

Perbandingan K-Nearest Neighbor dan Naive Bayes dalam Klasifikasi Genre Film Berdasarkan Sinopsis

Alfin Gimnastiar^{*}, Muhammad Ikhsan

Fakultas Sains dan Teknologi, Program Studi Ilmu Komputer, Universitas Islam Negeri Sumatera Utara, Medan, Indonesia

INFORMASI ARTIKEL

Diterima Redaksi: 18 Agustus 2026
Revisi Akhir: 01 September 2026
Diterbitkan Online: 25 September 2026

KATA KUNCI

K-Nearest Neighbor
Naive Bayes
TF-IDF
Klasifikasi Teks
Sinopsis Film

KORESPONDENSI^(*)

Phone: +62 823-6697-7233
E-mail: Alfingimnastiar@gmail.com

A B S T R A K

Pertumbuhan industri film dan semakin beragamnya konten menyebabkan pengelompokan film berdasarkan genre menjadi semakin menantang. Penelitian ini bertujuan untuk membandingkan kinerja algoritma K-Nearest Neighbor (KNN) dan Naive Bayes dalam mengklasifikasikan genre film berdasarkan teks sinopsis. Proses pengolahan teks dilakukan menggunakan pendekatan *Natural Language Processing* (NLP) dan ekstraksi fitur *Term Frequency-Inverse Document Frequency* (TF-IDF) dengan pendekatan *single-label classification*. Dataset diperoleh dari The Movie Database (TMDB) dan terdiri atas 1.000 sinopsis film berbahasa Inggris yang terbagi secara seimbang ke dalam lima genre, yaitu *Action*, *Horror*, *Romance*, *Comedy*, dan *Science Fiction*. Dataset kemudian dibagi menjadi 80% data latih dan 20% data uji. Hasil pengujian menunjukkan bahwa Naive Bayes dengan *Laplace Smoothing* ($\alpha = 1,0$) menghasilkan kinerja lebih baik dengan *Accuracy* 59,00%, *Precision* 58,44%, *Recall* 59,00%, dan *F1-score* 58,50%. Sementara itu, KNN ($K = 5$) dengan *Cosine Similarity* memperoleh *Accuracy* 44,00%, *Precision* 45,16%, *Recall* 44,00%, dan *F1-score* 44,06%. Performa Naive Bayes yang lebih baik berkaitan dengan kemampuannya dalam memanfaatkan pola probabilitas kosakata pada masing-masing genre, meskipun kedua model masih menghadapi kesulitan pada genre dengan kosakata yang saling beririsan. Berdasarkan hasil tersebut, Naive Bayes lebih efektif dibandingkan KNN untuk klasifikasi genre film pada dataset yang digunakan.

PENDAHULUAN

Perkembangan teknologi informasi telah mendorong pertumbuhan industri film digital dan *platform* layanan *streaming*, sehingga jumlah film yang tersedia terus meningkat. Kondisi tersebut memberikan lebih banyak pilihan bagi pengguna, tetapi juga membuat proses pencarian dan pemilihan film menjadi semakin kompleks. Klasifikasi berdasarkan genre dapat membantu pengelompokan dan penelusuran film serta mendukung pengembangan sistem rekomendasi [1]. Namun, proses klasifikasi genre tidak selalu sederhana karena satu film dapat memiliki lebih dari satu genre sehingga proses klasifikasi menjadi lebih kompleks dan berpotensi menimbulkan perbedaan hasil pada berbagai sumber data. Kondisi tersebut dapat memengaruhi konsistensi pengelompokan film dan menyulitkan pengguna dalam menemukan tontonan yang sesuai. Oleh karena itu, dibutuhkan pendekatan otomatis yang sistematis dan konsisten untuk melakukan klasifikasi genre film secara lebih terukur.

Sinopsis film dapat dimanfaatkan sebagai sumber data untuk melakukan klasifikasi karena memuat informasi mengenai tema, konflik, alur, dan karakter dalam bentuk teks tidak terstruktur. Data tersebut dapat diolah menggunakan *Natural Language Processing* (NLP) [2] dan direpresentasikan secara numerik menggunakan metode seperti *Term Frequency-Inverse Document Frequency* (TF-IDF) [3]. Dalam klasifikasi teks, K-Nearest Neighbor (KNN) menentukan kelas berdasarkan kedekatan dengan data berlabel [4], sedangkan Naive Bayes menggunakan pendekatan probabilistik dan

dikenal efisien untuk pengolahan data teks [5]. Perbedaan karakteristik kedua algoritma tersebut dapat menghasilkan performa yang berbeda bergantung pada *dataset* dan permasalahan yang digunakan [6].

Sejumlah penelitian telah membandingkan KNN dan Naive Bayes pada berbagai domain. Munawaroh dan Alamsyah [7] menunjukkan bahwa performa kedua algoritma dapat berbeda berdasarkan karakteristik data, sedangkan Prayogo dkk. [8] memperoleh akurasi KNN sebesar 85,00% dan Naive Bayes sebesar 81,66% pada klasifikasi judul artikel jurnal. Pada domain film, Buslim dkk. [9] membandingkan KNN, Naive Bayes, dan SVM pada klasifikasi genre berdasarkan sinopsis, di mana SVM memperoleh akurasi tertinggi sebesar 93% dengan menggunakan *dataset* yang tidak seimbang. Penelitian lain juga telah mengeksplorasi pendekatan *multi-label* [10], penggunaan arsitektur *Deep Learning* seperti LSTM [11], serta penerapan KNN pada genre anime dengan data seimbang [12]. Hasil tersebut menunjukkan bahwa performa algoritma sangat dipengaruhi oleh karakteristik data, domain, dan pendekatan klasifikasi yang digunakan.

Meskipun demikian, penelitian yang secara khusus membandingkan KNN dan Naive Bayes pada klasifikasi genre film berbasis sinopsis dengan *dataset* The Movie Database (TMDB) yang dikontrol dan seimbang masih terbatas. Oleh karena itu, penelitian ini menggunakan 1.000 sinopsis film berbahasa Inggris dari TMDB yang terbagi secara seimbang ke dalam lima genre, yaitu *Action*, *Horror*, *Romance*, *Comedy*, dan *Science Fiction*. Pendekatan *single-label classification* digunakan dengan TF-IDF sebagai metode ekstraksi fitur. Pemodelan klasifikasi diimplementasikan menggunakan parameter algoritma KNN berbasis *Cosine Similarity* ($K=5$) dan Naive Bayes dengan *Laplace Smoothing* ($\alpha=1,0$).

