Department of Digital Business
Journal of Artificial Intelligence and Digital Business (RIGGS)
Homepage: https://journal.ilmudata.co.id/index.php/RIGGS
Vol. 4 No. 4 (2026) pp: 10390-10397
P-ISSN: 2963-9298, e-ISSN: 2963-914X

Perbandingan Algoritma XGBoost dan Random Forest dalam Klasifikasi Surat Masuk Pemerintahan

Fadila Ananda Kartika Hidayat[1*], Neny Sulistianingsih[2], Rifqi Hammad[3]
[1]Program Studi Ilmu Komputer, Fakultas Teknik, Universitas Bumigora
[2]Magister Ilmu Komputer, Program Pasca Sarjana, Universitas Bumigora
[3]Program Studi Rekayasa Perangkat Lunak, Fakultas Teknik, Universitas Bumigora
fadilaananda2004@gmail.com*, neny.sulistianingsih@universitasbumigora.ac.id, rifqi.hammad@universitasbumigora.ac.id

Abstrak

Pengelolaan surat masuk pada lingkungan pemerintahan daerah berperan penting dalam mendukung efektivitas administrasi dan pengambilan keputusan oleh pimpinan daerah. Surat masuk merupakan salah satu bentuk komunikasi resmi yang harus dikelola secara sistematis agar informasi yang terkandung di dalamnya dapat segera ditindaklanjuti sesuai dengan bidang administrasi terkait. Volume surat yang tinggi sering menimbulkan berbagai kendala, terutama dalam proses identifikasi isi surat dan pengelompokan berdasarkan bidang administrasi yang berwenang. Kondisi tersebut berpotensi menyebabkan keterlambatan disposisi, kesalahan pengelompokan surat, serta menurunnya kualitas pelayanan administrasi apabila masih dilakukan secara manual. Penelitian ini bertujuan untuk membandingkan kinerja dua algoritma machine learning, yaitu Extreme Gradient Boosting (XGBoost) dan Random Forest, dalam melakukan klasifikasi surat masuk kepala daerah secara otomatis dan terstruktur. Data yang digunakan dalam penelitian ini meliputi arsip surat masuk pemerintah daerah periode 2021–2022 serta hasil ekstraksi dokumen surat berbentuk PDF yang diperoleh dari aplikasi persuratan SRIKANDI menggunakan pustaka pdfplumber untuk menghasilkan data teks yang dapat diolah.. Tahapan penelitian mencakup proses pra-pemprosesan data, pembagian data menjadi data latih dan data uji, pelatihan model, serta evaluasi kinerja model menggunakan indikator accuracy, precision, recall, dan F1-score. Berdasarkan hasil pengujian, algoritma XGBoost menunjukkan performa yang lebih unggul dengan nilai akurasi sebesar 81,87% dan F1-score 82,00%, dibandingkan Random Forest yang hanya mencapai akurasi 76,00% dan F1-score 76,03%. Dengan demikian, XGBoost dinilai lebih efektif untuk mendukung proses klasifikasi surat dalam implementasi e-government di lingkungan pemerintahan daerah.

Kata kunci: Klasifikasi Surat, Pemerintahan, XGBoost, Random Forest

1. Latar Belakang

Efisiensi administrasi publik merupakan salah satu pilar utama dalam mewujudkan tata kelola pemerintahan yang baik [1]. Sejalan dengan implementasi *e-government* di Indonesia, berbagai instansi pemerintah daerah terus berupaya melakukan transformasi digital guna meningkatkan efektivitas, akuntabilitas, serta transparansi pelayanan [2], [3], [4]. Salah satu aspek administrasi yang penting namun masih sering menghadapi tantangan adalah pengelolaan surat masuk dan tata naskah dinas [5]. Volume surat yang tinggi, bersamaan dengan proses klasifikasi manual, kerap menyebabkan keterlambatan, kesalahan disposisi, serta kesulitan dalam proses penelusuran arsip [6]. Pengelolaan surat yang kurang efisien berdampak langsung terhadap proses tindak lanjut surat dan kelancaran operasional birokrasi [7], [8].

