INFOTECH journal
ISSN : 2460-1861 (Print), 2615-4250 (Online)
Vol. 11 No. 2 Desember 2025, pp. 377-385

PENERAPAN ALGORITMA *NAÏVE BAYES* MENGGUNAKAN *PYTHON* UNTUK MENDETEKSI SMS SPAM DAN PROMO

MRamdani[1], Muhamad Rafi Hamdani[2], Ibnu Rizki Prayoga[3], Sri Budhi Lestari[4], Bimo Hakim Prabowo[5], Sigit Wibawa[6], Muhammad Muharrom[7]
[1,2,3,4,5,6,7] *Universitas Bina Sarana Informatika, Bekasi, Indonesia*
Responden: mramdani1230@gmail.com

ABSTRACT
SPAM detection for SMS service is important to prevent small inconveniences, such as wasted time, to something far more dangerous, such as phishing, malware, and fraud. This research uses the Naïve Bayes algorithm, which was implemented with the Python programming language as a classification method to predict whether a SMS message is classified as normal, spam, or advertisement. The dataset this research uses came from the Kaggle platform and will be split into two parts: 80% for training and 20% for testing. The result of testing will be a Confusion Matrix, accuracy, precision, recall, and f1-score, which can then be used to estimate how effective the model is in detecting normal, spam, or advertisement SMS. Based on the result of this research, the Naïve Bayes algorithm shows a good performance with an accuracy of 93% in classifying SMS into normal, spam, and advertisement. This result shows that the Naïve Bayes algorithm is effective in detecting spam and advertisements for SMS service.

Keywords: Naïve Bayes, Python, phishing, malware, Confusion Matrix, accuracy, precision, recall, f1-score

ABSTRAK
Deteksi pesan spam pada layanan SMS merupakan hal penting untuk mencegah mulai dari hal kecil seperti gangguan waktu hingga sesuatu yang berbahaya seperti *phishing, malware*, atau penipuan. Penelitian ini menggunakan algoritma *Naïve Bayes* yang di implementasikan dengan menggunakan bahasa *pemrograman Python* sebagai metode klasifikasi untuk memprediksi apakah sebuah pesan SMS tergolong normal, spam, atau promo. Dataset yang digunakan diperoleh dari platform *Kaggle* dan akan dibagi menjadi dua, yaitu 80% data untuk proses pelatihan (*training*) dan 20% data untuk pengujian (*testing*). Nantinya, hasil dari pengujian data berupa *Confusion Matrix, accuracy, precision, recall, dan f1-score* dapat digunakan untuk mengestimasi sebarapa efektif model untuk mendeteksi SMS normal, spam, ataupun promo. Berdasarkan hasil pengujian, algoritma *Naïve Bayes* menunjukkan performa yang baik dengan tingkat akurasi mencapai 93% dalam menklasifikasikan SMS normal, spam, dan promo. Hasil ini menunjukkan bahwa metode *Naïve Bayes* efektif digunakan dalam mendeteksi pesan spam dan promo pada layanan SMS.

***Kata Kunci**: Naïve Bayes, Python, phishing, malware, Confusion Matrix, accuracy, precision, recall, f1-score*

Riwayat Artikel :

Tanggal diterima : 31-10-2025

Tanggal revisi : 18-11-2025

Tanggal terbit : 01-12-2025

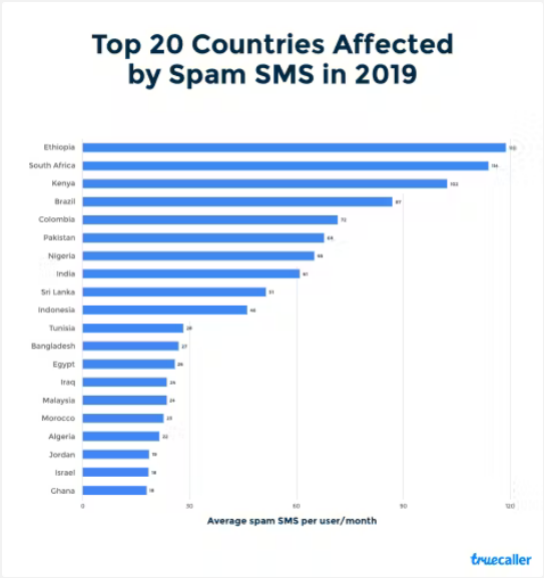
DOI :

https://doi.org/10.31949/infotech.v11i2.16392

1. PENDAHULUAN

Short Message Service (SMS) adalah layanan pada sebuah telepon genggam atau *smartphone* yang digunakan untuk mengirim pesan-pesan singkat (Amazon Web Service, n.d.) dan dapat dikirimkan serta diterima walaupun hanya dengan jaringan operator tanpa koneksi internet. Meski sudah banyak aplikasi berbasis teks yang menggunakan internet, SMS masih sering digunakan untuk mengirim pesan ke seseorang yang sudah dikenal atau mengirim sebuah promosi suatu produk maupun jasa (Ajat, 2023).

