JITET (Jurnal Informatika dan Teknik Elektro Terapan)
Vol. 13 No. 1, pISSN: 2303-0577 eISSN: 2830-7062 http://dx.doi.org/10.23960/jitet.v13i1.5506

PENERAPAN ALGORITMA NAIVE BAYES DENGAN *CHI-SQUARE* UNTUK KLASIFIKASI *SPAM* EMAIL BERBASIS KATA DAN FREKUENSI

Faiz Agil Firmansyah[1*], Ultach Enri[2], Iqbal Maulana[3]

1, 2, 3 Informatika, Fakultas Ilmu Komputer, Universitas Singaperbangsa Karawang;
Jl. H.S. Ronggowaluyo, Puseurjaya, Telukjambe Timur, Karawang, Jawa Barat 41361

Received: 24 Oktober 2024
Accepted: 14 Januari 2025
Published: 20 Januari 2025

Keywords:
chi-square, classification, email, naive bayes, spam email,

Corespondent Email:
faizfirmansyah3@gmail.com

Abstrak. Email telah menjadi alat komunikasi yang penting dalam kehidupan sehari-hari. Namun, kemudahan penggunaannya juga dimanfaatkan oleh pihak-pihak yang tidak bertanggung jawab untuk menyebarkan *spam*. Penelitian ini bertujuan untuk menerapkan metode algoritma Naive Bayes dengan *Chi-square* dalam klasifikasi *spam* email berbasis kata dan frekuensi. Dataset yang digunakan sebanyak 153 data yang digunakan dalam penelitian ini. Data ini diolah menggunakan metode klasifikasi dengan menggunakan algoritma Naive Bayes dengan *Chi-square* melalui proses *Knowledge Discovery in Databases* (KDD). Hasil menunjukkan bahwa nilai akurasi sebesar 81.00%, nilai *precision* sebesar 100%, nilai *recall* sebesar 65%, dan nilai *f1-score* sebesar 79% menggunakan Naive Bayes dengan *Chi-square*. Serta hasil evaluasi menggunakan kurva ROC menunjukkan bahwa nilai AUC mencapai 0.91 yang masuk ke dalam kategori sangat baik. Penelitian ini menunjukkan bahwa algoritma Naive Bayes dengan *Chi-square* berhasil mengklasifikasi *spam* email berbasis kata dan frekuensi.

Abstract. *Email has become an essential communication tool in everyday life. However, the ease of its use is also exploited by irresponsible parties to spread spam. This research aims to implement the Naive Bayes algorithm with Chi-square in classifying spam emails based on words and frequency. The dataset used in this research consists of 153 data. This data was processed using the classification method using the Naive Bayes algorithm with Chi-square through the Knowledge Discovery in Databases (KDD) process. The results show that the accuracy value is 81.00%, the precision value is 100%, the recall value is 65%, and the F1-score value is 79% using Naive Bayes with Chi-square. Furthermore, the evaluation results using the ROC curve show that the AUC value reaches 0.91, which is categorized as very good. This research shows that the Naive Bayes algorithm with Chi-square is successful in classifying spam emails based on words and frequency.*

1. PENDAHULUAN

Email telah menjadi alat komunikasi yang penting dalam kehidupan sehari-hari. Namun, kemudahan penggunaannya juga dimanfaatkan oleh pihak-pihak yang tidak bertanggung jawab untuk menyebarkan *spam* [1]. *Spam* email dapat berupa iklan yang tidak diinginkan, penipuan, *malware*, dan konten berbahaya lainnya [2]. Di samping itu, volume *spam* email terus meningkat setiap tahunnya, yang menyebabkan beban komputasi yang besar bagi *server* email dan mengganggu kenyamanan pengguna [3]. *Spammer* terus mengembangkan teknik dan strategi baru untuk menghindari deteksi *spam* email. Hal ini menyebabkan sistem klasifikasi

spam email yang statis menjadi tidak efektif dalam mendeteksi *spam* email baru.

Klasifikasi sendiri merupakan salah satu metode *data mining* yang dapat menemukan model dengan membedakan suatu konsep atau *class* yang labelnya belum diketahui. Klasifikasi dilakukan untuk menilai objek data dan memasukkannya ke dalam kelas tertentu dari sejumlah kelas yang tersedia [4].

Metode deteksi *spam* tradisional yang berbasis aturan seperti filter berbasis kata kunci dan *blacklist* tidak ideal untuk melawan *spam* yang selalu berkembang. Hal ini disebabkan oleh keterbatasannya dalam mengikuti pola kata-kata dan frekuensi kemunculan yang terus berubah dalam pesan *spam* [5]. Selain itu, keterbatasan utamanya terletak pada akurasinya yang rendah dalam mendeteksi *spam* email baru atau *spam* email yang terstruktur dengan baik [6]. Penyebabnya adalah metode tradisional hanya mengandalkan pola yang sudah ada dan tidak dapat beradaptasi dengan pola *spam* email yang terus berkembang.

Oleh karena itu, diperlukan pendekatan yang mampu mengatasi perubahan dinamis dalam strategi *spam* dan mampu menyesuaikan diri dengan pola baru yang muncul dari waktu ke waktu [7].

