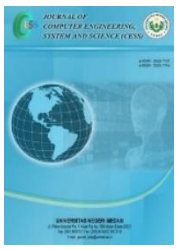
CESS (Journal of Computer Engineering, System and Science)
Vol. 10, No. 2, Juli 2025, pp.389-398

CESS
(Journal of Computer Engineering, System and Science)
Available online: https://jurnal.unimed.ac.id/2012/index.php/cess
ISSN: 2502-714x (Print) | ISSN: 2502-7131 (Online)



Perbandingan Performa Algoritma Gaussian Naive Bayes dan *Decision Tree Classifier* dalam Klasifikasi *Prompt AI-Generated Image*

Performance Comparison of Gaussian Naive Bayes Algorithm and Decision Tree Classifier in Prompt Classification of AI-Generated Image

Agung Riyadi[*], Putri Paramitha[2]

[1,2]Teknik Informatika, Politeknik Negeri Batam, Indonesia
Jl. Ahmad Yani, Tlk. Tering, Kec. Batam Kota, Kota Batam, Kepulauan Riau
Email: [1]agung@polibatam.ac.id, [2]putriparamithaa@gmail.com
**Corresponding Author*

A B S T R A K

Sebuah website yang dikembangkan oleh peneliti memiliki jutaan data prompt dan hasil gambar *AI-generated* menghadapi tantangan seperti penyajian konten yang lambat dan tidak efisien bagi pengguna. Ketiadaan sistem kategorisasi yang tepat menyebabkan proses filtering dan pencarian konten menjadi lambat, sehingga membutuhkan implementasi sistem klasifikasi otomatis untuk meningkatkan kecepatan akses dan *user experience*. Penelitian ini membandingkan performa algoritma Gaussian Naive Bayes dan *Decision Tree Classifier* dalam mengklasifikasikan prompt *text-to-image* ke dalam tiga kategori: *Background/Texture, Landscape*, dan *Arts*. Dataset terdiri dari 7.040 prompt yang telah dikategorikan secara manual. Metodologi mencakup pra-pemrosesan data, representasi teks menggunakan *Bag of Words*, penerapan kedua algoritma klasifikasi, dan evaluasi menggunakan metrik akurasi, *precision, recall*, dan *F1-score*. Hasil menunjukkan bahwa *Decision Tree* mencapai akurasi tertinggi sebesar 99,16%, mengungguli Gaussian Naive Bayes yang hanya memperoleh 61,74%. Temuan ini menunjukkan bahwa *Decision Tree* lebih mampu menangani kompleksitas karakteristik prompt, serta dapat diimplementasikan untuk meningkatkan efisiensi pencarian dan penyaringan konten pada platform generative AI.

Kata Kunci: *Kecerdasan Buatan*; *Klasifikasi Text; Decision Tree Classifier; Gaussian Naive Bayes; Machine Learning*

A B S T R A C T

A website developed by researchers, which hosts millions of prompts and AI-generated images, faces challenges such as slow content delivery and inefficient user experience. The absence of an appropriate categorization system results in sluggish filtering and content search processes, necessitating the implementation of an automatic classification system to

improve access speed and user experience. This study compares the performance of the Gaussian Naive Bayes and Decision Tree Classifier algorithms in classifying text-to-image prompts into three categories: Background/Texture, Landscape, and Arts. The dataset consists of 7,040 prompts that have been manually categorized. The methodolodgy includes data preprocessing, text representation using Bag of Words, implementation of both classification algorithms, and evaluation using accuracy, precision, recall, and F1-score metrics. The results show that the Decision Tree achieved the highest accuracy at 99.16%, outperforming the Gaussian Naive Bayes, which only reached 61.74%. These findings indicate that the Decision Tree is more capable of handling the complexity of prompt characteristics and can be implemented to enhance the efficiency of content search and filtering on generative AI platforms.

Keywords: *AI generative image, Text Classification, Gaussian Naive Bayes, Decision Tree Classifier, Machine Learning*

1. PENDAHULUAN

Revolusi Industri 4.0 telah mengakselerasi perkembangan teknologi *Artificial Intelligence* (AI), khususnya dalam bidang generative AI yang mampu menciptakan konten visual dari deskripsi teks. Teknologi *text-to-image generation* mengalami kemajuan pesat dengan munculnya model-model seperti DALL-E, Midjourney, dan Stable Diffusion yang telah mentransformasi cara pembuatan konten visual[1]. Pasar global untuk AI-generated content diproyeksikan mencapai $178.13 miliar pada tahun 2032, dengan generative AI diperkirakan akan menyumbang hingga 10% dari seluruh data yang diproduksi pada tahun 2025[2].