Sending and Posting Advertisement in Mass atau yang lebih dikenal dengan istilah SPAM adalah suatu kegiatan mengirim sebuah pesan oleh seseorang atau golongan tertentu dengan tujuan mempromosikan suatu produk atau layanan kepada penerima (Jagoan Hosting, 2023). Spam juga dapat berisi konten berbahaya seperti penipuan (*Scam*), *phishing*, atau bahkan spam yang berisi *malware*. Dalam konteks SMS, spam biasanya dilakukan oleh seseorang untuk mempromosikan suatu barang atau jasa, atau bahkan untuk memancing korban untuk mengunjungi website berbahaya dan mencuri data pribadi korban (*phishing*). Berdasarkan laporan *Truecaller* tahun 2019 (Gambar 1), Indonesia menempati posisi ke-10 sebagai negara dengan jumlah spam SMS terbanyak di dunia (Kim Fai Kok, 2019), pesan tersebut meliputi promosi judi online, penipuan (*scam*), serta *phishing* yang dapat mencuri informasi pribadi korban seperti info perbankan dan kartu kredit.



Gambar 1. Top 20 Countries Affected by Spam SMS in 2019. (Truecaller, 2019)

Oleh karena itu, penerapan *Machine Learning* sangat penting untuk dapat menklasifikasikan apakah sebuah SMS tergolong spam atau bukan. Berdasarkan penelitian sebelumnya, metode *rule-based filtering* telah terbukti tidak efektif untuk melawan taktik spam SMS yang terus berevolusi (Ahmed & Khalid, 2025).

Machine Learning merupakan suatu cabang dari kecerdasan buatan (Artificial Intelligence), *Machine Learning* dapat memberikan sebuah sistem kemampuan untuk belajar dan berkembang secara otomatis tanpa harus di program secara eksplisit (Sarker, 2021). Machine learning dikategorikan menjadi dua, yaitu *supervised learning* dan *unsupervised learning* (DasGupta et al., 2021). Supervised learning adalah proses Machine Learning dengan pengawasan, dimana dataset sudah diberi label sehingga algoritma mengetahui data *input* dan *output*. Sedangkan *unsupervised learning* adalah proses pembelajaran tanpa pengawasan dimana dataset tidak akan diberi label dan algoritma harus menemukan pola dan struktur data sendiri (Nasteski, 2017).

Algoritma *Naïve Bayes,* atau yang biasa juga disebut dengan *Naïve Bayes classifier (NBC),* bekerja dengan *Bayes' Theorem* untuk memprediksi probabilitas sebuah data yang ada di sebuah kelas berdasarkan ciri karakteristik atau fitur yang dimilikinya. *Bayes' Theorem* mangamsumsikan bahwa semua atribut karakteristik adalah independen satu sama lain, sehingga kemunculan suatu fitur tidak bergantung kepada fitur lainnya (Wahyuni et al., 2023). Keunggulan utama *Naïve Bayes* adalah kebutuhan data latih (*training data*) yang relatif sedikit untuk menghasilkan model klasifikasi yang baik. Algoritma *Naïve bayes* menggunakan rumus (1).

$$P\,(A|B) = \frac{P(B|A)\times P(A)}{P(B)} \qquad (1)$$

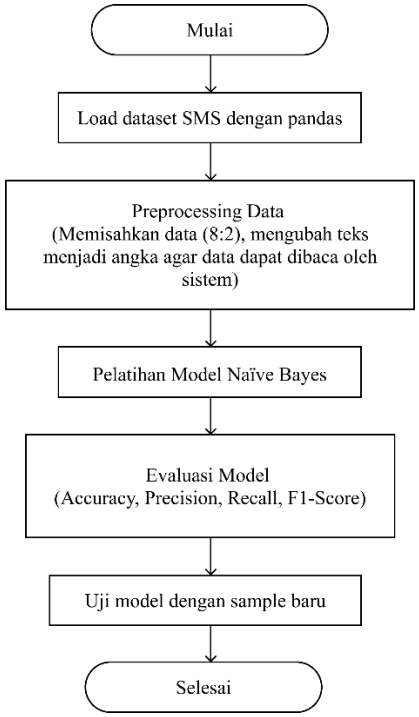
Dimana P(A) dan P(B) adalah probabilitas tanpa memperhatikan satu sama lain, P(A|B) adalah probabilitas A yang bergantung kepada B dan P(B|A) adalah probabilitas B yang bergantung kepada A. Didalam Naïve Bayes Classifier, A adalah hasil klasifikasi dari sebuat data dan B adalah serangkaian prediktor (Zhang, 2016).