Algoritma klasifikasi Naive Bayes terkenal dengan asumsinya yang unik, yaitu "kemurnian" (*naive*) antar fitur. Asumsi ini menyatakan bahwa fitur-fitur yang digunakan dalam proses klasifikasi tidak saling berhubungan satu sama lain. Berbeda dengan algoritma klasifikasi lain, Naive Bayes memanfaatkan perhitungan probabilitas untuk memprediksi kategori atau label yang tepat bagi data yang belum diklasifikasikan. Algoritma ini menghitung probabilitas kelas dan probabilitas fitur, kemudian menggabungkannya untuk menentukan kelas mana yang paling mungkin untuk data yang baru [8].

Pendekatan berbasis kata dan frekuensi digunakan untuk mengklasifikasikan email sebagai *spam* atau *non-spam* dengan memperhatikan kata-kata yang muncul dalam email dan seberapa sering kata-kata tersebut muncul. Setiap kata dalam email dianggap sebagai fitur yang dapat membedakan email spam dari yang bukan. Misalnya, kata-kata seperti "gratis", "diskon", "klik di sini", atau "uang" sering ditemukan dalam email spam, sehingga kehadiran kata-kata tersebut bisa menjadi indikator kuat bahwa email tersebut adalah *spam*. Frekuensi kemunculan suatu kata dalam email juga memengaruhi bobotnya dalam klasifikasi. Kata-kata yang sering muncul dalam email *spam* dan jarang ditemukan dalam email *non-spam* memiliki bobot yang lebih tinggi, sehingga membantu menentukan apakah email tersebut tergolong spam atau tidak [9].

Proses klasifikasi *text* sebelumnya pernah dilakukan juga oleh [10], namun penelitian bukan meneliti tentang *spam email*. Penelitian tersebut menggunakan Naïve Bayes dengan *Particle Swarm Optimization* (PSO) untuk *text mining sentiment restaurant*, dengan performa model yang dihasilkan Algoritma Naïve Bayes dengan PSO mendapatkan tingkat akurasi sebesar 83.80%. Penelitian ini juga menyarankan untuk menggunakan *chi-square* untuk membandingkan performanya.

2. TINJAUAN PUSTAKA

2.1 Spam Email

Menurut penelitian dari [11], *spam* email adalah pesan email yang tidak diminta dan tidak diinginkan oleh penerima. Penelitian ini menyatakan bahwa *spam* email dapat berupa iklan yang tidak relevan, penipuan *phishing*, *malware*, dan berbagai jenis konten yang tidak diinginkan lainnya.

2.2 Klasifikasi Email

Menurut penelitian oleh [12], klasifikasi *spam* email adalah proses mengkategorikan email sebagai *spam* atau bukan *spam*. Penelitian ini menunjukkan bahwa klasifikasi *spam* email dapat dilakukan dengan berbagai metode, yang masing-masing memiliki kelebihan dan kekurangan.

2.3 Naive Bayes

Algoritma Naive Bayes, menurut penelitian yang dilakukan oleh [12], adalah Algoritma klasifikasi probabilistik yang sederhana namun sangat efektif. Penelitian ini menggarisbawahi bahwa keefektifan Algoritma Naive Bayes berasal dari dasar teoritisnya yang kuat, yaitu teorema Bayes. Algoritma ini juga dikenal dengan prediksi yang cepat dan membutuhkan sedikit data pelatihan. Teorema Bayes secara matematis dinyatakan sebagai:

$$P(C|X) = \frac{P(X|C) \times P(C)}{P(X)} \qquad (1)$$

2.4 *Chi-square*

Chi-square adalah sebuah variabel acak yang didefinisikan sebagai jumlah kuadrat dari variabel-variabel acak normal standar yang saling independen. Sifat aditif dari variabel *chi-square* ini disebabkan oleh definisi tersebut. Distribusi probabilitasnya mengikuti distribusi gamma. Statistik *chi-square* untuk uji kebaikan-suai (*goodness-of-fit*) memiliki distribusi yang mendekati distribusi *chi-square* ketika ukuran sampelnya besar. Uji-uji hipotesis yang berkaitan dengan tabel kontingensi juga menggunakan statistik yang memiliki distribusi *chi-square* yang mendekati [13]. Perhitungan dari *Chi-square* dapat dilihat pada persamaan berikut:

$$\chi^2 = \Sigma [(O - E)^2 / E] \quad (2)$$

Keterangan:
χ^2: Nilai *chi-square*
O: Frekuensi observasi
E: Frekuensi yang diharapkan
Σ: Penjumlahan untuk semua sel dalam tabel kontingensi

2.5 *Knowledge Discovery in Databases* (KDD)