Penelitian ini bertujuan membandingkan performa KNN dan Naive Bayes dalam klasifikasi genre film berdasarkan sinopsis menggunakan metrik *Accuracy*, *Precision*, *Recall*, dan *F1-score*. Hasil penelitian diharapkan dapat memberikan gambaran mengenai algoritma yang lebih sesuai untuk klasifikasi genre film berbasis teks pada *dataset* dengan distribusi kelas yang seimbang.

TINJAUAN PUSTAKA

Preprocessing Teks

Preprocessing teks merupakan tahap awal dalam pengolahan data teks yang dilakukan untuk mengurangi *noise*, menghilangkan informasi yang tidak relevan, serta menyeragamkan bentuk kata agar teks lebih optimal dalam proses pengolahan oleh algoritma [13]. Pada penelitian ini, *preprocessing* terdiri atas lima tahapan utama, yaitu *case folding* untuk mengubah seluruh huruf menjadi *lowercase*, *cleaning* untuk menghapus karakter yang tidak diperlukan, *tokenization* untuk memisahkan teks menjadi token, *stopword removal* untuk menghilangkan kata-kata umum yang kurang informatif, dan *lemmatization* untuk mengubah kata ke bentuk dasarnya [14].

Term Frequency-Inverse Document Frequency (TF-IDF)

Term Frequency-Inverse Document Frequency (TF-IDF) merupakan metode pembobotan kata yang bertujuan untuk menentukan tingkat kepentingan suatu kata di dalam sebuah dokumen. Metode ini bekerja dengan mengubah teks menjadi vektor representasi numerik [15]. Secara konseptual, TF-IDF menyeimbangkan frekuensi kemunculan kata pada suatu dokumen tertentu (*Term Frequency*) dengan penyebaran kata tersebut pada seluruh kumpulan dokumen atau *corpus* (*Inverse Document Frequency*). Pendekatan ini mampu memberikan bobot lebih tinggi pada kata-kata yang sangat relevan dan khas dalam suatu dokumen, sekaligus menurunkan bobot kata yang terlalu sering muncul di banyak dokumen [16].

K-Nearest Neighbor (KNN)

K-Nearest Neighbor (KNN) merupakan metode *supervised learning* yang bersifat *non-parametric* dan mengklasifikasikan data berdasarkan kedekatan dengan data latih di dalam ruang fitur, tanpa membangun model pada tahap pelatihan. Algoritma ini bekerja dengan menentukan nilai K , menghitung tingkat kemiripan antara data uji dan seluruh data latih, lalu menetapkan kelas hasil prediksi berdasarkan mayoritas kelas dari K tetangga terdekat [17]. Pada data teks berdimensi tinggi, kemiripan antar dokumen umumnya diukur menggunakan *Cosine Similarity* karena menghitung sudut antar vektor sehingga tidak dipengaruhi oleh perbedaan panjang dokumen, menjadikannya lebih stabil dibandingkan *Euclidean Distance* untuk klasifikasi teks berbasis TF-IDF [18].

Naive Bayes

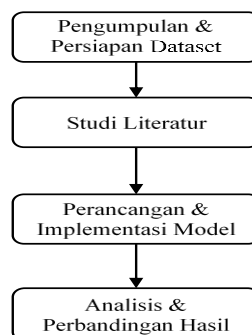
Naive Bayes merupakan algoritma klasifikasi berbasis probabilitas yang menentukan kelas suatu dokumen dengan melihat distribusi probabilitas kata-kata di dalamnya. Algoritma ini menggunakan Teorema Bayes dengan asumsi bahwa setiap fitur bersifat independen satu sama lain [19]. Probabilitas awal (*prior*) untuk setiap kelas diperoleh dari proporsi jumlah data latih pada masing-masing kelas. Selanjutnya, probabilitas bersyarat (*likelihood*) dihitung dengan menerapkan *Laplace Smoothing* untuk mencegah munculnya nilai probabilitas nol ketika suatu kata tidak ditemukan pada data pelatihan di kelas tertentu. Salah satu variannya, yaitu Multinomial Naive Bayes, dapat diterapkan secara stabil dan efisien pada pengolahan teks yang telah direpresentasikan dalam bentuk nilai desimal TF-IDF [20].

Evaluasi Model

Evaluasi kinerja algoritma klasifikasi umumnya dilakukan menggunakan *Confusion Matrix*. *Confusion Matrix* digunakan untuk membandingkan kelas aktual dengan kelas hasil prediksi [21]. *Accuracy* menunjukkan proporsi prediksi yang benar secara keseluruhan, *Precision* menunjukkan ketepatan prediksi suatu kelas, *Recall* menunjukkan kemampuan model dalam mengenali data yang termasuk dalam suatu kelas, sedangkan *F1-score* menggabungkan *Precision* dan *Recall* untuk memberikan gambaran kinerja model secara lebih menyeluruh [22].

METODOLOGI

Penelitian ini merupakan penelitian kuantitatif eksperimental yang bertujuan menguji dan membandingkan performa algoritma K-Nearest Neighbor (KNN) dan Naive Bayes dalam klasifikasi genre film berdasarkan teks sinopsis. Secara garis besar, tahapan penelitian ditunjukkan pada Gambar dibawah.



Gambar 1. Kerangka alur penelitian

Berdasarkan kerangka alur tersebut, detail tahapan metodologi yang dilakukan dijabarkan sebagai berikut:

Pengumpulan Dan Persiapan Dataset

Data dikumpulkan secara daring melalui *platform* TMDB menggunakan *Application Programming Interface* (API) dengan bahasa pemrograman Python. *Dataset* yang diperoleh meliputi judul film, sinopsis (*overview*), dan informasi genre. Pada pendekatan *single-label classification*, genre yang berada pada indeks pertama dari data TMDB ditetapkan sebagai label utama untuk setiap film. Data yang terkumpul kemudian diseleksi dan disusun secara seimbang sehingga masing-masing genre memiliki 200 data film.

