Jurnal Informatika Dan Rekayasa Komputer (JAKAKOM)
Available Online at https://ejournal.unama.ac.id/index.php/jakakom
Volume 5, Nomor 2, September 2025,
ISSN 2808-5469 (media cetak), ISSN 2808-5000 (media online)
UNAMA, DOI 10.33998/jakakom.v5i2

Perbandingan Algoritma Naive Bayes Dan K-Nearest Neighbor Pada Klasifikasi Email

Muhammad Athorik Alfayyed[1], Jasmir[1], Dodi Sandra[2*]

[1] Fakultas Ilmu Komputer, Program Studi Teknik Informatika, Universitas Dinamika Bangsa, Jambi, Indonesia
Email: [1]muhammadtorik905@gmail.com, [2]ijay_jasmir@yahoo.com, [3]doedy235@gmail.com
Email Penulis Korespondensi: doedy235@gmail.com

Artikel Info :
Artikel History :
Submitted : 10-05-2025
Accepted : 12-09-2025
Published : 30-09-2025

Kata Kunci: Spam *Email*, *Naive Bayes*, *K-Nearest Neighbor*, TF-IDF, Google Colab.

Abstrak−Penelitian ini membandingkan algoritma *Naive Bayes* dan *K-Nearest Neighbor* (KNN) dalam klasifikasi *email* menggunakan fitur TF-IDF. Dataset terdiri atas 5.572 email yang diproses melalui pelatihan dan evaluasi model dengan validasi silang 5-fold. Metode evaluasi mencakup akurasi, precision, recall, F1-score, dan ROC/AUC. Hasil menunjukkan *Naive Bayes* unggul dengan akurasi 98,38% dan AUC 0,98 dibandingkan KNN yang mencapai akurasi 92,47% dan AUC 0,82. *Naive Bayes* lebih stabil dalam recall pada kelas spam (89%), sementara KNN memiliki precision sempurna (100%) namun rendah recall (44%).

.

Keyword: Email Spam, Naive Bayes, K-Nearest Neighbor, TF-IDF, Google Colab

Abstract− This study compares the Naive Bayes and K-Nearest Neighbor (KNN) algorithms in email classification using the TF-IDF feature. The dataset consists of 5,572 emails processed through training and model evaluation with 5-fold cross validation. Evaluation methods include accuracy, precision, recall, F1-score, and ROC/AUC. The results show that Naive Bayes is superior with an accuracy of 98.38% and an AUC of 0.98 compared to KNN which achieves an accuracy of 92.47% and an AUC of 0.82. Naive Bayes is more stable in recall on spam class (89%), while KNN has perfect precision (100%) but low recall (44%).

.

1. PENDAHULUAN

Perkembangan teknologi saat ini telah mengalami kemajuan yang signifikan, terutama dalam bidang internet. Melalui internet, informasi dan berita dapat diakses serta diterima oleh masyarakat secara luas. Bahkan, internet memungkinkan individu untuk saling mengirim dan menerima pesan secara digital melalui fasilitas yang dikenal sebagai email. Namun, tidak semua orang menggunakan email dengan bijaksana, sehingga dapat menimbulkan dampak negatif bagi pihak lain. Salah satu masalah utama yang muncul adalah spam email, yaitu pesan yang tidak diinginkan dan berpotensi mengganggu kenyamanan pengguna.

Email sendiri merupakan entitas penting dalam komunikasi digital. Meskipun perannya sangat signifikan sebagai alat komunikasi daring, penggunaannya tidak terlepas dari tantangan, terutama terkait dengan kelebihan informasi dan pengelolaan pesan [1]. Selain digunakan untuk mengirimkan teks, email juga kerap dimanfaatkan untuk beriklan maupun mengirimkan file data. Saat ini, komunikasi melalui media elektronik menjadi salah satu cara paling efektif, dengan syarat utama adanya koneksi internet [2].

Untuk mengatasi persoalan seperti spam, diperlukan suatu pendekatan yang mampu mengelompokkan atau memisahkan email berdasarkan jenisnya, misalnya membedakan antara pesan normal dengan pesan spam. Pendekatan ini dikenal dengan klasifikasi. Secara umum, klasifikasi adalah teknik untuk membentuk model dari data pelatihan, yang kemudian digunakan untuk menganalisis data baru dan menentukan kelas yang sesuai [3]. Klasifikasi dapat dilakukan secara manual oleh manusia, namun seiring meningkatnya jumlah data, metode manual menjadi tidak efisien. Oleh karena itu, bantuan teknologi melalui berbagai algoritma klasifikasi sangat dibutuhkan. Beberapa algoritma yang umum digunakan antara lain Naïve Bayes, Support Vector Machine (SVM), dan Decision Tree [4].

Dalam konteks deteksi spam email, berbagai algoritma klasifikasi telah banyak diterapkan, seperti Decision Tree, K-Nearest Neighbor (KNN), Naïve Bayes, ID3, dan C4.5. Di antara teknik-teknik tersebut, Naïve Bayes dikenal sebagai pendekatan statistik yang sederhana namun efektif, dengan tingkat kesalahan rendah dan akurasi tinggi. Algoritma ini bekerja dengan memanfaatkan probabilitas dan frekuensi kata dalam dokumen untuk menentukan apakah sebuah email termasuk spam atau bukan [5].

Selain itu, K-Nearest Neighbors (KNN) juga merupakan algoritma pembelajaran mesin yang sering digunakan dalam klasifikasi teks. KNN bekerja dengan mengidentifikasi sejumlah tetangga terdekat dalam ruang fitur untuk menentukan kelas suatu data. Keunggulan utama KNN adalah kesederhanaannya sekaligus

Jurnal Informatika Dan Rekayasa Komputer (JAKAKOM)
Available Online at https://ejournal.unama.ac.id/index.php/jakakom
Volume 5, Nomor 2, September 2025,
ISSN 2808-5469 (media cetak), ISSN 2808-5000 (media online)
UNAMA, DOI 10.33998/jakakom.v5i2

kemampuannya mencapai akurasi tinggi. Dalam aplikasi seperti identifikasi email spam, KNN dapat menentukan kategori email berdasarkan fitur tertentu, misalnya frekuensi kemunculan kata kunci. Meskipun demikian, performa KNN sangat bergantung pada pemilihan nilai k yang tepat serta kualitas praproses data [6].