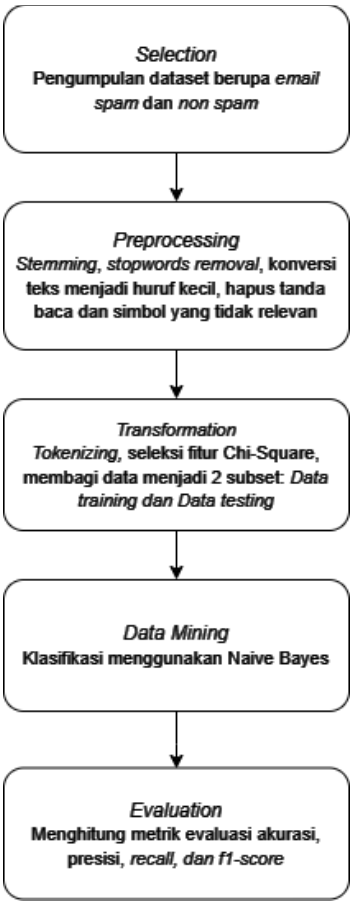
Knowledge Discovery in Databases (KDD) adalah suatu proses non-trivial yang bertujuan untuk menemukan dan mengidentifikasi pola dalam sekumpulan data. KDD memiliki hubungan erat dengan teknik integrasi, penemuan ilmiah, interpretasi, dan visualisasi data. KDD tidak hanya sebatas menemukan pola, tetapi juga tentang memahami pola tersebut dan mengkomunikasikannya dengan cara yang efektif [14].

2.6 Prinsip Pareto

Prinsip Pareto, yang juga disebut Aturan 80/20, merupakan konsep manajemen yang menyatakan bahwa 80% hasil atau output dihasilkan oleh 20% dari usaha atau input. Artinya, sebagian kecil faktor berkontribusi pada sebagian besar dampak. Walaupun proporsi pastinya bisa berbeda tergantung pada konteks, konsep ini secara umum diterapkan di berbagai bidang, termasuk bisnis, produksi, dan kehidupan sehari-hari [15].

3. METODE PENELITIAN

Metode penelitian yang digunakan dalam penelitian ini adalah *Knowledge Discovery in Database* (KDD). Metode ini digunakan pada penelitian ini karena menyediakan kerangka kerja yang tepat dan teknik yang sesuai serta dapat membantu menggali pengetahuan yang berguna dari data *email*, membangun model klasifikasi yang kuat, dan menginterpretasikan hasil secara efektif.



Gambar 1. Rancangan Penelitian

Metode *Knowledge Discovery in Database* (KDD) yang terdiri dari 5 tahapan yaitu *Selection*, *Preprocessing*, *Transformation*, *Data Mining* dan *Evaluation*.

3.1. Selection

Untuk memulai proses klasifikasi *spam* email, langkah pertama adalah mengumpulkan dataset email yang telah dikategorikan sebagai *spam* atau bukan *spam*. Dataset diperoleh dari *Hugging Face*, dan *Colby Computer Science Edu* sebanyak 153 data. Setelah dataset terkumpul, langkah selanjutnya adalah memastikan kualitasnya dengan melakukan pembersihan dan *preprocessing* data. Dengan memastikan kualitas dataset melalui proses pembersihan dan *preprocessing*, diharapkan

dapat meningkatkan kinerja model klasifikasi *spam* email yang akan dibangun.

3.2. Preprocessing

Setelah mengumpulkan dataset, langkah selanjutnya yang dilakukan adalah *preprocessing* data untuk membersihkan dan mempersiapkan data email. Proses ini melibatkan beberapa langkah, termasuk konversi teks email menjadi huruf kecil untuk konsistensi, penghapusan tanda baca dan simbol yang tidak relevan, penghilangan kata-kata berhenti (*stopwords*) yang tidak memiliki makna signifikan dalam klasifikasi dan juga proses *stemming*. *Stemming* adalah proses memperoleh kata dasar dengan menghapus imbuhan pada kata, termasuk awalan (prefix), sisipan (infix), akhiran (suffix), serta kombinasi awalan dan akhiran (confix) [16].

3.3. Transformation

Pada tahap transformasi data dimulai dengan memisahkan kata dalam setiap email menjadi token-token. Kemudian, setiap email direpresentasikan sebagai vektor fitur di mana setiap kata memiliki indeks dalam kamus kata unik, dengan nilai yang menunjukkan frekuensi kemunculan kata. Setelah normalisasi frekuensi, seleksi fitur akan dilakukan menggunakan metode *chi-square* untuk mengidentifikasi kata-kata atau fitur yang paling relevan untuk membedakan antara email *spam* dan non-*spam*. Setelah seleksi fitur, data dibagi menjadi dua subset dengan perbandingan 80:20, perbandingan 80:20 digunakan karena memiliki rasio data yang optimal untuk mengidentifikasi *email spam*. Prinsip Pareto mendukung pendekatan ini, di mana fokus pada fitur-fitur utama yang berdampak besar dapat meningkatkan efisiensi dan akurasi klasifikasi [17].

