Jurnal Manajemen Informatika & Sistem Informasi (MISI)
Volume 9, Nomor 1, Januari 2026.
ISSN 2614-1701 (media cetak)
ISSN 2614-3739 (media online)
DOI : 10.36595/misi.v5i2
http://e-journal.stmiklombok.ac.id/index.php/misi

IMPLEMENTASI ALGORITMA SVM PADA BOT WHATSAPP UNTUK DETEKSI PESAN SPAM

Reza Bahtiar Saputra[1], Muhamad Azwar[2], Lilik Widyawati[3], Husain[4], kurniadin Abdul Latif[5]

[123,]Program Studi Teknologi Informasi Universitas Bumigora, [45,]Program Studi Ilmu Komputer Universitas Bumigora

Jln. Ismail Marzuki No.22, Universitas Bumigora, Cilinaya, Cakranegara, Mataram 83127
[1] rezaosd339@gmail.com, [2] azwar@universitasbumigora.ac.id, [3] lilikwidya@universitasbumigora.ac.id, [4] husain@universitasbumigora.ac.id, [5] kurniadin@universitasbumigora.ac.id

Abstract

This study examines the classification of Indonesian-language WhatsApp messages into three categories—normal, fraudulent, and promotional—using the Support Vector Machine (SVM) algorithm. A total of 1,320 messages were collected from various sources with representative class distribution. Text preprocessing and feature extraction were performed using standard techniques to generate numerical representations suitable for model training. The SVM model was trained with parameter optimization and evaluated using common classification metrics. The model achieved an accuracy exceeding 90%, indicating the effectiveness of this approach in identifying spam messages. The resulting system was integrated into a WhatsApp bot built with Node.js and a Flask API to enable real-time message detection. This research contributes to the development of more efficient message filtering mechanisms that enhance the security of digital communication.

Keywords: *CRISP-DM, Spam Detection, WhatsApp Bot, Support Vector Machine.*

Abstrak

Penelitian ini mengkaji proses klasifikasi pesan WhatsApp berbahasa Indonesia ke dalam tiga kategori, yaitu normal, penipuan, dan promosi dengan memanfaatkan algoritma Support Vector Machine (SVM). Sebanyak 1320 pesan dikumpulkan dari beragam sumber dengan proporsi kelas yang representatif. Prapengolahan teks dan ekstraksi fitur diterapkan menggunakan teknik standar untuk menghasilkan representasi numerik yang dapat diproses oleh model. Pelatihan SVM dilakukan dengan optimasi parameter dan evaluasi menggunakan metrik klasifikasi umum. Model menunjukkan akurasi lebih dari 90%, menandakan bahwa pendekatan ini efektif dalam mengidentifikasi pesan spam. Sistem yang dihasilkan diintegrasikan ke dalam bot WhatsApp berbasis Node.js dan API Flask guna memungkinkan deteksi pesan secara real-time. Penelitian ini memberikan kontribusi terhadap pengembangan mekanisme penyaringan pesan yang lebih efisien dan mendukung keamanan komunikasi digital.

Kata kunci: CRISP-DM, Deteksi Spam, Bot WhatsApp, Support Vector Machine.

1. PENDAHULUAN

Perkembangan teknologi komunikasi digital telah menjadikan aplikasi pesan instan sebagai sarana utama interaksi masyarakat global. WhatsApp, dengan lebih dari 2 miliar pengguna di seluruh dunia, menempati posisi sebagai salah satu aplikasi paling populer dan banyak digunakan untuk keperluan pribadi maupun pekerjaan[1][2]. Salah satu keunggulan utama WhatsApp adalah kemudahan dalam penggunaan serta kemampuannya mengirim pesan dengan

Jurnal Manajemen Informatika & Sistem Informasi (MISI)
Volume 9, Nomor 1, Januari 2026.
ISSN 2614-1701 (media cetak)
ISSN 2614-3739 (media online)
DOI : 10.36595/misi.v5i2
http://e-journal.stmiklombok.ac.id/index.php/misi

cepat dan efisien. Akan tetapi, seiring bertambahnya jumlah pengguna dan meningkatnya popularitas aplikasi ini, muncul tantangan baru yang berkaitan dengan aspek keamanan dan kenyamanan, khususnya meningkatnya peredaran pesan spam.

Pesan spam sendiri merupakan pengiriman pesan tanpa izin penerima yang berpotensi menimbulkan gangguan, penipuan, hingga pencurian data pribadi [3]. Berdasarkan data yang dipublikasikan dalam *Truecaller Insights Report* 2020, Indonesia menempati peringkat teratas sebagai negara dengan jumlah pesan spam paling tinggi di Asia pada tahun 2020. Dari keseluruhan spam yang teridentifikasi, mayoritas berasal dari layanan keuangan yang menyumbang hingga 52%, kemudian diikuti oleh spam asuransi sebesar 25%, operator seluler sebesar 11%, spam berupa penipuan (*scam*) sebesar 9%, dan terakhir tagihan hutang yang menyumbang 3%. Laporan ini memperlihatkan bahwa spam di Indonesia lebih banyak terkait dengan aktivitas finansial dan komersial [3].

Selain menimbulkan gangguan bagi pengguna, spam juga dapat menjadi indikasi awal adanya aktivitas berbahaya yang mengarah pada kejahatan siber, termasuk praktik penipuan serta pencurian informasi pribadi yang dilakukan oleh pihak-pihak yang tidak memiliki tanggung jawab [4], seperti dalam penelitian Prasatra [2] menyebutkan modus yang digunakan pelaku kejahatan dalam menyebarkan spam pada WhatsApp umumnya dilakukan melalui pengiriman berbagai tautan yang sering kali menarik perhatian pengguna. Maraknya kasus penipuan digital sejak 2022–2024 seperti modus kurir palsu, undangan pernikahan fiktif, file APK berbahaya, penyamaran layanan resmi, hingga pemerasan melalui video call sex yang menunjukkan bahwa spam pada WhatsApp kini semakin variatif dan berbahaya.

Berbagai penelitian sebelumnya telah mengkaji deteksi spam berbasis machine learning. Penelitian Reviantika [5], penelitian tersebut menggunakan algoritma *Logistic Regression* untuk klasifikasi SMS spam dan non-spam dengan 1143 data (566 spam dan 577 non-spam). Proses dilakukan melalui *preprocessing teks* dan pembobotan TF-IDF, lalu dievaluasi menggunakan *confusion matrix*. Hasilnya menunjukkan akurasi mencapai 95%, yang lebih baik dibandingkan *Naïve Bayes* dengan akurasi 93%.

Kemudian penelitian lainnya yang dilakukan oleh Dwiyansaputra [6] mengombinasikan TF-IDF dan *Stochastic Gradient Descent* (SGD) *Classifier* dan memperoleh akurasi hingga 97%. Walaupun akurasi model cukup tinggi, proses deteksi umumnya dilakukan secara manual atau tidak *real-time*, sehingga pesan spam dapat terbaca pengguna sebelum teridentifikasi. Untuk menjawab kelemahan tersebut, penggunaan bot WhatsApp berbasis algoritma *Machine Leraning* dinilai efektif karena mampu melakukan deteksi secara otomatis saat pesan diterima, bekerja optimal pada data teks berdimensi tinggi, dan tetap stabil meskipun jumlah data relatif terbatas [3].