Menurut penelitian sebelumnya, algoritma *Naïve Bayes* lebih unggul daripada algoritma lainnya, kecuali *Logistic Regression*. *Naïve Bayes* dapat melampaui beberapa algoritma lain seperti *Keyword-Based Algorithm, Support Vector Machine (SVM), Random Forest,* dan *Decision Tree* dalam konteks tertentu (Androutsopoulos et al., 2000)(Ajat, 2023)(Pranckevičius & Marcinkevičius, 2017). Namun, algoritma machine learning dapat memiliki performa berbeda-beda tergantung dari *task* yang dikerjakan. Pada penelitian sebelumnya juga yang mengukur kemampuan algoritma *Naïve Bayes* pada teks SMS berbahasa inggris dengan data berjumlah 1.364 pesan ham (non-spam), sebanyak 1.336 pesan diklasifikasikan dengan benar sebagai ham sedangkan 28 pesan lainnya salah terdeteksi sebagai spam (Vijay & Kumar, 2021), ini

menunjukan algoritma Naïve Bayes ampuh dalam menjalani *task* yang berbasis teks, sehingga penelitian ini akan menggunakan *Naïve Bayes* untuk menklasifikasi pesan SMS ke normal, spam, dan promo.

2. METODE

2.1. Tahapan Penelitian



Gambar 2. Tahapan Penelitian

Tahapan penelitian ini dilakukan dengan beberapa langkah utama yang ditunjukkan di gambar 2.

a. ***Load* Dataset SMS Menggunakan *Pandas***
Sebelum diolah, dataset berbentuk *CSV* akan dimuat terlebih dahulu menggunakan *library Python* bernama *pandas* agar data dapat dibaca oleh *Python*.

b. ***Preprocessing Data***
Setelah dataset dimuat oleh *Python*, dataset akan dirubah kedalam bentuk matriks agar dapat dipahami oleh model *Machine Learning*. Setelah dirubah, dataset akan di bagi menjadi dua bagian yaitu 80% untuk *training* dan 20% untuk *testing* dengan menggunakan *train_test_split*

c. ***Pelatihan Model Naïve Bayes***
Model *Multinomial Naïve Bayes* yang dimuat oleh *scikit-learn* akan mulai dilatih dengan menggunakan dataset training yang sudah di *split* tadi.

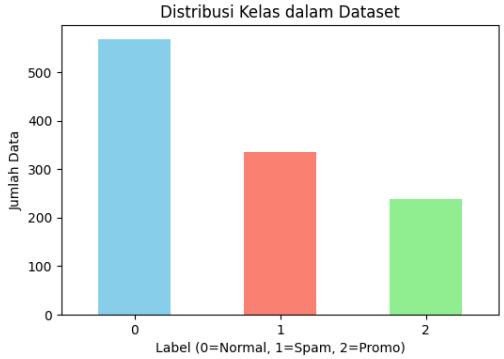
d. ***Evaluasi Model***
Setelah dilatih, model nantinya akan di evaluasi menggunakan dataset testing dan model akan menampilkan nilai *accuracy, precision, recall, dan F1-Score.*

e. ***Uji Model Dengan Sample Baru***
Setelah mendapatkan nilai *accuracy, precision, recall, dan F1-Score*. Model akan diuji untuk menklasifikasikan data baru berjumlah 20 pesan SMS untuk melihat apakah model mampu dengan benar menklasifikasikan data yang berada diluar dataset atau tidak.

2.2. Data Penelitian

a. **Sumber Data**
Data dari penelitian ini bersumber dari platform *Kaggle* dengan judul "Dataset SMS Spam Indonesia" dan dibuat oleh akun *Kaggle* dengan nama Bob Steward dan dapat diakses di: https://www.kaggle.com/datasets/bobsteward/dataset-sms-spam-indonesia (Bob Steward, 2024).

Dataset berjumlah sebanyak 1.143 data dan berisi kumpulan SMS dalam bahasa Indonesia yang telah dikelompokkan ke dalam tiga kelas yaitu 0 (normal), 1 (spam), dan 2 (promo). Distribusi kelas dari dataset tersebut ditunjukkan pada Gambar 3.