3.4. Data Mining

Pada tahap *data mining*, proses dilakukan dengan menggunakan teknik klasifikasi yang melibatkan algoritma *Naïve Bayes*. Proses untuk memodelkan klasifikasi *spam email* akan menggunakan *Naïve Bayes* dengan penggunaan data pelatihan untuk mengenali pola-pola yang terkait dengan *email spam* dan *email non-spam*. Langkah selanjutnya adalah mengoptimalkan parameter Algoritma Naive Bayes, seperti *smoothing*, untuk meningkatkan kemampuannya dalam membedakan antara keduanya. Pengaturan *smoothing* yang tepat sangat penting untuk mengatasi masalah *sparse data*, di mana kemungkinan beberapa kata tidak muncul dalam dataset pelatihan.

3.5. Evaluation

Setelah melatih model klasifikasi *spam* email, langkah selanjutnya adalah mengevaluasi kinerjanya pada data uji yang tidak pernah dilihat sebelumnya. Dalam proses evaluasi, metrik evaluasi akurasi, presisi, *recall*, dan F1-score dihitung untuk mengukur seberapa baik model dapat membedakan antara email *spam* dan bukan *spam*. Analisis hasil evaluasi dilakukan untuk memahami kekuatan (*strengths*) dan kelemahan (*weaknesses*) model klasifikasi *spam* email yang telah dikembangkan. Ini memungkinkan untuk mengidentifikasi area di mana model berfungsi dengan baik, serta area di mana perbaikan atau peningkatan diperlukan untuk meningkatkan kinerja dan keandalannya dalam mengklasifikasikan email.

4. HASIL DAN PEMBAHASAN

Pada penelitian ini dibantu menggunakan bahasa pemrograman Python dan beberapa *libraries* yang dimilikinya seperti Pandas, Sklearn, Matplotlib, Numpy, dan Seaborn. *Libraries* ini membantu dan memudahkan dalam pembuatan dataset, pengolahan dataset, permodelan, dan menampilkan grafik. Jadi sebagian besar tabel dan grafik yang dihasilkan pada bab ini dihasilkan dari *libraries* tersebut.

4.1 Selection

Dataset yang digunakan adalah data yang diperoleh dari *Hugging Face Repository*, dan *Colby Computer Science Edu* yang telah dilabeli "*spam*" dan "not *spam*" oleh penyedia data. Jumlah dataset yang digunakan dalam penelitian ini sebanyak 153 data. Terdapat 3 atribut dalam data tersebut seperti *Title*, *Text*, dan *Type*. Ketiga atribut tersebut akan digunakan dalam penelitian ini.

Tabel 1 Informasi Dataset

title	text	type
?? the secrets to SUCCESS	Hi James, Have you claim your complimentary gift yet? I've compiled in here a special astrology gift that predicts	*spam*

	everything about you in the future? This is your enabler to take the correct actions now. >> Click here to claim your copy now >> Claim yours now, and thank me later. Love, Heather	
?? You Earned 500 GCLoot Points	alt_text Congratulations, you just earned 500 You completed the following offer: View Points History To stop recieving notifications when you earn points, please visit your profile and click "Preferences" to change your setting. Room 1203, 12th Floor, Tower 3 China Hong Kong City, 33 Canton Road Tsim Sha Tsui, Kowloon, Hong Kong Unsubscribe	not *spam*

Pada Tabel 1 merupakan informasi dataset yang digunakan dalam penelitian ini yang berisi 3 *attributes* yaitu "title", "text", dan "type". Lalu, pada Tabel 2 perbandingan *label class* klasifikasi yaitu "*Spam*" dan "Not *Spam*".

Tabel 2 Perbandingan Label Class

Tipe	Persentase	Jumlah
Spam	56%	85
Not *Spam*	44%	67

4.2 Preprocessing

4.2.1 Lowercasing for Feature Reduction

Langkah ini bertujuan untuk menormalisasi data dan mengurangi dimensi fitur. Dengan mengubah semua menjadi huruf kecil, variasi kapitalisasi yang tidak memberikan informasi berarti untuk proses klasifikasi dapat dihilangkan.

Tabel 3 *Lowercasing*

Sebelum	Sesudah
?? the secrets to SUCCESS	?? the secrets to success
?? You Earned 500 GCLoot Points	?? you earned 500 gcloot points
?? Your GitHub launch code	?? your github launch code

4.2.2 Removing Punctuation

Menghapus tanda baca untuk menyederhanakan teks dan mengurangi kompleksitas model. Dengan menghapus tanda baca, jumlah fitur yang perlu dipelajari oleh algoritma dapat dikurangi. Selain itu, tanda baca yang berbeda dapat memiliki arti yang berbeda dalam konteks yang berbeda.

Tabel 4 *Removing Punctuation*

Sebelum	Sesudah
?? your github launch code	your github launch code
[the virtual reward center] re: ** clarifications	the virtual reward center re clarifications
10-1 mlb expert inside, plus everything you need	101 mlb expert inside plus everything you need

4.2.3 Removing Irrelevant Symbols

Menghapus simbol-simbol yang tidak relevan. Langkah ini dilakukan untuk membersihkan data dan meningkatkan kualitas fitur data, simbol-simbol seringkali digunakan dalam *email spam* untuk menarik perhatian atau mengelabui filter *spam*.