Tujuan penelitian ini adalah mengembangkan model deteksi spam berbasis *Support Vector Machine* (SVM) yang mampu mengklasifikasikan pesan WhatsApp ke dalam kategori Normal, Penipuan, dan Promosi secara akurat, serta mengintegrasikannya ke dalam bot WhatsApp agar proses deteksi dapat berjalan otomatis dan *real-time*. Selain itu, penelitian ini juga bertujuan mengevaluasi performa model melalui proses *training* dan *testing*, termasuk melakukan analisis terhadap kesalahan klasifikasi untuk mengetahui efektivitas sistem dalam mendeteksi pesan spam serta potensi peningkatan kinerjanya di masa mendatang.

Bot yang dikembangkan dalam penelitian ini bekerja secara otomatis tanpa campur tangan pengguna, memberikan peringatan terhadap pesan yang terdeteksi sebagai spam, serta dapat melakukan tindakan lanjutan seperti blokir atau pelabelan pesan. Dengan kemampuan tersebut, sistem diharapkan mampu mengurangi risiko interaksi awal pengguna dengan pesan berbahaya dan meningkatkan keamanan komunikasi digital.

2. TINJAUAN PUSTAKA

2.1 Klasifikasi Teks

Klasifikasi merupakan proses membangun sebuah model atau fungsi yang bertujuan untuk menjelaskan serta merepresentasikan suatu konsep atau kelas tertentu dari sekumpulan data. Model ini dibentuk melalui analisis terhadap *training dataset*, yaitu data yang sudah diberi label sesuai dengan kelasnya [7]. Dalam penelitian lain menyebutkan, klasifikasi adalah salah satu metode dalam *data mining* yang berfungsi untuk mengelompokkan sebuah objek ke dalam kelas tertentu [8][9][10]. Dalam klasifikasi teks, terdapat dua jenis utama, yaitu klasifikasi biner

Jurnal Manajemen Informatika & Sistem Informasi (MISI)
Volume 9, Nomor 1, Januari 2026.
ISSN 2614-1701 (media cetak)
ISSN 2614-3739 (media online)
DOI : 10.36595/misi.v5i2
http://e-journal.stmiklombok.ac.id/index.php/misi

dan multi-kelas. Klasifikasi biner membagi teks ke dalam dua kategori berlawanan, seperti spam dan non-spam, sehingga sering digunakan dalam deteksi pesan spam. Sementara itu, klasifikasi multi-kelas memungkinkan teks masuk ke lebih dari dua kategori, misalnya topik berita, jenis produk, atau genre dokumen[11].

2.2 WhatsApp Bot

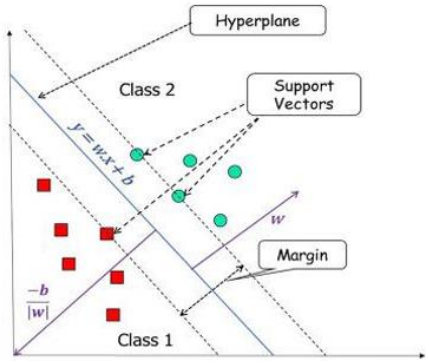
WhatsApp Bot adalah sebuah aplikasi atau program yang dibuat khusus untuk memberikan respon otomatis terhadap pesan yang diterima [12]. Pada penelitian yang dilakukan P. Alia et al.[13] tahun 2024, yang mengimplementasikan bot whatsApp dengan bantuan proses NLP yang digunakan untuk menjawab otomatis berbagai pertanayaan pelaanggan seputar produk, harga, alamat, dan informasi lain yaang sering ditanyakan sehingga layanan menjadi lebih cepat dan efisien bagi penjual maupun pembeli.

2.3 Python

Python merupakan bahasa pemrograman berorientasi objek yang dapat dijalankan dan digunakan secara interaktif [14]. Python dirancang dengan filosofi yang menekankan keterbacaan kode, memiliki sintaks yang jelas, serta didukung oleh pustaka standar yang luas dan lengkap, sehingga mampu menggabungkan kemudahan penulisan dengan fungsionalitas yang kuat [15]. Dalam penelitian J. Tan et al. tahun (2024), python merupakan bahasa pemrograman tingkat tinggi yang banyak dimanfaatkan dalam pengembangan perangkat lunak, analisis data, hingga kecerdasan buatan. [16].

2.4 SVM

SVM (*Support Vector Machine*) adalah algoritma *machine learning* yang digunakan untuk klasifikasi dan regresi dengan membangun *hyperplane* sebagai batas pemisah antar kelas.

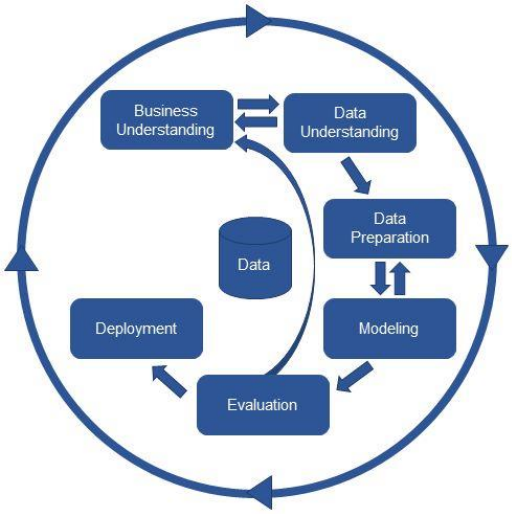


Gambar 1. Algotima SVM [3]

Pada penelitian oleh M. Arif Sofyan et al. (2024), menggunakan algoritma SVM untuk melakukan klasifikasi pesan pada SMS dan diimplementasikan kedalam aplikasi berbasasis *streamlit*[3]. Sedangkan pada penelitian ini, memanfaatkan algoritma SVM ke dalam sistem otomatis berbasis bot untuk melakukan klasifkasi pesan spam.

2.5 CRISP-DM

CRISP-DM (*Cross-Industry Standard Process for Data Mining*) adalah sebuah kerangka kerja standar yang digunakan untuk menyelesaikan masalah berbasis data melalui enam tahapan utama, yaitu *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modeling*, *Evaluation*, dan *Deployment* [17]. Metode CRISP-DM bertujuan untuk menggali serta menemukan pola-pola yang bermakna dan relevan dari data yang dianalisis [18].



Gambar 2. Tahapan CRISP-DM [3]

Keunggulan utama dari CRISP-DM terletak pada sifatnya yang iteratif atau berulang, sehingga memungkinkan peneliti untuk kembali ke tahap

Jurnal Manajemen Informatika & Sistem Informasi (MISI)
Volume 9, Nomor 1, Januari 2026.
ISSN 2614-1701 (media cetak)
ISSN 2614-3739 (media online)
DOI : 10.36595/misi.v5i2
http://e-journal.stmiklombok.ac.id/index.php/misi

sebelumnya apabila hasil belum sesuai harapan. Selain itu, kerangka kerja ini juga bersifat fleksibel sehingga dapat diterapkan pada berbagai jenis permasalahan data, khususnya dalam bidang *data mining*.

