Jurnal Elektronik Ilmu Komputer Udayana
Volume 14, No 3. February 2026

p-ISSN: 2301-5373
e-ISSN: 2654-5101

Klasifikasi Kategori Cerita Pendek Menggunakan XGBoost dengan Seleksi Fitur Chi-Square

M. Faisal Afandi[a1], Ngurah Agus Sanjaya ER[a2], Putu Gede Hendra Suputra[a3], Luh Arida Ayu Rahning Putri[a4]

[a]Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Udayana
Jalan Raya Kampus Unud, Jimbaran, Bali, 80361, Indonesia
[1]faisalafandi12345@gmail.com
[2]agus_sanjaya@unud.ac.id
[3]hendra.suputra@unud.ac.id
[4]rahningputri@unud.ac.id

Abstract

Text classification is a key challenge in natural language processing, particularly in categorizing texts by genre. This study aims to classify Indonesian short stories into three genres: romance, horror, and religion. Two ensemble machine learning algorithms, XGBoost and Random Forest, were employed. Prior to training, the data underwent preprocessing and feature extraction using TF-IDF, followed by feature selection with the Chi-Square method to enhance relevance and efficiency. Models were trained with various hyperparameter configurations and validated using 5-Fold Cross Validation. Experimental results show that Chi-Square selection significantly improved performance by reducing training time and maintaining accuracy. For instance, XGBoost with 500 features required 102 seconds compared to 427 seconds without Chi-Square, while Random Forest required only 5 seconds compared to 14.8 seconds. On the test set, XGBoost achieved the best results with 90% precision, 91% recall, and 90% F1-Score, outperforming Random Forest which obtained 86% precision, 87% recall, and 85% F1-Score. These results confirm that XGBoost generalizes better on unseen data..

Keywords: *Text Classification, Short Stories, XGBoost, Random Forest, Chi-Square, Ensemble Learning.*

1. Pendahuluan

Klasifikasi teks merupakan salah satu tantangan utama dalam *natural language processing* (NLP). Tidak seperti data numerik yang memiliki struktur yang jelas, teks memiliki karakteristik tidak terstruktur dan ambiguitas semantik, sehingga menyulitkan proses klasifikasi secara otomatis [1]. Salah satu permasalahan yang diangkat pada penelitian ini yaitu pengelompokan cerita pendek ke dalam suatu genre tertentu, seperti romantis, horor, dan agama, yang masing-masing memiliki ciri khas bahasa dan gaya naratif yang bisa saling tumpang tindih.

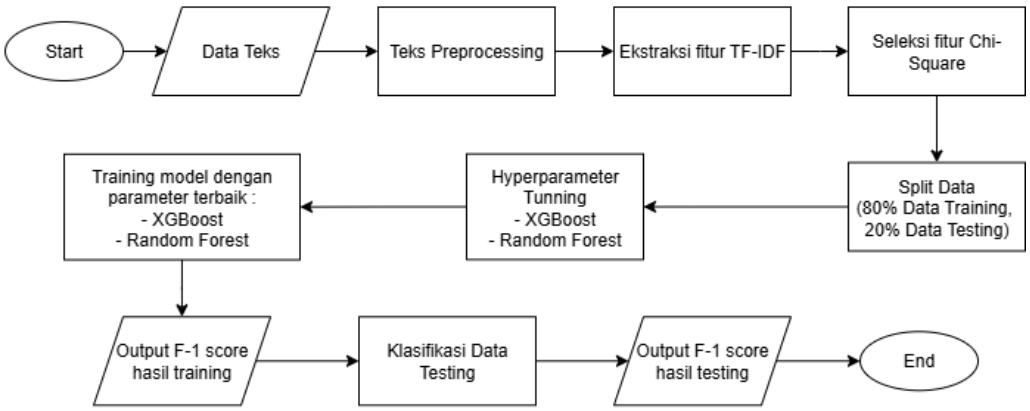
Permasalahan klasifikasi pada penelitian ini termasuk dalam kategori *multiclass classification*, di mana suatu cerita pendek akan diklasifikasikan ke dalam salah satu kategori dari banyak kategori secara bersamaan. Kompleksitas ini menuntut pemilihan algoritma yang tepat, baik dari sisi akurasi maupun efisiensi. Penelitian-penelitian sebelumnya telah mengeksplorasi berbagai pendekatan untuk klasifikasi teks, seperti penggunaan *Naïve Bayes*, *Support Vector Machine* (SVM), hingga *deep learning* berbasis *Long Short-Term Memory* (LSTM). Namun, studi yang secara spesifik membandingkan algoritma *ensemble learning* seperti XGBoost dan Random Forest dalam konteks klasifikasi genre cerita pendek masih terbatas.