4.2.4 Stopwords Removal

Stopwords removal bertujuan untuk meningkatkan efisiensi komputasi, hal ini akan membuat model pembelajaran mesin lebih mudah untuk mempelajari pola yang relevan dan menghasilkan keputusan yang lebih akurat.

Tabel 5 *Stopwords Removal*

Sebelum	Sesudah
hi james\n\nhave you claim your complimentary gift	hi james claim complimentary gift

\nalttext\ncongratulations you just earned	alttext congratulations earned 500 completed
heres your github launch code mortyj420\n	heres github launch code mortyj420

4.2.5 Stemming

Langkah terakhir pada tahap *pre-processing* yaitu *stemming*. *Stemming* dilakukan untuk menyatukan kata-kata yang memiliki akar kata yang sama sehingga algoritma klasifikasi dapat lebih mudah mengenali pola kata yang serupa, meskipun ditulis dalam bentuk yang berbeda.

Tabel 6 *Stopwords Removal*

Sebelum	Sesudah
earned 500 gcloot points	earn 500 gcloot point
alttext congratulations earned 500 completed	alttext congratul earn 500 complet
virtual reward center clarifications	virtual reward center clarif

4.3 Transformation

Pada tahap ini dataset diubah agar sesuai dengan kebutuhan penelitian. Proses awal transformasi data adalah mengubah nilai pada kolom “type” menjadi kategorikal numerik menggunakan teknik *Label Encoder*, teknik ini umumnya mengenkode label target dengan nilai antara 0 hingga n_classes-1.

Tabel 7 Transformasi Kolom *Type*

Sebelum	Sesudah
Spam	1
Not *Spam*	0

Setelah dilakukan transformasi pada kolom “type”, langkah selanjutnya adalah vektorisasi teks yang telah diproses pada tahap *preprocessing* untuk normalisasi frekuensi. Vektorisasi teks dilakukan sebagai persiapan untuk pelatihan model Naive Bayes. Vektor-vektor yang dihasilkan akan menjadi input bagi algoritma Naive Bayes untuk membangun model prediksi.

Tabel 8 Contoh Tampilan Hasil Vektorisasi Teks

	children	chill	china	chino	cholesterol
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0

Selanjutnya, dilakukan seleksi fitur menggunakan metode *chi-square* yang berguna untuk menemukan kata-kata yang paling sering muncul dalam email *spam* dan jarang muncul dalam email non-*spam*, atau sebaliknya.

Tabel 9 Hasil *Chi-square*

Frekuensi	Fitur	Skor *Chi-square*	P-Value
2468	logo	42.339623	7.672235e-11
3193	pt	25.660377	4.071004e-07
1515	email	23.228018	1.438857e-06
1107	compani	21.118535	4.317320e-06
396	account	19.289219	1.123389e-05

Kata “logo” memiliki skor *chi-square* tertinggi dan p-value terkecil yang menunjukkan bahwa kata "logo" sangat terkait dengan salah satu kelas antara *spam* atau *not spam*. Begitu pula dengan kata "pt", "email", "company", dan "account" memiliki skor *chi-square* yang tinggi dan p-value yang sangat kecil, menunjukkan bahwa kata-kata ini juga sangat relevan dalam membedakan antara kedua kelas.

Langkah terakhir pada transformasi data yaitu membagi data menjadi dua subset menggunakan prinsip pareto dengan perbandingan 80:20.

Tabel 10 Pembagian Dua Subset

Tipe	Persentase	Jumlah
Data Training	80%	121
Data Testing	20%	31

4.4 Data Mining

Pada tahap ini, pemodelan dataset *spam* email dilakukan dengan membagi dan menguji data menggunakan teknik *train-test split*. Data dipisahkan menjadi 80% untuk data training (123 data) dan 20% untuk data testing (31 data). Proses pemodelan klasifikasi dilakukan menggunakan algoritma Naive Bayes, yang dipilih karena kemampuannya dalam menangani klasifikasi pada dataset *spam* email.

Tabel 11 Prediksi Pada Data Uji

Data Uji Ke-	0	1	Prediksi
1	1.000000e+00	2.309836e-16	not spam
2	9.999967e-01	3.250502e-06	not spam
3	9.879623e-01	1.203771e-02	not spam
4	9.972801e-01	2.719929e-03	not spam
5	6.669074e-08	9.999999e-01	spam
dst.	.....	.....	.....

Model Naive Bayes yang dihasilkan dari data *training* dan dilatih dengan data *testing* menampilkan probabilitas target pada setiap fitur yang ditunjukkan pada kolom "0" dan "1", kolom tersebut merepresentasikan label *class* "Not *Spam*" dan "*Spam*". Nilai di kedua kolom tersebut dihitung menggunakan persamaan (1). Hasil probabilitas di kedua kolom dibandingkan, dan nilai yang lebih besar menjadi dasar untuk prediksi yang ditampilkan di kolom "Prediksi", yang berisi "not *spam*" atau "*spam*".

Sebelum menemukan hasil prediksi, langkah yang dilakukan adalah menentukan probabilitas prior masing-masing kelas, yaitu kelas 0 ("Not Spam") dan kelas 1 ("Spam"), pada data training yang terdiri dari 123 data. Probabilitas ini dihitung menggunakan rumus dasar probabilitas pada data latih.