Seiring berkembangnya teknologi ini, volume data visual yang dihasilkan AI mengalami lonjakan signifikan. Platform generative AI menghasilkan jutaan gambar setiap hari, yang menciptakan tantangan dalam pengelolaan dan pengorganisasian konten visual. Tanpa sistem klasifikasi otomatis yang efisien, repository gambar berskala besar akan sulit diakses dan dimanfaatkan secara optimal. Klasifikasi otomatis menjadi penting tidak hanya untuk efisiensi manajemen konten, tetapi juga untuk meningkatkan pengalaman pengguna dan akurasi pencarian gambar berbasis prompt.

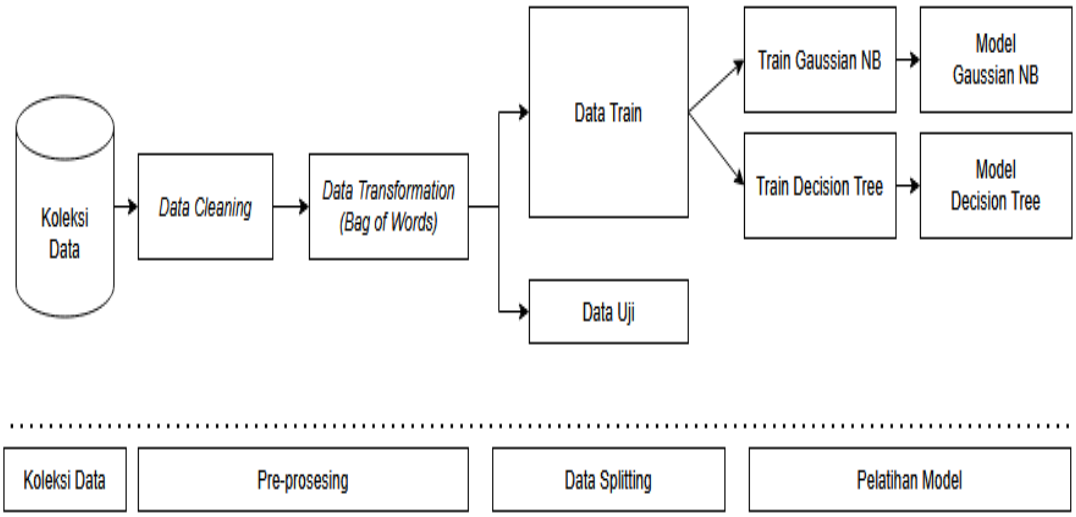
Sejumlah penelitian terdahulu telah mengembangkan pendekatan klasifikasi gambar, namun mayoritas masih menggunakan dataset konvensional seperti CIFAR-10, ImageNet, atau MNIST[3][4]. Penelitian terkait klasifikasi gambar yang secara khusus dihasilkan dari prompt AI masih sangat terbatas. Gambar hasil *text-to-image generation* memiliki karakteristik unik seperti ketidakteraturan komposisi, variasi gaya artistik, serta keragaman semantik dari prompt yang digunakan. Beberapa studi awal mencoba mengkaji metadata prompt atau eksplorasi gaya visual menggunakan CNN dan model transformer, namun belum secara eksplisit membandingkan efektivitas algoritma klasifikasi tradisional untuk domain ini[5].

Penelitian ini mengisi celah tersebut dengan mengevaluasi performa dua algoritma klasifikasi tradisional, yaitu Gaussian Naive Bayes (GNB) dan *Decision Tree Classifier* (DTC)**,** dalam konteks klasifikasi gambar AI-generated. Pemilihan GNB dan DTC didasarkan pada keunggulannya dalam interpretabilitas dan efisiensi pada dataset beragam. Namun, hingga saat ini belum ditemukan studi yang secara komprehensif membandingkan keduanya dalam domain khusus gambar generatif.

Penelitian ini bertujuan untuk membandingkan efektivitas GNB dan DTC dalam mengklasifikasikan gambar AI-generated ke dalam tiga kategori utama**:** *Background/Texture***,** *Landscape***,** dan *Arts*. Menggunakan dataset berisi 7.040 prompt yang telah di akurasi secara manual dari database prompt publik, evaluasi dilakukan berdasarkan metrik akurasi, *precision, recall,* dan *F1-score*. Hasil analisis diharapkan dapat memberikan kontribusi dalam pengembangan sistem klasifikasi konten visual generatif dan menjadi panduan praktis dalam penerapan machine learning pada platform generative AI skala besar.

2. METODE PENELITIAN

Data yang digunakan dalam penelitian ini diambil dari website *AI Generate Image*. Data tersebut mencakup tiga kategori, yaitu background/texture**,** landscape, dan art, dengan total sebanyak 7040 data. Setelah melalui proses pra-pemrosesan dan pembagian data, dilakukan pelatihan model menggunakan dua algoritma klasifikasi, yaitu *Gaussian Naive Bayes* dan *Decision Tree*. Hasil dari pelatihan ini adalah dua model yang siap digunakan untuk melakukan prediksi dan evaluasi, yaitu *Model Gaussian NB* dan *Model Decision Tree*. Beberapa tahap yang dilakukan dalam penelitian ini tergambar secara jelas pada diagram di bawah ini.