3. METODOLOGI PENELITIAN

Penelitian menggunakan Metodologi CRISP-DM (*Cross Industry Standard Process for Data Mining*) terdiri dari enam langkah utama yang saling berhubungan. Pertama, *Business Understanding* yang berfokus pada identifikasi kebutuhan, tujuan, dan perumusan masalah penelitian. Kedua, *Data Understanding* dilakukan dengan mengumpulkan data serta mengevaluasi kualitas dan karakteristiknya. Ketiga, *Data Preparation* mencakup proses pembersihan, pemilihan, hingga transformasi data agar siap dipakai dalam tahap selanjutnya. Keempat, *Modeling* yaitu pembangunan model menggunakan algoritma yang sesuai, misalnya untuk melakukan klasifikasi teks. Setelah itu, tahap *Evaluation* digunakan untuk menilai performa model dengan metrik evaluasi seperti akurasi, presisi, recall, dan F1-score, guna memastikan hasil sesuai dengan tujuan penelitian. Terakhir, *Deployment* dilakukan dengan menerapkan model yang sudah teruji ke dalam sistem nyata sehingga dapat digunakan secara praktis oleh pengguna.

3.1. Business Understanding

Pada tahap *Business Understanding*, penelitian ini diawali dari permasalahan yang dihadapi pengguna WhatsApp, yaitu maraknya pesan spam yang sering disalahgunakan untuk penipuan maupun penyebaran tautan berbahaya sehingga berpotensi merugikan pengguna. Untuk menjawab kebutuhan tersebut, penelitian difokuskan pada pengembangan bot WhatsApp yang mampu mendeteksi serta mengklasifikasikan pesan masuk ke dalam kategori spam atau bukan spam. Pemahaman ini kemudian menjadi dasar dalam merumuskan tujuan penelitian, yaitu menciptakan solusi keamanan tambahan yang praktis, efisien, serta mudah digunakan dalam aktivitas komunikasi sehari-hari. Dengan demikian, tahap ini tidak hanya menitikberatkan pada aspek teknis pengembangan sistem, tetapi juga pada nilai kebermanfaatannya bagi pengguna dalam meningkatkan keamanan dan kenyamanan berkomunikasi melalui WhatsApp.

3.2. Data Understanding

Data yang digunakan dalam penelitian ini berasal dari sebuah blog *WordPress* milik Yudi Wibisono, seorang dosen di Prodi Ilkom UPI yang berfokus pada bidang *Natural Language Processing* (NLP) serta penerapan *machine learning* pada bursa saham (*quantitative trading*), dan juga berperan sebagai konsultan NLP dan *machine learning* di beberapa perusahaan. Dataset awal berjumlah 1.143 pesan yang terbagi ke dalam tiga kategori, yaitu pesan normal sebanyak 569 (label 0), pesan penipuan sebanyak 335 (label 1), dan pesan promosi sebanyak 239 (label 2). Untuk memperkaya data, ditambahkan kumpulan pesan dari arsip pribadi berupa SMS dan WhatsApp, serta pesan dari beberapa rekan dengan izin mereka. Perluasan ini bertujuan untuk meningkatkan kualitas model dan kinerja sistem. Setelah proses penambahan data, total dataset menjadi 1.320 pesan, terdiri dari 566 pesan normal (43%), 459 pesan penipuan (34%), dan 293 pesan promosi (23%).

3.3. Data Preparation

Sebelum digunakan pada tahap penelitian, data terlebih dahulu melalui pemrosesan sederhana untuk memastikan data yang lebih bersih dan seragam sehingga mendukung peningkatan akurasi serta kinerja model. Tahapan yang akan dilakukan adalah *case folding*, *punctuation removal*, *Character Filtering*, *whitespace normalization*, dan *normalisasi*.

3.4. Modeling

Pada tahap *Modeling*, penelitian ini menggunakan algoritma SVM (*Support Vector Machine*) sebagai metode klasifikasi utama karena kemampuannya menangani data non-linier, efektif pada data berdimensi tinggi, dan tetap andal meski jumlah data relatif kecil. Untuk mendapatkan performa terbaik, dilakukan pencarian kombinasi *hyperparameter*

Jurnal Manajemen Informatika & Sistem Informasi (MISI)
Volume 9, Nomor 1, Januari 2026.
ISSN 2614-1701 (media cetak)
ISSN 2614-3739 (media online)
DOI : 10.36595/misi.v5i2
http://e-journal.stmiklombok.ac.id/index.php/misi

menggunakan *Grid Search*, yang menghasilkan pilihan kernel *linear* dengan parameter **C**. Kernel *linear* dipilih karena mampu menentukan pemisah terbaik tanpa perlu transformasi ke ruang berdimensi tinggi, sedangkan parameter C berperan dalam mengatur keseimbangan antara kompleksitas model dan toleransi kesalahan. Nilai C yang tinggi berpotensi meningkatkan akurasi namun rawan *overfitting*, sementara nilai C yang rendah lebih toleran terhadap kesalahan tetapi bisa menyebabkan *underfitting*.

Table 1. Hasil Dari Grid Search

No	Kernel	C	Gamma	F1-Weight
1	Linear	0.1	-	0.842223
2	Linear	1	-	0.949017
3	Linear	10	-	0.948983
4	RBF	100	-	0.948983
5	RBF	0.1	Scale	0.624904
6	RBF	1	Scale	0.613929
7	RBF	10	Scale	0.741670
8	RBF	100	Scale	0.751965

Penggunaan F1-score (*weighted*) sebagai metrik utama dalam *Grid Search* dipilih karena dataset memiliki distribusi kelas yang tidak seimbang, di mana pesan normal lebih banyak dibandingkan pesan penipuan dan promosi. Jika hanya mengandalkan akurasi, hasil model bisa bias terhadap kelas mayoritas meskipun kinerjanya pada kelas minoritas rendah. F1-score lebih adil karena menggabungkan *precision* dan *recall*, serta dengan *weighted average* setiap kelas tetap berkontribusi sesuai proporsi datanya. Dengan demikian, pemilihan parameter terbaik menggunakan F1-score mampu menggambarkan performa model secara lebih representatif pada kondisi data tidak seimbang.

3.5. Evaluation

Pada tahap evaluasi, kinerja model dianalisis menggunakan **Confusion Matrix**, yaitu sebuah tabel yang menggambarkan hasil prediksi model dengan membandingkannya terhadap label sebenarnya. Matriks ini terdiri dari empat variabel utama, yakni *True Positive* (TP) yang menunjukkan jumlah data positif yang diprediksi benar, *True Negative* (TN) untuk data negatif yang diprediksi benar, *False Positive* (FP) untuk data negatif yang salah diprediksi positif, serta *False Negative* (FN) untuk data positif yang salah diprediksi negatif.

