Jurnal Teknologi Informasi dan Multimedia
p-ISSN : 2715-2529
e-ISSN : 2684-9151
https://journal.sekawan-org.id/index.php/jtim

Pengembangan Deteksi Pesan Spam pada Website Inti Everspring Indonesia Menggunakan Algoritma *Support Vector Machine*

Syafaat Akbar [1], Mamluatul Hani'ah [2] dan Imam Fahrur Rozi [3]

[1] Prodi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang, Indonesia

* Korespondensi: mamluatulhaniah@polinema.ac.id

Sitasi: S. Akbar, M. Hani'ah, and I. F. Rozi, "Pengembangan Deteksi Pesan Spam pada *Website* Inti Ever-spring Indonesia Menggunakan Algoritma *Support Vector Machine"*, Jurnal Teknologi Informasi Dan Multimedia, vol. 8, no. 2, pp. 209-219, 2026. https://doi.org/10.35746/jtim.v8i2.872

Diterima: 11-09-2025
Direvisi: 08-12-2025
Disetujui: 16-12-2025

Abstract: The development of information technology has driven the growth of email-based communication in business environments, including at Inti Everspring Indonesia. However, the high volume of incoming emails increases the potential for spam messages that may disrupt work effectiveness and data security. This study develops a spam detection system on the company's website by applying the Support Vector Machine (SVM) algorithm. SVM was selected because of its ability to perform text classification efficiently. The dataset used in this research comes from the company's internal emails, consisting of labeled spam and non-spam messages. Since the dataset is imbalanced, an oversampling process was applied, followed by text preprocessing steps including case folding, tokenization, removal of stop words, symbols, numbers, and stemming. The model was then trained using the SVM algorithm, and its performance was evaluated using several metrics: accuracy, recall, precision, and F1-score. Based on the experiments, the SVM-based spam detection model achieved 100% precision, 100% recall, and a 100% F1-score. To validate the reliability of the algorithm, SVM performance was compared with BERT and Naïve Bayes. BERT achieved 96% accuracy, and Naïve Bayes achieved 97% accuracy. These results indicate that SVM is capable of classifying messages accurately, and SVM outperforms both algorithms.

Keywords: spam detection; Naïve Bayes; email classification; text processing; Inti Everspring Indonesia

Abstrak: Perkembangan teknologi informasi mendorong pertumbuhan komunikasi melalui email dalam lingkungan bisnis, termasuk di perusahaan Inti *Everspring* Indonesia. Namun, tingginya volume email yang diterima meningkatkan potensi masuknya pesan spam yang dapat mengganggu efektivitas kerja dan keamanan data. Penelitian ini mengembangkan sistem deteksi spam pada *website* perusahaan dengan menerapkan algoritma *Support Vector Machine* (SVM). Metode SVM dipilih karena mampu melakukan klasifikasi teks secara efisien. *Dataset* yang digunakan berasal dari email internal perusahaan, terdiri dari email spam dan non-spam yang telah diberi label. Karena *dataset* bersifat tidak seimbang, dilakukan proses *oversampling*, diikuti dengan tahapan *preprocessing* teks meliputi *case folding*, tokenisasi, penghapusan stop *word*, simbol, angka, dan *stemming*. Selanjutnya, dilakukan pelatihan model menggunakan algoritma SVM serta evaluasi kinerja model melalui pengujian beberapa metrik yaitu akurasi, *recall*, presisi, dan F1-score. Berdasarkan uji coba didapatkan hasil bahwa model deteksi spam menggunakan SVM memperoleh nilai presisi 100%, *recall* 100%, dan F1-score 100. Untuk membuktikan keandalan algoritma, dilakukan perbandingan kinerja dengan BERT dan *Naïve Bayes*. Metode BERT menghasilkan akurasi 96% sedangkan *Naïve Bayes* menghasilkan akurasi 97% yang menunjukkan bahwa SVM menghasilkan performa lebih baik dibandingkan kedua algoritma tersebut sehingga algoritma SVM mampu mengklasifikasikan pesan secara akurat.

Kata kunci: deteksi spam; SVM; klasifikasi email; pengolahan teks; Inti *Everspring* Indonesia

1. Pendahuluan

Website Inti *Everspring* Indonesia (IEI) merupakan media digital profil perusahaan yang bergerak di bidang *agrochemical*. Dalam situs tersebut, pengunjung dapat mengakses informasi terkait produk-produk yang ditawarkan perusahaan. Untuk mendukung komunikasi antara pengunjung dan perusahaan, *website* menyediakan fitur kontak email. Email ialah media komunikasi digital yang sering digunakan sebagai alat bertukar informasi. Namun, seiring meningkatnya penggunaan, email juga menjadi sasaran berbagai gangguan seperti spam yang dapat mengganggu aktivitas sehari-hari pengguna [1]. Selain itu, email yang dianggap sebagai media komunikasi yang aman dan efisien juga menjadi target utama bagi pelaku kejahatan siber dalam mengakses informasi pribadi dan perusahaan secara ilegal [2]. Email yang tidak diinginkan ini umumnya dikirimkan oleh pihak tertentu untuk mendapatkan keuntungan tanpa seizin penerima, baik dalam bentuk komersial maupun manipulatif [2]. Salah satu kasus spam juga tercatat terjadi di media sosial Twitter, di mana akun palsu atau akun yang dikompromikan digunakan untuk menyebarkan konten spam [1]. *Website* Inti *Everspring* Indonesia (IEI) mengandalkan filter bawaan dari *MDaemon* dengan metode penyaringan berbasis kata kunci manual untuk menyaring email yang masuk. Hanya saja, pendekatan ini tidak cukup efektif karena memungkinkan email spam tetap masuk dan mengganggu proses kerja admin. Lebih jauh, email spam juga dapat menjadi media penyebaran *phishing*, penipuan, dan ancaman keamanan lainnya [3]. Spam masih menjadi ancaman serius dalam komunikasi berbasis email. Banyak organisasi menghadapi lonjakan serangan spam yang kian meningkat setiap tahun. Studi menunjukkan bahwa jenis spam paling berbahaya melibatkan penipuan identitas dan akun palsu, dengan spam *phishing* menyumbang 73% dari total serangan. Statistik mencatat bahwa dari setiap 12.500.000 email spam yang dikirim, setidaknya satu berhasil mendapatkan balasan [4]. Oleh karena itu, diperlukan sistem deteksi spam yang lebih akurat dan adaptif.