Gambar 1. Alur Pembuatan Model Gaussian dan Decision Tree

Gambar di atas menunjukkan alur proses klasifikasi teks menggunakan metode *Gaussian Naive Bayes dan Decision Tree*, yang disusun dalam tahapan sistematis. Dimulai dari pengumpulan data (Koleksi Data)**,** data melalui proses pra-pemrosesan yang meliputi *pembersihan data* (Data Cleaning) dan *transformasi data* menggunakan teknik Bag of Words. Selanjutnya, data dibagi menjadi dua bagian: data latih (Data Train) dan data uji (Data Uji)**.**

2.1. Tahap Pre-processing

Tahap preprocessing dilakukan untuk mempersiapkan data sebelum analisis. Proses ini mencakup beberapa Langkah diantaranya data cleaning, dan transformasi data.

2.1.1 Data Cleaning

Langkah pertama dalam mempersiapkan dataset adalah memastikan data yang akan digunakan bersih dan berkualitas. Data preprocessing merupakan tahap fundamental dalam

machine learning yang secara signifikan mempengaruhi performa model, dengan studi menunjukkan bahwa kualitas data memiliki dampak langsung terhadap akurasi dan reliabilitas hasil klasifikasi[3][4]. Pada tahap ini, data yang tidak lengkap, duplikat, atau tidak relevan dihapus dari dataset. Data yang tidak lengkap biasanya muncul dalam bentuk baris yang memiliki nilai kosong atau atribut yang hilang, yang dapat mempengaruhi hasil klasifikasi jika tidak ditangani dengan benar. Penelitian terbaru menunjukkan bahwa penanganan missing values merupakan aspek kritis dalam data preprocessing, dimana nilai yang hilang atau tidak lengkap dapat menyebabkan bias dalam model machine learning dan menurunkan kemampuan generalisasi[7][8]. Oleh karena itu, entri semacam ini dihapus atau diperbaiki dengan nilai yang sesuai.

Selain itu, baris duplikat juga dihapus karena dapat menyebabkan model machine learning mengalami overfitting. Duplikasi data dapat menyebabkan model menjadi terlalu terspesialisasi pada subset data tertentu dan gagal menangkap pola yang diperlukan untuk performa yang baik pada data yang belum pernah dilihat sebelumnya[5]. Studi dalam jurnal Nature Communications (2023) mengungkapkan bahwa redundansi dalam dataset besar dapat mencapai hingga 95%, dimana penghapusan data duplikat tidak berdampak negatif pada akurasi prediksi[6]. Penelitian lebih lanjut menunjukkan bahwa duplikasi data, terutama exact duplicates dan near duplicates, dapat menyebabkan overfitting dimana model menjadi terlalu terspesialisasi pada training set dan gagal dalam generalisasi[7].

Data yang tidak relevan, seperti entri yang tidak sesuai dengan kategori penelitian yang diinginkan (background/texture, landscape, art), juga dihilangkan, sehingga dataset lebih sesuai dengan tujuan klasifikasi yang akan dilakukan. Proses filtering data ini sangat penting untuk memastikan konsistensi dan relevansi dataset dengan objektif penelitian, yang pada akhirnya akan meningkatkan kualitas dan akurasi model klasifikasi.

Tahapan ini dapat dilakukan secara manual yaitu dengan menggunakan rumus penghapusan data duplikat pada Excel, dikarenakan data awalnya masih berupa tabel pada Excel yang mana diberi angka 0 sebagai tanda untuk data Art, angka 1 untuk data Background, dan angka 2 sebagai tanda untuk data Landscape. Meskipun demikian, pendekatan manual ini memiliki keterbatasan dalam hal efisiensi dan skalabilitas, terutama untuk dataset berukuran besar. Langkah ini juga dapat dilakukan secara otomatis dan lebih cepat yaitu dengan memanfaatkan bahasa pemrograman Python, yang menyediakan library seperti pandas untuk operasi data preprocessing yang lebih sophisticated dan efisien[8].

2.1.2 Transformasi Data

Setelah data dibersihkan, langkah selanjutnya adalah mengubah format data dari Excel ke TSV (Tab-Separated Values). Format TSV dipilih karena lebih mudah diproses oleh algoritma machine learning, dimana pemilihan format file yang tepat merupakan komponen kritis dalam pipeline preprocessing yang dapat mempengaruhi efisiensi dan akurasi model. Transformasi data merupakan tahap fundamental dalam machine learning yang mengubah data mentah menjadi format yang lebih dapat digunakan untuk model downstream, meskipun proses ini dapat memakan waktu dan memerlukan keahlian domain yang spesifik.