Tabel 12 Probabilitas Setiap Kelas

Kelas (type)	Jumlah	Probabilitas
0	53	0.438017
1	68	0.561983

Setelah mengetahui probabilitas setiap kelas, perlu juga untuk mengetahui probabilitas *likelihood* pada data *training* untuk kelas *spam* dan *non spam* menggunakan fitur yang telah diseleksi *chi-square* dengan menggunakan rumus probabilitas *likelihood*.

Tabel 13 Probabilitas Likelihood untuk Kelas *Spam*

Frekuensi	Kata	Probabilitas
33	logo	0.00008
20	pt	0.00008
81	email	0.00202
55	compani	0.00121
58	account	0.00137

Tabel 14 Probabilitas Likelihood untuk Kelas *Not Spam*

Frekuensi	Kata	Probabilitas
81	logo	0.00311
80	pt	0.00192
117	email	0.00530
74	compani	0.00384
58	account	0.00393

Selanjutnya menghitung probabilitas *posterior* untuk mengetahui prediksi pada data uji menggunakan probabilitas *prior* dan probabilitas *likelihood* yang telah diketahui sebelumnya. Langkah pertama untuk mengetahui probabilitas *posterior*, yaitu menghitung probabilitas total untuk kata "com" atau P(email).

$$P(email) = P(email|Spam)\ x\ P(Spam) + P(email|NotSpam)\ x\ P(NotSpam)$$

$$P(email) = (0.00202\ x\ 0.561983) + (0.00530\ x\ 0.438017) = 0.003457$$

Setelah didapatkan probabilitas untuk kata "email", baru dapat menghitung probabilitas *posterior* atau P(*spam*|email).

$$P(Spam|email) = \frac{P(email|Spam)\ \times\ P(Spam)}{P(email)}$$

$$P(Spam|email) = \frac{0.00202\ x\ 0.561983}{0.003457} \approx 0.3276$$

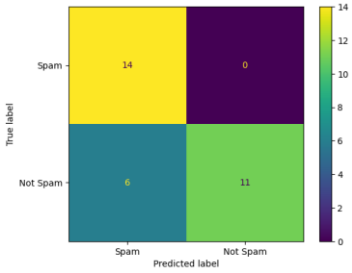
Perhitungan di atas merupakan salah satu contoh nilai probabilitas yang menunjukkan bahwa jika sebuah *email* mengandung kata "com", maka kemungkinan *email* tersebut adalah *spam* sebesar 32.76%.

Setelah menghitung probabilitas posterior untuk kedua kelas, hasilnya dibandingkan. Jika probabilitas pada kolom "1" (spam) lebih tinggi dari kolom "0" (not spam), maka model akan memprediksi email tersebut sebagai *spam*. Sebaliknya, jika nilai pada kolom "0" lebih tinggi, email diprediksi sebagai *not spam.* Perhitungan di atas nantinya akan digunakan untuk mendapat nilai pada kolom "0" dan "1" pada Tabel 4.11, yang berisi ringkasan kinerja model pada berbagai dataset uji.

4.5 Evaluation

Setelah proses pemodelan, model dianalisis dan dievaluasi dengan menggunakan *confusion matrix*, *classification report*, grafik *Receiver Operation Curve* (ROC), nilai *Area Under The Curve* (AUC), Selain itu, performa model juga diuji menggunakan *K-Fold Cross Validation*.

Pada model Naive Bayes, *confusion matrix* menunjukkan 14 *True Negatives* (TN), 11 *True Positives* (TP), 0 *False Positives* (FP), dan 6 *False Negatives* (FN), dengan total keseluruhan 31. Hasil ini diperoleh dari membandingkan prediksi model dari data *training* terhadap data *test*.



Gambar 2. *Confussion Matrix*

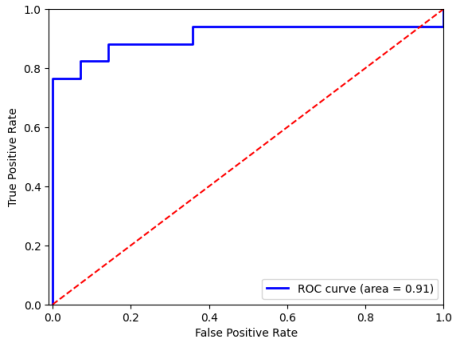
Melalui *confusion matrix*, dapat dihitung metrik evaluasi lain, yaitu *classification report*, yang menampilkan nilai *accuracy*, *precision*, *recall*, *f1-score*, dan *support*.

Tabel 14 *Classification Report*

	Precision	**Recall**	**F1-Score**	**support**
0	0,70	1.00	0.82	14
1	1.00	0.65	0.79	17
Accuracy			0.81	31
Macro avg	0.85	0.82	0.80	31
Weighted avg	0.86	0.81	0.80	31

Berdasarkan analisis *classification report*, model ini mencapai akurasi sekitar 81% dan memiliki *precisionl* yang baik untuk kelas 1 (*Spam*), yaitu sebesar 1.00. Kolom *support* mewakili jumlah data sebenarnya dalam kelas tersebut yang diharapkan untuk diklasifikasikan.