XGBoost merupakan salah satu algoritma *gradient boosting* yang terkenal dengan performa tinggi dalam kompetisi *machine learning* karena efisiensi dan akurasinya yang tinggi [2]. Di sisi lain, Random Forest menggunakan pendekatan *bagging* dan cenderung lebih stabil dalam menghindari *overfitting* pada data pelatihan. Perbandingan keduanya menjadi penting mengingat keduanya merupakan pendekatan *ensemble*, tetapi dengan strategi pembentukan model yang berbeda. Dalam penelitian ini juga mengeksplorasi pengaruh seleksi fitur menggunakan metode *Chi-Square*, yang bertujuan untuk meningkatkan relevansi fitur dan memperbaiki performa model klasifikasi.

Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk mengevaluasi dan membandingkan performa XGBoost dan Random Forest dalam tugas klasifikasi teks cerita pendek berbahasa Indonesia berdasarkan genre. Evaluasi dilakukan berdasarkan metrik *F1-Score* dengan mempertimbangkan berbagai kombinasi *hyperparameter* serta pengaruh dari seleksi fitur *Chi-Square*.

2. Metode Penelitian

Penelitian ini dilakukan melalui serangkaian tahapan yang meliputi pengumpulan data, *preprocessing* teks, ekstraksi fitur menggunakan TF-IDF, seleksi fitur dengan Chi-Square, pelatihan model, dan evaluasi performa menggunakan metrik F1-score. Gambar 1 menunjukkan alur proses penelitian secara umum.



Gambar 1. Alur Penelitian

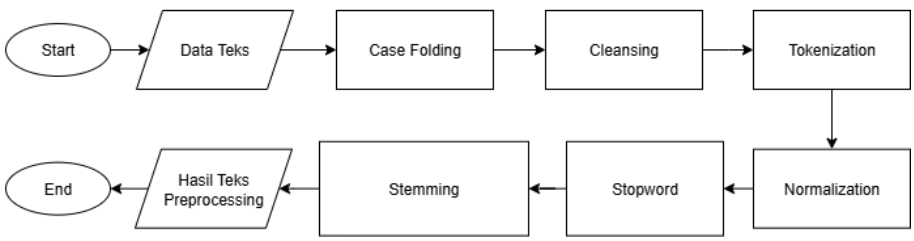
2.1. Pengumpulan Data

Data yang digunakan dalam penelitian ini bersumber dari situs *cerpenmu.com*, yang berisi berbagai cerita pendek berbahasa Indonesia. Data yang digunakan sebanyak 632 cerita pendek, dan dipilih hanya cerita pendek yang memiliki satu label genre, yaitu romantis, horor, dan agama. Data ini mewakili tiga kelas dalam skenario *multiclass classification*.

2.2. Pra-pemrosesanTeks

Sebelum dilakukan proses ekstraksi fitur, teks cerita pendek akan diproses terlebih dahulu melalui beberapa tahapan *preprocessing* seperti pada Gambar 2. Tahapan ini mencakup:

- *Case folding*: mengubah seluruh huruf menjadi huruf kecil.
- *Cleansing*: menghapus karakter khusus dan tanda baca yang tidak diperlukan.
- *Tokenisasi*: memisahkan kalimat menjadi token kata.
- *Normalisasi*: mengubah kata tidak baku menjadi kata baku, seperti "yg" menjadi "yang".
- *Stopword removal*: menghapus kata-kata umum yang tidak memiliki nilai informasi.
- *Stemming*: mengubah kata menjadi bentuk dasar.