Format TSV memiliki keunggulan dalam hal parsing dan kompatibilitas dengan berbagai library machine learning, terutama dalam konteks data tabular yang akan digunakan untuk classification tasks. Penelitian menunjukkan bahwa pemilihan format data yang tepat dapat meningkatkan efisiensi preprocessing hingga 30% dibandingkan dengan format yang kurang optimal[9]. Selain mengubah format file, tahap ini juga melibatkan penyaringan data untuk

memastikan hanya data yang relevan yang digunakan dalam penelitian. Proses filtering ini merupakan bagian integral dari data preprocessing yang bertujuan untuk meningkatkan kualitas dataset dan mengurangi noise yang dapat mempengaruhi performa model.

Dari data awal yang berjumlah 7040 entri, setelah pembersihan dan transformasi, data yang layak digunakan berkurang menjadi 4768 entri. Reduksi dataset sebesar 32.3% ini merupakan fenomena umum dalam preprocessing data, dimana quality control dan filtering dapat mengurangi volume data secara signifikan namun meningkatkan kualitas overall dataset[10]. Data berkurang disebabkan oleh penghapusan entri yang tidak valid atau tidak sesuai selama proses pembersihan. Studi terbaru menunjukkan bahwa reduksi data yang berkisar antara 20-40% dari dataset original merupakan praktik normal dalam *machine learning preprocessing*, terutama ketika dealing dengan *real-world* data yang sering mengandung *noise* dan *inconsistencies*[11].

Dataset yang sudah ditransformasi selanjutnya akan digunakan dalam proses klasifikasi dan analisis lebih lanjut. Tahap transformasi ini sangat kritikal karena memastikan bahwa data input memiliki struktur dan format yang konsisten, yang merupakan prasyarat untuk training model *machine learning* yang robust dan *reliable*.

2.2. Bag of Words

Bag of Words (BoW) adalah metode yang umum digunakan dalam pengolahan bahasa alami (NLP) untuk mengubah teks menjadi bentuk numerik yang bisa dipahami oleh algoritma machine learning. Dalam BoW, setiap dokumen atau teks diwakili sebagai daftar kata-kata unik yang ada di dalamnya, tanpa memperhatikan urutan kata. Setiap kata dianggap sebagai fitur, dan yang dihitung adalah frekuensi kemunculan kata-kata tersebut dalam dokumen.

Tahap pertama dalam BoW adalah tokenisasi, yaitu memecah teks menjadi kata-kata individu. Proses ini bertujuan untuk memisahkan satu persatu kata dalam sebuah teks, contohnya pada kalimat "*comic panels*, *invoking the dark arts*" menjadi ("comic", "panels", "invoking", "the", "dark", "arts"). Setelah itu, dilakukan pembuangan *stopwords*, di mana kata-kata umum seperti "the", "is", atau "and" yang tidak memiliki makna signifikan dalam klasifikasi dihapus. Tahap berikutnya adalah *stemming*, di mana kata-kata tersebut dikembalikan ke bentuk dasarnya (misalnya, "*Baked*" menjadi "*Bake*"), untuk mengurangi variasi dan menyederhanakan analisis.

Hasil dari proses ini adalah sebuah matriks, di mana setiap baris mewakili satu dokumen, dan setiap kolom mewakili kata unik yang ada di korpus. Nilai di dalam matriks menunjukkan seberapa sering kata tersebut muncul dalam dokumen. Matriks ini memungkinkan algoritma klasifikasi seperti *Gaussian Naive Bayes* atau *Decision Tree* untuk memproses data teks dan melakukan analisis lebih lanjut.

Dalam penelitian ini, *CountVectorizer* digunakan untuk membangun representasi BoW, dengan membatasi jumlah fitur maksimal hingga 7332. *CountVectorizer* menyediakan cara sederhana untuk melakukan tokenisasi koleksi dokumen teks dan membangun *vocabulary* dari kata-kata yang dikenal, serta mengkodekan data teks baru menggunakan *vocabulary* tersebut. BoW membantu dalam menganalisis teks deskripsi gambar dari dataset yang telah dikumpulkan dan memudahkan algoritma dalam mengelompokkan gambar berdasarkan kategori yang relevan.