		Predicted Class	
		(1) Positive	(0) Negative
Actual Class	(1) Positive	TP (True Positive)	FN (False Negative)
	(0) Negative	FP (False Positive)	TN (True Negative)

Gambar 3. Confusion Matrix

Keempat matriks tersebut dapat dihitung menggunakan rumus berikut:

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$\text{Presisi} = \frac{TP}{TP+FP} \quad (2)$$

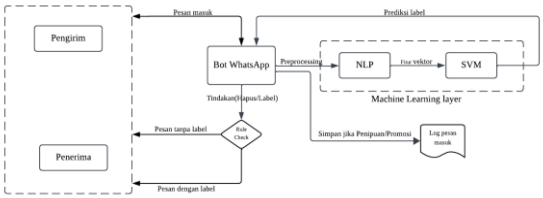
$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

$$\text{F1-Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

3.6. Deployment

Pada tahapan ini, sistem dirancang dengan dua komponen utama, yaitu model klasifikasi pesan spam berbasis *Support Vector Machine* (SVM) yang diimplementasikan melalui Flask API, serta bot WhatsApp yang dikembangkan menggunakan *library* whatsapp-web.js. Sistem ini dijalankan secara lokal pada dua lingkungan berbeda, yakni Python dan Node.js, yang saling terhubung melalui integrasi komunikasi menggunakan HTTP API.

Gambaran secara umum dari proses deteksi pesan spam yang akan dilakukan bisa dilihat pada gambar dibawah ini:



Gambar 4. Proses Secara Umum Deteksi Spam

Jurnal Manajemen Informatika & Sistem Informasi (MISI)
Volume 9, Nomor 1, Januari 2026.
ISSN 2614-1701 (media cetak)
ISSN 2614-3739 (media online)
DOI : 10.36595/misi.v5i2
http://e-journal.stmiklombok.ac.id/index.php/misi

1. Pesan Masuk

Pengirim mengirimkan pesan melalui WhatsApp, dan pesan tersebut diterima oleh sistem Bot WhatsApp.

2. Processing

Pesan yang diterima diproses terlebih dahulu untuk dilakukan pembersihan, *vectorizer*, dan lainnya.

3. Kalsifikasi

Hasil dari *processing* berupa vektor fitur dikirim ke model SVM yang telah dilatih untuk melakukan klasifikasi pesan ke dalam tiga kategori: Normal, Penipuan dan Promosi.

4. Prediksi Label

SVM mengembalikan label hasil klasifikasi ke Bot WhatsApp.

5. Rule Check:
   - Jika Normal, maka pesan akan diteruskan ke penerima tanpa label.
   - Jika Promosi atau Penipuan, maka bot akan menambahkan labael peringatan pada pesan sebelum diteruskan ke penerima, atau dapat dihapus tergantung tindakan yang diberikan.

6. Log Pesan

Jika pesan diklasifikasikan sebagai Penipuan atau Promosi, maka isi pesan dan labelnya disimpan ke dalam file log (CSV) untuk keperluan pelatihan ulang model dengan data baru.

7. Pesan Terkirim ke Penerima

Pesan akhir, baik berlabel atau tidak akan diteruskan ke penerima sesuai hasil dan aturan dari tindakan yang diberikan

Struktur direktori terbagi menjadi beberapa folder dan file utama yang dapat dilihat pada gambar dibawah ini:

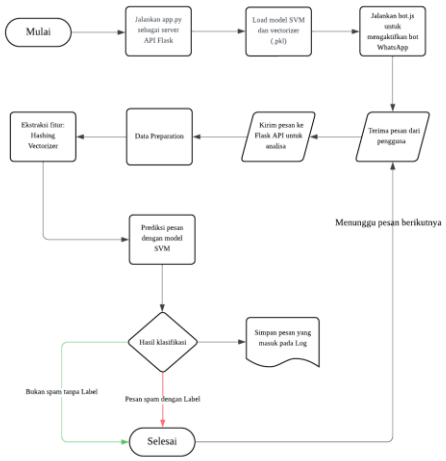
```
.
├── BOTWA_project/
│   ├── APIpy/
│   │   ├── data/
│   │   │   ├── dataset
│   │   │   └── kamus_normalisasi
│   │   ├── model/
│   │   │   ├── svm_model_with_extra_features.pkl
│   │   │   └── tfid_vectorizer_with_extra_features.pkl
│   │   ├── utils/
│   │   │   ├── ekstraksi_fitur.py
│   │   │   ├── processing.py
│   │   │   └── training.py
│   │   ├── app.py
│   │   └── requirements.txt
│   └── main/
│       ├── logs/
│       │   └── log_pesan.csv
│       └── bot.js
├── node_modules
├── package-lock.json
└── package.json
```

Gambar 5. Direktori Folder Proyek

Struktur pada gambar tersebut terdiri dari dua bagian utama, yaitu **APIpy/** yang berfungsi sebagai backend berbasis Python, dan **main/** yang digunakan untuk bot WhatsApp berbasis Node.js. Pada folder **APIpy/** terdapat subfolder **data/** yang berisi dataset serta kamus normalisasi, **model/** yang menyimpan hasil pelatihan berupa file model dan vectorizer, serta **utils/** yang memuat modul-modul pendukung seperti *training.py*, *processing.py*, dan *ekstraksi_fitur.py*. File inti **app.py** digunakan untuk menjalankan API dengan Flask, sementara **requirements.txt** berisi daftar dependensi Python yang diperlukan.

Di sisi lain, folder **main/** memuat file **bot.js** sebagai program utama bot WhatsApp serta subfolder **logs/** yang menyimpan hasil klasifikasi pesan dalam bentuk file CSV. Untuk kebutuhan Node.js, konfigurasi dependensi terdapat pada **package.json**, sedangkan semua dependensi yang terpasang tersimpan di dalam **node_modules/**. Dengan susunan ini, sistem mampu mendukung alur kerja mulai dari pelatihan model, penyimpanan, integrasi dengan API, hingga implementasi dan pengujian langsung melalui bot WhatsApp.

Berikut merupakan alur dari sistem deteksi pesan spam yang dapat dilihat pada gambar dibawah ini:



Gambar 6. Alur Dari Sistem Deteksi Pesan Spam

Alur proses pada gambar tersebut dimulai dengan menjalankan app.py sebagai server Flask yang berperan sebagai backend sistem. Setelah itu, model SVM beserta *Hashing Vectorizer* yang telah dilatih sebelumnya dimuat dari file berformat *.pkl*. Selanjutnya, file bot.js dijalankan untuk mengaktifkan bot WhatsApp sehingga sistem siap menerima pesan dari pengguna. Setiap pesan yang

Jurnal Manajemen Informatika & Sistem Informasi (MISI)
Volume 9, Nomor 1, Januari 2026.
ISSN 2614-1701 (media cetak)
ISSN 2614-3739 (media online)
DOI : 10.36595/misi.v5i2
http://e-journal.stmiklombok.ac.id/index.php/misi

masuk kemudian dikirim ke Flask API untuk dianalisis melalui tahap *data preparation*, yang meliputi case folding, penghapusan tanda baca, penyaringan karakter, normalisasi spasi, serta normalisasi teks guna memastikan data lebih bersih. Teks yang telah diproses selanjutnya diekstraksi fiturnya menggunakan *Hashing Vectorizer*, lalu diprediksi oleh model SVM untuk menentukan kategori pesan, yaitu normal, penipuan, atau promosi. Berdasarkan hasil klasifikasi, pesan normal diteruskan tanpa label, sementara pesan yang terindikasi spam akan diberi label khusus dan dicatat ke dalam log sistem untuk kebutuhan evaluasi maupun pelatihan ulang model. Setelah satu siklus selesai, sistem kembali dalam kondisi menunggu pesan berikutnya, sehingga keseluruhan proses berjalan secara otomatis, *real-time*, dan berkelanjutan.