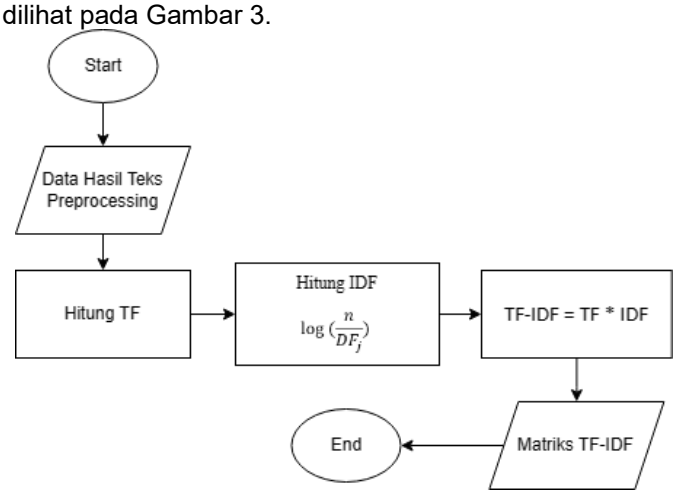


Gambar 2. Alur Pra-pemrosesanTeks

p-ISSN: 2301-5373
e-ISSN: 2654-5101

2.3. Ekstraksi Fitur TF-IDF

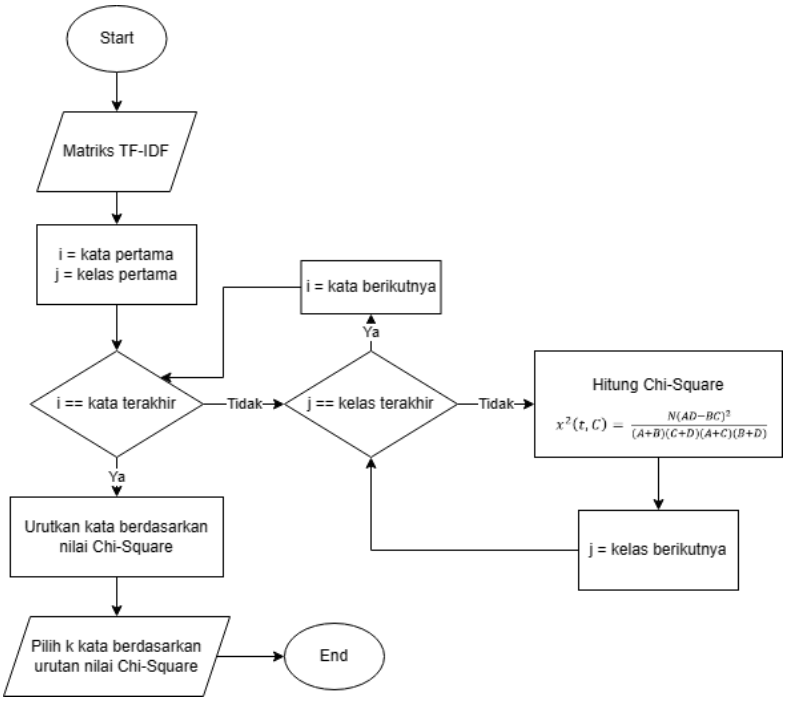
Setelah teks *preprocessing* dilakukan, tahap berikutnya adalah ekstraksi fitur menggunakan metode TF-IDF (*Term Frequency-Inverse Document Frequency*). Metode ini memberikan bobot terhadap kata-kata dalam dokumen berdasarkan frekuensi kemunculannya dan seberapa unik kata tersebut di seluruh dokumen [3]. Hasil dari proses ini adalah matriks TF-IDF yang merepresentasikan setiap dokumen dalam bentuk vektor numerik. Proses ekstraksi fitur TF-IDF dapat dilihat pada Gambar 3.



Gambar 3. Ekstraksi Fitur TF-IDF

2.4. Seleksi Fitur Chi-Square

Untuk meningkatkan performa klasifikasi dan mengurangi dimensi fitur, dilakukan seleksi fitur menggunakan metode Chi-Square. Teknik ini mengukur seberapa kuat hubungan antara setiap kata (fitur) dengan kelas target. Fitur dengan nilai Chi-Square tertinggi dipilih sebagai fitur yang paling relevan. Proses seleksi fitur Chi-Square dapat dilihat pada Gambar 4.