Studi Literatur

Studi literatur dilakukan untuk mengkaji teori, metode, dan penelitian terdahulu yang berkaitan dengan penelitian. Kajian ini mencakup *Natural Language Processing* (NLP), *preprocessing* teks, ekstraksi fitur TF-IDF, algoritma K-Nearest Neighbor (KNN) dan Naive Bayes, serta metode evaluasi klasifikasi dll.

Perancangan dan Implementasi Model

Tahapan ini terdiri atas dua fase utama, yaitu merancang alur kerja sistem dan mengimplementasikannya ke dalam kode program.

Perancangan

Perancangan dilakukan dengan menyusun alur proses klasifikasi yang meliputi pembacaan *dataset*, pembagian data menjadi data latih dan data uji, *preprocessing* teks, ekstraksi fitur, penerapan model klasifikasi, serta pengujian dan evaluasi. Alur ini kemudian menjadi dasar dalam membangun sistem klasifikasi genre film berdasarkan sinopsis.

Implementasi

Flowchart yang telah dirancang kemudian diimplementasikan ke dalam program menggunakan bahasa Python. Untuk mencegah kebocoran data, proses diawali dengan membagi *dataset* menjadi 80% data latih dan 20% data uji. Selanjutnya, teks diolah dengan *library* Natural Language Toolkit (NLTK) melalui tahapan *case folding*, *cleaning*, *tokenization*, *stopword removal*, dan *lemmatization*. Teks hasil *preprocessing* selanjutnya diubah menjadi representasi numerik menggunakan TF-IDF melalui fungsi ‘TfidfVectorizer’ pada *library* ‘Scikit-Learn’. Pada prinsipnya, pembobotan TF-IDF terdiri atas *Term Frequency* (TF), *Inverse Document Frequency* (IDF), dan bobot TF-IDF. Nilai TF menunjukkan frekuensi kemunculan suatu *term* dalam dokumen dan dapat dihitung menggunakan Persamaan (1):

$$TF(t, d) = \frac{f_{t,d}}{|d|} \quad (1)$$

Selanjutnya, nilai IDF digunakan untuk menunjukkan tingkat kepentingan suatu *term* berdasarkan penyebarannya pada keseluruhan *corpus*, sebagaimana dinyatakan pada Persamaan (2):

$$IDF(t) = \log\left(\frac{N}{df_t}\right) \quad (2)$$

Bobot akhir suatu *term* diperoleh dengan mengalikan nilai TF dan IDF seperti pada Persamaan (3)

$$w(t, d) = TF(t, d) \times IDF(t) \quad (3)$$

Matriks TF-IDF yang dihasilkan selanjutnya digunakan sebagai masukan bagi dua model klasifikasi. Pada algoritma K-Nearest Neighbor (KNN), model diimplementasikan menggunakan ‘KNeighborsClassifier’ dengan jumlah tetangga terdekat K=5 dan metrik ‘cosine’. Pada prinsipnya, tingkat kemiripan antara dua vektor dapat direpresentasikan menggunakan *Cosine Similarity* yang dinyatakan pada Persamaan (4):

$$\cos(\theta_{ij}) = \frac{\sum_k (d_{ik} d_{jk})}{\sqrt{\sum_k d_{ik}^2} \sqrt{\sum_k d_{jk}^2}} \quad (4)$$

Selanjutnya, klasifikasi menggunakan Multinomial Naive Bayes diimplementasikan melalui fungsi ‘MultinomialNB’ pada ‘Scikit-Learn’ dengan parameter $\alpha = 1,0$ sebagai Laplace Smoothing. Meskipun secara teori algoritma ini lebih sesuai untuk data berupa jumlah kemunculan kata, penggunaannya pada nilai TF-IDF yang berbentuk desimal tetap dapat memberikan hasil klasifikasi teks yang baik [20]. Pada prinsipnya, proses klasifikasi melibatkan probabilitas prior, likelihood, dan posterior. Probabilitas prior suatu kelas dinyatakan pada Persamaan (5):

$$P(C) = \frac{N_c}{N} \quad (5)$$

Probabilitas *likelihood* digunakan untuk menunjukkan peluang kemunculan suatu fitur pada kelas tertentu, secara umum rumusnya dapat dinyatakan pada Persamaan (6):

$$P(x_i|C) = \frac{w_{i,c} + 1}{\sum_{k=1}^n w_{k,c} + |V|} \quad (6)$$

Selanjutnya, probabilitas *prior* dan *likelihood* digunakan untuk memperoleh probabilitas *posterior* dengan asumsi independensi antarfitur, sebagaimana dinyatakan pada Persamaan (7):

$$P(C|X) \propto P(C) \prod_{i=1}^n P(x_i|C) \quad (7)$$

Setelah klasifikasi dilakukan, kedua model digunakan untuk memprediksi genre pada data uji. Hasil prediksi kemudian dibandingkan dengan label aktual menggunakan fungsi evaluasi dari 'Scikit-Learn'. Evaluasi menghasilkan *Confusion Matrix* serta nilai *Accuracy*, *Precision*, *Recall*, dan *F1-score* untuk mengukur kinerja masing-masing model. Nilai *Precision*, *Recall*, dan *F1-score*.

Analisis dan Perbandingan Hasil

Berdasarkan hasil metrik evaluasi yang telah diperoleh pada tahap implementasi, kinerja klasifikasi model KNN dan Naive Bayes kemudian dianalisis dan dibandingkan secara menyeluruh. Tahapan ini bertujuan untuk membedah karakteristik prediksi berdasarkan pola kesalahan pada *Confusion Matrix*, serta menentukan algoritma mana yang memberikan performa klasifikasi paling optimal pada klasifikasi genre film dengan distribusi kelas yang seimbang.

HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil dari setiap tahapan penelitian, mulai dari pengumpulan data, perancangan dan implementasi sistem klasifikasi, hingga analisis perbandingan kinerja algoritma KNN dan Naive Bayes.