Pendekatan konvensional seperti *rule-based filtering* tidak mampu mendeteksi spam yang kompleks karena tidak melibatkan proses pembelajaran dari pola data[4]. Oleh karena itu, diperlukan pendekatan lain dengan kemampuan untuk mempelajari pola dari data historis. Pendekatan yang memanfaatkan *machine learning* menjadi salah satu solusi yang relevan karena mampu melakukan proses pembelajaran terhadap data teks[4]. Salah satu algoritma *machine learning* yang memiliki performa unggul dalam klasifikasi teks, termasuk deteksi spam, adalah SVM. Menurut penelitian [5], SVM dikenal memiliki akurasi tinggi dan performa yang stabil dalam berbagai skenario klasifikasi teks. Dibandingkan dengan algoritma lain seperti *Naïve Bayes* dan *Random Forest*, SVM sering kali memberikan hasil yang lebih konsisten karena kemampuannya memaksimalkan margin pemisah antar kelas, sehingga lebih efektif dalam mengatasi data yang tidak seimbang maupun berdimensi tinggi. Selain itu, penelitian[6], membandingkan kinerja *Naïve Bayes* dan SVM dalam menganalisis opini tentang kendaraan listrik di media sosial "X". Hasilnya menunjukkan bahwa SVM mampu mengungguli *Naïve Bayes Classifier* (NBC) secara signifikan dalam klasifikasi opini publik terhadap topik tersebut. Penelitian [7] menunjukkan bahwa NBC dinilai lebih unggul dibandingkan metode klasifikasi lain dari segi akurasi maupun efisiensi komputasi dalam analisis sentimen. Dalam penelitian [8] yang membahas kepuasan pelanggan terhadap layanan Traveloka menunjukkan bahwa dengan menggunakan model SVM memperoleh akurasi tertinggi dibandingkan model *logistic regression dan naive bayes*. SVM mendapatkan akurasi model sebesar 84,58%. Penelitian [9] membuktikan bahwa model SVM dengan pembagian data 70:30 menghasilkan akurasi model sebesar 85.98%. Hal ini menunjukkan bahwa model SVM adalah pemodelan yang memiliki performa unggul dalam tugas klasifikasi teks.

Berdasarkan uraian tersebut, penelitian ini mengusulkan sistem deteksi spam pada *website* Inti *Everspring* Indonesia melalui penerapan algoritma *Support Vector Machine* (SVM). Sistem ini dirancang untuk menyaring pesan spam secara otomatis dengan menganalisis kata, frasa, dan pola linguistik yang mencurigakan. Dengan demikian, sistem diharapkan mampu meningkatkan efisiensi dalam penyaringan pesan, mengurangi beban manual, serta meningkatkan keamanan komunikasi Perusahaan[10].

Beberapa penelitian sebelumnya telah membahas berbagai pendekatan dalam mendeteksi spam, khususnya pada pesan berbasis teks seperti email maupun media sosial. Penelitian [11] melakukan evaluasi terhadap empat model berbeda pada masing-masing algoritma, hasil penelitian tersebut menunjukkan bahwa algoritma NBC memiliki performa terbaik pada model *Complement NB*, dengan nilai presisi 95,4%, nilai *recall* 95,4%, dan nilai f1-score 95,4%, serta nilai akurasi 93,1%. Sementara itu, algoritma SVM mencapai hasil tertinggi pada *Sigmoid*, yaitu nilai presisi 95,6%, nilai *recall* 100%, nilai f1-score 97,7%, serta nilai akurasi 96,5%. Berdasarkan temuan tersebut, dapat ditarik kesimpulan bahwasanya SVM mempunyai kinerja yang jauh lebih mendominasi daripada NBC dengan konteks mengklasifikasikan hoaks terkait masalah kesehatan. Sementara itu, penelitian [12] meneliti deteksi spam pada platform Twitter menggunakan algoritma *Naïve Bayes*. Hasil studi menunjukkan jika algoritma tersebut mampu mencapai akurasi 95,57%, terutama karena pendekatan yang digunakan mengabaikan duplikat *tweet* sehingga menghasilkan data yang lebih bersih dan bervariasi.