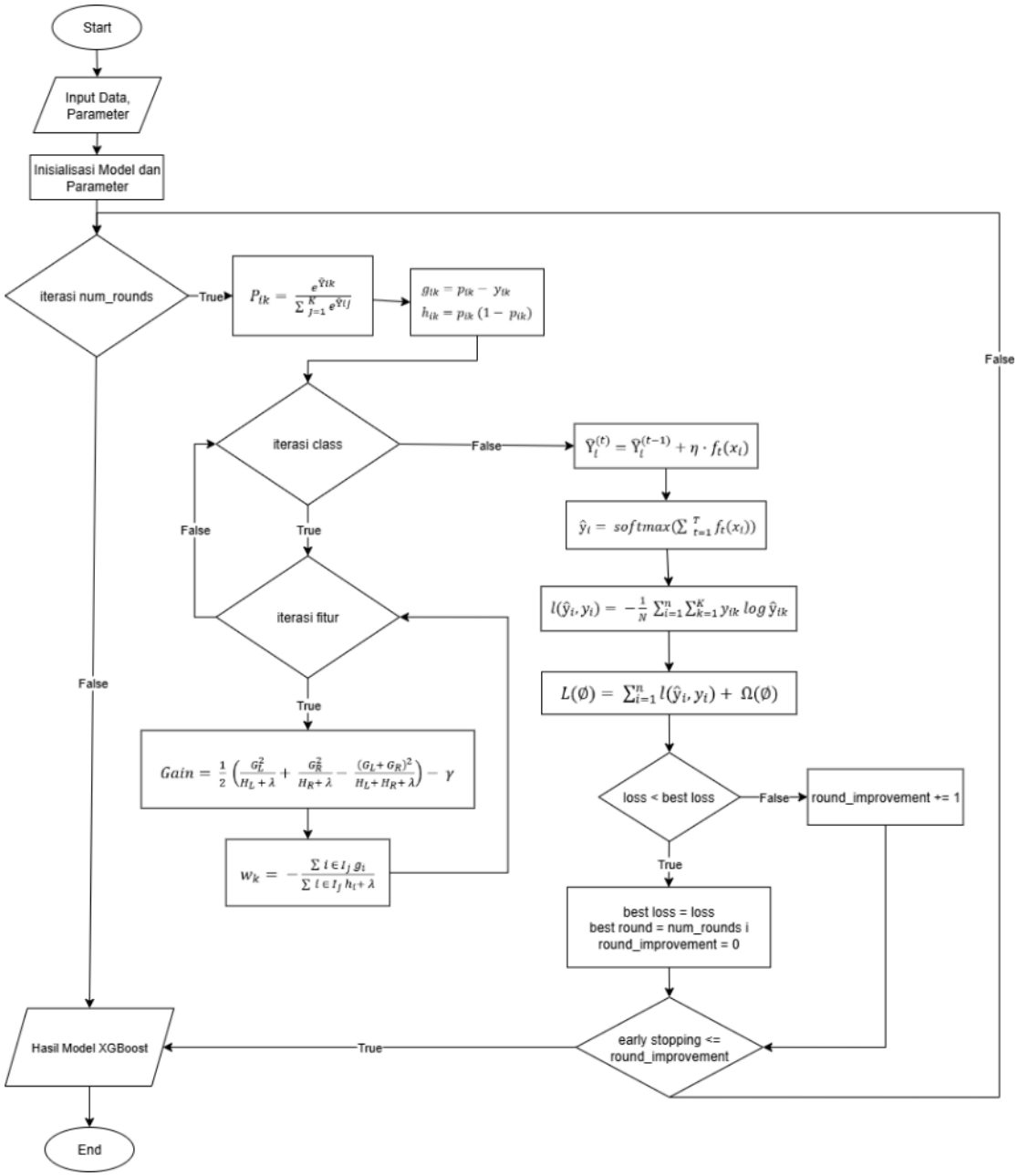


Gambar 4. Seleksi Chi-Square

2.5. Klasifikasi Teks XGBoost

XGBoost membangun model *gradient boosting decision tree* yang dioptimalkan secara bertahap untuk memperbaiki kesalahan prediksi dari model sebelumnya. Proses pelatihan

dilakukan dengan membagi data menjadi data latih dan data uji, serta menerapkan teknik *cross-validation*.