Untuk mengatasi tantangan tersebut, pengembangan sistem klasifikasi surat dengan pendekatan *machine learning* menjadi solusi yang relevan [9]. Dalam konteks *machine learning*, metode *ensemble tree* (ansambel pohon) memiliki performa yang baik sehingga banyak diterapkan pada berbagai klasifikasi. Penelitian ini berfokus pada perbandingan dua algoritma ensemble, yaitu *Random Forest (RF)* dan *Extreme Gradient Boosting (XGBoost)*. Kedua algoritma ini dipilih karena mendefinisikan dua pendekatan ensemble yang berbeda, Random Forest menerapkan *bagging* dengan membangun pohon secara independen [10], sedangkan *XGBoost* menggunakan boosting dengan membangun pohon secara berurutan [11].

Kedua algoritma tersebut akan digunakan untuk klasifikasi teks. Agar konten teks dapat diproses secara otomatis oleh model diperlukan konversi data menjadi representasi fitur terstruktur. Metode *Term Frequency–Inverse*

Document Frequency (TF-IDF) dipilih sebagai teknik ekstraksi fitur, karena merupakan pendekatan yang umum dan efektif dalam pemodelan klasifikasi teks [12], [13]. TF-IDF memberikan bobot pada kata berdasarkan tingkat kepentingannya dalam dokumen, sehingga model mampu menangkap pola linguistik dan konteks secara lebih akurat [14]. Representasi vektor hasil TF-IDF inilah yang kemudian akan digunakan sebagai input untuk melatih dan membandingkan model *Random Forest* dan *XGBoost.*

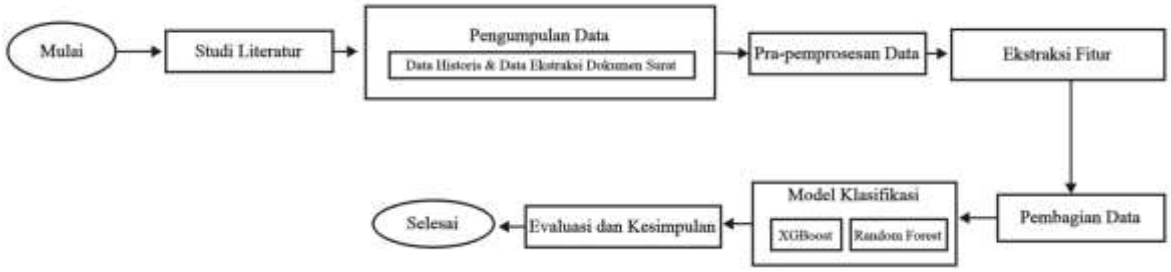
Terdapat beberapa studi komparatif yang mengukur kinerja kedua model ini dengan hasil yang bervariasi. Untuk analisis sentimen rumah makan, Wati et al. [15] menyatakan *XGBoost* (*Accuracy* 68%) lebih superior dibandingkan *Random Forest* (*Accuracy* 63%). Dalam konteks data performa UMKM, Erkamim et al. [16] juga menemukan *XGBoost* (*F1-Score* 0,950) sedikit lebih baik dari *Random Forest* (*F1-Score* 0,944), meskipun *Accuracy*-nya identik. Namun, keunggulan *XGBoost* tidak bersifat mutlak. Dalam klasifikasi data kesehatan mental, Wardana dan Rahim [11] menemukan keunggulan *XGBoost* (*Accuracy* 99,82%) sangat tipis dibandingkan *Random Forest* (*Accuracy* 99,04%). Sebaliknya, penelitian oleh Yulianti et al. [17] tentang klasifikasi penerima BPNT menunjukkan bahwa setelah *undersampling*, Random Forest justru lebih unggul dalam metrik *Recall* (80,01%) dibandingkan *XGBoost* (74,04%). Bahkan dalam klasifikasi berita oleh Juarto dan Yulianto [18], kedua model (*Random Forest Accuracy* 88% dan *XGBoost Accuracy* 89%) dikalahkan oleh model *boosting* lain. Sementara itu, penelitian yang relevan dengan metodologi ini (TF-IDF + *XGBoost*) oleh Hendrawan et al. [19] untuk sentimen produk lokal, berhasil mencapai F1-Score 0,940, dan Damaryanti dan Supriyanto [20] berhasil menerapkan *XGBoost* untuk klasifikasi arsip sistem Srikandi dengan *Accuracy* 77%.