Pengumpulan Data

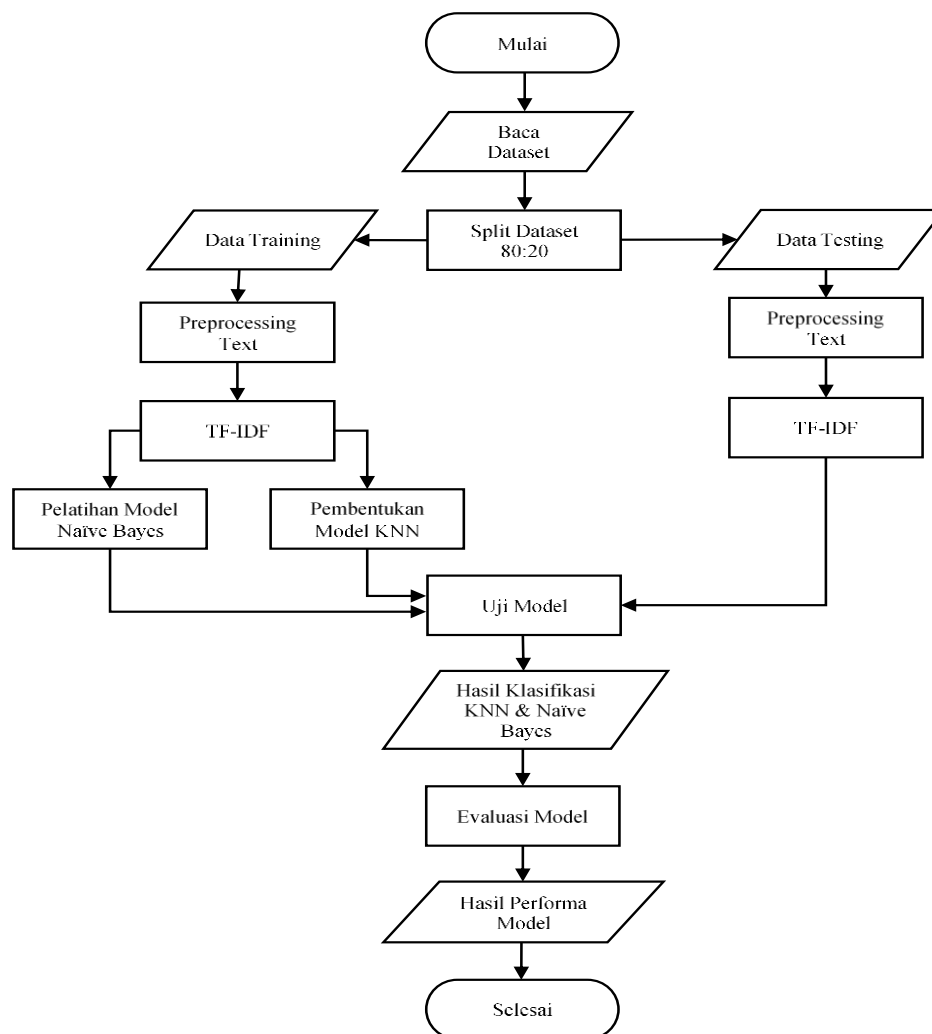
Data penelitian diperoleh secara otomatis melalui API Key dari *platform* The Movie Database (TMDB) menggunakan bahasa pemrograman Python. Pengambilan data difokuskan pada lima kelas genre utama, yaitu *Action*, *Horror*, *Romance*, *Comedy*, dan *Science Fiction*. Setiap film diberikan satu label kelas (*single-label*) berdasarkan indeks genre pertamanya. Total data yang berhasil diekstraksi adalah 1.000 sinopsis film yang terdistribusi secara seimbang (200 data per genre). Gambar 2 menampilkan cuplikan dari *dataset* mentah yang berhasil dikumpulkan.

id	judul	sinopsis	genre
D1	Disclosure Day	A cybersecurity expert becomes a whistleblower after uncovering secrets about aliens, putting him on the run from a corporation. Meanwhile, a meteorologist experiencing strange phenomena joins forces with him to prove there's life beyond our understanding.	Science Fiction
D2	Obsession	After breaking the mysterious "One Wish Willow" to win his crush's heart, a hopeless romantic finds himself getting exactly what he asked for but soon discovers that some desires come at a dark, sinister price.	Horror
D3	Backrooms	A strange doorway appears in the basement of a furniture showroom.	Horror
D4	Supergirl	When an unexpected and ruthless adversary strikes too close to home, Kara Zor-El, aka Supergirl, reluctantly joins forces with an unlikely companion on an epic, interstellar journey of vengeance and justice.	Action
D5	Scary Movie	Twenty-six years after outrunning a suspiciously familiar masked killer, the Core Four are back in the killer's crosshairs and no horror movie IP is safe.	Comedy
D6	The Mandalorian and Grogu	The evil Empire has fallen, and Imperial warlords remain scattered throughout the galaxy. As the fledgling New Republic works to protect everything the Rebellion fought for, they have enlisted the help of legendary Mandalorian bounty hunter Din Djarin and his young apprentice Grogu.	Action
D7	Masters of the Universe	After being separated for 15 years, the Sword of Power leads Prince Adam back to Eternia, where he discovers his home shattered under the fiendish rule of Skeletor. To save his family and his world, Adam must join forces with his closest allies, Teela and Duncan/Man-At-Arms, and embrace his true destiny as He-Man — the most powerful man in the universe.	Action
D8	Project Hail Mary	Science teacher Ryland Grace wakes up on a spaceship light years from home with no recollection of who he is or how he got there. As his memory returns, he begins to uncover his mission: solve the riddle of the mysterious substance causing the sun to die out. He must call on his scientific knowledge and unorthodox ideas to save everything on Earth from extinction.	Science Fiction
D9	The Devil Wears Prada 2	Andy Sachs returns to Runway as Miranda Priestly navigates a new media landscape and Runway's position within. The duo reconnect with former assistant Emily Charlton, now the head of a luxury brand that possesses funding which could ensure Runway's survival.	Comedy
D10	Passenger	After a young couple witnesses a gruesome highway accident, they soon realize they did not leave the crash scene alone, as a demonic presence called the Passenger that won't stop until it claims them both turns their van life adventure into a nightmare.	Horror
...	...	...	...
D1000	Timecrimes	A man accidentally gets into a time machine and travels back in time nearly an hour. Finding himself will be the first of a series of disasters of unforeseeable consequences.	Science Fiction

Gambar 2. Cuplikan *Dataset* Mentah

Perancangan Sistem

Untuk mengolah *dataset* mentah menjadi model klasifikasi yang dapat dievaluasi, dirancang alur sistem yang mencakup pembagian data, *preprocessing*, hingga evaluasi performa model. Sistem tersebut kemudian diterapkan menggunakan bahasa Python dengan bantuan berbagai *library* yang sesuai. Alur kerja sistem ditampilkan dalam bentuk *flowchart* pada Gambar 3.