Gambar 3. Distribusi Kelas Dalam File "Dataset SMS Spam Indonesia"

Dari gambar tersebut dapat disimpulkan bahwa kelas SMS normal memiliki jumlah data paling banyak yang diikuti oleh spam dan promo.

b. **Bentuk Data**

Dataset berbentuk *CSV (Comma Seperated Value)* yang dapat dibuka dengan *Microsoft Excel* dengan dua kolom utama yaitu teks dan label sesuai contoh Tabel 1

Tabel 1. Contoh Data Dalam File "Dataset SMS Spam Indonesia".

Label	Teks
2	[PROMO] Beli paket Flash mulai 1GB di MY TELKOMSEL APP dpt EXTRA kuota 2GB 4G LTE dan EXTRA nelpon hingga 100mnt/1hr. Buruan, cek di tsel.me/mytsel1 S&K
1	ANDA MAU MENANG TOGEL 100% Tembus pasang shio 7 tunggal angka 07.31.19.43....ingat maharnya pulsa 100rb dikirim ke no 0823...xxx khusus putaran SGP SENIN

1	Segera dptkan Ijazah D3-S1-S2 Asli Terdaftar & Akreditasi Cepat, Kilat Instan Yang Resmi Silahkan Ke: www.buatijazahs1.wordpress.com Atau Hub:085222999475.Tks
0	Ternyata formatnya banyak berantakan dari odp ke google sheet. Saya sdg rapikan sambil buat soal. Besok subuh harap cek lagi ya.

2.3. Preprocessing Data

Sistem di *Python environment* hanya dapat membaca matriks sebuah data dan tidak dapat membaca teks, oleh karena itu data harus melewati tahapan ini agar data yang berbentuk teks dapat diubah kedalam bentuk matriks yang nantinya akan dibaca oleh model. Tahapan yang dilakukan meliputi:

a. *Load* dataset menggunakan *pandas*.
b. Pemilihan kolom teks dan label.
c. Bagi data menjadi *training* dan *testing* menggunakan *train_test_split* dari *scikit-learn* dengan rasio 80% data latih dan 20% data uji.
d. Ubah teks menjadi matriks dengan menggunakan *CountVectorizer* agar dapat dipahami sistem.

Contoh kode *Python* dapat dilihat di Tabel 2.

Tabel 2. Contoh Kode *Python* Untuk *Preprocessing* Data.

Kode	Penjelasan
df = pd.read_csv('dataset_sms_spam_v1.csv', encoding='utf-8')	Kode ini digunakan untuk memuat dataset berbentuk *CSV* kedalam *Python* agar dapat digunakan.
X = df['Teks'] y = df['label']	Kode ini digunakan untuk memisah tabel berdasarkan teks (X) dan label (y).
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)	Kode ini digunakan untuk membagi dataset menjadi *training* dan *testing* dimana test_size=0.2, berarti data *testing* berjumlah 20% dan data *training* 80% dari total data.
vectorizer = CountVectorizer() X_train_counts = vectorizer.fit_transform(X_train)	Kode ini digunakan untuk mengubah teks kedalam bentuk yang dapat dipahami oleh sistem yaitu bentuk matriks.
X_test_counts = vectorizer.transform(X_test)	

2.4. *Preprocessing Data*

Algoritma yang digunakan untuk melatih model adalah *Multinomial Naïve Bayes* karena dapat bekerja dengan baik untuk mendeteksi frekuensi kata. Contoh kode *Python* dapat dilihat di Tabel 3.

Tabel 3. Contoh Kode *Python* Untuk Preprocessing Data.

Kode	Penjelasan
model = MultinomialNB()	Kode ini digunakan untuk memuat model *Multinomial Naïve Bayes.*
model.fit(X_train_counts, y_train)	Kode ini digunakan untuk melatih model.

2.5. Evaluasi Model (*Accuracy, Precision, Recall, F1-Score)*

Evaluasi digunakan untuk mengukur performa model dalam menklasifikasikan SMS. Metode evaluasi yang digunakan meliputi:

a. Akurasi (*Accuracy*): Menunjukkan seberapa sering model memprediksi dengan benar dibandingkan dengan semua prediksi yang dilakukan. Rumus akurasi dapat dilihat di rumus (2).

$$Accuracy = \frac{correct\ classification}{total\ classification} \quad (2)$$

b. *Precision*: Menunjukkan seberapa banyak prediksi positif yang benar-benar positf dan bukan *false-positive.* Rumus *precision* dapat dilihat di rumus (3).

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

c. *Recall*: Menunjukkan seberapa banyak data positif yang dapat ditemukan oleh model. Rumus *recall* dapat dilihat di rumus (4).

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

d. *F1-Score*: Menunjukkan rata-rata harmonik dari *precision* dan *recall* yang dapat memberikan gambaran seimbang antara kedua metrik tersebut. Rumus *F1-Score* dapat dilihat di rumus (5).