Meskipun banyak studi menunjukkan bahwa *XGBoost* unggul, implementasi terbaik tetap dipengaruhi oleh karakteristik data yang digunakan. Namun demikian, dalam praktiknya, efektivitas sistem *e-office* yang berperan sebagai bagian dari penerapan *e-government* [8] serta aplikasi kearsipan seperti Srikandi [20] sangat bergantung pada ketepatan klasifikasi pada tahapan awal pemrosesan surat.

Oleh karena itu, penelitian ini bertujuan untuk membandingkan performa algoritma *Random Forest* dan *XGBoost* dalam klasifikasi surat masuk pemerintah. Penelitian ini menggunakan metode TF-IDF sebagai teknik ekstraksi fitur dari data surat yang digunakan. Evaluasi model dilakukan berdasarkan indikator evaluasi berupa *Accuracy, Precision, Recall,* dan *F1-Score* untuk menentukan model yang paling efektif dan berpotensi diimplementasikan dalam mendukung transformasi digital serta peningkatan efisiensi administrasi di pemerintah daerah.

2. Metode Penelitian

Penelitian ini dilakukan untuk membandingkan kinerja model *XGBoost* dan *Random Forest* dalam melakukan klasifikasi surat masuk kepala daerah. Pendekatan penelitian bersifat kuantitatif dengan memanfaatkan data surat historis serta data rill yang diperoleh melalui proses ekstraksi dokumen. Secara umum, tahapan metodologi dalam penelitian ini meliputi studi literatur, pengumpulan data surat, preprocessing data teks, proses ekstraksi fitur menggunakan metode TF-IDF, pembagian data, pembangunan model klasifikasi menggunakan algoritma *XGBoost* dan *Random Forest*, evaluasi hasil klasifikasi, serta penyusunan kesimpulan akhir berdasarkan hasil evaluasi kinerja model. Tahapan penelitian tersebut digambarkan secara umum pada Gambar 1.



Gambar 1. Metodologi Penelitian

2.1. Studi Literatur

Tahap studi literatur ini bertujuan mengidentifikasi metodologi klasifikasi yang paling sesuai untuk data surat masuk kepala daerah. Data penelitian bersifat tekstual dikumpulkan dari arsip historis dan hasil ekstraksi dokumen PDF. *Library pdfplumber* terbukti efektif untuk mengekstraksi teks dari PDF, yang merupakan tahap pengumpulan data yang akan diolah lebih lanjut dalam alur kerja *Natural Language Processing* (NLP) [9]. Mengingat data input

DOI: https://doi.org/10.31004/riggs.v4i4.5044
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

berupa kolom 'asal surat' dan 'perihal' yang berbentuk teks, metode ekstraksi fitur *Term Frequency–Inverse Document Frequency* (TF-IDF) dipilih karena kemampuannya menilai kepentingan kata dalam sebuah dokumen relatif terhadap seluruh korpus [13]. Metode ini efektif memberikan bobot informatif pada kata-kata kunci [14], yang esensial untuk klasifikasi teks.

Untuk klasifikasi, algoritma berbasis *ensemble tree* dipilih karena kinerjanya yang stabil pada dataset berskala menengah hingga besar. *Random Forest* (RF) digunakan sebagai model yang terbukti mampu menangani data berdimensi tinggi dengan konsisten [10]. Sebagai pembanding, model *Extreme Gradient Boosting* (*XGBoost*) dipilih karena sifat boosting-nya yang membangun model secara bertahap untuk memperbaiki kesalahan prediksi dari iterasi sebelumnya [11]. Berbagai studi komparatif secara konsisten menunjukkan bahwa *XGBoost* memiliki keunggulan performa (*accuracy* dan *F1-score*) dibandingkan *Random Forest* dalam tugas klasifikasi serupa [15], [16], [17]. Berdasarkan studi literatur ini, kedua model dipilih untuk dibandingkan guna menentukan pendekatan paling akurat untuk klasifikasi surat masuk pemerintah.