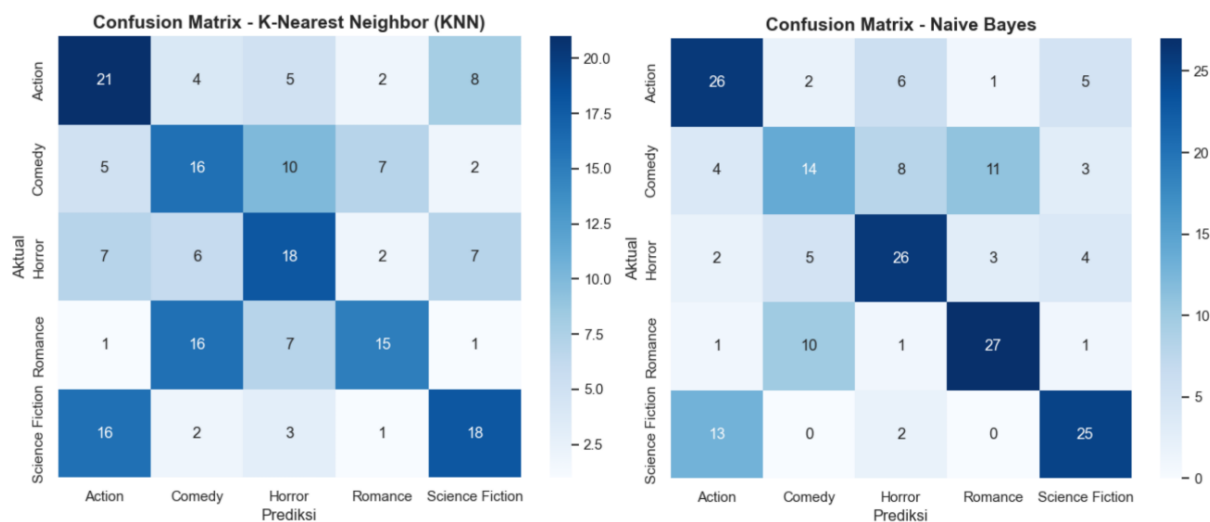


Gambar 3. Flowchart Sistem Klasifikasi

Hasil Implementasi

Berdasarkan *flowchart* tersebut, proses implementasi diawali dengan pembagian *dataset* menggunakan *Stratified Train-Test Split*, yang menghasilkan 80% data latih (800 data) dan 20% data uji (200 data). Sinopsis kemudian melalui tahapan *preprocessing* secara berurutan, meliputi *case folding*, *cleaning*, *tokenization*, *stopword removal*, dan *lemmatization*. Hasil *preprocessing* selanjutnya diubah menjadi matriks numerik menggunakan pembobotan TF-IDF.

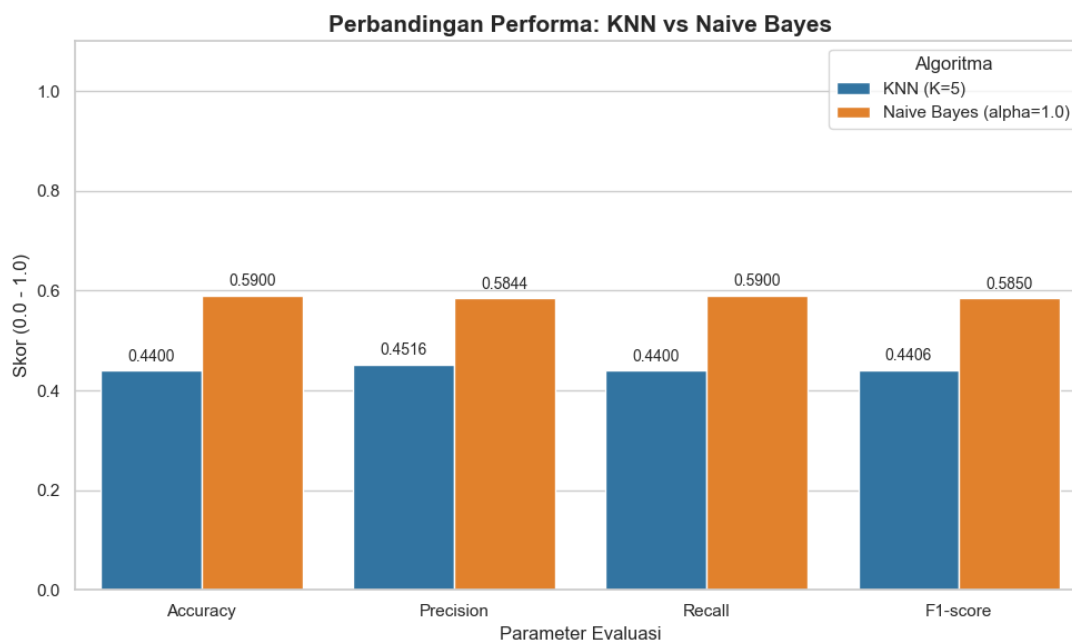
Matriks TF-IDF kemudian digunakan sebagai masukan untuk algoritma KNN dengan 5 tetangga terdekat dan metrik *Cosine Similarity*, serta Multinomial Naïve Bayes dengan *Laplace Smoothing* dan $\alpha = 1,0$. Prediksi dari kedua model terhadap 200 data uji kemudian dievaluasi menggunakan *Confusion Matrix* untuk melihat kesesuaian antara label prediksi dan label aktual. Hasil klasifikasi masing-masing model ditampilkan pada Gambar 4 berikut.



