adalah pilihan optimal untuk klasifikasi teks, diikuti oleh *Random Forest* dan *Naïve Bayes*. Pendekatan ini menekankan pentingnya penggunaan kombinasi n-gram dan teknik ekstraksi fitur yang tepat untuk meningkatkan kinerja model secara signifikan.

Skenario kedua menggunakan kombinasi terbaik dari skenario pertama dan menambahkan fitur kontekstual menggunakan regex untuk mendeteksi kata-kata khusus, meningkatkan kemampuan model dalam mendeteksi pola dalam teks (Yang et al., 2020).

Tabel 2. Hasil Metrik Kombinasi Fitur Kontekstual *Non-IT Support*

Kombinasi Algoritme dan Fitur	Precision	Recall	F-Score	Accuracy
<i>Random Forest</i> - TF-IDF Unigram - MS	96.78%	96.79%	96.78%	96.79%
<i>Decision Tree</i> TF-IDF Unigram - MP	94.37%	94.36%	94.37%	94.37%
SVM TF - Unibitri - Data	97.80%	97.82%	97.80%	97.81%
<i>Naïve Bayes</i> TF-Unibitri - MS	96.25%	96.34%	96.25%	96.26%



Gambar 4. Hasil *Confusion matrix* Data untuk Label *Non-IT Support*

Tabel 2 dan **gambar 4** di atas menganalisis kinerja model klasifikasi teks dengan fitur kontekstual untuk label *Non-IT Support*. Gambar *Confusion matrix* menunjukkan bahwa model SVM dengan fitur TF-Unibitri memiliki prediksi yang sangat akurat, dengan 1097 *true positives* untuk *Non-IT Support* dan hanya 28 *False Positives*. Untuk *IT Support*, terdapat 727 *true positives* dan 13 *false negatives*, menunjukkan keandalan tinggi dalam prediksi kedua kelas.

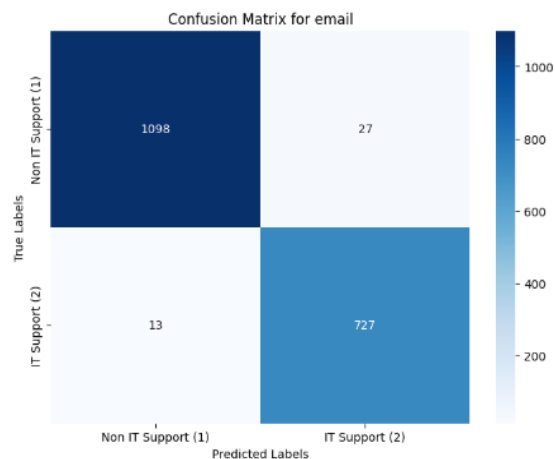
Tabel kinerja model menunjukkan bahwa SVM dengan kombinasi fitur TF-Unibitri dan data kontekstual memiliki performa terbaik dengan *Precision* 97.80%,

Recall 97.82%, *F-Score* 97.80%, dan Akurasi 97.81%. *Random Forest* dengan fitur TF-IDF Unigram dan MS juga menunjukkan performa tinggi dengan *Precision*, *Recall*, *F-Score*, dan Akurasi semuanya 96.79%. *Decision Tree* dengan fitur TF-IDF Unibitri dan MP, serta *Naïve Bayes* dengan TF-Unibitri dan MS, menunjukkan performa yang baik, meskipun sedikit di bawah SVM dan *Random Forest*.

Kesimpulannya, model SVM dengan fitur kontekstual TF-Unibitri menunjukkan kinerja superior dalam klasifikasi teks *Non-IT Support*, diikuti oleh *Random Forest*. Pendekatan ini menekankan pentingnya fitur kontekstual dalam meningkatkan akurasi dan keandalan model klasifikasi teks.