Term Frequency inverse Document Frequncy (TFIDF) adalah metode yang populer dalam reprentasi teks untuk analisis sentimen dan klasifikasi dokumen [7]. Metode ini mengukur pentingnya kata dalam dokumen relatif terhadap frekuensi kata tersebut dalam seluruh koleksi dokumen, sehingga membantu dalam menentukan kata-kata yang paling representasif [8]. Penelitian terbaru menunjukkan bahwa TFIDF dapat meningkatkan akurasi klasifikasi dalam teks, terutama ketika digunakan bersama dengan teknik lain seperti Naive Bayes dan Latent Dirichlet Allocation(LDA)[8]. Misalnya, Kim dan Gil (2019) mengembangkan sistem klasifikasi artikel penelitian berdasarkan TFIDF dan LDA, yang menunjukkan peningkatan signifikan dalam pengelompokan artiker berdasarkan topik yang serupa.

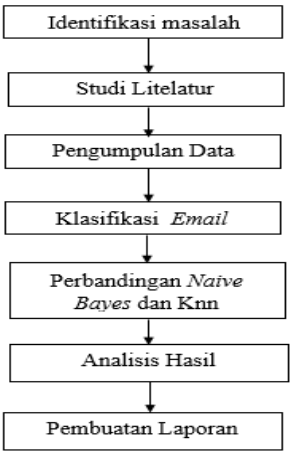
Penelitian sejenis menyampaikan hasil terkait penelitian tentang algoritma Naive Bayes dan K-Nearest Neighbor yaitu Algoritma Naive Bayes memberikan klasifikasi yang lebih baik dari pada algoritma C.45 dalam hal penentuan usia kelahiran. Algoritma Naive Bayes juga memberikan nilai yang lebih baik dalam hal ketepatan saat menentukan kelayakan calon anggota kredit koperasi, tetapi algoritma C.45 memberikan hasil yang lebih baik dalam hal recall dan ketepatan [8]. Pada penelitian terdahulu sudah ada beberapa yang mengembangkan metode ini salah satunya yaitumenghasilkan bahwa algoritma hasil K-Nearest Neighbor kinerja klasifikasi terbaik dicapai dengan persentase accuracy sebesar 72.91% dan precision mencapai 73,36% [9]. Kemudian ada yang menunjukan bahwa algoritma Naïve Bayes dan K-Nearest Neighbor dapat digunakan dalam mengklasifikasi teks, Pada penelitian ini akurasi k-Nearest Neighbor lebih baik dengan akurasi 80% dibandingkan Naïve Bayes yang mendapatkan akurasi 73% [10]. Hasil klasifikasi model Naive Bayes lebih baik daripada model KNN. Nilai ketepatan, yang merupakan tingkat ketepatan prediksi suatu sistem dengan menghitung prediksi positif dari total data yang diprediksi sistem, termasuk prediksi salah, diperoleh dari pengujian Naive Bayes dengan nilai 63,21%, sedangkan KNN dengan nilai 53,10% [11]. Membandingkan dua algoritma klasifikasi, naif bayes dan k-nn, dengan nilai k = 1 - k = 7, menunjukkan bahwa metode klasifikasi k-NN memberikan akurasi terbaik sebesar 85,57% pada k = 3 dan k = 5, dan presisi dan recall k-NN mulai stabil pada k = 7 dan seterusnya. Untuk naif bayes, nilai presisi tertinggi adalah 88,00% [12].

Dari uraian di atas, maka penelitian akan berfokus pada perbandigan algoritma Naive Bayes dan K-Nearest Neighbor pada klasifikasi spam email . Dengan judul penelitian "Perbandingan Algoritma Naive Bayes Dan K-Nearest Neighbor Pada Klasifikasi Email ".

2. METODOLOGI PENELITIAN

2.1 Kerangka Kerja Penelitian

Untuk membantu mempersiapkan penelitian ini diperlukan kerangka kerja yang jelas, langkah-langkah dalam kerangka ini adalah langkah-langkah yang akan dilakukan untuk mengatasi permasalahan yang akan dibahas. Adapun kerangka kerja penelitian yang digunakan adalah sebagai berikut.



Gambar 1. Blok Kerangka Kerja Penelitian

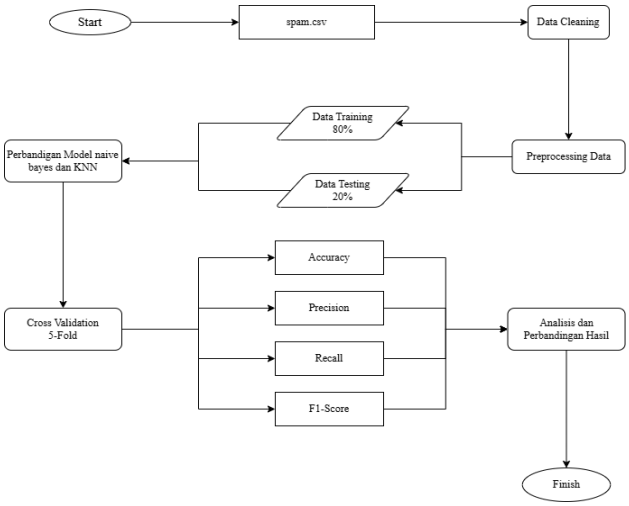
Berdasarkan kerangka kerja di atas maka dapat diuraikan pembahasan masing-masing tahap dalam penelitian adalah sebagai berikut :

Jurnal Informatika Dan Rekayasa Komputer (JAKAKOM)
Available Online at https://ejournal.unama.ac.id/index.php/jakakom
Volume 5, Nomor 2, September 2025,
ISSN 2808-5469 (media cetak), ISSN 2808-5000 (media online)
UNAMA, DOI 10.33998/jakakom.v5i2