Gambar 4. Confusion Matrix Algoritma KNN dan Naive Bayes

Analisis dan Perbandingan Performa Algoritma

Berdasarkan hasil implementasi, Naive Bayes menunjukkan kinerja yang lebih baik dibandingkan KNN dalam mengklasifikasikan genre film berdasarkan sinopsis. Model Naive Bayes memperoleh *Accuracy* sebesar 59,00%, *Precision* 58,44%, *Recall* 59,00%, dan *F1-score* 58,50%. Sementara itu, KNN menghasilkan *Accuracy* 44,00%, *Precision* 45,16%, *Recall* 44,00%, dan *F1-score* 44,06%. Perbandingan kedua model, termasuk selisih akurasi sebesar 15%, ditampilkan pada grafik batang pada Gambar 5. Hasil ini menunjukkan bahwa Naive Bayes dengan *Laplace Smoothing* lebih mampu mengenali pola kemunculan kata dalam sinopsis dibandingkan KNN yang menggunakan *Cosine Similarity*.



Gambar 5. Grafik Perbandingan Performa KNN dan Naive Bayes

Karakteristik klasifikasi dari kedua algoritma dibedah lebih mendalam berdasarkan pola prediksi yang dihasilkan oleh *Confusion Matrix*, yang ringkasannya disajikan pada Tabel 1.

Tabel 1. Analisis Hasil Klasifikasi Berdasarkan *Confusion Matrix*

Kategori Analisis	Naive Bayes	K-Nearest Neighbor
Genre Paling Akurat	Romance (27), Action (26), Horror (26)	Action (21), Horror (18), Comedy (16)

Kesalahan Klasifikasi Tertinggi 1	<i>Science Fiction</i> terprediksi <i>Action</i> (13 kasus)	<i>Science Fiction</i> terprediksi <i>Action</i> (16 kasus)
Kesalahan Klasifikasi Tertinggi 2	<i>Romance</i> terprediksi <i>Comedy</i> (10 kasus)	<i>Romance</i> terprediksi <i>Comedy</i> (16 kasus)

Berdasarkan Tabel 1, Naive Bayes menunjukkan kemampuan yang baik dalam mengenali genre *Romance*, *Action*, dan *Horror*. Ketiga genre tersebut cenderung memiliki kosakata yang lebih khas, seperti unsur supranatural atau ancaman, sehingga menghasilkan bobot TF-IDF yang lebih representatif dan membantu proses klasifikasi. Sebaliknya, sebagian besar kesalahan prediksi pada kedua model terjadi karena adanya kemiripan kosakata antargenre. Misalnya, *Science Fiction* dan *Action* memiliki kata-kata yang berkaitan dengan pertarungan dan penyelamatan, sedangkan *Romance* dan *Comedy* dapat sama-sama memuat unsur percintaan dan humor sehingga lebih sulit dibedakan.

Untuk memperkuat hasil evaluasi, dilakukan pengujian tambahan dengan memvariasikan jumlah tetangga (K) pada KNN dan nilai *Laplace Smoothing* (alpha) pada Naive Bayes. Hasil pengujian dan perbandingan setiap variasi parameter disajikan pada Tabel 2 dibawah.

Tabel 2. Perbandingan Performa Berdasarkan Variasi Parameter

Algoritma	Parameter	Accuracy	Precision	Recall	F1-score
K-Nearest Neighbor	K = 3	42,50%	44,51%	42,50%	42,25%
	K = 5	44,00%	45,16%	44,00%	44,06%
	K = 7	49,50%	50,12%	49,50%	49,02%
	K = 9	54,00%	53,88%	54,00%	53,69%
	K = 11	49,50%	49,29%	49,50%	49,14%
Naive Bayes	$\alpha = 0.5$	58,50%	57,89%	58,50%	58,00%
	$\alpha = 1$	59,00%	58,44%	59,00%	58,50%
	$\alpha = 1.5$	58,50%	57,96%	58,50%	57,95%
	$\alpha = 2$	59,00%	58,57%	59,00%	58,45%
	$\alpha = 2.5$	59,00%	58,32%	59,00%	58,54%

Hasil pada Tabel 2 menunjukkan bahwa performa KNN berubah seiring perubahan nilai K dan mencapai akurasi tertinggi sebesar 54,00% pada K = 9 sebelum kembali menurun pada K = 11. Peningkatan pada K = 9 diduga terjadi karena jumlah tetangga yang lebih besar membantu mengurangi pengaruh *noise* dari dokumen-dokumen yang memiliki kemiripan kata, seperti sisa-sisa *stopword* atau istilah umum yang saling beririsan antargenre, sehingga keputusan klasifikasi menjadi lebih stabil. Namun, ketika nilai K diperbesar terlalu jauh, tetangga yang kurang relevan dari kelas lain mulai ikut memengaruhi proses voting dan menyebabkan akurasi kembali menurun. Meskipun demikian, akurasi tertinggi KNN masih berada di bawah performa terendah Naive Bayes. Sebaliknya, Naive Bayes menunjukkan performa yang lebih stabil, dengan akurasi berada pada kisaran 58,50% hingga 59,00% pada berbagai nilai alpha.

Pengujian tambahan selanjutnya dilakukan dengan menggunakan metode ekstraksi fitur alternatif, yaitu *Bag of Words* (BoW) sebagai pembanding TF-IDF. Hasil pengujian tersebut disajikan pada Tabel 3.

Tabel 3. Perbandingan Performa Berdasarkan Metode Ekstraksi Fitur

Algoritma	Metrik	TF-IDF	BoW
K-Nearest Neighbor	Accuracy	44,00%	44,00%
	Precision	45,16%	45,11%
	Recall	44,00%	44,00%
	F1-score	44,06%	44,10%
Naive Bayes	Accuracy	59,00%	58,50%
	Precision	58,44%	58,05%
	Recall	59,00%	58,50%
	F1-score	58,50%	58,11%

Berdasarkan Tabel 3, penggunaan metode ekstraksi fitur yang berbeda tidak memberikan peningkatan performa yang signifikan pada kedua algoritma. Pada Naive Bayes, TF-IDF menghasilkan nilai yang sedikit lebih baik dibandingkan BoW, sedangkan pada KNN keduanya menghasilkan akurasi yang sama. Secara keseluruhan, hasil pengujian

menunjukkan bahwa Naive Bayes lebih efektif dibandingkan KNN dalam melakukan klasifikasi genre film berdasarkan teks sinopsis pada penelitian ini.