Tabel 3. Hasil Metrik Kombinasi Fitur Kontekstual *IT Support*

Kombinasi Algoritme dan Fitur	Precision	Recall	F-Score	Accuracy
<i>Random Forest</i> - TF-IDF Unigram - Akses	96.84%	96.85%	96.84%	96.84%
<i>Decision Tree</i> TF-IDF Unibi Email	94.42%	94.42%	94.42%	94.42%
SVM TF-Unibi - Email	97.86%	97.87%	97.86%	97.86%
<i>Naïve Bayes</i> TF-Unibitri - MSP	96.25%	96.34%	96.25%	96.26%



Gambar 5. Hasil *Confusion matrix* Data untuk Label *IT Support*

Tabel 3 dan **gambar 5** di atas menunjukkan kinerja model klasifikasi teks dengan fitur kontekstual untuk label *IT Support*. *Confusion matrix* menggambarkan bahwa model SVM dengan fitur TF-Unibi dan email menunjukkan performa yang sangat baik, dengan 1098 *true positives* untuk *Non-IT Support* dan hanya 27 *False Positives*. Untuk *IT Support*, terdapat 727 *true positives* dan hanya 13 *false negatives*, yang menunjukkan akurasi tinggi dalam prediksi.

Tabel kinerja model menunjukkan bahwa SVM dengan kombinasi fitur TF-Unibi dan email memiliki performa terbaik dengan *Precision* 97.86%, *Recall* 97.87%, *F-Score* 97.86%, dan Akurasi 97.86%. *Random Forest* dengan fitur TF-IDF Unigram dan akses juga menunjukkan hasil yang baik dengan *Precision*, *Recall*, *F-Score*, dan Akurasi semuanya 96.84%. *Decision Tree* dengan fitur TF-IDF Unibi dan

email serta *Naïve Bayes* dengan TF-Unibitri dan MSP menunjukkan performa yang cukup baik, meskipun lebih rendah dibandingkan SVM dan *Random Forest*.

Kesimpulannya, model SVM dengan fitur kontekstual TF-Unibi dan email menunjukkan kinerja terbaik dalam klasifikasi teks *IT Support*, diikuti oleh *Random Forest*. Pendekatan ini menekankan pentingnya penggunaan fitur kontekstual untuk meningkatkan akurasi dan keandalan model klasifikasi teks.

Formulasi metrik kombinasi fitur kontekstual dan TF-IDF dilakukan dengan membandingkan hasil analisis *balance* data terhadap fitur kontekstual. Proses ini melibatkan pengukuran perbedaan antara fitur kontekstual dengan hasil analisis TF-IDF. Dengan menghitung selisih antara fitur kontekstual dan nilai TF-IDF, kita dapat mengevaluasi sejauh mana kedua metrik ini berkontribusi terhadap hasil analisis *balance* data. Pendekatan ini membantu mengidentifikasi ketidaksesuaian atau kesenjangan antara data yang telah di-*balance* dengan konteks sebenarnya, sehingga menghasilkan model yang lebih akurat dan relevan dengan kondisi dan variabel kontekstual yang ada.

Tabel 4. Perbandingan Metrik Kombinasi Fitur Kontekstual dan TF-IDF *Non-IT Support*

Kombinasi Algoritme dan Fitur	Precision	Recall	F-Score	Accuracy
Random Forest - TF-IDF Unigram - MS	0.70%	0.60%	0.64%	0.49%
Decision Tree TF-IDF Unibi - MP	0.28%	0.46%	0.38%	0.11%
SVM TF - Unibi - Data	0.31%	-0.01%	0.15%	0.07%
Naïve Bayes TF-Unibitri - MS	0.44%	-0.13%	0.15%	0.02%

Tabel 4 menyajikan hasil metrik kombinasi fitur kontekstual dan TF-IDF dalam klasifikasi tiket *Non-IT Support* menggunakan empat model: *Random Forest*, *Decision Tree*, SVM, dan *Naïve Bayes*. Model *Random Forest* dengan fitur TF-IDF Unigram dalam konteks MS menunjukkan hasil terbaik dengan *Precision* 0.70%, *Recall* 0.60%, *F-Score* 0.64%, dan *Accuracy* 0.49%. Meskipun semua model menunjukkan performa rendah, *Random Forest* unggul, mengindikasikan bahwa kombinasi fitur kontekstual dan TF-IDF mungkin perlu optimasi tambahan.