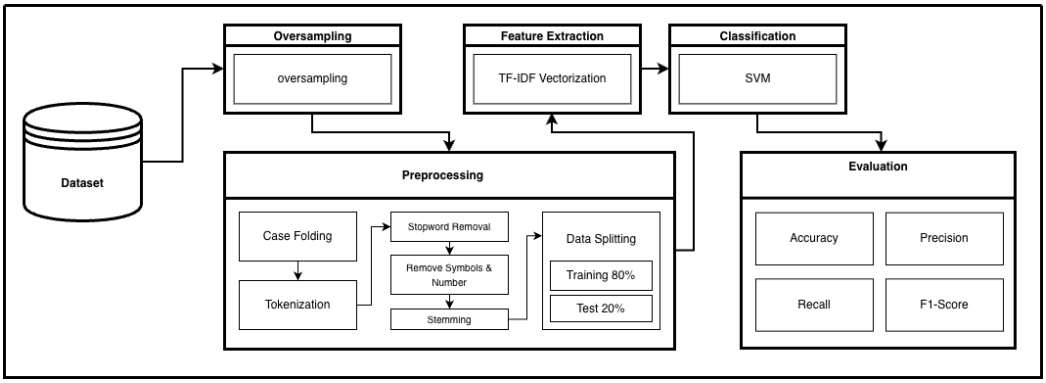
Selain algoritma klasik, model berbasis *transformer* seperti BERT juga telah diterapkan dalam tugas deteksi spam dan menunjukkan performa yang sangat baik, terutama dalam memahami konteks bahasa alami yang lebih kompleks [4]. Namun, dalam kondisi dengan keterbatasan sumber daya komputasi serta kebutuhan akan efisiensi, algoritma yang lebih ringan seperti *Naïve Bayes* tetap relevan untuk digunakan [12]. Penelitian [10] mengeksplorasi integrasi penghapusan stop *words*, ekstraksi fitur TF-IDF, dan *stemming* dalam *pipeline* deteksi spam, yang dieksekusi pada GPU Tesla P100. Dengan pendekatan tersebut, mereka melaporkan bahwa akurasi pada data pelatihan mencapai 99,67%, sedangkan akurasi pada data pengujian adalah 99,03% tercatat lebih tinggi daripada NB konvensional tanpa praproses, yang hanya mendapatkan akurasi pelatihan sebesar 99,45% dan akurasi pengujian sebesar 98,85%. Berbeda dengan penelitian [10] yang tidak menerapkan teknik penyeimbangan data, penelitian ini menggunakan metode *oversampling* untuk menangani ketidakseimbangan kelas pada data-taset, sehingga model memiliki kemampuan generalisasi yang lebih baik.

Adapun pada konteks praktis, sistem filter spam yang saat ini digunakan oleh Inti *Everspring* Indonesia masih bergantung pada *MDaemon*, sebuah *mail* server yang menggunakan metode penyaringan berbasis kata kunci secara manual. Pendekatan tersebut memiliki keterbatasan karena hanya mampu mendeteksi spam sederhana, sementara pesan spam modern sering kali menggunakan pola bahasa yang lebih halus, variatif, dan kontekstual. Maka dari itu, diperlukan pendekatan yang lebih adaptif dan cerdas, seperti metode *machine learning*, untuk mengatasi keterbatasan sistem penyaringan spam berbasis aturan konvensional.

2. Metode Penelitian

Metode ini dirancang sebagai alat mengembangkan sistem deteksi spam berbasis algoritma SVM pada *website* Inti *Everspring* Indonesia. Alur penelitian ditunjukkan pada Gambar 1, yang meliputi empat tahapan utama, yaitu *oversampling*, *preprocessing*, ekstraksi fitur, klasifikasi, dan evaluasi. Tahap awal dimulai dengan pengumpulan *dataset* email yang telah diberi label spam dan non-spam. Mengingat data bersifat tidak seimbang, dilakukan proses *oversampling* untuk menyeimbangkan distribusi kelas. Selanjutnya, data melalui praproses yang mencakup *case folding*, tokenisasi, penghapusan *stopword*, penghapusan *symbol*, dan angka, serta *stemming*. Data yang sudah melalui tahap

pemrosesan kemudian dibagi menjadi dua bagian, yaitu data latih sebesar 80% dan data uji sebesar 20%. Proses ekstraksi fitur dilakukan menggunakan *Term Frequency–Inverse Document Frequency* (TF-IDF), yang hasilnya digunakan untuk melatih model SVM. Performa model akan dihitung dengan metrik akurasi, presisi, *recall*, dan F1-score.



Gambar 1. Metode Penelitian

2.1. Pengumpulan Data

Data yang akan digunakan didapat berdasarkan arsip email perusahaan selama satu tahun terakhir, yaitu dari Januari hingga Desember 2024 yaitu memperoleh total 243 data dengan perbandingan 193 email spam dan 50 email non-spam. Teknik pengumpulan data dilakukan dengan cara meminta izin akses langsung ke admin pengolah email, kemudian diadakan *meeting* untuk membahasnya. Setelah sudah diberikan izin, dilakukan klasifikasi untuk menentukan label spam atau non-spam. Pelabelan manual dilakukan oleh peneliti yang langsung divalidasi oleh admin pengolah email Inti *Everspring* Indonesia.

2.2. Oversampling

Oversampling berfungsi untuk menyeimbangkan distribusi data pada kelas minoritas dilakukan dengan mensintesis data baru pada kelas tersebut hingga jumlahnya setara dengan jumlah data pada kelas mayoritas. Hal ini digunakan untuk menangani ketidakseimbangan pada *dataset* [13]. Ketidakseimbangan ini biasanya karena jumlah dokumen pada suatu kelas lebih sedikit daripada kelas lain, sehingga menjadikan kurangnya efektivitas model.