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (5)$$

Keterangan:

TP = *True Positive*, FP = *False Positive*, FN = *False Negative*.

2.6. Pengujian Model

Setelah model dilatih, akan dilakukan pengujian terhadap beberapa sampel SMS baru diluar dataset untuk mengetahui apakah model dapat memprediksi pesan SMS dengan benar atau tidak. Contoh kode *Python* dapat dilihat di Tabel 4.

Tabel 4. Contoh Kode *Python* Untuk Pengujian Model.

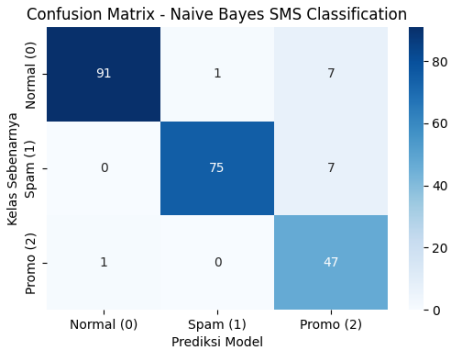
Kode	Penjelasan
sms_samples = ["Pakai XL tdk perlu repot setting bisa langsung internetan.", "GRATIS internetan 5MB berlaku utk 7 hari hanya dengan isi ulang Rp10rb.", "Pinjaman mudah tanpa jaminan. Proses cepat! Hubungi 081xxxxxxx.",]	Kode ini digunakan untuk membuat *array of samples* yang nantinya akan digunakan untuk *testing* model.
sample_counts = vectorizer.transform(sms_samples) prediction = model.predict(sample_counts)	Kode ini pertama akan mengubah *array of samples* tadi menjadi sebuah matriks yang nantinya akan di prediksi oleh model.

3. HASIL DAN PEMBAHASAN

Pada penelitian ini, pengelolaan data SMS menggunakan *Python* sebagai bahasa pemrograman, terminal *VS Code* sebagai *output* dalam bentuk *CLI* (*Command-line Interface*), dan juga menggunakan *library matplotlib* untuk menggambarkan *Confusion Matrix* model. Sebelum melakukan *training*, dataset yang berjumlah sebesar 1.143 akan dibagi menjadi dua bagian yaitu 80% untuk *training* dan 20% untuk *testing*, model juga akan di tes dengan 20 data sampel SMS yang berada diluar dataset.

3.1. Hasil *Confusion Matrix Naïve Bayes*

Confusion Matrix adalah tabel yang digunakan untuk mengevaluasi kinerja model *Machine Learning*, tabel ini membandingkan nilai prediksi model dengan nilai aktual sehingga dapat memberikan gambaran rinci seberapa tepat model menklasifikasikan data. Dari *Confusion Matrix* juga, *accuracy, precision, recall, dan f1-score* dapat dihitung (Adhitya et al., 2023).



Gambar 4. Hasil Confusion Matrix Algoritma Naïve Bayes

Dari Gambar 4 dapat disimpulkan bahwa:

a. Model berhasil menklasifikasikan dengan benar 91 SMS normal (0) dari total 99 SMS Normal .
b. Model berhasil menklasifikasikan dengan benar 75 SMS spam (1) dari total 82 SMS spam.
c. Model berhasil menklasifikasikan dengan benar 47 SMS promo (2) dari total 48 SMS promo.

Kesalahan terbesar model terdapat dalam kemampuannya untuk mendeteksi kelas promo dimana kelas spam dan normal salah diklasifikasikan sebagai promo. Hal ini menunjukkan bahwa model masih kesulitan untuk membedakan antara SMS promo dengan SMS normal dan spam dikarenakan kemiripan fitur atau karakteristik kelas promo dengan normal dan spam.

Secara keseluruhan, model mencapai akurasi sekitar 93% yang berarti jika ada 100 SMS, model dapat menklasifikasikan dengan benar 93 SMS.

3.2. *Accuracy, Precision, Recall, dan F1-Score model*

Evaluasi model dilakukan di *environment Python* untuk mendapatkan nilai *accuracy, precision, recall, dan F1-score*. Hasil evaluasi ditunjukkan pada laporan klasifikasi berikut (Gambar 5).

```
Laporan Klasifikasi:
              precision    recall  f1-score   support

           0       0.99      0.92      0.95        99
           1       0.99      0.91      0.95        82
           2       0.77      0.98      0.86        48

    accuracy                           0.93       229
   macro avg       0.92      0.94      0.92       229
weighted avg       0.94      0.93      0.93       229
```

Gambar 5. Laporan Hasil Klasifikasi Model di Python

Berdasarkan Gambar 5, dapat disimpulkan bahwa kelas normal (0) memiliki rata-rata (*mean*) dari nilai *precision, recall, dan f1-score* sebesar 0.953, yang merupakan nilai tertinggi dibandingkan kelas lainnya. Nilai rata-rata tersebut diperoleh dari hasil perhitungan (0.99 + 0.92 + 0.95)/3. Kelas spam (1) yang memiliki rata-rata hampir mirip dengan normal

sebesar 0.95, sedangkan kelas promo (2) memiliki rata-rata paling rendah sebesar 0.87.