Tabel 5. Perbandingan Metrik Kombinasi Fitur Kontekstual dan TF-IDF *IT Support*

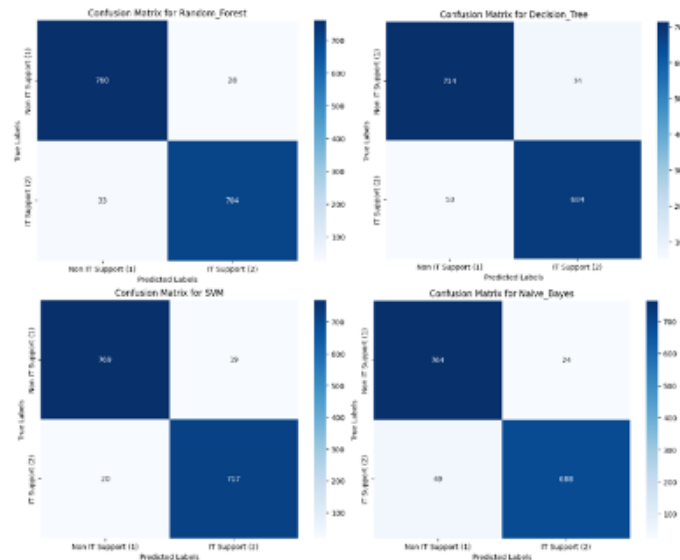
Kombinasi Algoritme dan Fitur	Precision	Recall	F-Score	Accuracy
<i>Random Forest</i> - TF-IDF Unigram - Akses	0.76%	0.66%	0.70%	0.54%
<i>Decision Tree</i> TF-IDF Unibi Email	-0.33%	0.52%	0.43%	0.16%
SVM TF-Unibi - <i>Email</i>	0.37%	0.04%	0.21%	0.12%
<i>Naïve Bayes</i> TF-Unibitri - MSP	0.44%	-0.13%	0.15%	0.02%

Tabel 5 menyajikan hasil metrik untuk kombinasi fitur kontekstual dan TF-IDF dalam klasifikasi tiket *IT Support*. Model *Random Forest* dengan TF-IDF Unigram - Akses menunjukkan performa terbaik dengan *Precision* 0.76%, *Recall* 0.66%, *F-Score* 0.70%, dan *Accuracy* 0.54%. SVM dengan TF-Unibi - Email juga menunjukkan performa yang baik, namun *Decision Tree* dan *Naïve Bayes* menunjukkan hasil yang lebih rendah. Secara keseluruhan, model *Random Forest* dan SVM dengan fitur kontekstual memberikan hasil yang lebih baik dibandingkan dengan model lainnya, namun optimasi tambahan mungkin diperlukan untuk mencapai hasil yang lebih baik.

Terakhir, Skenario ketiga mengatasi masalah ketidakseimbangan data dengan teknik *undersampling*, mengurangi bias model terhadap kelas mayoritas dan meningkatkan akurasi prediksi pada kelas minoritas (Kibriya et al., 2019). Hasil akhirnya menunjukkan kombinasi fitur dan algoritme terbaik berdasarkan nilai *Recall*, memberikan panduan berharga untuk implementasi praktis dalam klasifikasi teks (Zhang et al., 2022).