Pada penelitian ini, jumlah data non-spam lebih sedikit dibandingkan data spam. Ketidakseimbangan ini dapat mempengaruhi kinerja model klasifikasi. Oleh karena itu, dilakukan proses *oversampling* pada data non-spam. Langkah pertama adalah memisahkan data berdasarkan label spam dan non-spam. Setelah diketahui jumlah data spam, sistem akan menerapkan teknik *oversampling* pada kelas non-spam menggunakan fungsi *resample*() dari *Scikit-learn*. *Oversampling* dilakukan dengan pengembalian (*replace*=*True*) untuk meningkatkan jumlah data non-spam hingga seimbang dengan data spam. Setelah proses *oversampling* selesai, data non-spam hasil duplikasi digabung kembali dengan data spam untuk membentuk *dataset* yang seimbang dan siap digunakan dalam pelatihan model.

2.3. Preprocessing

Preprocessing meliputi tahapan normalisasi huruf kecil (*case folding*), tokenisasi, *stoword removal*, *remove symbol and number*, dan *stemming*. Tahapan ini dimaksudkan sebagai cara menghapus kata atau karakter yang tidak mempunyai nilai penting, sehingga penelaahan dapat difokuskan pada kata-kata yang relevan dan bermakna.

1. *Case Folding*, yaitu proses mengubah seluruh karakter dalam dokumen menjadi huruf kecil *(lowercase).* [14].
2. Tokenisasi dilakukan terhadap setiap teks masukan menjadi kata, istilah, simbol, tanda baca, atau elemen lain yang memiliki makna dan disebut sebagai *token* [15].
3. *Stopword Removal* digunakan untuk meningkatkan kinerja dalam mendeteksi email spam, stop *word* memainkan peran penting dalam mempercepat proses pencarian dengan mengecualikan beberapa kata dari email yang dianggap kurang bernilai. Tidak ada satu daftar stop *word* global yang digunakan oleh alat pemrosesan bahasa alami. Kata-kata yang paling umum *seperti 'is', 'at', 'on', 'the'*, dan '*am'* biasanya ditentukan sebagai stop *word* dengan mengurutkan istilah berdasarkan frekuensinya secara menurun, lalu mengambil istilah yang paling sering muncul untuk disaring sebagai stop *word* [10].
4. Menghapus simbol dan angka bertujuan untuk membersihkan teks dari karakter non-huruf, seperti tanda baca, simbol khusus, dan angka, sehingga akan menyisakan kata-kata yang valid untuk proses analisis.
5. *Stemming* dilakukan untuk menghasilkan kata dasar dari setiap *token*. Penelitian ini menggunakan *Porter Stemmer*. Ini adalah metode untuk menghapus akhiran morfologis dan infleksional dari kata-kata dalam bahasa Inggris. *Stemmer* ini bekerja secara linear dalam lima langkah dengan menerapkan aturan pada setiap langkahnya [14].

2.3.1. TF-IDF

TF-IDF yang merupakan ekstraksi fitur pada data teks berfungsi untuk mengukur seberapa penting suatu kata dalam sebuah dokumen dibandingkan dengan seluruh dokumen dalam kumpulan data (*corpus*). TF-IDF merupakan gabungan dari TF dan IDF [16], TF memberikan bobot berdasarkan frekuensi kemunculan suatu kata dalam dokumen tertentu sedangkan IDF memberikan bobot berdasarkan keunikan atau kelangkaan kata di seluruh kumpulan dokumen.

$$TF_{i,j} = \frac{f_{i,j}}{\sum_k f_{i,j}} \quad (1)[16]$$

Keterangan:
$TF_{i,j}$ = frekuensi relatif kata ke-i pada dokumen ke-j
$f_{i,j}$ = jumlah kemunculan kata ke-i dalam dokumen ke-j
$\sum_k f_{i,j}$ = jumlah seluruh kata dalam dokumen ke-j

Di mana $TF_{i,j}$ adalah frekuensi istilah ke-i dalam dokumen ke-j, dan $\sum_k f_{i,j}$ adalah jumlah total kata dalam dokumen j [16].

Namun, *term frequency* (TF) saja tidak mampu mengukur pentingnya istilah yang hanya muncul sesekali dalam beberapa dokumen. Untuk mengukur istilah yang jarang muncul namun signifikan, digunakan istilah *inverse document frequency* (IDF) seperti dalam persamaan (2) [16]:

$$IDF(i, D) = \log\left(\frac{N+1}{n_i+1}\right) + 1 \quad (2)[16]$$

Keterangan:
N = jumlah dokumen
n_i = jumlah dokumen yang memiliki *term* i
+ 1 di luar log = Penyesuaian hasil agar tetap positif atau stabil

Untuk menghitung nilai akhir TF-IDF, cukup mengalikan persamaan (1) dan (2) seperti dalam persamaan (3) [16]. Hasil perhitungan ini berfungsi untuk menyaring kata-

kata yang bersifat umum sekaligus mempertahankan kata-kata yang dianggap penting dalam dokumen.