KESIMPULAN DAN SARAN

Kesimpulan

Berdasarkan hasil penelitian, Naive Bayes menunjukkan kinerja yang lebih baik dan stabil dibandingkan K-Nearest Neighbor (KNN) dalam mengklasifikasikan genre film berdasarkan teks sinopsis pada 1.000 data seimbang dari The Movie Database (TMDB). Dengan ekstraksi fitur TF-IDF dan pembagian data 80:20, Naive Bayes dengan *Laplace Smoothing* ($\alpha = 1,0$) memperoleh *Accuracy* 59,00%, *Precision* 58,44%, *Recall* 59,00%, dan *F1-score* 58,50%. Sementara itu, KNN ($K = 5$) menghasilkan *Accuracy* 44,00%, *Precision* 45,16%, *Recall* 44,00%, dan *F1-score* 44,06%. Berdasarkan *Confusion Matrix*, genre *Romance*, *Action*, dan *Horror* cenderung lebih mudah dikenali karena memiliki kosakata yang khas. Kesalahan prediksi lebih banyak terjadi antara *Science Fiction* dan *Action*, serta *Romance* dan *Comedy* karena adanya kemiripan kosakata pada sinopsis.

Saran

Penelitian selanjutnya dapat menggunakan *dataset* yang lebih besar dan menerapkan pendekatan *multi-label classification*, karena satu film umumnya dapat memiliki lebih dari satu genre. Representasi teks juga dapat dikembangkan menggunakan metode berbasis semantik, seperti Word2Vec, GloVe, atau BERT, untuk menangkap hubungan makna antarkata dengan lebih baik. Selain itu, pencarian *hyperparameter* menggunakan *Grid Search* atau *Random Search* dapat diterapkan untuk memperoleh parameter yang lebih optimal. Performa tersebut juga dapat dibandingkan dengan algoritma lain atau metode berbasis *deep learning*.

DAFTAR PUSTAKA

- [1] J. M. A. Saputra, M. L. Huizen, and B. D. Arianto, "Sistem Rekomendasi Film pada Platform Streaming Menggunakan Metode Content-Based Filtering," *Jurnal Transformatika*, vol. 22, no. 1, pp. 10–21, Jul. 2024, doi: 10.26623/transformatika.v22i1.7041.
- [2] V. Agustina and A. Herliana, "Analisis Sentimen Publik atas Kebijakan Efisiensi Anggaran 2025 dengan Text Mining dan Natural Language Processing," *Jurnal Media Informatika*, vol. 6, no. 3, pp. 2182–2194, Jul. 2025, doi: 10.55338/JUMIN.V6I3.6301.
- [3] R. R. Putra, N. A. Putri, and A. D. Putra, *Teknik Cosine Similarity Dan TF-IDF Dalam Analisis Data*. Payakumbuh: PT. Serasi Media Teknologi, 2024.
- [4] B. P. Candra and A. F. Mujaki, "Klasifikasi Jenis Surat Pada Sistem E-Arsip Menggunakan Metode K-Nearest Neighbor," *Jurnal Ilmiah Sistem Informasi*, vol. 5, no. 1, pp. 449–459, Jan. 2026, doi: 10.51903/336mw957.
- [5] L. Zhang, "Features extraction based on Naive Bayes algorithm and TF-IDF for news classification," *PLoS One*, vol. 20, no. 7, p. e0327347, Jul. 2025, doi: 10.1371/journal.pone.0327347.
- [6] Sutriawan, S. Mutmainnah, T. A. Lorusae, and S. Ramadhan, "Model Text Embedding dan TF-IDF+Ngram untuk Meningkatkan Kinerja Algoritma Binary Classifier pada Klasifikasi SMS Palsu," *Jurnal Sistem Informasi Triguna Dharma (JURSI TGD)*, vol. 4, no. 1, pp. 55–64, Jan. 2025, doi: 10.53513/jursi.v4i1.10582.
- [7] K. Munawaroh and A. Alamsyah, "Performance Comparison of SVM, Naïve Bayes, and KNN Algorithms for Analysis of Public Opinion Sentiment Against COVID-19 Vaccination on Twitter," *Journal of Advances in Information Systems and Technology*, vol. 4, no. 2, pp. 113–125, Mar. 2022, doi: 10.15294/jaist.v4i2.59493.
- [8] A. Prayogo, F. Fauziah, and W. Winarsih, "PERBANDINGAN ALGORITMA NAÏVE BAYES DAN K-NEAREST NEIGHBOR PADA KLASIFIKASI JUDUL ARTIKEL PADA JURNAL ILMIAH," *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 8, no. 4, pp. 1327–1338, Nov. 2023, doi: 10.29100/JIPI.V8I4.4141.
- [9] N. Buslim, L. K. Oh, M. H. A. Hardy, and Y. Wijaya, "Comparative Analysis of KNN, Naïve Bayes and SVM Algorithms for Movie Genres Classification Based on Synopsis.," *JURNAL TEKNIK INFORMATIKA*, vol. 15, no. 2, pp. 169–177, Dec. 2022, doi: 10.15408/jti.v15i2.29302.