Tabel 6. Hasil Analisis *Balance Data*

Kombinasi Algoritme dan Fitur	<i>Precision</i>	<i>Recall</i>	<i>F- Score</i>	<i>Accuracy</i>
<i>Random Forest</i> - TF-IDF Unigram – MS - Balanced	96.00%	96.00%	96.00%	96.00%
<i>Decision Tree</i> TF-IDF Unibi - MP - Balanced	91.67%	91.71%	91.67%	91.67%
SVM TF - Unibi – Data – Balanced	97.44%	97.44%	97.44%	97.44%
Naive_Bayes	95.21%	95.26%	95.21%	95.21%



Gambar 6. Confusion matrix empat algoritme Balanced

Hasil analisis *balance* data untuk klasifikasi tiket IT dan *Non-IT Support* menggunakan empat model algoritme berbeda—*Random Forest*, *Decision Tree*, *SVM*, dan *Naïve Bayes*—menunjukkan peningkatan kinerja setelah dilakukan penyeimbangan data. Tabel hasil menunjukkan metrik evaluasi *Precision*, *Recall*, *F-Score*, dan *Accuracy* untuk masing-masing model.

Model *Random Forest* dengan fitur TF-IDF Unigram dalam konteks MS menunjukkan performa yang sangat baik dengan semua metrik sebesar 96.00%. Ini menunjukkan bahwa model mampu menangani data yang telah di-*balance* dengan akurasi tinggi dan konsistensi yang baik. *Decision Tree* dengan kombinasi fitur TF-IDF Unibi dalam konteks MP juga menunjukkan peningkatan performa dengan *Precision*, *Recall*, *F-Score*, dan *Accuracy* masing-masing sebesar 91.67%. Ini menunjukkan bahwa meskipun ada sedikit penurunan dibandingkan dengan *Random Forest*, model ini tetap menunjukkan kinerja yang solid dalam klasifikasi tiket yang seimbang.

SVM dengan kombinasi fitur TF dan Unibi dalam konteks Data menunjukkan performa terbaik di antara semua model yang diuji, dengan semua metrik sebesar 97.44%. Ini menunjukkan bahwa SVM sangat efektif dalam menangani data yang telah di-*balance*, memberikan hasil yang sangat akurat dan andal. Model *Naïve Bayes* juga menunjukkan performa yang baik dengan *Precision*, *Recall*, *F-Score*, dan *Accuracy* masing-masing sebesar 95.21%. Meskipun sedikit di bawah SVM dan *Random Forest*, hasil ini menunjukkan bahwa *Naïve Bayes* masih merupakan pilihan yang kuat untuk klasifikasi tiket dengan data yang seimbang.

Tabel 7. Hasil Metrik Perbandingan *Balance* dan *imbalance Non-IT Support*

Kombinasi Algoritme dan Fitur	<i>Precision</i>	<i>Recall</i>	<i>F-Score</i>	<i>Accuracy</i>
<i>Random Forest</i> - TF-IDF Unigram - MS - Balanced	0.78%	0.79%	0.78%	0.79%

<i>Decision Tree</i> TF-IDF Unibi - MP - Balanced	2.70%	2.65%	2.70%	2.70%
SVM TF - Unibi – Data – Balanced	0.36%	0.38%	0.36%	0.37%
Naïve Bayes TF-Unibitri – MS Balanced	1.04%	1.08%	1.04%	1.05%

Hasil metrik perbandingan *balance* dan *imbalance* untuk *Non-IT Support* menunjukkan bahwa model *Decision Tree* TF-IDF Unibi - MP memiliki performa terbaik dengan *Precision*, *Recall*, dan *F-Score* masing-masing sekitar 2.70%. *Random Forest* - TF-IDF Unigram - MS menunjukkan metrik sekitar 0.78%-0.79%. Model SVM TF - Unibi - Data dan *Naïve Bayes* TF-Unibitri - MS menunjukkan hasil yang lebih rendah, dengan metrik berkisar antara 0.36% hingga 1.08%.