$$TF\text{-}IDF(i,j,d,D) = TF_{i,j}.IDF(i,D) \quad (3)[16]$$

Keterangan:
· = Operasi perkalian antara TF dan IDF

2.3.2. Pelatihan Model

Algoritma ini juga dikenal dengan nama SVM *Classification*, adalah sebuah teknik untuk melakukan klasifikasi berbasis *supervised learning*. SVM bekerja dengan cara mempelajari pola dari data latih untuk kemudian digunakan dalam memprediksi kelas pada data uji [17]. Menurut penelitian [6], SVM merupakan metode klasifikasi berbasis *supervised learning* yang dapat memperkirakan kelasnya dengan memanfaatkan pola yang diperoleh dari tahap pelatihan. Proses klasifikasi pada SVM dilakukan melalui pembentukan *hyperplane* yang berfungsi memisahkan data antara positif dan negatif. *Hyperplane* dianggap optimal adalah yang mempunyai jarak margin terlebar terhadap data *train* terdekat dari masing-masing kelas, sebab margin yang lebih luas umumnya mampu menurunkan tingkat kesalahan generalisasi. Dengan demikian, pemisahan terbaik ditentukan melalui pengukuran margin *hyperplane* dan pencarian titik maksimalnya. [18]. SVM dikenal memiliki tingkat akurasi yang tinggi, terutama dalam tugas klasifikasi seperti analisis sentimen, serta mampu menangani data kompleks secara efektif. Rumus perhitungan dasar SVM ditunjukkan pada Persamaan (4) [17]:

$$f(X_d) = \sum_{i=1}^{ns} a_i y_i x_i x_d + b \quad (4)[17]$$

Pada persamaan (4), terdapat beberapa variabel penting yang digunakan dalam proses klasifikasi. Variabel ns menyatakan seberapa banyak support vector, a_i merupakan bobot dari masing-masing titiknya, y_i menunjukkan kelas data, x_i adalah vektor data, sedangkan x_d adalah data baru yang dikelompokkan. Adapun b merupakan nilai bias atau kesalahan yang diperoleh dari proses pelatihan model. Dalam penelitian ini, digunakan algoritma SVM dengan *kernel* RBF dan nilai parameter C=1.

2.3.3. Evaluasi Model

Kinerja model yang dilatih diukur menggunakan beberapa metrik seperti akurasi, presisi, recall, dan F1-score dengan memanfaatkan *Confusion Matrix* seperti pada tabel 1.

Tabel 1. *Confusion Matrix*

Confusion Matrix	***Actually Positive (1)***	***Actually Negative (0)***
Predicted Positive (1)	*True Positives (TPs)*	*False Positives (FPs)*
Predicted Negative (0)	*False Negatives (FNs)*	*True Negatives (TNs)*

Keterangan
TP: jumlah data positif yang berhasil diprediksi dengan benar sebagai positif.
TN: jumlah data negatif yang berhasil diprediksi dengan benar sebagai negatif.
FP: jumlah data negatif yang keliru diprediksi sebagai positif.
FN: jumlah data positif yang keliru diprediksi sebagai negatif.

1. Akurasi

Akurasi menghitung prediksi yang benar dari total data:

$$akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)[19]$$

2. Presisi

Menunjukkan proporsi email yang benar-benar spam di antara semua email yang diprediksi sebagai spam, dapat dilihat pada persamaan (6) [19]:

$$precision = \frac{TP}{TP + FP} \quad (6)[19]$$

3. Recall

Recall mengukur kemampuan model untuk mendeteksi semua email spam yang sebenarnya, dapat dilihat pada persamaan (7) [19]:

$$recall = \frac{TP}{TP + FN} \quad (7)[19]$$

4. F1-Score

Rata-rata persamaan (6) [19] dan persamaan (7) [19], yang memberikan keseimbangan antara keduanya:

$$F1\text{-}Score = 2 \times \frac{Precision \cdot Recall}{Precision + Recall} \quad (8)[19]$$

3. Pembahasan

Data yang diolah dalam penelitian ini didapatkan dari arsip email perusahaan selama periode Januari hingga Desember 2024 dengan jumlah total 243 data, terdiri dari 193 email spam dan 50 email non-spam. Pemilihan *dataset* internal ini bertujuan untuk memastikan relevansi data terhadap kondisi nyata yang dihadapi oleh perusahaan.

Tahapan penelitian meliputi beberapa langkah utama. Pertama, dilakukan *oversampling* untuk menyeimbangkan distribusi kelas antara email spam dan non-spam. *Oversampling* diterapkan guna mengurangi bias model terhadap kelas mayoritas [13]. Selanjutnya, dilakukan *preprocessing* teks yang mencakup *case folding*, *tokenization*, *stopwords removal*, penghapusan simbol dan angka, serta *stemming* untuk mengembalikan kata ke bentuk dasarnya. Data yang telah diproses kemudian ditransformasi menggunakan metode TF-IDF sehingga dapat direpresentasikan dalam bentuk numerik. Tabel 2 merupakan contoh *preprocessing* data yang dilakukan pada penelitian ini.