Nilai pada laporan klasifikasi tersebut merupakan hasil pembulatan otomatis oleh sistem *Python*. Pada bagian selanjutnya, akan dilakukan perhitungan manual untuk mendapat nilai *accuracy, precision, recall,* dan *F1-score* yang lebih akurat.

a. **Akurasi *(Accuracy)***

Berdasarkan *Confusion Matrix* Gambar 4, diketahui bahwa total data yang tepat diklasifikasikan oleh model adalah 91 + 75 + 47 = 213 dan total keseluruhan data testing adalah 99 + 82 + 48 = 229. Maka dengan menggunakan rumus yang terdapat di bab 2.6, akurasi dapat dihitung dengan:

$$Accuracy = \frac{213}{229} = 0.93013$$

Demikian, model memiliki akurasi sebesar 0.93013 (93.01%).

b. ***Precision***

Perhitungan *precision* dapat dilakukan dengan menggunakan rumus yang terdapat di bab 2.6. Berdasarkan hasil *Confusion Matrix* yang terdapat di Gambar 4, diketahui bahwa nilai *True Positive* dan *False Positive* masing-masing kelas sebagai berikut (Table 5).

Tabel 5. Nilai *True Positive* dan *False Positive* Setiap Kelas

Kelas	TP	FP
Normal (0)	91	1
Spam (1)	75	1
Promo (2)	47	14

Dari Tabel 5, dapat dihitung nilai *precision* setiap kelas:
Normal:

$$Precision = \frac{91}{91+1}$$

$$Precision = \frac{91}{92}$$

$$Precision = 0.98913$$

Spam:

$$Precision = \frac{75}{75+1}$$

$$Precision = \frac{75}{76}$$

$$Precision = 0.98684$$

Promo:

$$Precision = \frac{47}{47+14}$$

$$Precision = \frac{47}{61}$$

$$Precision = 0.77049$$

Tabel 6. Hasil Perhitungan *Precision* Setiap Kelas

Kelas	Nilai Precision
Normal (0)	0.98913 (98.91%)
Spam (1)	0.98684 (98.68%)
Promo (2)	0.77049 (77.04%)

Bedasarkan Tabel 6, dapat diketahui bahwa kelas normal memiliki *precision* tertinggi sebesar 0.98913 (98.91%) yang diikuti oleh spam sebesar 0.98684 (98.68%). Sedangkan kelas promo memiliki *precision* terendah di 0.77049 (77.04%). Ini artinya model dapat dengan presisi untuk menklasifikasikan SMS spam dan normal, sedangkan model kesulitan untuk kelas promo.

c. ***Recall***

Perhitungan *recall* untuk masing-masing kelas menggunakan rumus yang terdapat di bab 2.6. Dengan menggunakan hasil *Confusion Matrix* di Gambar 4, dapat diketahui bahwa nilai *True Positive* dan *False Negative* masing-masing kelas sesuai dengan (Tabel 7).

Tabel 7. Nilai *True Positive* dan *False Negative* Setiap Kelas

Kelas	TP	FP
Normal (0)	91	8
Spam (1)	75	7
Promo (2)	47	1

Dari Tabel 7, dapat dihitung nilai *recall* setiap kelas sebagai berikut:
Normal:

$$Recall = \frac{91}{91+8}$$

$$Recall = \frac{91}{99}$$

$$Recall = 0.91919$$

Spam:

$$Recall = \frac{75}{75+7}$$

$$Recall = \frac{75}{82}$$

$$Recall = 0.91463$$

Promo:

$$Recall = \frac{47}{47+1}$$

$$Recall = \frac{47}{48}$$

$$Recall = 0.97916$$

Tabel 8. Hasil Perhitungan *Recall* Setiap Kelas

Kode	Nilai Precision
Normal (0)	0.91919 (91.91%)
Spam (1)	0.91463 (91.46%)
Promo (2)	0.97916 (97.91%)

Dari Tabel 8, diketahui bahwa kelas promo memiliki *recall* tertinggi sebesar 0.97916 (97.91%). Hal ini menunjukkan bahwa model hampir dapat mengenali seluruh data positif pada kelas promo, dengan *False Negative* lebih sedikit dibandingkan kelas lainnya.