1. Identifikasi Masalah
   Penelitian dimulai dengan mengidentifikasi masalah utama, yaitu tingginya jumlah spam yang mengganggu aktivitas digital, serta menentukan relevansi perbandingan klasifikasi spam dengan algoritma berbasis machine learning. Pertanyaan penelitian difokuskan pada evaluasi performa kedua algoritma, penerapan fitur TF-IDF, serta efektivitas dan efisiensi masing-masing metode.
2. Studi Litelatur
   Langkah selanjutnya adalah melakukan studi literatur untuk menganalisis teori terkait algoritma *Naive Bayes* dan KNN, serta metode TF-IDF sebagai teknik representasi teks. Selain itu, penelitian sebelumnya yang relevan ditinjau untuk mengidentifikasi kelebihan dan kekurangan metode yang digunakan, sekaligus menjadi dasar ilmiah untuk penelitian ini.
3. Pengumpulan Data
   Data dikumpulkan dari *dataset email* yang telah diberi label spam dan bukan spam, diambil dari kaggel. *Dataset* ini diproses lebih lanjut melalui langkah praproses data, termasuk penghapusan *stopwords*, normalisasi teks, dan representasi data menggunakan TF-IDF. Tahapan ini bertujuan untuk memastikan bahwa data siap untuk analisis lebih lanjut.
4. Klasifikasi *Email*
   Pada tahap klasifikasi, algoritma *Naive Bayes* dan KNN diterapkan pada *dataset*. Parameter algoritma, seperti nilai kkk pada KNN, dioptimalkan untuk mencapai hasil terbaik. *Dataset* dibagi menjadi set latih dan set uji untuk memastikan evaluasi performa yang obyektif.
5. Perbandingan *Naive Bayes* dan *K-Nearest Neighbor*.
   Hasil perbandingan dievaluasi menggunakan metrik seperti akurasi, *precision*, *recall*, dan F1-score. Membandingkan performa kedua algoritma berdasarkan hasil pengujian.
6. Analisis Hasil
   Setelah memperoleh hasil, penelitian berfokus pada analisis untuk membandingkan kekuatan dan kelemahan kedua algoritma. Analisis ini bertujuan untuk menentukan algoritma yang lebih efektif dalam konteks perbandingan klasifikasi *email*, serta mengidentifikasi implikasi temuan terhadap pengembangan teknologi perbandingan klasifikasi teks.
7. Pembuatan Laporan
   Setelah penelitian selesai, laporan dibuat yang mencakup latar belakang, metode penelitian, hasil evaluasi, analisis, kesimpulan, dan saran. Untuk mendukung transparansi dan akurasi laporan, dokumen tambahan, seperti *dataset* atau kode program, dilampirkan. Diharapkan bahwa hasil penelitian ini akan memberikan wawasan tentang algoritma yang paling cocok untuk membandingkan klasifikasi email dan berfungsi sebagai referensi untuk penelitian berikutnya.

2.2 Alur Eksperimen

Untuk mendukung pelaksanaan eksperimen, diperlukan suatu alur eksperimen yang terstruktur guna memfasilitasi proses pengujian dan evaluasi. Alur eksperimen ini berfungsi sebagai panduan yang sistematis dalam melaksanakan setiap tahap eksperimen, sehingga memungkinkan peneliti untuk memperoleh hasil yang terukur. Adapun alur eksperimen yang digunakan dalam penelitian ini dapat dilihat pada gambar berikut :



Gambar 2. Alur Eksperimen

3. HASIL DAN PEMBAHASAN

Jurnal Informatika Dan Rekayasa Komputer (JAKAKOM)
Available Online at https://ejournal.unama.ac.id/index.php/jakakom
Volume 5, Nomor 2, September 2025,
ISSN 2808-5469 (media cetak), ISSN 2808-5000 (media online)
UNAMA, DOI 10.33998/jakakom.v5i2

Machine Learning bidang kecerdasan buatan, atau pembelajaran mesin, adalah bidang ilmiah yang mencakup pembuatan algoritme yang memungkinkan komputer belajar dari data empiris, seperti database data sensor. Sebagai bidang kecerdasan buatan yang berkembang pesat, pembelajaran mesin telah meningkatkan efisiensi dalam menyelesaikan masalah klasifikasi, regresi, pengelompokan, dan deteksi anomali di berbagai domain. belajar dari data untuk menjadi lebih cerdas [13].

3.1 *Algoritma Naïve Bayes*

Naïve Bayes Classifier untuk menyelesaikan kasus Supervised Learning-di mana Label, Kelas, atau Target digunakan sebagai referensi dalam kumpulan data-Pengklasifikasi *Naïve Bayes* menggunakan statistik dan teori probabilitas [14]. Keuntungan menggunakan metode *Naive Bayes* adalah hanya membutuhkan sedikit data pelatihan untuk menentukan estimasi parameter yang dibutuhkan dalam proses klasifikasi, dan dapat bekerja lebih baik dalam situasi dunia nyata yang kompleks karena hal ini [15].

3.1.1 Pelatihan Algoritma *Naive Bayes*

Pelatihan prosedur klasifikasi probabilistik yang mudah dipahami tetapi efektif yang dikenal sebagai model *Naive Bayes*. Proses pelatihan ini dijalankan pada platform Google Colab dengan menggunakan bahasa pemrograman Python dan beberapa pustaka utama, termasuk Pandas untuk manipulasi data, *NumPy* untuk operasi numerik, dan *Matplotlib* untuk visualisasi data. Model ini dilatih pada data yang telah diproses sebelumnya untuk menemukan pola dan hubungan antara karakteristik dan label kelas. Tujuan pelatihan ini adalah untuk memungkinkan model menggunakan pengetahuan yang telah diperolehnya untuk memperkirakan kelas data baru dengan benar. Gambar berikut. menunjukkan kode yang digunakan untuk melatih model *Naive Bayes*.

```
# Model Naive Bayes
nb = MultinomialNB()
nb.fit(X_train_tfidf, y_train)
y_pred_nb = nb.predict(X_test_tfidf)
```

Gambar 3. Kode Pelatihan Algoritma Naive Bayes

Pada gambar diatas proses pelatihan model *Naive Bayes* pada kode tersebut dimulai dengan inisialisasi model Multinomial Naive Bayes menggunakan kelas MultinomialNB dari library scikit-learn, yang cocok untuk data dengan distribusi multinomial seperti teks yang telah diolah. Selanjutnya, model dilatih menggunakan data pelatihan yang telah dikonversi menjadi representasi numerik melalui metode TF-IDF (*Term Frequency-Inverse Document Frequency*), yang disimpan dalam variabel X_train_tfidf, dan label targetnya dalam y_train. Proses pelatihan ini dilakukan dengan metode fit, di mana model mempelajari hubungan antara data teks dalam bentuk vektor dan label yang sesuai. Setelah pelatihan selesai, model digunakan untuk memprediksi label dari data uji, yang juga telah diubah menjadi vektor TF-IDF (X_test_tfidf), dengan menggunakan metode predict. Hasil prediksi ini disimpan dalam variabel y_pred_nb. Proses ini memungkinkan model untuk membuat prediksi berdasarkan pola yang telah dipelajari selama pelatihan.