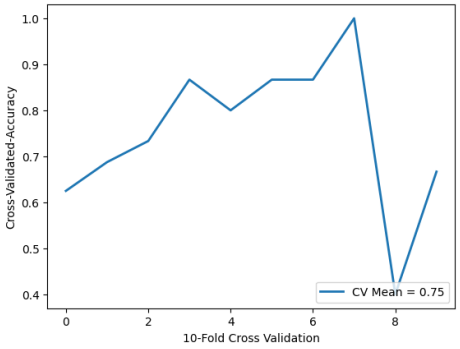
Selanjutnya, evaluasi dilakukan menggunakan metrik evaluasi ROC, yang membandingkan nilai *False Positive Rate* (FPR) dan *True Positive Rate* (TPR) yang diperoleh dari *confusion matrix* sebelumnya. Setelah mengetahui nilai dari kedua parameter tersebut, perbandingannya divisualisasikan dalam bentuk grafik.



Gambar 2. ROC

Berdasarkan Gambar 2 diketahui bahwa nilai AUC adalah 0,91. Nilai ini diperoleh dengan menghitung area di bawah kurva ROC. Hal ini mengindikasikan bahwa model klasifikasi yang digunakan memiliki kinerja yang sangat baik dalam membedakan antara kelas positif dan negatif [18].

Evaluasi juga dilakukan menggunakan K-Fold Cross Validation. Hal ini dikarenakan ingin mendapatkan estimasi akurasi model yang lebih reliabel dan menghindari masalah *overfitting*.



Gambar 3. *10-Fold Cross Validation*

Pada Gambar 3 terlihat performa model di setiap iterasi dengan nilai K = 10. Iterasi ke-8 menunjukkan akurasi terburuk, sementara iterasi ke-7 memiliki akurasi terbaik. Rata-rata akurasi di setiap iterasi adalah 0,75, seperti yang ditunjukkan Gambar 3. Hasil ini membantu memastikan bahwa model tidak hanya bekerja baik pada satu subset data, tetapi juga memiliki performa yang stabil di berbagai bagian dataset.

Berdasarkan evaluasi yang telah dilakukan, penelitian ini memiliki kelebihan, penggunaan

algoritma Naive Bayes yang dikombinasikan dengan seleksi fitur menggunakan *Chi-square*, yang menghasilkan akurasi 81% dan AUC sebesar 0,91, menunjukkan performa yang sangat baik dalam klasifikasi spam email.

Namun, penelitian ini juga memiliki beberapa kekurangan. Dataset yang digunakan relatif kecil, hanya 153 data, sehingga dapat membatasi generalisasi model. Selain itu, asumsi independensi antar fitur pada Naive Bayes tidak selalu realistis, yang juga dapat mempengaruhi keakuratan prediksi. Nilai *recall* yang hanya 65% menunjukkan model masih kurang efektif dalam mendeteksi semua spam (*false negatives*). Selain itu, penelitian ini tidak membandingkan kinerjanya dengan algoritma lain seperti SVM atau *Random Forest*, yang bisa memberikan evaluasi yang lebih komprehensif.

5. KESIMPULAN

Berdasarkan hasil dari penelitian yang telah dilakukan, didapatkan kesimpulan mengenai mengenai algoritma Naive Bayes dalam mengklasifikasi *spam* email berbasis kata dan frekuensi serta saran untuk penelitian selanjutnya.

a. Penelitian ini menggunakan proses *Knowledge Discovery in Databases* (KDD) untuk membangun model prediksi *spam* email. *Data mining* menggunakan algoritma Naive Bayes untuk mengeksplorasi pola dalam email. Pembagian data untuk *modelling* adalah 80% data *training* dan 20% data *testing* dari total 153 data.
b. Untuk melakukan evaluasi performa model menggunakan metrik evaluasi, tentukan TP, TN, FP, FN dengan *confusion matrix* yang didapatkan dengan membandingkan prediksi model dari data *training* terhadap data *test*. Setelah melakukan *confusion matrix*, dilakukan *classification report* untuk menampilkan nilai *accuracy*, *precision*, *recall*, dan *f1-score*. Kemudian didapatkan 0,81 untuk nilai *accuracy*, 1,00 untuk nilai *precision*, 0,65 untuk nilai *recall*, dan 0,79 untuk nilai *f1-score*. Nilai-nilai tersebut didapat menggunakan rumus *confusion matrix*.
c. Algoritma Naive Bayes menunjukkan kinerja yang baik dalam mengklasifikasikan email *spam* dan non-*spam* berdasarkan kata dan frekuensinya, meskipun terdapat beberapa kelemahan. Dari hasil analisis menggunakan ROC memberikan nilai AUC sebesar 0,91, yang menunjukkan kemampuan model untuk membedakan kelas masuk ke dalam kategori sangat baik (90%-100%).
d. Untuk penelitian selanjutnya diharapkan untuk menggunakan dataset terbaru dan lebih besar serta mengeksplorasi metode *preprocessing* lain yang lebih canggih dalam klasifikasi *spam* email. Selain itu diharapkan juga agar dapat membandingkan algoritma Naive Bayes dengan algoritma *Machine Learning* lainnya dan metode evaluasi yang lebih inovatif, sehingga dapat menghasilkan model dengan performa yang lebih optimal.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada pihak-pihak terkait yang telah memberi dukungan terhadap penelitian ini.