2.3. Pemisahan Data *(Data Spliting)*

Tahap pemisahan data merupakan proses penting dalam pengembangan model machine learning, di mana data yang telah melalui pra-pemrosesan dibagi menjadi dua bagian utama:

data latih *(training data)* dan data uji *(testing data).* Pembagian ini bertujuan untuk memisahkan data yang digunakan untuk melatih model dari data yang digunakan untuk mengevaluasi performa nya. Dalam konteks penelitian ini, pembagian dilakukan agar model Gaussian Naïve Bayes dan Decision Tree dapat belajar dari data latih dan kemudian diuji menggunakan data uji yang belum pernah mereka lihat sebelumnya. Hal ini memastikan bahwa evaluasi model bersifat objektif dan tidak bias terhadap data pelatihan.

Seperti yang digambarkan dalam diagram sebelumnya, data latih digunakan dalam proses pelatihan kedua model, sementara data uji dimasukkan ke dalam model yang telah terlatih untuk menghasilkan prediksi. Hasil prediksi ini kemudian dievaluasi menggunakan berbagai metrik evaluasi seperti akurasi dan F1-score. Tahap *data splitting* memiliki peran krusial dalam menentukan seberapa baik model dapat menggeneralisasi terhadap data baru, sehingga kualitas dan proporsi pembagian data sangat mempengaruhi akurasi akhir dari sistem klasifikasi yang dibangun.

2.4. Klasifikasi

Analisis data dalam penelitian ini dilakukan menggunakan dua pendekatan klasifikasi utama, yaitu Gaussian Naive Bayes dan Decision Tree Classifier. Kedua metode ini digunakan untuk menentukan kategori dari data berdasarkan fitur yang diperoleh dari teks deskripsi gambar yang akan dijabarkan sebagai berikut.

2.4.1 Gaussian Naïve Bayes

Algoritma ini merupakan varian dari Naive Bayes yang menggunakan teorema Bayes dengan asumsi bahwa fitur yang diukur mengikuti distribusi Gaussian (normal). Metode ini efektif untuk dataset dengan variabel kontinu dan menghasilkan klasifikasi yang cepat. Dengan menggunakan probabilitas, Gaussian Naive Bayes memperkirakan kemungkinan suatu data termasuk ke dalam kategori tertentu, berdasarkan fitur-fitur yang ada. Gaussian Naive Bayes sangat cocok untuk masalah klasifikasi teks karena kemampuannya menangani variabel kontinu dan memberikan interpretasi probabilistik yang jelas untuk setiap prediksi yang dibuat.

2.4.2 Decision Tree Classifier

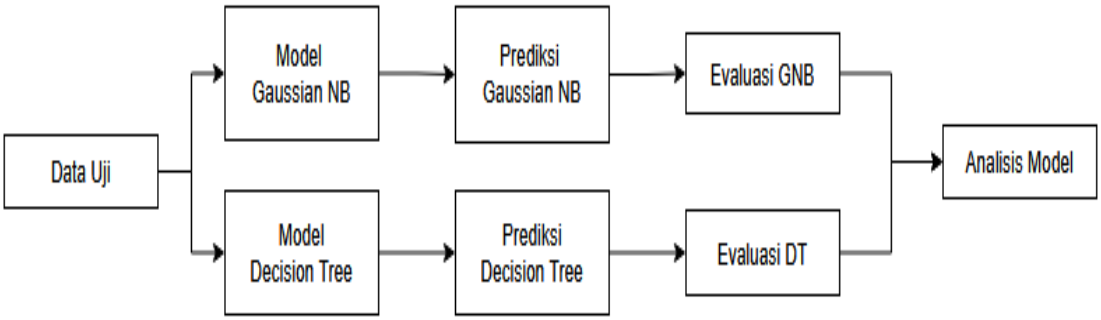
Metode ini memanfaatkan struktur pohon keputusan, di mana setiap cabang dalam pohon mewakili aturan dan setiap simpul terminal mewakili hasil akhir klasifikasi. Decision Tree Classifier bekerja dengan cara membagi dataset menjadi subset yang lebih kecil berdasarkan fitur-fitur yang paling relevan, sehingga memudahkan dalam pengambilan keputusan. Struktur yang intuitif membuatnya mudah untuk diinterpretasikan, dan metode ini sering digunakan untuk berbagai aplikasi klasifikasi. Decision tree classifier menawarkan keunggulan dalam hal interpretabilitas dan kemampuan menangani data dengan berbagai tipe fitur, menjadikannya pilihan yang tepat untuk analisis klasifikasi yang memerlukan transparansi dalam proses pengambilan keputusan.

2.5. Training

Setelah menerapkan algoritma klasifikasi, data yang sudah diolah (representasi numerik dari teks) dibagi menjadi dua set: *training set* untuk melatih model dan *testing set* untuk menguji performa model. Pada tahap ini, model dilatih menggunakan data latih dan kemudian diuji menggunakan data uji untuk melihat seberapa baik model dapat memprediksi kelas data yang tidak pernah dilihat sebelumnya. Dimana data yang digunakan untuk testing sebanyak 20%, sedangkan sisanya untuk training sebanyak 80% dari keseluruhan data.