3.1.2 Evaluasi *Naive Bayes*

Tujuan dari tahap evaluasi kinerja adalah untuk mengevaluasi kemampuan model *Naive Bayes* untuk mengklasifikasikan data baru. Ini dicapai dengan membandingkan label data pengujian dengan prediksi model. Beberapa ukuran evaluasi, termasuk akurasi, presisi, perolehan kembali, dan skor F1, digunakan untuk mengukur seberapa baik model dapat mengidentifikasi data. Untuk menjalankan prosedur evaluasi ini, kode yang ditunjukkan pada Gambar berikut.

```
data['label'] = data['label'].map({'ham': 0, 'spam': 1})
accuracy = accuracy_score(y_test, y_pred_nb) # Changed y_pred to y_pred_nb
precision = precision_score(y_test, y_pred_nb, pos_label=1) # Changed y_pred to y_pred_nb
recall = recall_score(y_test, y_pred_nb, pos_label=1) # Changed y_pred to y_pred_nb
f1 = f1_score(y_test, y_pred_nb, pos_label=1) # Changed y_pred to y_pred_nb

print("=== Evaluation Metrics for Naive Bayes ===")
print(f"Accuracy: {accuracy:.2f}")
print(f"Precision: {precision:.2f}")
print(f"Recall: {recall:.2f}")
print(f"F1-Score: {f1:.2f}")
```

Gambar 4. Kode Evaluasi Kinerja Naïve Bayes

Pada gambar atas, kode tersebut dirancang untuk mengevaluasi kinerja model *Naive Bayes* dengan menggunakan empat metrik evaluasi utama: akurasi, presisi, recall, dan F1-Score. Langkah awal dalam proses ini adalah mengonversi label kategori pada kolom *label* dari dataset menjadi format numerik, di mana kategori *ham* direpresentasikan sebagai 0 dan *spam* sebagai 1. Konversi ini diperlukan agar label dapat digunakan dalam

Available Online at https://ejournal.unama.ac.id/index.php/jakakom
Volume 5, Nomor 2, September 2025,
ISSN 2808-5469 (media cetak), ISSN 2808-5000 (media online)
UNAMA, DOI 10.33998/jakakom.v5i2

penghitungan metrik evaluasi.

Selanjutnya, **akurasi** dihitung menggunakan fungsi `accuracy_score`, yang menggambarkan proporsi prediksi yang benar terhadap keseluruhan data uji. **Presisi** dihitung dengan fungsi `precision_score`, yang menunjukkan proporsi prediksi *spam* yang benar dari seluruh prediksi yang dikategorikan sebagai *spam*. Pada perhitungan ini, kategori *spam* dianggap sebagai label positif. **Recall** dihitung dengan fungsi `recall_score` untuk mengukur seberapa banyak data yang benar-benar termasuk kategori *spam* berhasil diidentifikasi oleh model secara tepat. Terakhir, **F1-Score** dihitung sebagai rata-rata harmonis antara presisi dan recall untuk memberikan keseimbangan antara kedua metrik, terutama dalam kasus ketidakseimbangan kelas antara kategori positif dan negatif.

Hasil evaluasi berupa nilai akurasi, presisi, recall, dan F1-Score disajikan dalam format dua angka desimal. Penyajian ini bertujuan untuk memberikan deskripsi yang lebih jelas dan ringkas mengenai performa model secara keseluruhan.

3.1.3 Laporan Klasifikasi *Naive Bayes*

Setelah menghitung metrik evaluasi seperti akurasi, precision, recall, dan F1-score, langkah selanjutnya adalah menyusun laporan klasifikasi untuk memberikan gambaran yang lebih komprehensif mengenai kinerja model *Naive Bayes* dalam mengklasifikasikan data. Laporan ini menyajikan metrik evaluasi secara rinci untuk setiap kelas, termasuk precision, recall, F1-score, dan jumlah data yang digunakan (support), sehingga mempermudah interpretasi terhadap kemampuan model dalam menangani masing-masing kelas. Hasil laporan klasifikasi naive bayes dapat dilihat pada gambar berikut.

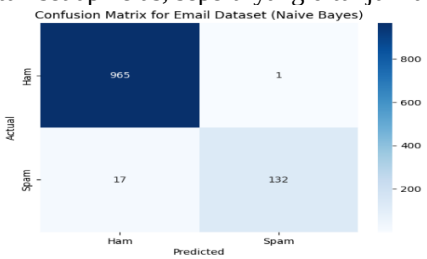
```
=== Naive Bayes ===
Accuracy: 0.9838565022421525
              precision    recall  f1-score   support

           0       0.98      1.00      0.99       966
           1       0.99      0.89      0.94       149

    accuracy                           0.98      1115
   macro avg       0.99      0.94      0.96      1115
weighted avg       0.98      0.98      0.98      1115
```

Gambar 5. Laporan Klasifikasi Naive Bayes

Pada gambar di atas hasil evaluasi menunjukkan bahwa model *Naive Bayes* memiliki akurasi keseluruhan sebesar 98.38%, yang berarti sebagian besar prediksi yang dilakukan oleh model adalah benar. Pada kelas ham (label 0), model mencapai precision sebesar 0.98, recall sempurna 1.00, dan F1-Score 0.99, menunjukkan performa yang sangat baik dalam mengidentifikasi data ham. Sebaliknya, pada kelas spam (label 1), precision mencapai 0.99, tetapi recall lebih rendah, yaitu 0.89, yang berarti ada sekitar 11% data spam yang tidak teridentifikasi oleh model. F1-Score untuk kelas ini adalah 0.94, yang masih menunjukkan kinerja yang cukup baik. Dengan jumlah sampel sebanyak 966 untuk kelas ham dan 149 untuk kelas spam, model lebih unggul dalam menangani kelas dengan lebih banyak data. Rata-rata makro dari precision, recall, dan F1-Score adalah masing-masing 0.99, 0.94, dan 0.96, sedangkan rata-rata tertimbang, yang mempertimbangkan jumlah sampel di setiap kelas, adalah 0.98 untuk semua metrik tersebut. Secara keseluruhan, model memiliki performa yang sangat baik, tetapi menunjukkan sedikit kelemahan pada recall untuk kelas spam, yang berarti ada beberapa data spam yang tidak terdeteksi.

3.1.4 Confusion Matrix *Naive Bayes*

Confusion Matrix dalam *Naive Bayes* memberikan gambaran umum tentang kapasitas model untuk membedakan antara kelas hasil confusin matrix ditampilkan sebagai matriks yang menunjukkan jumlah prediksi akurat dan tidak akurat untuk setiap kelas, seperti yang ditunjukkan pada Gambar berikut.