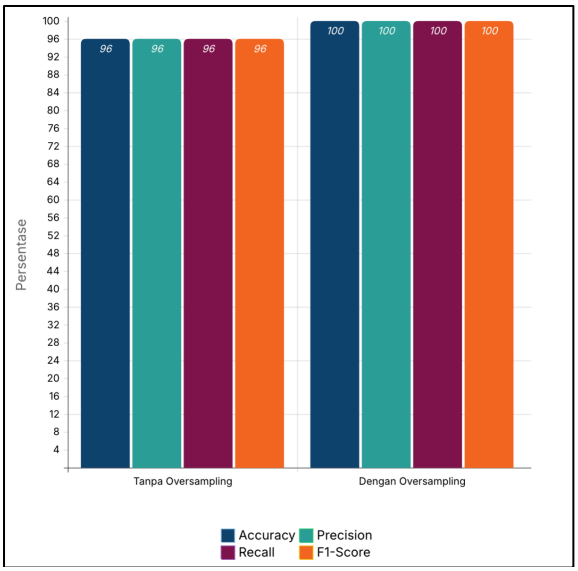
Tabel 2. *Contoh Preprocessing*

Tahap Preprocessing	***Sebelum***	***Sesudah***
Case Folding	*Congratulations! You've won $10,000! Click here to claim your prize now! Limited time offer*	*congratulations! you've won $10,000! click here to claim your prize now! limited time offer.*
Tokenization	*congratulations! you've won $10,000! click here to claim your prize now! limited time offer.*	*['congratulations', '!', 'you', ''', 've', 'won', '$', '10,000', '!', 'click', 'here', 'to', 'claim', 'your', 'prize', 'now', '!', 'limited', 'time', 'offer', '.']*
Stopwords removal	*['congratulations', '!', 'you', ''', 've', 'won', '$', '10,000', '!', 'click', 'here', 'to', 'claim', 'your', 'prize', 'now', '!', 'limited', 'time', 'offer', '.']*	*['congratulations', '!', ''', '$', '10,000', '!', 'click', 'claim', 'prize', '!', 'limited', 'time', 'offer', '.']*
Remove Symbols & Number	*['congratulations', '!', ''', '$', '10,000', '!', 'click', 'claim',*	*['congratulations', 'click', 'claim', 'prize', 'limited', 'time', 'offer']*

Tahap Preprocessing	***Sebelum***	***Sesudah***
	'prize', '!', 'limited', 'time', 'offer', '.']	
Stemming	*['congratulations', 'click', 'claim', 'prize', 'limited', 'time', 'offer']*	*['congratul', 'click', 'claim', 'prize', 'limit', 'time', 'offer']*

Setelah data dilakukan *prepocessing*, selanjutnya dilakukan ekstraksi fitur menggunakan TF-IDF. Hasil TF-IDF inilah yang menjadi *input* algoritma SVM yang kemudian dilakukan proses pelatihan dan pengujian. Eksperimen dilakukan dalam tiga skenario, yaitu (1) klasifikasi tanpa *oversampling*, (2) klasifikasi setelah dilakukan *oversampling*, (3) membandingkan performa hasil algoritma SVM dengan algoritma *Naïve Bayes* dan *Bidirectional Encoder Representations from Transformers* (BERT) yang bertujuan untuk keperluan analisis komparatif. Pemilihan *Naïve Bayes* didasarkan pada kemampuannya menangani *dataset* berukuran kecil dengan dimensi tinggi, sedangkan BERT dipilih karena keunggulannya dalam memahami konteks semantik melalui *contextual embedding*. Proses pelatihan memanfaatkan 80% data dengan *hyperparameter* SVM berupa kernel RBF, nilai C = 1, dan nilai gamma = scale. Sedangkan algoritma pembanding digunakan *Multinomial Naive Bayes* dengan alpha = 0.1 sebagai *Laplace smoothing*. Pada model BERT, digunakan *IndoBERT-base* dengan konfigurasi *batch* size 16, learning rate 0.00002, max sequence length 128.

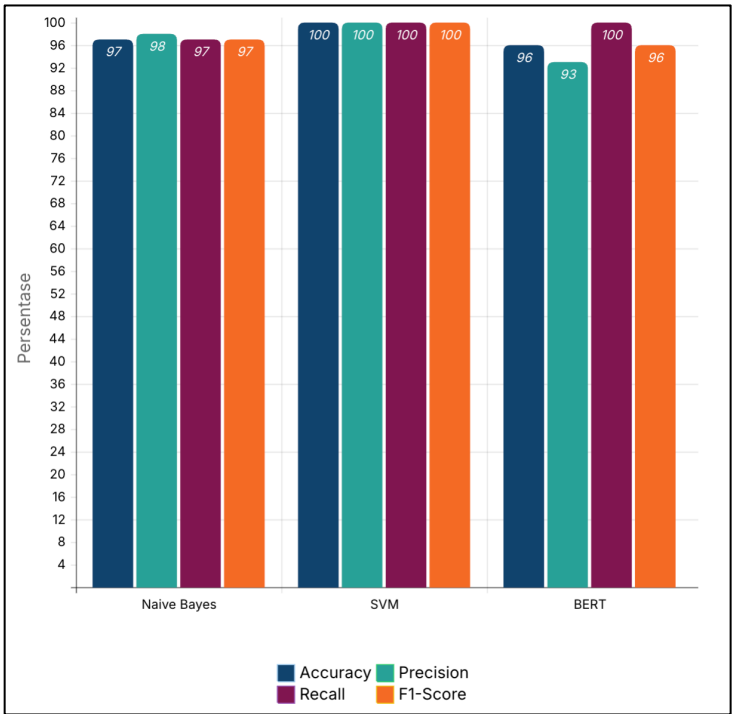
Hasil pengujian awal ditampilkan pada Gambar 2, yang memperlihatkan perbandingan kinerja SVM pada *dataset* tanpa *oversampling* dan dengan *oversampling*. Meskipun terdapat perbedaan distribusi kelas, performa SVM tidak menunjukkan peningkatan signifikan setelah dilakukan *oversampling*. Hal ini dikarenakan karakteristik *dataset* yang cukup jelas membedakan antara email spam dan non-spam. Secara umum, email non-spam berisi kosakata yang relevan dengan aktivitas bisnis seperti *product* atau order, sedangkan email spam sering mengandung kata-kata promosi seperti *advertisement*, promo, atau *click here*.