2.2. Pengumpulan Data

Data yang digunakan dalam penelitian ini terdiri dari dua sumber utama yang dikombinasikan, yaitu data historis surat masuk dan data surat hasil ekstraksi dokumen PDF yang diperoleh dari Biro Administrasi Pimpinan Sekretariat Daerah Pemerintah Provinsi Nusa Tenggara Barat. Data historis surat merupakan data arsip administratif yang telah terdokumentasi sebelumnya oleh instansi, dan digunakan sebagai sumber referensi untuk membangun dataset awal. Sementara itu, data surat rill diperoleh melalui proses ekstraksi konten teks dari dokumen surat PDF menggunakan library *pdfplumber* dari *Python* sehingga memungkinkan data surat yang sebelumnya hanya tersedia dalam bentuk dokumen digital non-terstruktur dapat dikonversi menjadi data teks yang dapat diolah secara komputasional [9]. Data historis dan data hasil ekstraksi ini kemudian dikombinasikan sebagai bahan dasar penyusunan dataset klasifikasi surat masuk. Data yang diperoleh terdiri dari atribut Tanggal Terima, Asal Surat, Tanggal Surat, Nomor Surat, Perihal, dan Kategori Bidang. Namun, dalam penelitian ini, proses klasifikasi difokuskan pada atribut Asal Surat dan Perihal yang merupakan data teks dan menjadi dasar untuk analisis serta atribut Kategori Bidang sebagai label atau kelas yang menjadi tujuan dari proses klasifikasi.

2.3. Pra-pemprosesan Data

Tabel Tahap Pra-pemprosesan data bertujuan untuk membersihkan data teks mentah guna memastikan kualitas data yang baik sebelum masuk ke tahap ekstraksi fitur. Proses ini dilakukan agar data menjadi lebih terstruktur dan siap untuk diproses lebih lanjut [10]. Langkah awal dilakukan dengan penanganan data yang tidak relevan, yaitu menghapus data kosong (*missing values*) dan data duplikat. Penanganan *missing values* penting untuk menjaga integritas data dan mencegah kesalahan dalam pemodelan [21], sedangkan penghapusan duplikat diperlukan agar model tidak menerima pola yang berulang yang dapat memunculkan bias dalam proses klasifikasi [19].

Selanjutnya, dilakukan normalisasi format teks melalui *case folding*. Tahap ini mengubah seluruh teks pada fitur 'asal surat' dan 'perihal' menjadi huruf kecil (*lowercase*) [11], [18]. Penyeragaman ini memastikan bahwa kata yang sama tetapi berbeda kapitalisasi diperlakukan sebagai satu fitur yang identik oleh model. Tahapan ini berperan signifikan dalam *text mining* karena membantu mengurangi jumlah fitur unik dalam data [18], serta merupakan prosedur umum yang digunakan dalam tahap persiapan data untuk klasifikasi teks [22].

2.4. Ekstraksi Fitur

Data yang telah dibersihkan diubah menjadi representasi vektor numerik agar dapat dipahami oleh model *machine learning* [18]. Tahap ini menerapkan metode *Term Frequency-Inverse Document Frequency (TF-IDF). TF-IDF* adalah teknik pembobotan statistik yang umum digunakan dalam text mining dan sistem temu kembali informasi untuk mengevaluasi seberapa penting suatu kata (*term*) dalam sebuah dokumen dalam konteks koleksi dokumen (korpus) [12]. Metode ini memberikan bobot tinggi pada kata-kata yang spesifik dan informatif, sekaligus mengurangi bobot kata-kata umum [13]. Bobot *TF-IDF* dihitung menggunakan persamaan (1) :

$$TF - IDF(t,d) = \left(\frac{f_{t,d}}{|d|}\right) \times \log\left(\frac{N}{df(t)}\right) \qquad (1)$$

DOI: https://doi.org/10.31004/riggs.v4i4.5044
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

Dengan:

- t : term/kata
- d : dokumen
- $f_{t,d}$: jumlah kemunculan kata *t* pada dokumen *d*
- $|d|$: total kata pada kolom *d*
- N : total dokumen (korpus)
- $df(t)$: jumlah dokumen yang mengandung kata *t*

Komponen pertama persamaan $\left(\frac{f_{t,d}}{|d|}\right)$ adalah Term Frequency (TF), yang mengukur frekuensi kemunculan sebuah kata dalam satu dokumen [11]. Komponen kedua, $\log\left(\frac{N}{df(t)}\right)$ adalah Inverse Document Frequency (IDF), yang mengukur keunikan kata tersebut di seluruh korpus [13],[14]. Matriks *TF-IDF* yang dihasilkan dari gabungan fitur 'asal surat' dan 'perihal' ini kemudian digunakan sebagai input numerik untuk melatih model klasifikasi *ensemble tree* (*Random Forest* dan *XGBoost*) [19].