Gambar 6. Confusion Matrix Naive Bayes

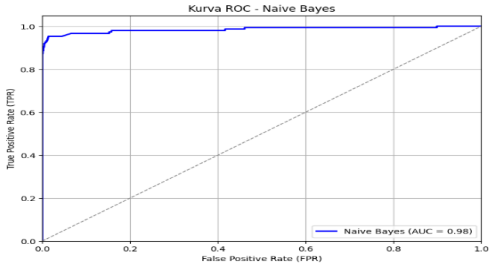
Confusion Matrix pada gambar di atas menunjukkan performa model *Naive Bayes* dalam mengklasifikasikan data menjadi dua kelas, yaitu "Ham" dan "Spam." Matriks ini menunjukkan bahwa sebanyak 132 data "Spam" berhasil diklasifikasikan dengan benar sebagai "Spam" (True Positive), sedangkan 965 data "Ham" berhasil diklasifikasikan dengan benar sebagai "Ham" (True Negative). Selain itu, terdapat hanya 1 data "Ham" yang salah diklasifikasikan sebagai "Spam" (False Positive), namun sebanyak 17 data "Spam" salah diklasifikasikan sebagai "Ham" (False Negative). Dari hasil ini dapat disimpulkan bahwa model

Jurnal Informatika Dan Rekayasa Komputer (JAKAKOM)
Available Online at https://ejournal.unama.ac.id/index.php/jakakom
Volume 5, Nomor 2, September 2025,
ISSN 2808-5469 (media cetak), ISSN 2808-5000 (media online)
UNAMA, DOI 10.33998/jakakom.v5i2

Naive Bayes memiliki performa yang sangat baik dalam mengklasifikasikan data "Ham" dengan tingkat kesalahan yang sangat kecil. Namun, pada data "Spam," model masih menghasilkan beberapa kesalahan berupa False Negative, di mana data "Spam" salah diidentifikasi sebagai "Ham." Secara keseluruhan, model ini menunjukkan kemampuan yang solid dalam menangani klasifikasi pada *dataset* email tersebut.

3.1.5 Kurva ROC/AUC *Naive Bayes*

Hasil kurva ROC/AUC untuk model *Naive Bayes* disajikan pada gabungan-gabungan. Kurva ROC (Receiver Operating Characteristic) menunjukkan hubungan antara Rate Positif Benar (TPR) dan Rate Negatif Benar (FPR), sedangkan AUC (Area Under Curve) menunjukkan seberapa baik model dapat membedakan kedua kelas tersebut secara keseluruhan. Hasil kurva ROC/AUC untuk model *Naive Bayes* dapat dilihat pada Gambar berikut.



Gambar 7. Kurva ROC/AUC Naïve Bayes

Berdasarkan gambar kurva ROC (Receiver Operating Characteristic) untuk model *Naive Bayes*, terlihat bahwa nilai Area Under the Curve (AUC) mencapai 0.98, yang menunjukkan performa model yang sangat baik dalam membedakan antara kelas positif dan negatif. Kurva ROC yang mendekati sudut kiri atas grafik mencerminkan tingkat True Positive Rate (TPR) yang tinggi dan False Positive Rate (FPR) yang rendah, menandakan bahwa model dapat mengidentifikasi data positif dengan akurasi tinggi sambil meminimalkan kesalahan dalam mengenali data negatif. Dengan AUC sebesar 0.98, model *Naive Bayes* memiliki kemampuan diskriminasi yang sangat baik, sehingga mampu membedakan kelas dengan akurat. Secara keseluruhan, model ini menunjukkan kinerja yang sangat baik dalam klasifikasi dan dapat diandalkan untuk *dataset* yang digunakan

3.1.6 Pengujian 5-Fold Cross Validation *naïve bayes*

Pengujian menggunakan 5-fold cross validation dilakukan untuk mengevaluasi performa model *Naive Bayes* secara lebih akurat dengan membagi *dataset* menjadi lima lipatan (fold) yang berbeda. Proses ini bertujuan untuk memastikan bahwa hasil evaluasi tidak bergantung pada satu pembagian *dataset* saja, sehingga memberikan gambaran yang lebih umum tentang kemampuan model dalam melakukan klasifikasi. Hasil dari pengujian ini akan disajikan pada gambar berikut.

```
Cross-Validation Accuracy Scores: [0.98318386 0.98318386 0.97306397 0.97643098 0.97194164]
Mean Accuracy: 0.98
```

Gambar 8. 5-Fold Cross Validation naïve bayes

Berdasarkan hasil cross-validation pada model *Naive Bayes* yang ditampilkan pada gambar, model menunjukkan performa akurasi yang sangat baik dengan nilai yang konsisten di setiap fold. Akurasi model berkisar antara 0.97 hingga 0.98, yang menunjukkan bahwa model mampu mengklasifikasikan data dengan tingkat ketepatan yang tinggi pada berbagai subset data latih dan uji. Rata-rata akurasi sebesar 0.98 mengindikasikan bahwa model *Naive Bayes* memiliki kemampuan yang andal dalam melakukan klasifikasi pada *dataset* ini, dengan tingkat kesalahan yang sangat rendah.

3.2 *K-Nearest Neighbor*

K-Nearest Neighbor, yang menggunakan pembelajaran terawasi, mengklasifikasikan hasil dari data masukan baru berdasarkan terdekat dalam data nilai[16].Dalam kelompok pembelajaran berbasis indikasi, algoritma *K-Nearest Neighbor* (K-NN) adalah salah satu teknik yang termasuk dalam kelompok pembelajaran berbasis indikasi. K-NN bekerja dengan menemukan sejumlah kkk objek dalam data pelatihan yang memiliki tingkat kedekatan atau kemiripan tertinggi dengan objek pada data baru atau data uji. Untuk menerapkan metode ini, diperlukan sistem klasifikasi yang mendukung proses pencarian dan pengambilan informasi dengan baik [17].

3.2.1 Pelatihan Algoritma *K-Nearest Neighbor*

Pelatihan model *K-Nearest Neighbor* (KNN), yang merupakan salah satu algoritma klasifikasi berbasis kedekatan atau jarak, dilakukan dengan pendekatan yang sederhana namun efektif. Proses pelatihan ini

Jurnal Informatika Dan Rekayasa Komputer (JAKAKOM)
Available Online at https://ejournal.unama.ac.id/index.php/jakakom
Volume 5, Nomor 2, September 2025,
ISSN 2808-5469 (media cetak), ISSN 2808-5000 (media online)
UNAMA, DOI 10.33998/jakakom.v5i2

dilakukan pada platform Google Colab menggunakan bahasa pemrograman *Python* dan memanfaatkan beberapa library utama, yaitu Pandas untuk manipulasi data, *NumPy* untuk operasi numerik, dan *Matplotlib* untuk visualisasi data. Model ini dilatih menggunakan data yang telah diproses sebelumnya untuk mempelajari pola distribusi dan hubungan antara data berdasarkan jarak antar titik. Pelatihan ini dilakukan dengan tujuan agar model dapat memprediksi kelas data baru secara akurat dengan membandingkan jaraknya terhadap sejumlah tetangga terdekat pada data pelatihan. Pada Gambar berikut terdapat kode yang digunakan untuk melatih model *K-Nearest Neighbor*.

```
# Model KNN
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train_tfidf, y_train)
y_pred_knn = knn.predict(X_test_tfidf)
```

Gambar 9. Pelatihan Algoritma K-Nearest Neighbor

Kode tersebut menunjukkan proses pelatihan dan prediksi menggunakan algoritma K-Nearest Neighbor (KNN) dengan KNeighborsClassifier dari scikit-learn, di mana model mempertimbangkan 5 tetangga terdekat (n_neighbors=5). Model dilatih dengan fit() menggunakan data X_train_tfidf dan y_train, lalu melakukan prediksi dengan predict() pada X_test_tfidf. KNN tidak membangun model probabilistik, melainkan menyimpan data pelatihan dan menentukan kelas berdasarkan mayoritas label dari 5 tetangga terdekat. Implementasi ini menunjukkan pendekatan sederhana namun efektif dalam klasifikasi.