Gambar 2. Perbandingan Algoritma SVM Tanpa *Oversampling* dan Dengan *Oversampling*

Selanjutnya, hasil perbandingan antara SVM, *Naïve Bayes*, dan BERT ditunjukkan pada Gambar 3. Dari hasil tersebut, algoritma SVM terbukti memberikan performa terbaik di hampir semua metrik evaluasi. Pada metrik akurasi SVM menghasilkan performa paling baik dengan nilai akurasi 100%, SVM mengungguli NB yang memiliki akurasi 97% dan BERT yang memiliki akurasi 96%. *Naïve Bayes* menunjukkan performa yang stabil, terutama pada *dataset* kecil, sedangkan BERT menghasilkan performa

terendah. Rendahnya performa BERT disebabkan oleh keterbatasan jumlah data, mengingat BERT membutuhkan *dataset* berukuran besar untuk mengoptimalkan representasi konteks [5].



Gambar 3. Perbandingan Hasil Evaluasi Model Dengan *Oversampling*

Berdasarkan hasil yang terdapat pada gambar 2 dan gambar 3 dengan membandingkan tiga algoritma terhadap data pesan *email spam* dan *non-spam* yang berjumlah 243 data dapat disimpulkan bahwa SVM merupakan algoritma yang memiliki performa yang unggul untuk deteksi spam. Keunggulan SVM terletak pada kemampuannya memisahkan data non-linear dengan margin yang optimal, sehingga tetap memiliki performa yang baik meskipun jumlah data relatif kecil. Sementara itu, meskipun *Naïve Bayes* memiliki kelebihan dari segi kesederhanaan dan efisiensi, akan tetapi algoritma ini tidak mampu melampaui akurasi yang dicapai oleh SVM. Sebaliknya, BERT yang secara teoritis lebih unggul dalam memahami konteks semantik justru tidak optimal pada penelitian ini karena jumlah data yang terbatas tidak cukup untuk memanfaatkan potensi penuh model berbasis *transformer*. Dengan demikian, hasil penelitian ini memperkuat temuan sebelumnya bahwa SVM sangat efektif untuk klasifikasi teks dengan *dataset* kecil hingga menengah. Namun, untuk penerapan di masa depan dengan data dalam skala besar dan bervariasi, penggunaan model berbasis *deep learning* seperti BERT tetap potensial, terutama apabila dipadukan dengan strategi *fine-tuning* dan *incremental learning*.

4. Kesimpulan

Pada penelitian ini digunakan algoritma SVM untuk deteksi email spam pada sistem *website* Inti *Everspring* Indonesia. Evaluasi kinerja dilakukan menggunakan metrik akurasi, presisi, *recall*, dan F1-score dengan membandingkan performa SVM terhadap algoritma *Naïve Bayes* dan BERT. Hasil pengujian menunjukkan bahwa SVM menghasilkan performa paling baik dengan nilai akurasi 100%, melampaui NB dengan akurasi 97% dan BERT dengan nilai akurasi 96%. Meskipun demikian, penelitian ini masih memiliki beberapa keterbatasan dimana *dataset* yang digunakan hanya berasal dari arsip email tahun 2024, sedangkan karakteristik dan pola email spam bersifat dinamis

dan akan terus berkembang seiring waktu. Keterbatasan tersebut berpotensi membatasi kemampuan model dalam mengenali pola email baru yang mungkin muncul pada periode berikutnya. Berdasarkan hal tersebut, pada penelitian selanjutnya diperlukan pemanfaatan *dataset* yang lebih besar dan berkelanjutan secara berkala berbasis metode *incremental learning[20]* agar model dapat diperbarui secara berkala menggunakan email terbaru. Dengan demikian, sistem diharapkan mampu meningkatkan cakupan pola yang dikenali serta meminimalkan kesalahan dalam proses penyaringan email spam di masa mendatang.

Ucapan Terima Kasih: Penulis menyampaikan terima kasih kepada Politeknik Negeri Malang atas dukungan dan fasilitas yang diberikan selama pelaksanaan penelitian ini. Ucapan terima kasih juga disampaikan kepada PT Inti Everspring Indonesia yang telah menyediakan data penelitian sehingga penelitian ini dapat terlaksana dengan baik

Referensi

[1] M. R. Ningsih, J. Unjung, H. al Farih, and M. A. Muslim, "Classification email spam using Naive Bayes algorithm and Chi-squared feature selection," J. Appl. Intell. Syst., vol. 9, no. 1, pp. 74–87, 2024, https://doi.org/10.62411/jais.v9i1.9695

[2] Z. A. Diekson, M. R. B. Prakoso, M. S. Q. Putra, M. S. A. F. Syaputra, S. Achmad, and R. Sutoyo, "Sentiment analysis for customer review: Case study of Traveloka," *Procedia Computer Science*, vol. 216, pp. 682–690, 2023, https://doi.org/10.1016/j.procs.2022.12.184

[3] R. R. Salam, M. F. Jamil, Y. Ibrahim, R. Rahmaddeni, S. Soni, and H. Herianto, "Analisis sentimen terhadap Bantuan Langsung Tunai (BLT) Bahan Bakar Minyak (BBM) menggunakan Support Vector Machine: Sentiment analysis of cash direct assistance distribution for fuel oil using Support Vector Machine," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 3, no. 1, pp. 27–35, 2023, https://doi.org/10.57152/malcom.v3i1.590