DAFTAR PUSTAKA

[1] A. Hidayat, "Aplikasi Teks Mining Untuk Mendeteksi Spam Pada Email Berbasis Naive Bayes," *Jurnal Teknologi Pintar*, vol. 2, no. 8, 2022.

[2] A. Wibisono, "Filtering Spam Email Menggunakan Metode Naive Bayes," *Jurnal Teknologi Pintar*, vol. 3, no. 4, 2023.

[3] M. A. M. Foqaha, "Email spam classification using hybrid approach of RBF neural network and particle swarm optimization," *International Journal of Network Security & Its Applications*, vol. 8, no. 4, pp. 17–28, 2016.

[4] M. R. Rahmaputri, D. S. Y. Kartika, and S. F. A. Wati, "KLASIFIKASI TINGKAT KEPUASAN PELANGGAN SAT & SUN : THE ALMEATY SERVICE MENGGUNAKAN NAIVE BAYES," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 3, pp. 2658–2663, Aug. 2024, doi: 10.23960/jitet.v12i3.4844.

[5] G. Borotić, L. Granoša, J. Kovačević, and M. Bagić Babac, "Effective Spam Detection with Machine Learning," *Croatian Regional Development Journal*, vol. 4, no. 2, pp. 43–64, 2023.

[6] B. Christanto and D. H. Setiabudi, "Penerapan Random Forest dalam Email Filtering untuk

Mendeteksi Spam," *Jurnal Infra*, vol. 8, no. 2, pp. 138–142, 2020.

[7] H. Iswanto, E. Seniwati, Y. Astuti, and D. Maulina, "Comparison of Algorithms on Machine Learning For Spam Email Classification," *IJISTECH (International Journal of Information System and Technology)*, vol. 5, no. 4, pp. 446–455, 2021.

[8] P. S. Zakaria, R. Julianto, and R. S. Bernada, "Implementasi Naive Bayes Menggunakan Python Dalam Klasifikasi Data," *Buletin Ilmiah Ilmu Komputer dan Multimedia (BIIKMA)*, vol. 1, no. 1, pp. 126–131, 2023.

[9] J. Mythili, B. Deebeshkumar, T. Eshwaramoorthy, and J. N. Ajay, "Enhancing Email Spam Detection with Temporal Naive Bayes Classifier," in *2024 International Conference on Communication, Computing and Internet of Things (IC3IoT)*, IEEE, 2024, pp. 1–6.

[10] R. Aulianita, A. M. B. Aji, and Y. E. Achyani, "TEXT MINING MENGGUNAKAN NAIVE BAYES BERBASIS PARTICLE SWARM OPTIMIZATION UNTUK SENTIMENT RESTAURANT," *JUTIM (Jurnal Teknik Informatika Musirawas)*, vol. 6, no. 1, pp. 21–29, 2021.

[11] T. Toma, S. Hassan, and M. Arifuzzaman, "An analysis of supervised machine learning algorithms for spam email detection," in *2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI)*, IEEE, 2021, pp. 1–5.

[12] T. Lv, P. Yan, H. Yuan, and W. He, "Spam filter based on naive Bayesian classifier," in *Journal of Physics: Conference Series*, IOP Publishing, 2020, p. 012054.

[13] H. O. Lancaster and E. Seneta, "Chi-square distribution," *Encyclopedia of biostatistics*, vol. 2, 2005.

[14] C. E. Purnomo and R. Rikendry, "Penerapan Metode C4. 5 Untuk Klasifikasi Warga Miskin Pada Desa Mengandung Sari," *Jurnal Teknologi dan Sistem Informasi*, vol. 2, no. 3, pp. 14–25, 2021.

[15] V. Pareto and J. Lopreato, *Vilfredo Pareto*. TY Crowell Company, 1965.

[16] T. Ridwansyah, "Implementasi text mining terhadap analisis sentimen masyarakat dunia di twitter terhadap Kota Medan menggunakan k-fold cross validation dan naïve bayes classifier," *KLIK: Kajian Ilmiah Informatika dan Komputer*, vol. 2, no. 5, pp. 178–185, 2022.

[17] I. P. Putri, "Analisis Performa Metode K-Nearest Neighbor (KNN) dan Crossvalidation pada Data Penyakit Cardiovascular," *Indonesian Journal of Data and Science*, vol. 2, no. 1, pp. 21–28, 2021.

[18] A. Nugroho, A. B. Gumelar, A. G. Sooai, D. Sarvasti, and P. L. Tahalele, "Perbandingan Performansi Algoritma Pengklasifikasian Terpandu Untuk Kasus Penyakit Kardiovaskular," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 4, no. 5, pp. 998–1006, 2020.