3.2.2 Evaluasi *K-Nearest Neighbor*

Pada tahap evaluasi kinerja model *K-Nearest Neighbor* (KNN), dilakukan penilaian untuk menilai seberapa baik model dalam mengklasifikasikan data baru. Proses ini dilakukan dengan membandingkan hasil prediksi model dengan label sebenarnya dari data pengujian. Metrik evaluasi yang digunakan meliputi akurasi, precision, recall, dan F1-score untuk mengukur sejauh mana model mampu melakukan klasifikasi dengan benar. Proses evaluasi ini dilakukan dengan menggunakan kode yang ditampilkan pada Gambar berikut.

```
# Evaluasi untuk KNN
accuracy_knn = accuracy_score(y_test, y_pred_knn)
precision_knn = precision_score(y_test, y_pred_knn, pos_label=1)
recall_knn = recall_score(y_test, y_pred_knn, pos_label=1)
f1_knn = f1_score(y_test, y_pred_knn, pos_label=1)

# Menampilkan hasil evaluasi
print("=== Evaluation Metrics for K-Nearest Neighbors (KNN) ===")
print(f"Accuracy: {accuracy_knn:.4f}")
print(f"Precision: {precision_knn:.4f}")
print(f"Recall: {recall_knn:.4f}")
print(f"F1-Score: {f1_knn:.4f}")
```

Gambar 10. kode evaluasi kinerja K-Nearest Neighbor

Kode tersebut menunjukkan proses evaluasi model *K-Nearest Neighbor* (KNN) menggunakan metrik evaluasi seperti akurasi, precision, recall, dan F1-score. Akurasi dihitung menggunakan fungsi accuracy_score, yang mengukur persentase prediksi yang benar dibandingkan dengan total prediksi. Precision dihitung menggunakan fungsi precision_score, yang mengukur proporsi data yang diprediksi sebagai positif (kelas 1) yang benar-benar positif, dengan parameter pos_label=1 untuk menentukan label positif. Recall dihitung menggunakan fungsi recall_score, yang menunjukkan sejauh mana model mampu menemukan semua data positif yang sebenarnya. Selanjutnya, F1-score dihitung menggunakan fungsi f1_score, yang merupakan rata-rata harmonis antara precision dan recall, memberikan gambaran seimbang tentang kinerja model, terutama pada *dataset* dengan ketidakseimbangan kelas. Hasil evaluasi ini kemudian ditampilkan dalam format angka desimal hingga empat digit di belakang koma untuk memberikan gambaran yang rinci tentang performa model KNN dalam klasifikasi data.

3.2.3 **Laporan Klasifikasi *K-Nearest Neighbor***

Setelah melaksanakan proses pelatihan menggunakan algoritma *K-Nearest Neighbor*, langkah selanjutnya adalah menyusun laporan hasil pelatihan untuk memberikan gambaran yang lebih komprehensif mengenai kinerja model dalam melakukan klasifikasi. Laporan ini menyajikan metrik evaluasi secara rinci, seperti akurasi, precision, recall, dan F1-score, serta mengidentifikasi performa model terhadap masing-masing kelas dalam data. Hasil laporan pelatihan *K-Nearest Neighbor* dapat dilihat pada gambar berikut.

```
=== K-Nearest Neighbors ===
Accuracy: 0.9246636771300448
              precision    recall  f1-score   support

           0       0.92      1.00      0.96       966
           1       1.00      0.44      0.61       149

    accuracy                           0.92      1115
   macro avg       0.96      0.72      0.78      1115
weighted avg       0.93      0.92      0.91      1115
```

Gambar 11. Laporan Klasifikasi K-Nearest Neighbor

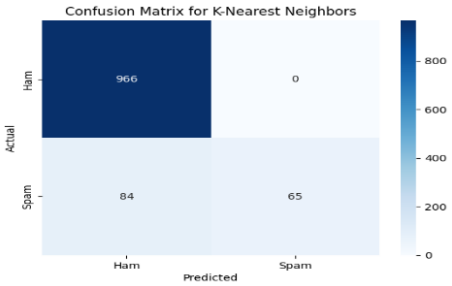
Model K-Nearest Neighbors (KNN) memiliki akurasi keseluruhan sebesar 92.47%, dengan kinerja yang sangat baik pada kelas 0 (precision 0.92, recall 1.00, F1-score 0.96). Namun, pada kelas 1, meskipun precision

Jurnal Informatika Dan Rekayasa Komputer (JAKAKOM)
Available Online at https://ejournal.unama.ac.id/index.php/jakakom
Volume 5, Nomor 2, September 2025,
ISSN 2808-5469 (media cetak), ISSN 2808-5000 (media online)
UNAMA, DOI 10.33998/jakakom.v5i2

mencapai 1.00, recall yang rendah (0.44) menyebabkan F1-score turun menjadi 0.61, menunjukkan kesulitan model dalam mengenali data kelas ini akibat ketidakseimbangan data. Evaluasi macro average menunjukkan precision 0.96, recall 0.72, dan F1-score 0.78, sedangkan weighted average mencerminkan dominasi kelas mayoritas dengan precision 0.93, recall 0.92, dan F1-score 0.91. Untuk meningkatkan performa kelas minoritas, disarankan teknik seperti oversampling, undersampling, pemberian bobot, atau penyesuaian parameter k. Jika hasil masih kurang optimal, dapat dipertimbangkan penggunaan algoritma alternatif.

3.2.4 Confusion Matrix *K-Nearest Neighbor*

Confusion Matrix pada *K-Nearest Neighbor* disajikan dalam bentuk matriks yang menunjukkan jumlah prediksi yang benar dan salah untuk setiap kelas, sehingga memberikan gambaran tentang kemampuan model dalam membedakan antara kelas-kelas yang berbeda; Confusion Matrix ini dapat dilihat pada Gambar berikut.