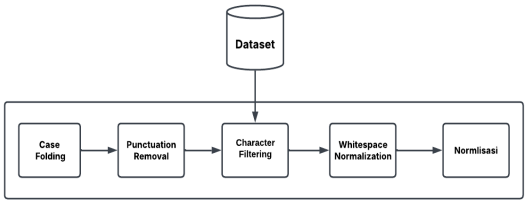
4. HASIL DAN PEMBAHASAN

4.1. Data Preparation

Tahapan dalam data preparation yang terdiri dari 5 proses yaitu *case folding*, *punctuation removal*, *character filtering*, *whitespace normalization*, dan *normalisasi*.

Input: "Penawaran Spesial, Obral Paket Data 1.5 GB(00-24)+nelpon sepuasnya ke nmr Indosat hanya Rp20rb utk 1minggu.Daftar skrg ketik YA ke 929 berlaku hari ini RTX1"

Processing:



Gambar 7. Tahapan Dalam Data Preparation

Table 2. Case Folding

Input	Output
Penawaran Spesial, Obral Paket Data 1.5 GB(00-24)+nelpon sepuasnya ke nmr Indosat hanya Rp20rb utk 1minggu.Daftar skrg ketik YA ke 929 berlaku hari ini RTX1	"penawaran spesial, obral paket data 1.5 gb(00-24)+nelpon sepuasnya ke nmr indosat hanya rp20rb utk 1 minggu. daftar skrg ketik ya ke 929 berlaku hari ini rtx1"

Table 3. Punctuation Removal

Input	Output
penawaran spesial, obral paket data 1.5 gb(00-24)+nelpon sepuasnya ke nmr indosat hanya rp20rb utk 1 minggu. daftar skrg ketik ya ke 929 berlaku hari ini rtx1	"penawaran spesial obral paket data 1 5 gb(00-24)+nelpon sepuasnya ke nmr indosat hanya rp20rb utk 1 minggu daftar skrg ketik ya ke 929 berlaku hari ini rtx1"

Table 4. Character Filtering

Input	Output
penawaran spesial obral paket data 1 5 gb(00-24)+nelpon sepuasnya ke nmr indosat hanya rp20rb utk 1 minggu. daftar skrg ketik ya ke 929 berlaku hari ini rtx1	"penawaran spesial obral paket data 1 5 gb 00 24 nelpon sepuasnya ke nmr indosat hanya rp20rb utk 1 minggu. daftar skrg ketik ya ke 929 berlaku hari ini rtx1"

Table 5. Whitespace Normalization

Input	Output
penawaran spesial obral paket data 1 5 gb 00 24 nelpon sepuasnya ke nmr indosat hanya rp20rb utk 1 minggu daftar skrg ketik ya ke 929 berlaku hari ini rtx1	"penawaran spesial obral paket data 1 5 gb 00 24 nelpon sepuasnya ke nmr indosat hanya rp20rb utk 1 minggu daftar skrg ketik ya ke 929 berlaku hari ini rtx1"

Table 6. Normalisasi

Input	Output
penawaran spesial obral paket data 1 5 gb 00 24 nelpon sepuasnya ke nmr indosat hanya rp20rb utk 1 minggu daftar skrg ketik ya ke 929 berlaku hari ini rtx1	"penawaran spesial obral paket data 1 5 gb 00 24 nelpon sepuas ke nomor indosat hanya rp20ribu untuk 1 minggu daftar sekarang ketik ya ke 929 berlaku hari ini rtx1"

Jurnal Manajemen Informatika & Sistem Informasi (MISI)
Volume 9, Nomor 1, Januari 2026.
ISSN 2614-1701 (media cetak)
ISSN 2614-3739 (media online)
DOI : 10.36595/misi.v5i2
http://e-journal.stmiklombok.ac.id/index.php/misi

Tahapan *normalisasi* diatas menggunakan kamus normalisasi yang dibuat manual hasil dari *review* dataset yang digunakan.

Kode dari proses *data preparation* dapat dilihat pada gambar dibawah ini:

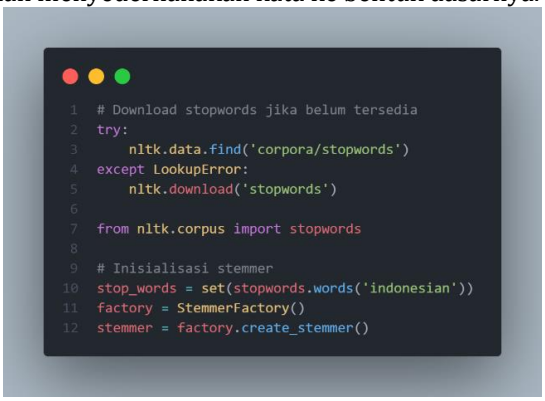
Gambar 8. Kode Tahapan Data Preparation

4.2. Modeling

Pada tahap ini, seluruh kebutuhan untuk membangun model dipersiapkan dengan melakukan instalasi dan import berbagai pustaka, mulai dari core machine learning, pengolahan data, preprocessing teks, hingga visualisasi dan komponen pendukung lainnya.

Inisialisasi Stemmer dan Stopword

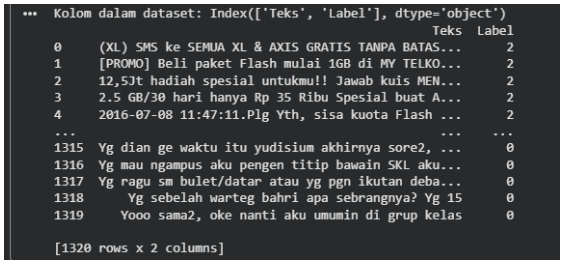
Menyiapkan *stopwords* bahasa Indonesia dan menginisialisasi *stemmer* dari Sastrawi yang digunakan untuk menghapus kata tidak penting dan menyederhanakan kata ke bentuk dasarnya.



Gambar 9. Kode Inisialisasi Stemmer dan Stopword

Load Dataset

Load dataset yang sudah disiapkan dan berikut ini adalah tampilan dari dataset yang digunakan.



Gambar 10. Isi dan Jumlah Dataset

Load Kamus Normalisasi

Kamus normalisasi yang dipakai disusun secara mandiri dengan mengumpulkan kata-kata tidak baku dari dataset, seperti typo dan singkatan. Data tersebut awalnya disimpan dalam file Excel, lalu dikonversi menjadi *dictionary* agar dapat digunakan dalam proses normalisasi teks pada tahap prapemrosesan.