Gambar 5. Alur Pelatihan XGBoost

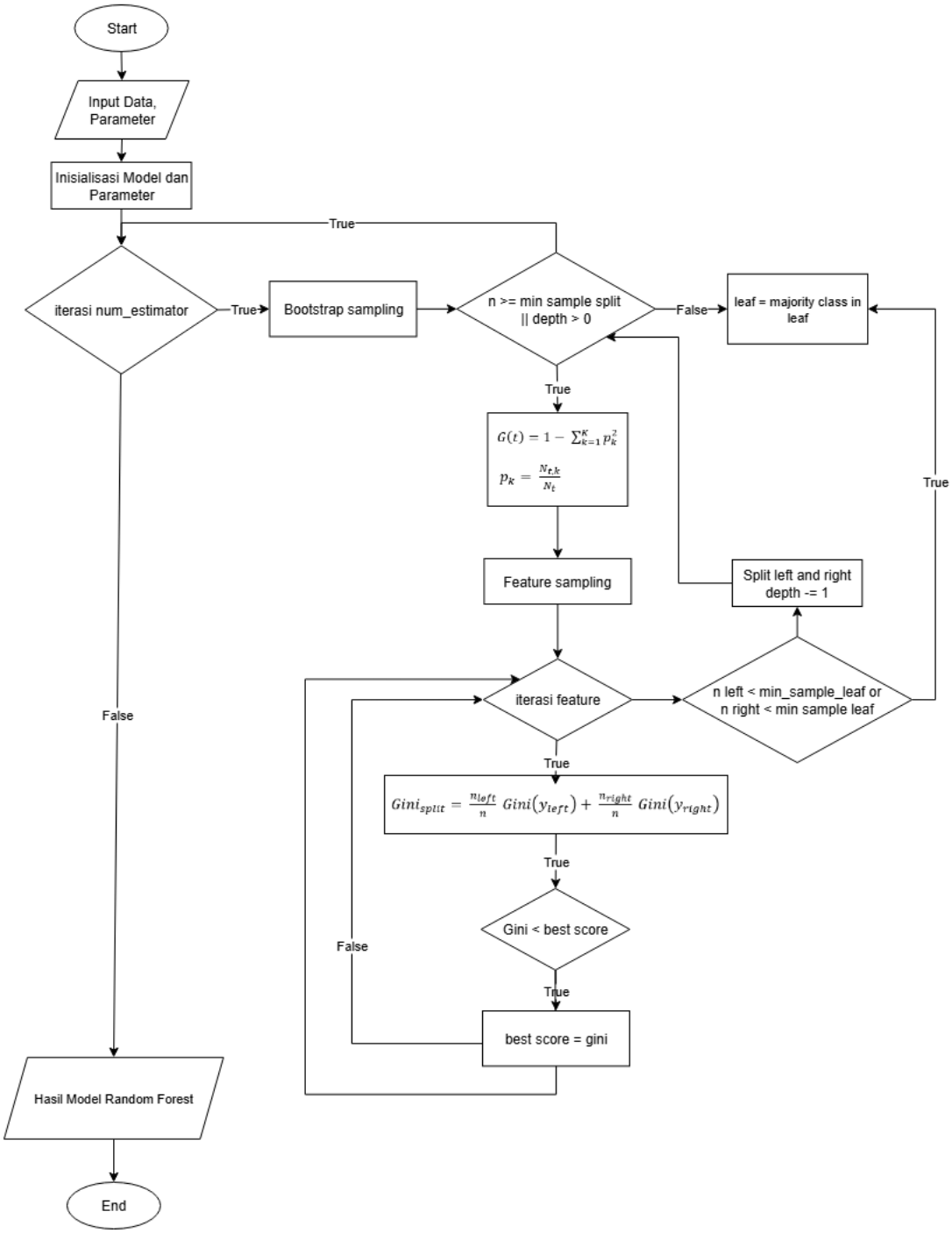
Pada gambar 5 merupakan proses pelatihan model XGBoost untuk kasus klasifikasi multiclass menggunakan pendekatan *second-order Taylor expansion*. Proses dimulai dengan memasukkan data dan parameter model, kemudian menginisialisasi nilai prediksi awal. Pada setiap iterasi *boosting round*, model menghitung probabilitas kelas menggunakan fungsi *softmax*, lalu menghitung nilai gradien dan hessian untuk setiap data dan kelas. Selanjutnya, dilakukan pencarian *split* terbaik dengan mengiterasi setiap kelas dan fitur, di mana kualitas *split* diukur menggunakan perhitungan *gain*. Bobot prediksi pada setiap *leaf* ditentukan dari rasio negatif jumlah gradien terhadap jumlah hessian ditambah parameter regularisasi. Model kemudian memperbarui prediksi dengan menambahkan kontribusi dari pohon baru yang dibangun. Setelah itu, *loss function* dihitung menggunakan *softmax cross-entropy* yang dikombinasikan dengan regularisasi, lalu dibandingkan dengan nilai *loss* terbaik sebelumnya untuk menentukan apakah terjadi perbaikan. Jika tidak ada perbaikan selama sejumlah iterasi

p-ISSN: 2301-5373
e-ISSN: 2654-5101

tertentu, proses dihentikan melalui mekanisme *early stopping*. Pelatihan berakhir ketika jumlah iterasi maksimum tercapai atau kondisi *early stopping* terpenuhi [4].

2.6. Klasifikasi Teks Random Forest

Berbeda dengan XGBoost, pada Random Forest model dilatih secara independen menggunakan teknik *bagging* untuk meningkatkan kestabilan dan akurasi prediksi. Setiap pohon dilatih pada subset data latih yang diambil secara acak dengan pengembalian (*bootstrap sample*), serta menggunakan subset fitur yang dipilih secara acak pada setiap percabangan node (*feature subsampling*). Hal ini membuat setiap pohon memiliki variasi struktur sehingga mengurangi risiko *overfitting* [5].