d. ***F1-Score***

Nilai *F1-Score* masing-masing kelas dapat dihitung menggunakan rumus yang terdapat di bab 2.6. Dengan nilai *precision* dan *recall* yang telah dihitung sebelumnya, kita dapat menghitung hasil *F1-Score* masing-masing kelas sebagai berikut.

$$F1 = 2 \times \frac{0.98913 \times 0.91919}{0.98913 + 0.91919}$$

$$F1 = 2 \times \frac{0.90919}{1.90832}$$

$$F1 = 2 \times 0.47643$$

$$F1 = 0.95286$$

Spam:

$$F1 = 2 \times \frac{0.98684 \times 0.91463}{0.98684 + 0.91463}$$

$$F1 = 2 \times \frac{0.90259}{1.90147}$$

$$F1 = 2 \times 0.47468$$

$$F1 = 0.94936$$

Promo:

$$F1 = 2 \times \frac{0.77049 \times 0.97916}{0.77049 + 0.97916}$$

$$F1 = 2 \times \frac{0.75443}{1.74965}$$

$$F1 = 2 \times 0.43118$$

$$F1 = 0.86237$$

Tabel 9. Hasil Perhitungan *F1-Score* Setiap Kelas

Kode	Nilai Precision
Normal (0)	0.95286 (95.28%)
Spam (1)	0.94936 (94.93%)
Promo (2)	0.86237 (86.23%)

Dari Tabel 9 dapat disimpulkan bahwa kelas normal memiliki keseimbangan yang baik antara *precision* dan *recall* dengan nilai sebesar 0.95286 (95.28%), yang diikuti oleh spam dengan nilai tidak jauh beda dari normal yaitu 0.94936 (94.93%). Sedangkan kelas promo mendapat nilai terendah sebesar 0.86237 (86.23%) karena nilai *precision* jauh lebih rendah dibandingkan dengan nilai *recall*.

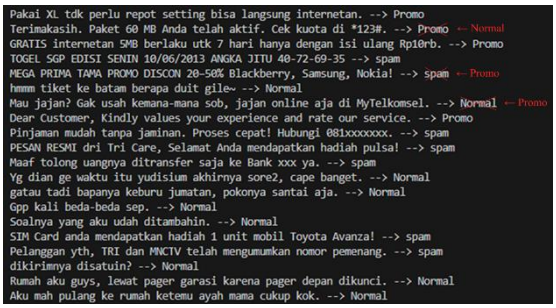
Dari hasil nilai-nilai tersebut dapat diketahui bahwa model kesulitan membedakan antara kelas normal dan spam dengan kelas promo. Dengan nilai *F1-Score* kelas promo 86.23%, kelas promo mempunyai rata-rata harmonik terkecil daripada kelas lainnya dikarenakan nilai *precision* kelas promo yang jauh lebih rendah dengan nilai *recall*-nya. Meskipun memiliki kelemahan didalam klasifikasi sms promo, model mampu dengan akurat memprediksi kelas normal dan spam dengan nilai *precision* masing-masing yaitu 98.91% untuk normal dan 98.68% untuk spam.

3.3. Testing Model Terhadap Data Baru Diluar *Training*

Hasil pengujian terhadap 20 data sampel SMS baru diluar dataset (Gambar 6) menunjukan bahwa model *Multinomial Naïve Bayes* mampu menklasifikasikan dengan benar 17 data sampel dari total 20 data sampel. Jika memakai rumus akurasi yang ada di bab 2.6, maka dapat dihitung:

$$Accuracy = \frac{17}{20} = 0.85$$

Dengan demikian, dapat disimpulkan bahwa model *Multinomial Naïve Bayes* memiliki kemampuan generalisasi yang baik dengan tingkat akurasi sebesar 0.85 (85%) terhadap data baru yang tidak dilibatkan dalam proses *training*.



Gambar 6. Hasil Testing Sampel SMS Diluar Dataset

4. KESIMPULAN

SMS merupakan salah satu fitur komunikasi pada *smartphone* yang masih belum dapat ditinggalkan hingga saat ini, meskipun sudah banyak bermunculan aplikasi pesan instan yang menggunakan internet seperti *Telegram*, *Whatsapp*, dan lainnya. SMS sering kali menjadi pilihan ketika seseorang berada diluar jangkauan jaringan internet seperti di daerah pedalaman desa ataupun kawasan terpencil.