Gambar 11. Kode Dan Isi Kamus Normalisasi

Training Data

Membagi dataset menjadi dua bagian yaitu *data training* dan *data testing*. *Data Training* digunakan untuk melatih model dan *data testing* digunakan untuk melihat sebaik apa kinerja model.

Training dan *testing* dibagi menjadi 2 tahap, yaitu **70:30** dan **80:20**.

Training & Testing 70:30

Jumlah data *training* 924/70% dari 1320 total data.
Jumlah data *testing* 396/30% dari 1320 total data.

Training & Testing 80:20

Jumlah data *testing* 1056/80% dari 1320 total data.
Jumlah data *training* 264/20% dari 1320 total data.

Processing Teks

Contoh hasil dari *pre-processing* teks dapat dilihat pada gambar dibawah ini:

Jurnal Manajemen Informatika & Sistem Informasi (MISI)
Volume 9, Nomor 1, Januari 2026.
ISSN 2614-1701 (media cetak)
ISSN 2614-3739 (media online)
DOI : 10.36595/misi.v5i2
http://e-journal.stmiklombok.ac.id/index.php/misi

Gambar 12. Hasil Processing Teks

4.3. Evaluation

Setelah model selesai dilatih, langkah berikutnya adalah mengevaluasi performa model klasifikasi untuk mengetahui seberapa baik kemampuannya dalam membedakan kategori pesan. Proses evaluasi ini menggunakan data uji yang diperoleh dari pembagian dataset dengan proporsi 80:20 karena lebih stabil.

Tabel 7. Hasil *Confusion Matrix* rasio 80:20

Akurasi	0.9545454545454546
Presisi	0.9550237154571831
Recall	0.9545454545454546
F1-Score	0.9546071070038011

Berikut hasil evaluasi dari training model:

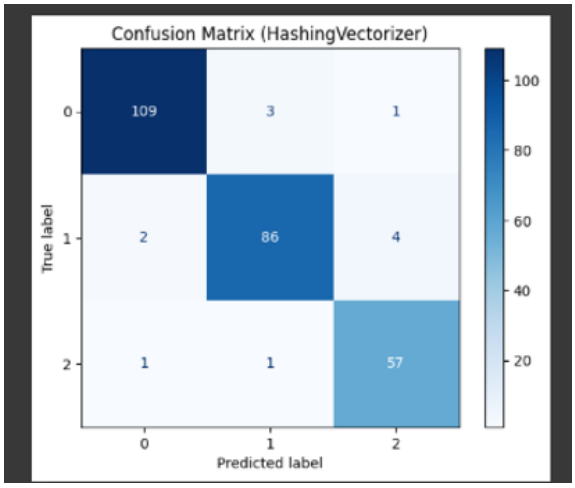
```
Kolom dalam dataset: Index(['Teks', 'Label'], dtype='object')
Akurasi: 0.9545454545454546
Presisi: 0.9550237154571831
Recall: 0.9545454545454546
F1 Score: 0.9546071070038011

Laporan Klasifikasi:
              precision    recall  f1-score   support

           0       0.97      0.96      0.97       113
           1       0.96      0.93      0.95        92
           2       0.92      0.97      0.94        59

    accuracy                           0.95       264
   macro avg       0.95      0.96      0.95       264
weighted avg       0.96      0.95      0.95       264
```

Gambar 13. Hasil Evaluasi



Gambar 14. Hasil Dalam Bentuk Confussion Matrix

4.4. Deployment

Pada tahap deployment, sistem yang telah dibangun diimplementasikan agar dapat berjalan secara nyata. Model klasifikasi pesan spam berbasis SVM diintegrasikan dengan Flask API sebagai *backend* dan dihubungkan dengan WhatsApp bot menggunakan *library whatsapp-web.js*. Proses ini memungkinkan setiap pesan yang masuk ke WhatsApp dikirim ke server untuk diproses melalui tahapan *preprocessing* teks, diubah menjadi representasi numerik dengan *Hashing Vectorizer*, lalu diprediksi oleh model. Hasil klasifikasi selanjutnya ditindaklanjuti, misalnya pesan normal diteruskan sedangkan pesan spam (Penipuan atau Promosi) diberi label peringatan dan disimpan dalam *log*. Dengan demikian, sistem dapat bekerja secara otomatis, real-time, dan berkelanjutan dalam mendeteksi spam di WhatsApp.

Berikut langkah dari implementasi model dan integrai bot WhatsApp:

1. **Import Library:**

   from flask import Flask, request, jsonify
   Import joblib
   Import numpy as np

   Kode diatas digunakan untuk membuat API server, menanganai permintaan HTTP, mengirim respon dalam format JSON, memuat model *machine learning* dalam format *.pkl* hasil dari proses pelatihan, dan manipulasi array numerik

Jurnal Manajemen Informatika & Sistem Informasi (MISI)
Volume 9, Nomor 1, Januari 2026.
ISSN 2614-1701 (media cetak)
ISSN 2614-3739 (media online)
DOI : 10.36595/misi.v5i2
http://e-journal.stmiklombok.ac.id/index.php/misi

Load Model:
model = joblib.load('model/v9/hashsing_vextorizer.pkl')
vecrotizer = joblib.load(model/v9/svm_model.pkl)
Kode diatas untuk memuat model dan vectorier hasil dari pelatihan untuk digunakan saat API menerima permintaan prediksi.
Endpoint Prediksi:
@app.route('/predict', methods=[POST])
def predict():
Kode pada route untuk menentukan ednpoint yanag hanya menerima metode POST yang digunakan oleh bot WhatsApp untuk mengirim pesana pengguna dan mendapatkan prediksi klasifikasi, kemudian pada predict yang didalamnya terdiri dari beberap bagian, yaitu:
data = request.json
text = data.get('message', '')
Digunakan untuk mengambil data JSON dari permintaan dan ekstraksi dari *field message*.
hash_features = vectorizer.transform([processed])
Digunakan untuk mengubah teks menjadi fitur numerik yang hasilnya berupa *sparse matrix* dengan ukuran (1 x 5000) → 1 dokumen, 5000 fitur.
hash_array = hash_features.toaarray()
Digunakan untuk meruba *sparse matrix* menjadi *dense array*, yang dimana model SVM membutuhkan input dalam bentuk numerik.
return jsonify({})
Digunakan untuk mengirim hasil predikssi ke bot WhatsApp dalam bentuk JSON.

2. **Import Library WhatsApp Bot:**
const {Client} = require('whatsapp-web.js')
const qrcode = require('qrcode-terminal')
const axios = require('axios')
const fs = require('fs')
const path = require('path')
kode diatas digunakan untuk membuat dan mengontrol bot WhatsApp Web, membuat dan menampilkan QR Code ke terminla, *library* HTTP untuk *request* ke API FLASK, untuk membaca, menulis dan membuatt log file pesan masuk.