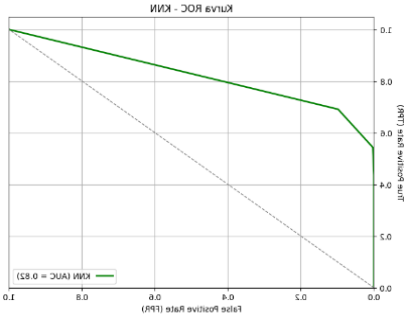


Gambar 12. Confusion Matrix K-Nearest Neighbor

Model K-Nearest Neighbors (KNN) menunjukkan kemampuan yang cukup baik dalam membedakan kelas positif dan negatif dengan nilai AUC sebesar 0.82, yang masuk dalam kategori baik meskipun tidak sempurna. Kurva ROC mengindikasikan bahwa model dapat mencapai True Positive Rate (TPR) yang tinggi, tetapi juga mengalami peningkatan False Positive Rate (FPR). Secara keseluruhan, KNN efektif dalam klasifikasi data, namun masih dapat ditingkatkan melalui penyesuaian parameter jumlah tetangga (n_neighbors) atau preprocessing data yang lebih optimal.

3.2.5 **Kurva ROC/AUC *K-Nearest Neighbor***

Kurva ROC/AUC pada *K-Nearest Neighbors* (K-NN) disajikan untuk menggambarkan kemampuan model dalam membedakan antara kelas positif dan negatif di berbagai tingkat threshold. ROC (Receiver Operating Characteristic) menunjukkan hubungan antara True Positive Rate (TPR) dan False Positive Rate (FPR), sedangkan AUC (Area Under Curve) mengukur seberapa baik model dapat mengklasifikasikan kedua kelas tersebut secara keseluruhan yang dapat dilihat pada Gambar 4.16



Gambar 13. Kurva ROC/AUC K-Nearest Neighbor

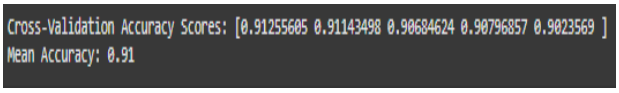
Berdasarkan kurva ROC (Receiver Operating Characteristic) untuk model *K-Nearest Neighbors* (KNN), model menunjukkan kemampuan yang cukup baik dalam membedakan antara kelas positif dan negatif. Nilai AUC (Area Under Curve) sebesar 0.82 mengindikasikan bahwa performa model berada dalam kategori baik, meskipun tidak sempurna. Kurva ROC menggambarkan hubungan antara True Positive Rate (TPR) dan False Positive Rate (FPR) pada berbagai threshold, dengan hasil yang menunjukkan bahwa model mampu mencapai TPR yang tinggi meskipun disertai peningkatan FPR. Secara keseluruhan, model KNN memiliki efektivitas yang baik dalam mengklasifikasikan data, namun masih ada ruang untuk perbaikan lebih lanjut, seperti dengan menyesuaikan parameter jumlah tetangga (n_neighbors) atau melakukan *preprocessing* data yang lebih optimal untuk meningkatkan performa model.

3.2.6 Pengujian 5-Fold Cross Validation *K-Nearest Neighbor*

Pengujian menggunakan 5-fold cross-validation dilakukan untuk mengevaluasi performa model *K-*

Available Online at https://ejournal.unama.ac.id/index.php/jakakom
Volume 5, Nomor 2, September 2025,
ISSN 2808-5469 (media cetak), ISSN 2808-5000 (media online)
UNAMA, DOI 10.33998/jakakom.v5i2

Nearest Neighbors secara lebih akurat dengan membagi *dataset* menjadi lima lipatan (fold) yang berbeda. Proses ini bertujuan untuk memastikan bahwa hasil evaluasi tidak bergantung pada satu pembagian *dataset* saja, sehingga memberikan gambaran yang lebih umum tentang kemampuan model *K-Nearest Neighbors* dalam melakukan klasifikasi. Dengan metode ini, setiap subset data digunakan sebagai data uji secara bergantian, sementara subset lainnya digunakan sebagai data latih, sehingga menghasilkan evaluasi yang lebih stabil dan reliabel terhadap performa model. Hasil dari pengujian ini memberikan gambaran yang lebih menyeluruh mengenai kinerja model *K-Nearest Neighbors* dalam menangani data klasifikasi. hasil dari pengujian ini akan disajikan pada gambar 4.17.



Gambar 14. 5-Fold Cross Validation K-Nearest Neighbor

Berdasarkan hasil cross-validation pada model *K-Nearest Neighbor* yang ditampilkan pada gambar, akurasi model berada dalam rentang 0.90 hingga 0.91 pada setiap fold. Hal ini menunjukkan bahwa model memiliki performa yang cukup stabil dan konsisten dalam melakukan klasifikasi pada berbagai subset data latih dan uji. Rata-rata akurasi sebesar 0.91 menunjukkan bahwa model *K-Nearest Neighbor* dapat membuat prediksi yang baik dengan tingkat ketepatan yang cukup tinggi.

3.3 Perbandingan *Naive Bayes dan K-Nearest Neighbor*

Dalam penelitian ini, dilakukan perbandingan antara algoritma Naive Bayes dan K-Nearest Neighbor (KNN) dalam klasifikasi email berdasarkan beberapa metrik evaluasi, yaitu akurasi, presisi, recall, F1-score, serta kurva ROC/AUC. Berikut adalah rangkuman dari hasil perbandingan yang telah dilakukan :

1. Perbandingan Akurasi
   Hasil pengujian menunjukkan bahwa algoritma Naive Bayes memiliki akurasi sebesar 98.38%, sedangkan K-Nearest Neighbor (KNN) hanya mencapai 92.46%. Dengan selisih akurasi sebesar 5.92%, Naive Bayes terbukti lebih efektif dalam klasifikasi email dibandingkan KNN.
2. Perbandingan Presisi
   Naive Bayes memiliki presisi lebih tinggi pada kelas 0 (0.98 dibandingkan 0.92 milik KNN), menunjukkan kemampuannya yang lebih baik dalam mengidentifikasi email yang benar-benar termasuk dalam kelas ini. Namun, pada kelas 1, KNN lebih unggul dengan presisi sempurna 1.00 dibandingkan 0.99 pada Naive Bayes, menunjukkan bahwa semua prediksi kelas 1 pada KNN benar tanpa kesalahan.
3. Perbandingan Recall
   Kedua model memiliki recall sempurna sebesar 1.00 untuk kelas 0, tetapi terdapat perbedaan signifikan pada kelas 1. Naive Bayes mencapai recall 0.89, sedangkan KNN hanya 0.44, yang berarti KNN mengalami kesulitan dalam mengidentifikasi data yang benar-benar termasuk dalam kelas 1. Dengan demikian, Naive Bayes lebih andal dalam menangkap seluruh data yang relevan pada kelas 1.
4. Perbandingan F1-Score
   Naive Bayes memiliki F1-score lebih tinggi dibandingkan KNN pada kedua kelas, dengan nilai 0.99 untuk kelas 0 dan 0.94 untuk kelas 1, sedangkan KNN hanya mencapai 0.96 dan 0.61. Perbedaan signifikan pada kelas 1 menunjukkan bahwa Naive Bayes lebih baik dalam menjaga keseimbangan antara presisi dan recall, terutama dalam mengklasifikasikan data pada kelas ini.
5. Perbandingan Kurva ROC/AUC
   Naive Bayes menunjukkan performa yang lebih baik dalam membedakan kelas positif dan negatif, dengan nilai AUC sebesar 0.98 dibandingkan 0.82 pada KNN. Kurva ROC Naive Bayes juga lebih mendekati sudut kiri atas grafik, menunjukkan kemampuannya yang lebih baik dalam meminimalkan kesalahan klasifikasi dan membedakan data dengan lebih akurat.
6. Analisis Kelebihan dan Kekurangan Naive Bayes dan K-Nearest Neighbor
   Naive Bayes unggul dalam kecepatan dan efisiensi komputasi serta bekerja baik dalam klasifikasi teks, tetapi memiliki keterbatasan akibat asumsi independensi antar fitur. Sementara itu, KNN mudah diimplementasikan dan efektif untuk pola data yang kompleks, namun memerlukan komputasi tinggi untuk dataset besar dan sensitif terhadap pemilihan jumlah tetangga (k).