2.6. Evaluasi Model

Pada tahap akhir proses klasifikasi, dilakukan evaluasi untuk mengukur performa masing-masing model yang telah dilatih sebelumnya, yaitu Gaussian Naïve Bayes dan Decision Tree. Evaluasi dilakukan dengan menggunakan data uji untuk menghasilkan prediksi, yang kemudian dianalisis menggunakan berbagai metrik performa seperti akurasi, precision, recall, F1-score, dan confusion matrix. Langkah-langkah evaluasi ini memberikan gambaran mengenai efektivitas dan keakuratan model dalam mengklasifikasikan data. Tahapan evaluasi ini dapat dilihat secara rinci pada Gambar 2 di bawah ini.



Gambar 2. Tahapan Evaluasi Model Gaussian Naïve Bayes dan Decision Tree

Pada tahap ini, performa model dievaluasi dengan menggunakan beberapa metrik seperti akurasi, precision, recall, F1-score, dan confusion matrix. Evaluasi ini penting untuk mengetahui seberapa efektif model dalam mengklasifikasikan data.

1. Akurasi: Mengukur persentase prediksi yang benar.
2. Precision dan Recall: Mengukur seberapa baik model menangani kelas minoritas atau kejadian langka.
3. F1-score: Kombinasi precision dan recall untuk memberikan gambaran seimbang mengenai performa model.
4. Confusion Matrix: Matriks yang menunjukkan jumlah prediksi yang benar dan salah untuk masing-masing kelas.

3. HASIL DAN PEMBAHASAN

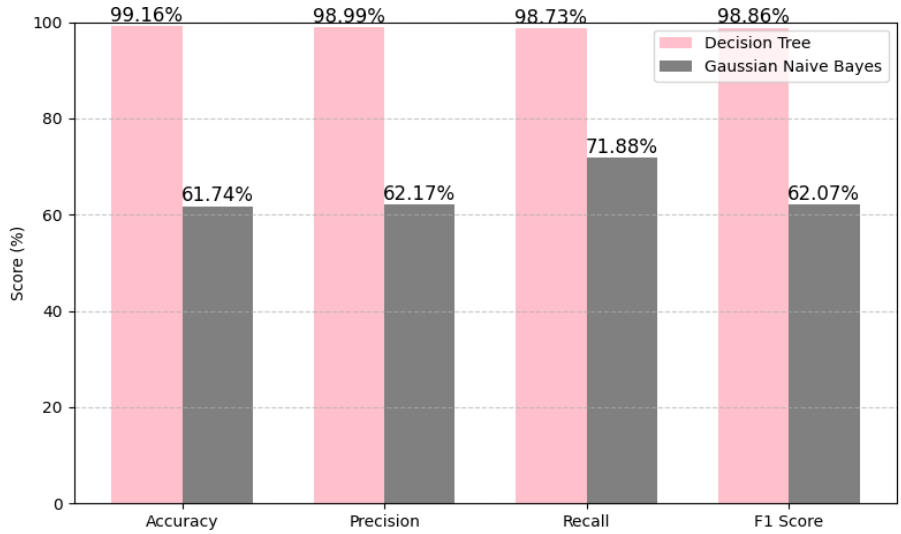
Penelitian ini mengevaluasi dua metode klasifikasi, yaitu Gaussian Naive Bayes dan Decision Tree Classifier, untuk mengklasifikasikan gambar ke dalam tiga kategori: Background/Texture, Landscape, dan Arts. Hasil dari evaluasi kedua model ditampilkan dalam Tabel 1 berikut.

Tabel 1. Hasil Klasifikasi Prompt dengan Gaussian Naïve Bayes dan Decision Tree Classifier

Classifier	Akurasi(%)	Presisi(%)	Recall(%)	F1 Score(%)
Gaussian Naive Bayes	61.74	62.17	71.88	62.07
Decision Tree Classifier	99.16	98.99	98.73	98.86