[4] R. Fatima, M. M. S. Fareed, S. Ullah, G. Ahmad, and S. Mahmood, "An optimized approach for detection and classification of spam email's using ensemble methods," Wireless Pers. Commun., pp. 1–27, 2024, https://doi.org/10.1007/s11277-024-11628-9

[5] Y. R. Hutagaol and Y. Arifin, "Klasifikasi spam email berbasis semantik menggunakan metode BERT," INTECOMS J. Inf. Technol. Comput. Sci., vol. 7, no. 5, pp. 1823–1836, 2024, https://doi.org/10.31539/intecoms.v7i5.12515

[6] T. Sahmoud and M. Mikki, "Spam detection using BERT," arXiv preprint arXiv:2206.02443, 2022. https://doi.org/10.48550/arXiv.2206.02443

[7] R. Meléndez, M. Ptaszynski, and F. Masui, "Comparative investigation of traditional machine-learning models and transformer models for phishing email detection," Electronics, vol. 13, no. 24, p. 4877, 2024. https://doi.org/10.3390/electronics13244877

[8] A. Wantoro and D. Saputra, "Perbandingan metode Naive Bayes dan Support Vector Machine (SVM) pada analisis kendaraan listrik pada media sosial 'X'," J. Inform. Polinema, vol. 11, no. 2, pp. 227–234, 2025, https://doi.org/10.33795/jip.v11i2.6458

[9] R. Apriani and D. Gustian, "Analisis sentimen dengan Naïve Bayes terhadap komentar aplikasi Tokopedia," J. Rekayasa Teknol. Nusa Putra, vol. 6, no. 1, pp. 54–62, 2019, https://doi.org/10.52005/rekayasa.v6i1.86

[10] M. Jaiswal and S. Das, "Detecting spam e-mails using stop word TF-IDF and stemming algorithm with Naïve Bayes classifier on the multicore GPU," Int. J. Electr. Comput. Eng. (IJECE), vol. 11, no. 4, pp. 3168–3175, Aug. 2021, https://doi.org/10.11591/ijece.v11i4.pp3168-3175

[11] R. R. Sani, Y. A. Pratiwi, S. Winarno, E. D. Udayanti, and F. Alzami, "Analisis perbandingan algoritma Naive Bayes classifier dan Support Vector Machine untuk klasifikasi berita hoax pada berita online Indonesia," J. Masy. Informatika, vol. 13, no. 2, pp. 85–98, Nov. 2022, https://doi.org/10.14710/jmasif.13.2.47983

[12] D. Wahyuningtyas, "Spam detection on Twitter using Naive Bayes algorithm," J. Ilmu Komput. Apl. (JIKA), vol. 7, no. 1, pp. 31–40, 2020, https://doi.org/10.29244/jika.7.1.31-40

[13] I. Sitohang, T. H. Saragih, D. Kartini, R. A. Nugroho, and M. R. Faisal, "Implementasi SMOTE dan extreme learning machines pada klasifikasi dataset microarray," J. Inform. Polinema, vol. 8, no. 4, pp. 9–16, 2022, https://doi.org/10.33795/jip.v8i4.1029

[14] M. Rahayu, A. Luthfiarta, L. Cahyaningrum, and A. N. Azzahra, "Pengaruh oversampling dan cross validation pada model machine learning untuk sentimen analisis kebijakan luaran kelulusan mahasiswa," J. Media Inform. Budidarma, vol. 8, no. 1, pp. 163–172, Jan. 2024, https://doi.org/10.30865/mib.v8i1.7012

[15] S. Vijayarani and R. Janani, "String matching algorithms for retrieving information from desktop—comparative analysis," in Proc. Int. Conf. Inventive Comput. Technol. (ICICT), vol. 3, pp. 1–6, Aug. 2016, https://doi.org/10.1109/INVENTIVE.2016.7830233

[16] R. Wongso, F. A. Luwinda, B. C. Trisnajaya, and O. Rusli, "News article text classification in Indonesian language," Procedia Comput. Sci., vol. 116, pp. 137–143, 2017, https://doi.org/10.1016/j.procs.2017.10.039

[17] A. Subadi and K. Kusrini, "Diagnosa stunting berdasarkan gejala medis menggunakan algoritma Naive Bayes, SVM dan K-NN," J. Inform. Polinema, vol. 10, no. 4, pp. 501–510, 2024, https://doi.org/10.33795/jip.v10i4.5628

[18] H. Apriyani and K. Kurniati, "Perbandingan metode Naive Bayes dan Support Vector Machine dalam klasifikasi penyakit diabetes melitus," J. Inf. Technol. Ampera, vol. 1, no. 3, pp. 133–143, 2020, https://doi.org/10.51519/journalita.volume1.issue3.year2020.page133-143

[19] A. Fadli, T. Limbong, R. Priskila, and V. H. Pranatawijaya, "Penggunaan algoritma Naive Bayes untuk memprediksi kelulusan mahasiswa," J. Mahas. Tek. Inform. (JATI), vol. 8, no. 3, pp. 3773–3779, Jun. 2024, https://doi.org/10.36040/jati.v8i3.9791

[20] N. Capuano, L. Greco, P. Ritrovato, and M. Vento, "Sentiment analysis for customer relationship management: an incremental learning approach," *Applied Intelligence 2020 51:6*, vol. 51, no. 6, pp. 3339–3352, Nov. 2020, https://doi.org/10.1007/S10489-020-01984-X