4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, maka dapat dirumuskan beberapa kesimpulan sebagai berikut. *Naive Bayes* menunjukkan akurasi tertinggi sebesar 98.38%. Model ini juga unggul pada metrik precision, recall, dan F1-score, terutama dalam menangani data kelas mayoritas (ham). Kurva

Available Online at https://ejournal.unama.ac.id/index.php/jakakom
Volume 5, Nomor 2, September 2025,
ISSN 2808-5469 (media cetak), ISSN 2808-5000 (media online)
UNAMA, DOI 10.33998/jakakom.v5i2

ROC/AUC Naive Bayes mencapai nilai 0.98, menunjukkan performa klasifikasi yang sangat baik dan hampir sempurna. *K-Nearest Neighbor* memiliki akurasi 92.46%, yang lebih rendah dibandingkan *Naive Bayes*. Meskipun precision pada kelas minoritas (spam) mencapai nilai sempurna 1.00, nilai recall hanya sebesar 0.44, menunjukkan bahwa model kurang mampu mendeteksi spam dengan baik. Kurva ROC/AUC *K-Nearest Neighbor* sebesar 0.82, menunjukkan performa klasifikasi yang baik namun kurang optimal dibandingkan *Naive Bayes*.

REFERENCES

[1] A. H. Awadallah, S. Dumais, And T. Alrashed, "The Lifetime Of Email Messages: A Large-Scale Analysis Of Email Revisitation," In *Chiir 2018 - Proceedings Of The 2018 Conference On Human Information Interaction And Retrieval*, Association For Computing Machinery, Inc, Feb. 2018.

[2] H. Mukhtar, J. Al Amien, And M. A. Rucyat, "Filtering Spam Email Menggunakan Algoritma Naïve Bayes," *Jurnal . (Computer Science And Information Technology)*, Vol. 3, No. 1, Pp. 9–19, May 2022.

[3] P. Putra, A. M. H Pardede, And S. Syahputra, "Analisis Metode K-Nearest Neighbour (Knn) Dalam Klasifikasi Data Iris Bunga," *Jurnal Teknik Informatika Kaputama (Jtik)*, Vol. 6, No. 1, 2022.

[4] A. P. Wibawa, M. Guntur, A. Purnama, M. Fathony Akbar, And F. A. Dwiyanto, "Metode-Metode Klasifikasi," *Prosiding Seminar Ilmu Komputer Dan Teknologi Informasi*, Vol. 3, No. 1, 2018.

[5] A. Wibisono Informatika, "Filtering Spam Email Menggunakan Metode Naive Bayes," *Teknologipintar.Org*, Vol. 3, No. 4, Pp. 2023–2024.

[6] E. Firasari, N. Khasanah, U. Khultsum, D. N. Kholifah, R. Komarudin, And W. Widyastuty, "Comparation Of K-Nearest Neighboor (K-Nn) And Naive Bayes Algorithm For The Classification Of The Poor In Recipients Of Social Assistance," In *Journal Of Physics: Conference Series*, Iop Publishing Ltd, Nov. 2020.

[7] M. M. Danyal, S. S. Khan, M. Khan, S. Ullah, M. B. Ghaffar, And W. Khan, "Sentiment Analysis Of Movie Reviews Based On Nb Using Tf–Idf And Count," *Soc Netw Anal Min*, Vol. 14, No. 1, Dec. 2024.

[8] S. W. Kim And J. M. Gil, "Research Paper Classification Systems Based On Tf-Idf And Lda Schemes," *Human-Centric Computing And Information Sciences*, Vol. 9, No. 1, Dec. 2019

[9] Y. I. Kurniawan, "Perbandingan Algoritma Naive Bayes Dan C.45 Dalam Klasifikasi Data Mining," *Jurnal Teknologi Informasi Dan Ilmu Komputer*, Vol. 5, No. 4, Pp. 455–464, Oct. 2018

[10] H. Ma'rifah, A. P. Wibawa, and M. I. Akbar, "Klasifikasi Artikel Ilmiah Dengan Berbagai Skenario Preprocessing," Sains, Apl. Komputasi dan Teknol.Inf., vol. 2, no. 2, p. 70, 2020, doi:10.30872/jsakti.v2i2.2681..

[11] A. Indriani, "Analisa Perbandingan Metode Naïve Bayes Classifier Dan K-Nearest Neighbor Terhadap Klasifikasi Data," Sebatik, vol. 24, no. 1, pp. 1–7, 2020, doi: 10.46984/sebatik.v24i1.909.

[12] S. A. Pangestu, C. Aminuallah, S. I. Akuntansi, And S. Informasi, "Klasifikasi Opini Publik Tentang Covid-19 Di Twitter Menggunakan Metode Pengklasifikasi Naive Bayes Dan Knn."

[13] R. Marsuciati, G. Gumelar, And R. Prietno, "Klasifikasi Metode Naïve Bayes Dan K-Nearest Neighbor Untuk Menentukan Keluarga Tidak Mampu. 2020

[14] A. A. Hania, "Artificial Intelligence, Machine Learning, Neural Network, Dan Deep Learning," 2017

[15] M. K. Anam And D. A. Jakaria, "Sistem Prediksi Harga Kripto Dengan Metode Regresi," Vol. 10, No. 2, Pp. 467–479, 2023

[16] T. Wahyono, U. Kristen, And S. Wacana, "Fundamental Of Python For Machine Learning: Dasar-Dasar Pemrograman Python Untuk Machine Learning Dan Kecerdasan Buatan," 2018

[17] D. Aplikasinya Tim Penulis *Et Al.*, *Data Mining*. 2021