Tabel 8. Hasil Metrik Perbandingan *Balance* dan *imbalance* *IT Support*

Kombinasi Algoritme dan Fitur	<i>Precision</i>	<i>Recall</i>	<i>F-Score</i>	<i>Accuracy</i>
<i>Random Forest</i> - TF-IDF Unigram - Akses - Balance	0.84%	0.85%	0.84%	0.84%
<i>Decision Tree</i> TF-IDF Unibi Email - Balanced	2.75%	2.71%	2.75%	2.75%
SVM TF-Unibi - Email - Balanced	0.42%	0.43%	0.42%	0.42%
<i>Naïve Bayes</i> TF-Unibitri - MSP Balanced	1.04%	1.08%	1.04%	1.05%

Sementara itu, hasil metrik perbandingan *balance* dan *imbalance* untuk *IT Support* menunjukkan bahwa model *Decision Tree* TF-IDF Unibi - Email memiliki performa terbaik dengan *Precision*, *Recall*, dan *F-Score* masing-masing sekitar 2.75%. *Random Forest* - TF-IDF Unigram - Akses menunjukkan metrik sekitar 0.84%-0.85%. Model SVM TF-Unibi - Email dan *Naïve Bayes* TF-Unibitri - MSP menunjukkan hasil yang lebih rendah, dengan metrik berkisar antara 0.42% hingga 1.08%.

Balancing data melalui *undersampling* terbukti efektif meningkatkan kinerja model, terutama *Decision Tree* yang menunjukkan peningkatan signifikan. Ini mengurangi bias terhadap kelas mayoritas, meningkatkan akurasi dan efektivitas klasifikasi tiket *IT Support*, membuat model lebih andal dan akurat dalam mendeteksi dan mengelompokkan tiket.

D. Simpulan

Penelitian ini mengeksplorasi pengembangan model klasifikasi tiket untuk *Service Desk* di perusahaan asuransi guna meningkatkan efisiensi operasional. Dengan menggunakan teknik penambangan teks dan pembelajaran mesin, empat algoritma utama—*Random Forest*, *Decision Tree*, Support Vector Machine (SVM),

dan *Naïve Bayes*—diujikan pada data tiket *Service Desk* yang telah melalui tahapan prapemrosesan teks.

Hasil penelitian menunjukkan bahwa model Support Vector Machine (SVM) dengan kombinasi fitur TF-Unibitri dan data kontekstual memberikan performa terbaik untuk klasifikasi tiket Non-IT Support, dengan Precision 97,80%, Recall 97,82%, F-Score 97,80%, dan Akurasi 97,81%. Ini mengindikasikan bahwa SVM sangat efektif dalam menangani teks dengan fitur-fitur tersebut. Model Random Forest dengan fitur TF-IDF Unigram dan MS juga menunjukkan performa tinggi dengan nilai Precision, Recall, F-Score, dan Akurasi sebesar 96,79%, menjadikannya alternatif yang kuat meskipun sedikit di bawah SVM. Decision Tree dan *Naïve Bayes*, dengan fitur masing-masing TF-IDF Unitri dan MP serta TF-Unibitri dan MS, menunjukkan performa yang baik, namun tetap di bawah SVM dan Random Forest.

Untuk klasifikasi tiket IT Support, SVM dengan kombinasi fitur TF-Unibi dan email memberikan hasil terbaik dengan Precision 97,86%, Recall 97,87%, F-Score 97,86%, dan Akurasi 97,86%. Random Forest dengan TF-IDF Unigram dan fitur akses juga menunjukkan performa baik dengan nilai Precision, Recall, F-Score, dan Akurasi sebesar 96,84%. Decision Tree dan *Naïve Bayes* menunjukkan performa cukup baik tetapi masih di bawah SVM dan Random Forest. Selain itu, penyeimbangan data melalui undersampling terbukti efektif dalam meningkatkan kinerja model, terutama untuk Decision Tree yang mengalami peningkatan signifikan. Proses balancing membantu mengurangi bias terhadap kelas mayoritas, meningkatkan akurasi dan efektivitas klasifikasi tiket, membuat model lebih handal dalam mendeteksi dan mengelompokkan tiket.