2.5. Pembagian Data

Data yang telah direpresentasikan dalam bentuk vektor *TF-IDF* dibagi menjadi dua subset: data latih (*training data*) dan data uji (*testing data*) [11]. Data latih berfungsi untuk melatih model klasifikasi, sementara data uji digunakan untuk mengevaluasi kinerja model terhadap data baru yang belum pernah dilihat sebelumnya [19]. Untuk mendapatkan performa model yang paling optimal, penelitian ini melakukan perbandingan menggunakan empat skenario rasio data latih dan data uji yang berbeda, yaitu 70:30, 80:20, 85:15, dan 90:10. Kinerja dari keempat skenario rasio ini akan dievaluasi pada tahap pengujian model untuk menentukan pembagian data (*split data*) terbaik yang menghasilkan nilai evaluasi tertinggi.

2.6 Model Klasifikasi

Model klasifikasi yang digunakan dalam penelitian ini meliputi *XGBoost* dan *Random Forest*. Kedua model *ensemble* ini memiliki pendekatan yang berbeda dalam pembentukan pohon keputusan, sehingga perbandingan keduanya akan membantu menentukan metode yang lebih efektif untuk data yang digunakan.

XGBoost adalah algoritma *ensemble tree* (ansambel pohon) yang sangat efisien dan populer, yang didasarkan pada kerangka *Gradient Boosting Decision Tree* (GBDT) [16]. *XGBoost* bekerja dengan membangun pohon keputusan secara sekuensial (bertahap), di mana setiap pohon baru dilatih untuk mengoreksi kesalahan (residual) dari pohon-pohon sebelumnya [11]. Keunggulan utama *XGBoost* adalah kemampuannya mencegah *overfitting* melalui fungsi objektif yang ter-regularisasi [20]. Fungsi objektif (O) yang dioptimalkan oleh *XGBoost* didefinisikan pada persamaan (2):

$$O = \sum_{i=1}^{n} L\left(y_i, F(x_i)\right) + \sum_{k=1}^{t} R(f_k) + C \qquad (2)$$

Pada persamaan (2) fungsi $L\left(y_i, F(x_i)\right)$ berperan sebagai *loss function* yang digunakan untuk mengukur tingkat kesalahan prediksi yang dihasilkan model pada setiap data. Sementara itu, $R(f_k)$ merupakan fungsi regularisasi yang berfungsi mengendalikan kompleksitas model sehingga dapat meminimalkan potensi terjadinya *overfitting*. Untuk memperjelas definisi dari fungsi regularisasi $R(f_k)$ yang tercantum pada persamaan (2), bentuk persamaannya ditunjukkan pada persamaan (3) berikut.

$$R(f_k) = \alpha H + \frac{1}{2} n \sum_{j=1}^{H} \omega_j^2 \qquad (3)$$

Dengan:

- α : kompleksitas daun
- H : jumlah daun
- n : parameter penalti

DOI: https://doi.org/10.31004/riggs.v4i4.5044
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

- ω_j^2 : hasil output dari setiap simpul daun
- C : konstanta yang dapat dihilangkan secara efektif

Model kedua yang digunakan sebagai pembanding adalah *Random Forest*. Random Forest juga merupakan algoritma *ensemble tree* (ansambel pohon) yang populer [10], [11], [15], namun didasarkan pada kerangka *Bagging* (Bootstrap Aggregating). Berbeda dengan *XGBoost* yang membangun pohon secara sekuensial, *Random Forest* bekerja dengan membangun sejumlah besar pohon keputusan (*decision tree*) secara independen. Setiap pohon dilatih pada subset data acak yang berbeda (diambil melalui *bootstrap*) dan prediksi akhir ditentukan dari gabungan hasil seluruh pohon [10]. Untuk klasifikasi seperti dalam penelitian ini, prediksi *Random Forest* ditentukan oleh kelas dengan suara terbanyak (modus) dari semua pohon. Proses ini dipresentasikan pada persamaan (4):

$$\hat{y} = \arg\max_{c \in C} \sum_{i=1}^{T} 1(h_i(x) = c) \qquad (4)$$

Dengan

- $\hat{y}$: prediksi kelas (hasil *voting* terbanyak)
- $h_i(x)$: prediksi pohon ke-*i* untuk input *x*
- C : himpunan semua kelas
- T : jumlah pohon dalam hutan
- $1(\cdot)$: fungsi indikator (1 jika benar, 0 jika salah)