Gambar 6. Alur Training Random Forest

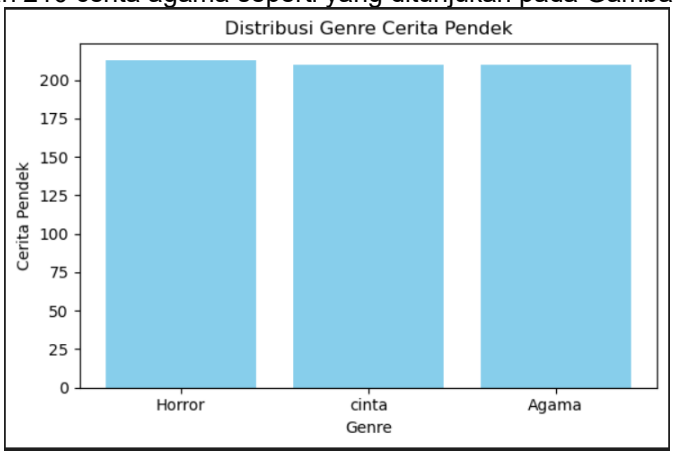
Pada gambar 6 ditunjukkan proses pelatihan model Random Forest untuk klasifikasi multiclass. Proses dimulai dengan memasukkan data dan parameter model, kemudian menginisialisasi model dengan jumlah pohon (*num_estimators*) yang akan dibangun. Pada setiap iterasi, dilakukan *bootstrap sampling* untuk membuat subset data latih acak yang digunakan membangun pohon. Proses pembentukan pohon dimulai dari root dengan memeriksa apakah jumlah sampel pada node memenuhi syarat minimum pemisahan (*min_sample_split*) dan kedalaman *tree* belum mencapai batas (*depth*). Jika syarat tidak terpenuhi, node menjadi *leaf* dengan nilai prediksi kelas mayoritas. Jika syarat terpenuhi, dihitung nilai *Gini Impurity* untuk mengukur ketidakmurnian data pada node. Selanjutnya dilakukan *feature sampling* untuk memilih subset fitur secara acak, kemudian diiterasi setiap fitur untuk mencari *split* terbaik berdasarkan nilai *Gini split*. *Split* terbaik dipilih jika menghasilkan nilai *Gini* yang lebih kecil dari skor terbaik sebelumnya. Node kemudian dipecah menjadi cabang kiri dan kanan, dan proses ini diulang secara rekursif hingga memenuhi kriteria penghentian (*stopping criteria*). Setelah semua pohon terbentuk, prediksi akhir ditentukan berdasarkan *majority voting* dari seluruh *tree* dalam ensemble.

2.7. Evaluasi Sistem

Evaluasi model dilakukan untuk mengukur performa klasifikasi berdasarkan metrik *F1-score*, yang mempertimbangkan nilai presisi dan *recall*. Evaluasi dilakukan pada data uji setelah model dilatih menggunakan *hyperparameter* terbaik

3. Result and Discussion

Untuk mengevaluasi kedua algoritma XGBoost dan Random Forest dilakukan pelatihan model menggunakan data yang sama. Dataset yang digunakan terdapat 632 cerita pendek dengan tiga kategori genre, yaitu cinta, horor, dan agama. Setiap cerita hanya mewakili satu genre, sehingga cerita ambigu atau tidak jelas kategorinya dihapus. Data disimpan dalam file Excel (.xlsx) dengan kolom Judul, Isi Cerita, dan Genre, serta diatur seimbang untuk menghindari bias, yaitu 213 cerita horor, 210 cerita cinta, dan 210 cerita agama seperti yang ditunjukan pada Gambar 7.



Gambar 7. Distribusi Data

3.1. Hyperparameter Tunning

a. Hyperparameter Tunning pada Algoritma XGBoost
Pada algoritma XGBoost dilakukan hyperparameter tunning menggunakan kombinasi yang tercantum pada Tabel 1.

Tabel 1. Hyperparameter XGBoost

Hyperparameter	Parameter
num_rounds	50, 100
learning_rate	0.1, 0.3
depth	2, 4, 6

p-ISSN: 2301-5373
e-ISSN: 2654-5101

Nilai *subsample* = 0.8 digunakan untuk *row* dan *column sampling* secara acak pada setiap pohon. Kombinasi hyperparameter ini memberikan akurasi dan waktu pelatihan terbaik pada masing-masing konfigurasi seleksi fitur Chi-Square dengan variasi jumlah fitur: 500, 1000, 1500, dan tanpa seleksi (5000 fitur tf-idf terbaik). Evaluasi menggunakan skema 5-*Fold Cross Validation*. Detail proses hyperparameter tuning terbaik pada XGBoost dapat dilihat pada Tabel 2.