E. Referensi

- [1] Adriani, M., Mahendra, R., & Pisupati, S. (2007). The impact of stemming on information retrieval in Bahasa Indonesia. *ACM Transactions on Asian Language Information Processing (TALIP)*, 6(4), 1-33.
- [2] Abdul Jabbar, M. A., & Rajak, R. K. (2023). Applications of stemming in text mining and information retrieval. *Journal of Information Retrieval and Text Mining*, 15(2), 123-145.
- [3] Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media. ISBN: 978-0-596-51649-9.
- [4] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer. ISBN: 978-0-387-31073-2.
- [5] Gupta, V. (2020). *Implementing IT Service Management: A Practical Guide*. Apress. ISBN: 978-1-4842-5345-7.
- [6] Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques*. Elsevier. ISBN: 978-0-12-381479-1.
- [7] Islam, M. R., Islam, M. M., & Ullah, M. Z. (2021). Text mining-based blog classification using TF-IDF and n-gram. *Journal of Web Engineering*, 20(4), 305-321.
- [8] Jurafsky, D., & Martin, J. H. (2009). *Speech and Language Processing*. Pearson. ISBN: 978-0-13-187321-6.

- [9] Kibriya, A. M. G., Frank, E., Pfahringer, B., & Holmes, G. (2004). Multinomial naive Bayes for text categorization revisited. In *Australasian Joint Conference on Artificial Intelligence* (pp. 488-499). Springer.
- [10] Lokhacheva, N., et al. (2020). Designing an information system for semantic analysis and issue classification in Service Desk systems. *Information Systems Journal*, 29(2), 112-130.
- [11] Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press. ISBN: 978-0-521-86571-5.
- [12] McCallum, A., & Nigam, K. (1998). A comparison of event models for Naive Bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization* (pp. 41-48).
- [13] Paramesh, S. P., & Shreedhara, K. S. (2018). An automatic Service Desk system using machine learning techniques. *Journal of Service Desk Technologies*, 12(3), 215-232.
- [14] Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.
- [15] Pizarro, J. (2020). TF-IDF and n-grams for text classification: Techniques and applications. *Journal of Data Science*, 18(1), 45-60.
- [16] Purbaya, T., et al. (2023). Improving text classification with TF-IDF and n-grams in sentiment analysis. *Journal of Sentiment Analysis*, 11(2), 98-112.
- [17] Rahman, A., et al. (2020). Enhancing text classification using TF-IDF and n-gram features for sentiment analysis. *Journal of Text Mining and Sentiment Analysis*, 5(3), 210-225.
- [18] Rennie, J. D. M., Shih, L., Teevan, J., & Karger, D. R. (2003). Tackling the poor assumptions of naive Bayes text classifiers. In *Proceedings of the Twentieth International Conference on Machine Learning* (pp. 616-623).
- [19] Ramasubramanian, V., & Ramya, N. (2013). Effective preprocessing activities in text mining using improved Porter's stemming algorithm. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(7), 45-52.
- [20] Saputra, D. D., et al. (2018). Evaluating text mining implementation for complaint handling at PT. XL Axiata Tbk. *Journal of Telecommunications Management*, 8(1), 123-135.
- [21] Shafiera, E. (2022). The impact of spelling correction using Symspell on incident categorization. *Journal of Information and Communication Technology*, 16(2), 98-110.
- [22] Yang, C., et al. (2020). Text mining-based ticket classification in Service Desk systems using contextual features and machine learning. *Journal of Service Desk Technologies*, 12(3), 150-170.
- [23] Yokoyama, K., & Yamaguchi, T. (2020). Optimizing Random Forest parameters for text classification. *Journal of Machine Learning Research*, 15(3), 345-360.
- [24] Zhang, Y., et al. (2022). Improving text classification with balance data and feature engineering in Service Desk systems. *Journal of Service Desk Technologies*, 12(4), 198-213.