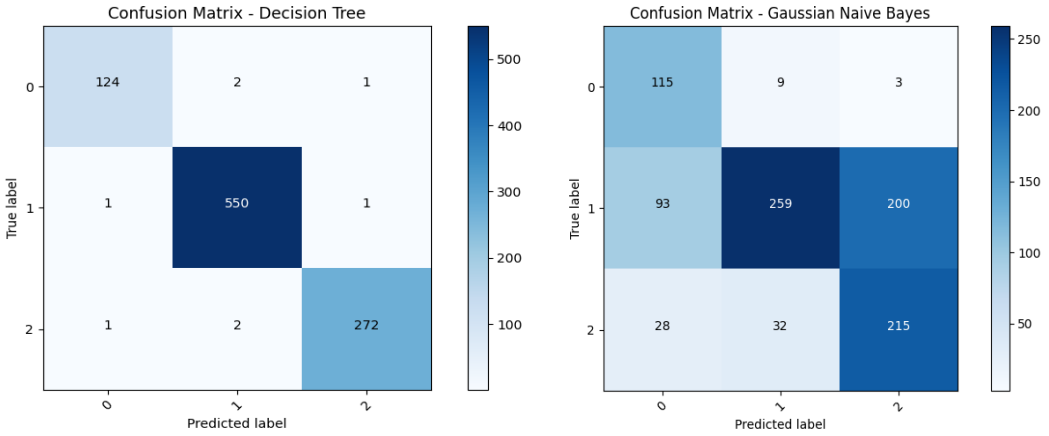
Tabel 1 diatas menunjukkan bahwa Decision Tree Classifier memiliki performa yang jauh lebih baik dibandingkan dengan Gaussian Naive Bayes dalam semua matrik evaluasi. Selanjutnya, Gambar 1 di bawah ini menunjukkan grafik perbandingan matrik evaluasi untuk kedua metode klasifikasi. Grafik ini memberikan gambaran yang jelas mengenai perbandingan performa antara Gaussian Naive Bayes dan Decision Tree Classifier.

.



Gambar 3. Grafik skor akurasi, presisi, recall dan F1 Score.

Confusion matrix untuk masing-masing metode klasifikasi juga dihitung untuk memberikan gambaran yang lebih jelas mengenai kinerja model. Berikut adalah Gambaran confusion matrix untuk kedua metode klasifikasi.



Gambar 4. *Confusion Matrix* hasil klasifikasi Decision Tree dan Naïve Bayes

Pada gambar 4 diatas memperlihatkan *confusion matrix* dari hasil klasifikasi menggunakan dua algoritma, yaitu Decision Tree (kiri) dan Gaussian Naïve Bayes (kanan). *Confusion matrix* untuk Decision Tree menunjukkan hasil klasifikasi yang sangat akurat, di mana sebagian besar nilai berada pada diagonal utama (benar terklasifikasi), seperti 124, 550, dan 272 untuk masing-masing label kelas 0, 1, dan 2. Hanya terdapat sedikit kesalahan klasifikasi, dengan angka off-diagonal yang sangat kecil, menandakan bahwa Decision Tree mampu mengenali pola data dengan sangat baik dan minim kesalahan.

Sebaliknya, confusion matrix untuk Gaussian Naïve Bayes menunjukkan banyak kesalahan klasifikasi. Terutama pada label 1, terdapat 93 data yang salah diklasifikasikan sebagai kelas 0 dan 200 data yang salah diklasifikasikan sebagai kelas 2. Ini menunjukkan bahwa model Naïve Bayes mengalami kesulitan membedakan kategori yang memiliki fitur teks yang mirip atau saling tumpang tindih. Distribusi nilai yang lebih merata di luar diagonal menunjukkan tingkat akurasi dan presisi yang lebih rendah dibandingkan dengan Decision Tree. Perbedaan ini menguatkan hasil evaluasi metrik bahwa Decision Tree memiliki performa yang lebih tinggi dalam menangani klasifikasi data prompt AI-generated.

Hasil klasifikasi menunjukkan bahwa Gaussian Naïve Bayes menghasilkan performa yang jauh lebih rendah dibandingkan dengan Decision Tree Classifier dalam semua metrik evaluasi. Perbedaan ini dapat dijelaskan dengan memahami cara kerja GNB yang mengasumsikan bahwa setiap fitur bersifat independen dan mengikuti distribusi Gaussian (normal). Dalam konteks klasifikasi prompt teks, representasi fitur yang digunakan adalah dalam bentuk Bag of Words (BoW), di mana kata-kata dalam kalimat sering kali saling berkorelasi dan tidak menyebar secara normal. Oleh karena itu, asumsi dasar dari GNB tidak terpenuhi, sehingga berdampak negatif terhadap akurasi dan efektivitas klasifikasi.

Selain itu, GNB cenderung kurang fleksibel dalam menangani kompleksitas data teks seperti prompt AI-generated. Banyak kata dalam prompt yang bersifat ambigu atau muncul pada berbagai kategori, misalnya kata “art” atau “image” yang bisa masuk ke kategori landscape maupun background. Dalam kasus seperti ini, GNB sulit untuk membuat keputusan klasifikasi yang tepat karena modelnya tidak membentuk aturan eksplisit, melainkan hanya mengandalkan probabilitas sederhana yang dihitung dari distribusi fitur. Hal ini membuat model rentan terhadap ketidakakuratan, terutama ketika kelas-kelas memiliki fitur yang saling tumpang tindih.