- [10] J. Akbar, E. Utami, and A. Yaqin, "Multi-Label Classification of Film Genres Based on Synopsis Using Support Vector Machine, Logistic Regression and Naïve Bayes Algorithms," *Institute of Electrical and Electronics Engineers*, pp. 250–255, 2022, doi: 10.1109/ICITISEE57756.2022.10057828.
- [11] R. Amelia and D. B. Santoso, "Prediksi Genre Film Dengan Klasifikasi Multi Kelas Sinopsis Menggunakan Jaringan LSTM," *INTECOMS: Journal of Information Technology and Computer Science*, vol. 6, no. 2, pp. 771–779, Sep. 2023, doi: 10.31539/intecom.v6i2.6961.
- [12] G. P. A. Brahmantha, E. Utami, and A. Yaqin, "KLASIFIKASI GENRE ANIME BERDASARKAN SINOPSIS MENGGUNAKAN ALGORITMA K-NEAREST NEIGHBORS," *Jurnal Manajemen Informatika dan Sistem Informasi*, vol. 7, no. 1, pp. 15–24, Feb. 2024, doi: 10.36595/MISI.V7I1.1016.
- [13] K. Rahayu, V. Fitria, D. Septhya, R. Rahmaddeni, and L. Efrizoni, "Klasifikasi Teks untuk Mendeteksi Depresi dan Kecemasan pada Pengguna Twitter Berbasis Machine Learning," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 3, no. 2, pp. 108–114, Sep. 2023, doi: 10.57152/MALCOM.V3I2.780.
- [14] A. P. Harahap, A. Karim, and R. Rohani, "Analisis Klasifikasi Sentimen Prediksi Rating Aplikasi Apple's AppStore Dengan Menggunakan Metode Algoritma Random Forest," *Building of Informatics, Technology and Science (BITS)*, vol. 6, no. 4, pp. 2371–2379, Mar. 2025, doi: 10.47065/BITS.V6I4.6812.
- [15] M. Utami, "Komparasi Metode Support Vector Machine, Naïve Bayes, dan Random Forest Pada Natural Language Processing Terhadap Berita Negatif," *JCOSIS (Journal Computer Science and Information Systems)*, vol. 2, no. 1, pp. 8–13, May 2025, doi: 10.61567/JCOSIS.V2I1.229.
- [16] M. S. Maulana, Y. Anshori, R. Azhar, R. Laila, and N. T. Lapatta, "IMPLEMENTASI PEMBOBOTAN TF-IDF PADA CHATBOT TELEGRAM UNTUK SISTEM LAYANAN INFORMASI," *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 10, no. 3, pp. 1869–1877, Aug. 2025, doi: 10.29100/JUPI.V10I3.6314.
- [17] M. A. Mazta, E. Saputra, and M. Razi, "PERBANDINGAN KINERJA TF-IDF DAN COUNT VECTORIZATION PADA SISTEM REKOMENDASI JUDUL SKRIPSI BERBASIS CONTENT-BASED FILTERING," *Jurnal Informatika Teknologi dan Sains (Jinteks)*, vol. 7, no. 4, pp. 1807–1816, Nov. 2025, doi: 10.51401/JINTEKS.V7I4.6741.
- [18] N. F. Octavia and Berlilana, "Penerapan Algoritma K-Nearest Neighbor untuk Analisis Sentimen Ulasan Produk Elektronik pada Platform E-Commerce," *Jurnal Algoritma*, vol. 22, no. 2, pp. 2110–2121–2110–2121, Nov. 2025, doi: 10.33364/ALGORITMA/V.22-2.3083.
- [19] A. Davina, M. Putri, N. Sulistianingsih, and R. Rismayati, "Pengaruh Teknik Representasi Teks Bag-of-Words dan TF-IDF terhadap Akurasi Klasifikasi Sentimen Teks Multi-Domain," *Jurnal Teknologi Informasi dan Multimedia*, vol. 7, no. 4, pp. 675–688, Oct. 2025, doi: 10.35746/jtim.v7i4.756.
- [20] I. M. B. Gautama, "Klasifikasi Data Saran Pemustaka di Perpustakaan STIKOM Bali Menggunakan TF-IDF dan Multinomial Naive Bayes," *Jurnal Sistem dan Informatika (JSI)*, vol. 17, no. 2, pp. 62–72, May 2023, doi: 10.30864/JSI.V17I2.490.
- [21] F. R. Valerian, M. Syarief, and D. A. Fatah, "KLASIFIKASI TINGKAT OBESITAS MENGGUNAKAN METODE GBM DAN CONFUSION MATRIX," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 2, pp. 2242–2249, Mar. 2025, doi: 10.36040/JATI.V9I2.13062.
- [22] M. Fadli and R. A. Saputra, "KLASIFIKASI DAN EVALUASI PERFORMA MODEL RANDOM FOREST UNTUK PREDIKSI STROKE," *Jurnal Teknik*, vol. 12, no. 2, Oct. 2023, doi: 10.31000/JT.V12I2.9099.

NOMENKLATUR

$TF(t, d)$	: nilai <i>Term Frequency</i> kata t pada dokumen d
$f_{t,d}$	: jumlah kemunculan kata (<i>term</i>) t dalam dokumen d
$ d $	: jumlah seluruh <i>term</i> t pada dokumen d
N	: jumlah total dokumen dalam <i>corpus</i>
df_t	: jumlah dokumen yang mengandung <i>Term</i> t
$TF(t, d)$	: nilai <i>Term frequency</i> dari poin sebelumnya
$IDF(t)$	: nilai <i>Inverse document frequency</i> dari poin sebelumnya
$w_{t,d}$	: bobot <i>term</i> t pada dokumen d
$\cos(\theta_{ij})$	: nilai <i>Cosine Similarity</i> antara dokumen i dan j
d_{ik}	: bobot TF-IDF fitur kata ke- k pada dokumen i (data uji)
d_{jk}	: bobot TF-IDF fitur kata ke- k pada dokumen j (data latih)

k	: indeks fitur kata
$P(C)$	: probabilitas awal (<i>prior</i>) kelas C
N_c	: jumlah data latih yang masuk ke kelas C
N	: jumlah seluruh data latih
$P(x_i C)$	: probabilitas kemunculan fitur ke- i (<i>likelihood</i>) pada kelas C .
$w_{i,c}$	: Jumlah total bobot TF-IDF fitur ke- i pada dokumen dalam kelas C .
$\sum_{k=1}^n w_{k,c}$	: total bobot TF-IDF seluruh fitur pada kelas C .
$ V $	: jumlah seluruh kata unik pada data latih.
$P(C X)$	: probabilitas dokumen X termasuk ke dalam kelas C (<i>posterior</i>).