Melalui penerapan *Machine Learning* dengan menggunakan model *Multinomial Naïve Bayes,* sistem klasifikasi SMS dapat dikembangkan untuk menjaga keamanan pengguna dari pesan yang bersifat spam atau promosi. Berdasarkan hasil pengujian dari dataset sebesar 1.143 yang dibagi menjadi 80% *training* dan 20% *testing*. Model *Multinomial Naïve Bayes* dapat dengan baik mengklasifikasikan pesan kedalam tiga kategori yaitu normal, spam, dan promo, dengan tingkat akurasi data *testing* mencapai 93%.

Mekipun model memiliki kelemahan dalam menklasifikasikan kelas promo, model dapat dilatih agar dapat meningkatkan performa dengan menambahkan lebih banyak data pelatihan (*training data*), penambahan data ini diharapkan dapat membuat model menjadi lebih akurat dalam menklasifikasikan data ke kelas yang tepat. Walaupun peningkatan akurasi yang diperoleh mungkin hanya sebesar 1-2%, hal ini tetap penting karena dapat membuat model lebih tepat dalam menklasifikasikan data kedalam kelas yang tepat sehingga model akan memiliki performa lebih baik secara keseluruhan dalam mendeteksi SMS spam maupun promo. Hasil ini dapat menjadikan landasan dasar untuk pengembangan sistem deteksi spam sms yang lebih lanjut di masa mendatang.

PUSTAKA

Adhitya, R. R., Witanti, W., Yuniarti, R., Jenderal, U., & Yani, A. (2023). *PERBANDINGAN METODE CART DAN NAÏVE BAYES UNTUK KLASIFIKASI*. *9*(2), 307–318.

Ahmed, A. B., & Khalid, H. (2025). ENHANCED SMS SPAM DETECTION USING BERNOULLI NAIVE BAYES WITH TF-IDF FJS. *FUDMA Journal of Sciences (FJS)*, *9*(1), 393–399.

Ajat, M. H. S. (2023). Klasifikasi Sms Spam Dengan Komparasi Metode Svm Dan Naïve Bayes. *METHODIKA: Jurnal Teknik Informatika Dan Sistem Informasi*, *9*(1), 31–34. https://doi.org/10.46880/mtk.v9i1.1694

Amazon Web Service. (n.d.). *Apa itu SMS? - Penjelasan Layanan Pesan Singkat - AWS*. Retrieved November 1, 2025, from https://aws.amazon.com/id/what-is/sms/

Androutsopoulos, I., Koutsias, J., Chandrinos, K. V., & Spyropoulos, C. D. (2000). Experimental comparison of Naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages. *SIGIR Forum (ACM Special Interest Group on Information Retrieval)*, 160–167. https://doi.org/10.1145/345508.345569

Bob Steward. (2024). *Dataset_SMS_Spam__Indonesia*. https://www.kaggle.com/datasets/bobsteward/dataset-sms-spam-indonesia

DasGupta, S., Saha, S., & Das, S. K. (2021). SMS spam detection using machine learning. *Journal of Physics: Conference Series*, *1797*(1). https://doi.org/10.1088/1742-6596/1797/1/012017

Jagoan Hosting. (2023, November 28). *Spam - Pengertian, Jenis, Contoh & Cara Mencegahnya*. https://www.jagoanhosting.com/blog/spam-adalah/#pengertian-spam

Kim Fai Kok. (2019, December 3). *Truecaller Insights: Top 20 Countries Affected by Spam Calls & SMS in 2019 - Truecaller Blog*. https://www.truecaller.com/blog/insights/truecaller-insights-top-20-countries-affected-by-spam-calls-sms-in-2019

Nasteski, V. (2017). An overview of the supervised machine learning methods. *Horizons.B*, *4*, 51–62. https://doi.org/10.20544/horizons.b.04.1.17.p05

Pranckevičius, T., & Marcinkevičius, V. (2017). Comparison of Naive Bayes, Random Forest, Decision Tree, Support Vector Machines, and Logistic Regression Classifiers for Text Reviews Classification. *Baltic Journal of Modern Computing*, *5*(2), 221–232. https://doi.org/10.22364/bjmc.2017.5.2.05

Sarker, I. H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*, *2*(3), 1–21. https://doi.org/10.1007/s42979-021-00592-x

Vijay, & Kumar, S. (2021). Spam SMS Detection Using Naive Bayes Classifier Abstract :

International Journal of Scientific Research and Engineering Development, *4*(1), 561–563.

Wahyuni, T., Susanti, D., Teknik, F., & Majalengka, U. (2023). *Analisis potensi bencana alam tanah longsor kabupaten majalengka menggunakan algoritma naïve bayes classifier 1,23*. *9*(2), 299–306.

Zhang, Z. (2016). Naïve bayes classification in R. *Annals of Translational Medicine*, *4*(12), 1–5. https://doi.org/10.21037/atm.2016.03.38