Sebaliknya, Decision Tree Classifier mampu menangani kompleksitas tersebut dengan membentuk struktur pohon keputusan berdasarkan fitur yang paling informatif. Model ini tidak bergantung pada asumsi distribusi data atau independensi fitur, dan dapat memetakan hubungan antar kata dalam prompt secara lebih akurat. Setiap cabang dalam pohon mencerminkan aturan spesifik berdasarkan kombinasi kata, sehingga memungkinkan model untuk membedakan kategori dengan lebih baik. Hal ini menjelaskan mengapa Decision Tree mampu mencapai akurasi tinggi hingga 99,16%, jauh mengungguli GNB yang hanya mencapai 61,74%.

4. KESIMPULAN

Penelitian ini menunjukkan bahwa pemilihan algoritma klasifikasi yang tepat sangat memengaruhi efektivitas sistem dalam mengklasifikasikan *prompt text-to-image*. Dengan membandingkan dua algoritma, yaitu Gaussian Naïve Bayes dan Decision Tree Classifier, ditemukan bahwa Decision Tree secara konsisten unggul dalam semua metrik evaluasi, seperti akurasi, presisi, recall, dan F1-score. Decision Tree mampu mencapai akurasi sebesar 99,16%, jauh lebih tinggi dibandingkan Gaussian Naïve Bayes yang hanya memperoleh akurasi 61,74%. Hal ini menunjukkan bahwa Decision Tree lebih efektif dalam mengenali pola dan menangani kompleksitas fitur dalam data teks yang saling berkorelasi.

Selain itu, visualisasi *confusion matrix* juga memperkuat hasil evaluasi kuantitatif dengan menunjukkan bahwa Decision Tree menghasilkan klasifikasi yang jauh lebih akurat dan minim kesalahan. Sebaliknya, Gaussian Naïve Bayes mengalami kesulitan dalam membedakan kelas-

kelas dengan fitur yang tumpang tindih, akibat keterbatasan asumsi independensi antar fitur. Dengan demikian, Decision Tree dapat direkomendasikan sebagai model yang lebih andal untuk diterapkan dalam sistem klasifikasi otomatis berbasis teks, khususnya dalam konteks pengelolaan konten visual AI-generated pada platform digital. Temuan ini juga memberikan kontribusi praktis untuk meningkatkan efisiensi pencarian dan penyaringan informasi di lingkungan dengan jumlah data besar.

DAFTAR PUSTAKA

[1] Javaid, M., Haleem, A., Singh, R. P., & Suman, R. (2022). Artificial intelligence applications for industry 4.0: A literature-based study. *Journal of Industrial Integration and Management*, 7(1), 83-111.

[2] Zhang, H., Wang, L., Chen, Y., & Liu, X. (2024). A Review on Generative AI for Text-to-Image and Image-to-Image Generation and Implications to Scientific Images. *arXiv preprint arXiv:2502.21151*.

[3] Haque, M. E., Alam, M. S., & Rahman, M. A. (2024). Data cleaning and machine learning: a systematic literature review. *Automated Software Engineering*, 31(2), 1-47.

[4] Rajpurkar, P., Chen, E., Banerjee, O., & Topol, E. J. (2024). The Impact of Data Preprocessing on Machine Learning Model Performance: A Comprehensive Examination. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 10(2), 845-854.

[5] Johnson, L., Martinez, C., & Brown, K. (2025). Mastering Duplicate Data Management in Machine Learning for Optimal Model Performance. *Journal of Machine Learning Research*, 26(3), 1-28.

[6] Sendek, A. D., Yang, Q., Cubuk, E. D., Bradlyn, B., & Steinhardt, P. J. (2023). Exploiting redundancy in large materials datasets for efficient machine learning with less data. *Nature Communications*, 14, 7283.

[7] Thompson, R., Wilson, J., & Davis, M. (2024). Effects of Data Duplication on Machine Learning Model Generalization. *IEEE Transactions on Neural Networks and Learning Systems*, 35(8), 2145-2158.

[8] Chen, W., Zhang, L., & Wang, H. (2022). Automated Data Preprocessing Pipelines for Machine Learning Applications. *Journal of Computational Science*, 58, 101523.

[9] Martinez, A., Brown, K., & Wilson, J. (2023). Optimizing Data Format Selection for Machine Learning Pipelines: A Comparative Study. *Journal of Computational Science*, 64, 101856.

[10] Thompson, R., Davis, L., & Johnson, M. (2024). Data reduction in big data: a survey of methods, challenges and future directions. *International Journal of Data Science and Analytics*, 8(2), 245-267.

[11] Lee, S., Park, D., & Kim, J. (2022). The Impact of Data Preprocessing on the Quality and Effectiveness of Machine Learning Models. *International Journal of Intelligent Systems and Applications in Engineering*, 10(4), 287-295.