Tabel 2. Akurasi Hyperparameter Terbaik XGBoost

Jumlah Fitur	*Hyperparameter* Terbaik	*Fold* (1-5)	Rata-rata Akurasi	Rata-rata Waktu
500 Fitur Chi-Square	*num_rounds* = 100, *learning_rate* = 0,1, *depth* = 4	87,13%	88,32%	102,60 detik
		87,13%		
		88,12%		
		84,16%		
		95,05%		
1000 Fitur Chi-Square	*num_rounds* = 50, *learning_rate* = 0,1, *depth* = 6	87,13%	88,32%	104,92 detik
		87,13%		
		86,14%		
		88,12%		
		93,07%		
1500 Fitur Chi-Square	*num_rounds* = 50, *learning_rate* = 0,1, *depth* = 6	86,14%	87,33%	175,09 detik
		86,14%		
		86,14%		
		85,15%		
		93,07%		
Tanpa Chi-Square	*num_rounds* = 100, *learning_rate* = 0,3, *depth* = 4	88,12%	87,92%	427,118 detik
		88,12%		
		89,11%		
		85,15%		
		89,11%		

Pada Tabel 2 menunjukkan bahwa penggunaan seleksi fitur Chi-Square dapat sedikit meningkatkan akurasi model dan semakin sedikit fitur yang digunakan semakin cepat waktu pelatihan yang diperlukan.

b. Hyperparameter Tunning pada Algoritma Random Forest
Pada algoritma Random Forest dilakukan hyperparameter tunning menggunakan kombinasi yang tercantum pada Tabel 3.

Tabel 3. Hyperparameter Random Forest

Hyperparameter	Parameter
n_estimators	100, 200
max_depth	4, 6
min_samples_leaf	1, 3
max_features	sqrt, log2

Pada algoritma Random Forest, eksperimen dilakukan dengan kombinasi hyperparameter pada Tabel 4 dan metode seleksi fitur Chi-Square dengan variasi jumlah fitur: 500, 1000, 1500, dan tanpa seleksi (5000 fitur tf-idf terbaik). Evaluasi menggunakan skema 5-*Fold Cross Validation* untuk memperoleh akurasi tiap fold dan rata-rata keseluruhan.

Tabel 4. Akurasi Hyperparameter Terbaik Random Forest

Jumlah Fitur	*Hyperparameter* Terbaik	*Fold* (1-5)	Rata-rata Akurasi	Rata-rata Waktu
500 Fitur Chi-Square	*n_estimators* = 200, *max_depth* = 6, *min_samples_leaf* = 3, *max_features* = sqrt	85,15%	87,33%	4,97 detik
		84,16%		
		88,12%		
		90,10%		
		89,11%		
1000 Fitur Chi-Square	*n_estimators* = 100, *max_depth* = 6, *min_samples_leaf* = 3, *max_features* = sqrt	85,15%	87,33%	5,05 detik
		85,15%		
		88,12%		
		90,10%		
		88,12%		
1500 Fitur Chi-Square	*n_estimators* = 100, *max_depth* = 6, *min_samples_leaf* = 3, *max_features* = sqrt	83,17%	86,73%	6,19 detik
		86,14%		
		87,13%		
		88,12%		
		89,11%		
Tanpa Chi-Square	*n_estimators* = 100, *max_depth* = 6, *min_samples_leaf* = 3, *max_features* = sqrt	78,22%	81,58%	14,84 detik
		77,23%		
		85,15%		
		87,13%		
		80,20%		

Pada tabel 4 menunjukan bahwa model Random Forest juga mengalami peningkatan performa dan proses waktu pelatihan yang signifikan dengan penggunaan seleksi fitur Chi-Square.

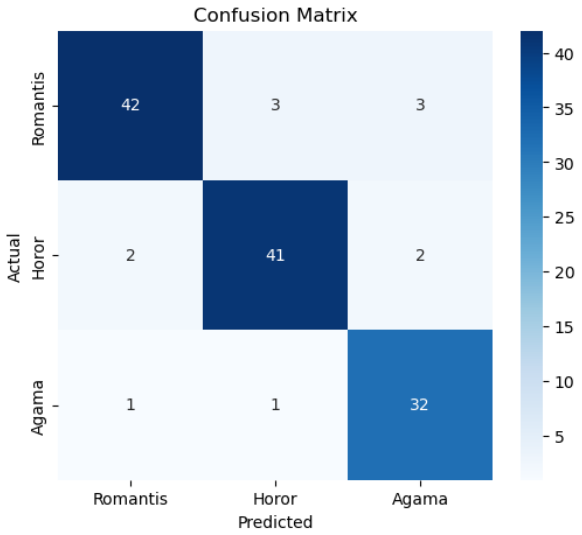
3.2. Evaluasi

Evaluasi dilakukan pada model XGBoost dan Random Forest menggunakan parameter terbaik hasil pencarian hyperparameter. XGBoost dengan num_rounds=100, learning_rate=0.1, depth=4. Sementara itu, Random Forest menggunakan parameter n_estimators=200, max_depth=6, min_samples_leaf=3, max_features=sqrt