Setiap ada pesan masuk, kode dibawah ini akan dipicu:

Gambar 15. Fungsi menerima pesan

Kode pada Gambar diatas untuk mengirim pesan ke API FLASK untuk klasifikasi, jika hasil klsifikasi spam (penipuan atau promosi) pesan disimpan ke dalam folder log, setelah itu bot akan mengirim pesan balasan sesuai jenis pesan spam yang diterima, jika pesan yang diterima merupakan pesan normal (bukan spam), pesan tersebut akan terikirm tanpa balasan atau label apapun dan prosesnya akan berjalan seperti biasa.

4.5. Hasil Implementasi

Berikut adalah cara kerja sistem dalam mendeteksi pesan yang masuk dari pengirim, jika pesan yang masuk dikategorikan sebagai pesan spam "**Penipuan**" maka bot tersebut secara otomatis akan memberikan peringatan berupa *replay* pesan dengan peringatan "**Pesan ini terindikasi sebagai *PENIPUAN*! Harap berhati-hati!**", begitu juga dengan pesan yang terindikasi sebagai pesan "**Promosi**" maka bot tersebut akan *mereplay* pesan dengan peringatan "**Pesan ini terindikias sebagai pesan *Promosi***", dan pesan tersebut akan dimasukkan ke dalam folder "log" untuk digunakan sebagai bahan untuk pelatihan ulang sistem.

Jurnal Manajemen Informatika & Sistem Informasi (MISI)
Volume 9, Nomor 1, Januari 2026.
ISSN 2614-1701 (media cetak)
ISSN 2614-3739 (media online)
DOI : 10.36595/misi.v5i2
http://e-journal.stmiklombok.ac.id/index.php/misi

Table 7. Tindakan Bot

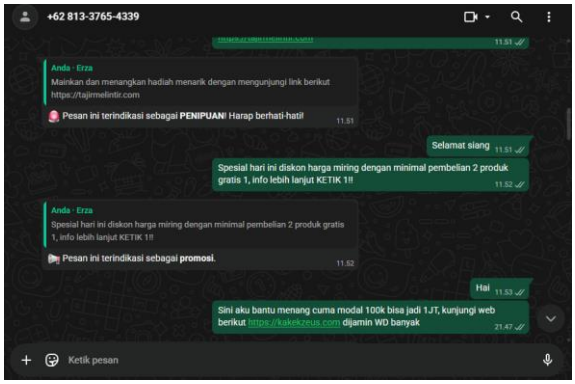
No	Isi pesan	Label	Tindakan
1	5 menit lagi belanja dan makan diskon puas s/d 50%, dengan applykartu kredit Bank Mega sekarang di sini myads.id/7tfPe	2	Pesan ini terindikias sebagai pesan *Promosi*
2	Ini gua agus temen SMA dulu, pinjam dulu 100k butuh banget ni besok gua ganti. TF ke no ini 087888765433	1	Pesan ini terindikasi sebagai *PENIPUAN*! Harap berhati-hati!
3	Jadi besok kita pergi belanja?	0	-

1. Analisa Kinerja Hasil

Dalam pengujian, bot mampu memberikan peringatan langsung pada pesan yang terindikasi berbahaya, misalnya pesan dengan tautan mencurigakan atau ajakan promosi. Pesan tersebut secara otomatis diberi label, seperti "**Pesan ini terindikasi sebagai promosi**" atau "**Pesan ini terindikasi sebagai PENIPUAN! Harap berhati-hati!**", sehingga pengguna dapat lebih waspada. Dilengkapi dengan fitur pencatatan (*logging system*) yang menyimpan pesan terklasifikasi spam untuk keperluan analisis maupun pelatihan ulang model.

2. Hasil

Gambar 16. Hasil pada Terminal



Gambar 17. Hasil pada WhatsApp

Berdasarkan hasil prediksi yang ditampilkan pada terminal dan aplikasi WhatsApp, dapat disimpulkan bahwa sistem bot berhasil menjalankan fungsinya dengan baik dalam mendeteksi pesan spam. Pesan yang dikategorikan sebagai promosi maupun penipuan ditandai secara jelas dengan label peringatan, sementara pesan normal tidak diberi label tambahan. Hasil prediksi memberikan informasi real-time berupa peringatan seperti "Pesan ini terindikasi sebagai promosi" atau "Pesan ini terindikasi sebagai PENIPUAN! Harap berhati-hati!". Hal ini menunjukkan bahwa integrasi antara model klasifikasi SVM dengan bot WhatsApp melalui Flask API berjalan efektif, serta mampu meningkatkan keamanan pengguna dengan memberikan deteksi dini terhadap potensi pesan berbahaya.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian, sistem deteksi pesan spam pada WhatsApp menggunakan algoritma *Support Vector Machine* (SVM) dengan kernel *linear* berhasil mencapai akurasi, presisi, recall, dan f1-score rata-rata sebesar 95%. Hal ini menunjukkan bahwa model mampu mengklasifikasikan pesan ke dalam kategori Normal, Penipuan, dan Promosi dengan tingkat ketepatan yang tinggi, sesuai dengan tujuan penelitian.

Keunggulan penelitian ini terletak pada integrasi model ke dalam bot WhatsApp yang dapat memberikan prediksi pesan secara langsung. Namun demikian, masih terdapat kelemahan berupa kesalahan klasifikasi pada sebagian pesan yang memiliki kemiripan kata antar kategori, sehingga perlu pengembangan lebih lanjut dengan memperbesar variasi dataset dan metode pra-pemrosesan yang lebih mendalam.

Jurnal Manajemen Informatika & Sistem Informasi (MISI)
Volume 9, Nomor 1, Januari 2026.
ISSN 2614-1701 (media cetak)
ISSN 2614-3739 (media online)
DOI : 10.36595/misi.v5i2
http://e-journal.stmiklombok.ac.id/index.php/misi

Untuk pengembangan penelitian ke depan, terdapat beberapa saran dan hal yang dapat dilakukan:

1. **Perluasan dataset** dengan jumlah data yang lebih besar dan distribusi yang seimbang antar kategori, sehingga model dapat belajar lebih optimal dan hasil klasifikasi menjadi lebih representatif.
2. **Eksperimen dengan Algoritma Lain** – Selain SVM, penelitian selanjutnya bisa membandingkan performa dengan algoritma lain seperti Random Forest, Naïve Bayes, atau model berbasis Deep Learning (misalnya LSTM atau Transformer).
3. **Pengayaan Fitur** – Fitur yang digunakan dapat diperluas, misalnya dengan analisis semantic (word embeddings seperti Word2Vec, FastText, atau BERT) agar sistem mampu menangkap konteks makna pesan lebih dalam.
4. **Peningkatan Evaluasi** – Selain akurasi, presisi, recall, dan F1-score, pengujian juga bisa ditambah dengan metrik lain seperti ROC-AUC, confusion matrix per kelas yang lebih detail, atau analisis kesalahan (error analysis) agar kelemahan sistem lebih mudah diidentifikasi.
5. **Integrasi dengan Sistem Anti-Spam Lain** – Bot dapat dikembangkan agar terhubung dengan sistem anti-spam eksternal atau basis data spam global untuk meningkatkan kemampuan deteksi terhadap pola serangan baru yang belum ada dalam dataset.
6. **Fitur Umpan Balik dari Pengguna** – Menambahkan mekanisme *feedback loop* dari pengguna untuk mengonfirmasi apakah pesan yang terdeteksi benar-benar spam atau bukan, sehingga data umpan balik ini dapat digunakan untuk melatih ulang model agar lebih adaptif dan akurat.
7. **Keamanan dan Privasi** – Penelitian selanjutnya dapat mempertimbangkan aspek keamanan dan privasi data pengguna, misalnya dengan menerapkan enkripsi pada pesan yang dianalisis atau mekanisme *anonymization* sehingga sistem tetap mematuhi standar perlindungan data.

6. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Universitas, dosen pembimbing, serta rekan-rekan yang telah memberikan dukungan, bimbingan, dan bantuan data sehingga penelitian ini dapat terselesaikan dengan baik.

DAFTAR PUSTAKA:

[1] A. Nur, R. Hasanah, R. A. Krestianti, and S. Wati, "Implementasi Algoritma Regresi Logistik untuk Binary Classification dalam Spam SMS dan WhatsApp," *Agustus*, vol. 7, pp. 2549–7952, 2023, [Online]. Available: https://proceeding.unpkediri.ac.id/index.php/inotex/

[2] K. Manajemen *et al.*, "Analisis Ancaman Pishing Melalui Aplikasi WhatsApp: Studi," vol. 1, no. 10, 2024, doi: 10.32672/jnkti.v7i3.7551.

[3] M. Arif Sofyan, N. Rahaningsih, and R. Danar Dana, "Deteksi Sms Spam Berbahasa Indonesia Menggunakan Algoritma Support Vector Machine," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 3, pp. 3071–3079, 2024, doi: 10.36040/jati.v8i3.9532.

[4] G. Sanhaji, J. Julian, and H. Syah, "WFraud Alert Sebagai Prediksi Pesan Penipuan WhatsApp Menggunakan Naïve Bayes," *J. Tekno Kompak*, vol. 18, no. 1, p. 113, 2024, doi: 10.33365/jtk.v18i1.3523.

[5] F. R. Suprihati, "Analisis Klasifikasi SMS Spam Menggunakan Logistic Regression," *J. Sist. Cerdas*, vol. 4, no. 3, pp. 155–160, 2021, doi: 10.37396/jsc.v4i3.166.

[6] R. Dwiyansaputra, G. S. Nugraha, F. Bimantoro, and A. Aranta, "Deteksi Sms Spam Berbahasa Indonesia Menggunakan Tf-Idf Dan Stochastic Gradient Descent Classifier," *J. Teknol. Informasi, Komput. dan Apl.*, vol. 3, no. 2, pp. 200–207, 2021.

[7] A. Purnama and D. Hamidin, "Metode Algoritma Logistic Regression dalam Klasifikasi Email Spam," *J. Software, Hardw. Inf. Technol.*, vol. 5, no. 1, pp. 39–47, 2025, doi: 10.24252/shift.v5i1.159.

[8] D. Rika Widianita, *KLASIFIKASI MESSAGE SPAM MENGGUNAKAN METODE NEIVE BAYES DAN MULTINOMIAL NAÏVE BAYES* VIII, no. I. 2023.

[9] A. R. Hanum *et al.*, "Analisis Kinerja Algoritma Klasifikasi Teks Bert dalam Mendeteksi Berita Hoaks," *J. Teknol. Inf.*

Jurnal Manajemen Informatika & Sistem Informasi (MISI)
Volume 9, Nomor 1, Januari 2026.
ISSN 2614-1701 (media cetak)
ISSN 2614-3739 (media online)
DOI : 10.36595/misi.v5i2
http://e-journal.stmiklombok.ac.id/index.php/misi

dan Ilmu Komput., vol. 11, no. 3, pp. 537–546, 2024, doi: 10.25126/jtiik.938093.

[10] M. I. U. Rosyidi and N. Rochmawati, "Teknik Bagging Pada Algoritma Klasifikasi Decision Tree dan SVM Untuk Klasifikasi SMS Berbahasa Indonesia," *J. Informatics Comput. Sci.*, vol. 5, no. 02, pp. 265–271, 2023, doi: 10.26740/jinacs.v5n02.p265-271.

[11] I. B. L. SAYIDINA AHMADAL QOSOSYI, "Implementasi bidirectional long-short term memory (bilstm) untuk kelasifikasi sentimen pada kasus pemilihan umum 2024," 2024.

[12] E. A. Widyaningrum, M. F. Fadrian, and W. Admaja, "Pengaruh Pelayanan Informasi Swamedikasi Online Berbasis Whatsapp Bot terhadap Pengetahuan Masyarakat," *Maj. Farmasetika*, vol. 8, no. 3, p. 235, 2023, doi: 10.24198/mfarmasetika.v8i3.43683.

[13] P. A. Alia, R. W. Febriana, J. S. Prayogo, and R. Kriswibowo, "Implementation Chatbot on Whatsapp Using Artificial Intelligence With Natural Language Processing Method," vol. 5, no. 1, pp. 8–14, 2024.

[14] A. Triono, A. Setia Budi, and R. Abdillah, "Agus Triono, Apri Setia Budi, Rafi Abdillah, dan Wahyudi. 'Implementasi Peretasan Sandi Vigenere Cipher Menggunakan Bahasa Pemrograman Python.' Jurnal JOCOTIS - Journal Science Informatica and Robotics, vol. 1, no. 1, September 2023, pp. 1-9. E-ISSN: xxxx," *J. JOCOTIS-Journal Sci. Inform. Robot. E-ISSN xxxx-xxxx*, vol. 1, no. 1, pp. 1–9, 2023, [Online]. Available: https://jurnal.ittc.web.id/index.php/jct/

[15] Sari, "Pemrograman dan bahasa Pemrograman," *STMIK-STIE Mikroskil*, no. June, pp. 1–91, 2021.

[16] J. Tan, Y. Chen, and S. Jiao, *Visual Studio Code in Introductory Computer Science Course: An Experience Report*, vol. 1, no. 1. Association for Computing Machinery, 2024. doi: 10.18260/1-2--48259.

[17] Herlawati, D. B. Srisulistiowati, S. C. Agustin, P. H. Syafina, N. Rachmatin, and S. Setiawati, "Metode Naïve Bayes dan Support Vector Machine untuk Mengolah Sentimen Ulasan dan Komentar di Platform Digital," *J. Students' Res. Comput. Sci.*, vol. 5, no. 2, pp. 197–212, 2024, doi: 10.31599/dby15h32.

[18] F. N. Dhewayani, D. Amelia, D. N. Alifah, B. N. Sari, and M. Jajuli, "Implementasi K-Means Clustering untuk Pengelompokkan Daerah Rawan Bencana Kebakaran Menggunakan Model CRISP-DM," *J. Teknol. dan Inf.*, vol. 12, no. 1, pp. 64–77, 2022, doi: 10.34010/jati.v12i1.6674.