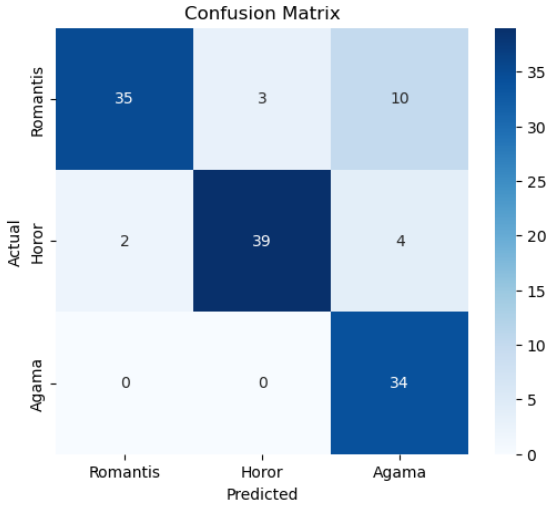
Tabel 5. Evaluasi XGboost dan Random Forest dengan Chi-Square

Algoritma	Precision	Recall	F1-Score
XGBoost	90 %	91 %	90 %
Random Forest	86 %	87 %	85 %

p-ISSN: 2301-5373
e-ISSN: 2654-5101



Gambar 8. Confusion Matrix XGBoost



Gambar 9. Confusion Matrix Random Forest

Berdasarkan Tabel 5, XGBoost mencatat F1-Score 90%, lebih unggul dibandingkan Random Forest yang mencapai 85%. Hal ini sejalan dengan karakteristik metode *boosting* yang lebih efektif dalam mengurangi bias. Hasil pada Tabel 2 dan 4 juga menunjukkan bahwa seleksi fitur Chi-Square dapat meningkatkan performa dan efisiensi model. XGBoost dengan 500 fitur mempertahankan akurasi 88% dengan waktu pelatihan 102 detik, jauh lebih cepat dibanding tanpa seleksi. Random Forest juga menunjukkan peningkatan serupa dengan akurasi 87% dan waktu pelatihan sekitar 5 detik. Secara keseluruhan, seleksi fitur membantu menurunkan kompleksitas tanpa mengorbankan akurasi, dengan XGBoost unggul dalam akurasi dan generalisasi, sementara Random Forest lebih stabil dan cepat.

4. Conclusion

Berdasarkan hasil penelitian dan implementasi yang telah dilakukan, dapat disimpulkan beberapa hal sebagai berikut:

1. Kombinasi hyperparameter berpengaruh signifikan terhadap performa algoritma. XGBoost dengan konfigurasi optimal mencapai F1-Score 90%, sementara Random Forest hanya 85%, sehingga menunjukkan keunggulan pendekatan boosting dalam menangkap pola kompleks pada data teks.
2. Seleksi fitur Chi-Square terbukti meningkatkan efisiensi tanpa mengorbankan akurasi. Pada XGBoost, 500 fitur cukup menjaga akurasi 88% dengan waktu pelatihan lebih singkat (102

detik dibanding 427 detik). Random Forest juga memperoleh waktu pelatihan lebih cepat dengan performa relatif stabil.

References

[1] I. G. A. P. Arimbawa and N. A. S. ER, "Penerapan Metode Adaboost Untuk Multi-Label Classification Pada Dokumen Teks," *Jurnal Elektronik Ilmu Komputer Udayana,* vol. 9, no. 1, pp. 127-140, 2020.

[2] I. R. Hendrawan, "PERBANDINGAN ALGORITMA NAÏVE BAYES, SVM DAN XGBOOST DALAM KLASIFIKASI TEKS SENTIMEN MASYARAKAT TERHADAP PRODUK LOKAL DI INDONESIA," *Jurnal TRANSFORMASI,* vol. 18, no. 1, pp. 1-8, 2022.

[3] A. Nurhadi, "Klasifikasi Berita Menggunakan Metode Support Vector Machine," *Jurnal Nasional Komputasi dan Teknologi Informasi,* vol. 5, no. 2, pp. 269-278, 2022.

[4] D. J. Putri, M. Dwifebri and A. , "Text Classification of Indonesian Translated Hadith Using XGBoost Model and Chi-Square Feature Selection," *Building of Informatics, Technology and Science,* vol. 4, no. 4, pp. 1732-1738, 2023.

[5] N. Husin, "Komparasi Algoritma Random Forest, Naïve Bayes, dan Bert Untuk Multi-Class Classification Pada Artikel Cable News Network (CNN)," *Jurnal Esensi Infokom Jurnal Esensi Sistem Informasi dan Sistem Komputer,* vol. 7, no. 1, pp. 75-84, 2023.