2.7 Evaluasi dan Kesimpulan

Evaluasi kinerja model klasifikasi (*Random Forest* dan *XGBoost*) dilakukan menggunakan empat indikator evaluasi yang umum digunakan dalam penelitian *machine learning* [10], [11], [15], [16], yaitu *Accuracy*, *Precision*, *Recall*, dan *F1-Score*. *Accuracy* digunakan untuk mengukur proporsi total prediksi yang benar dari keseluruhan data [11] ditunjukkan oleh persamaan (4). *Precision* mengukur tingkat ketepatan model dalam memprediksi kelas positif ditunjukkan oleh persamaan (5), sementara *Recall* mengukur kemampuan model untuk menemukan kembali semua sampel positif yang sebenarnya yang berhasil diprediksi benar dari seluruh kasus positif ditunjukkan oleh persamaan (6) [10], [15]. *F1-Score* digunakan sebagai indikator penyeimbang karena merupakan rata-rata harmonik dari *Precision* dan *Recall*, yang memberikan evaluasi yang lebih kuat, terutama jika distribusi kelas tidak merata ditunjukkan oleh persamaan (7) [15], [16]. Keempat indikator evaluasi ditampilkan pada persamaan berikut.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \qquad (4)$$

$$\text{Precision} = \frac{TP}{TP + FP} \qquad (5)$$

$$\text{Recall} = \frac{TP}{TP + FN} \qquad (6)$$

$$\text{F1-Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \qquad (7)$$

DOI: https://doi.org/10.31004/riggs.v4i4.5044
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

Dengan

- *TP* (*True Positive*): Jumlah data positif (kategori surat yang benar) yang diprediksi benar.
- *TN* (*True Negative*): Jumlah data negatif (kategori surat yang salah) yang diprediksi benar.
- *FP* (*False Positive*): Jumlah data negatif yang salah diprediksi sebagai positif.
- *FN* (*False Negative*): Jumlah data positif yang salah diprediksi sebagai negatif.

Kesimpulan dari penelitian ini akan diperoleh berdasarkan hasil evaluasi komparatif. Model (*XGBoost* atau *Random Forest*) dan rasio *split data* (70:30, 80:20, 85:15, atau 90:10) yang secara konsisten menghasilkan nilai *F1-Score* dan *Accuracy* tertinggi akan ditentukan sebagai pendekatan yang paling efektif untuk klasifikasi surat masuk pemerintah daerah.

3. Hasil dan Diskusi

3.1 Hasil

Proses pengujian dilakukan menggunakan data historis surat dan hasil ekstraksi dokumen surat yang dikombinasikan berjumlah 4.058, dan telah melalui tahap pra-pemprosesan data sehingga tersisa 3.998 data bersih. Selanjutnya, fitur dari data tersebut diekstraksi menggunakan TF-IDF. Evaluasi model dilakukan dengan membandingkan kinerja pada empat skenario rasio split data yang berbeda, yaitu 70:30, 80:20, 85:15, dan 90:10. Perbandingan antara model *XGBoost* dan *Random Forest* kemudian dilakukan berdasarkan konfigurasi internal dan hasil performa yang diperoleh. Tabel 1. menyajikan parameter internal yang digunakan untuk membangun kedua model, sehingga memberikan gambaran mengenai konfigurasi dan perbedaan arsitektur masing-masing metode *ensemble*.

Tabel 1. Parameter internal model

No	Parameter Hasil	*XGBoost*	*Random Forest*
1	Jumlah Pohon Terbentuk	1000	200
2	Fitur Dominan	keyword_density (0,13)	subject_lenght (21,99)
3	Rata-rata Gain per Fitur	2,218	-
4	Nilai Training Loss Akhir	0,243	-
5	Jumlah Fitur Kontributif	9	10

Model *XGBoost* dilatih dengan 1000 pohon (*n_estimators*), sedangkan *Random Forest* menggunakan 200 pohon. Perbedaan ini mencerminkan arsitektur *boosting* pada *XGBoost* yang membangun pohon secara sekuensial untuk mengoreksi kesalahan [11], [16]. Parameter 'Rata-rata Gain per Fitur' (2.218) dan 'Nilai Training Loss Akhir' (0.243) adalah parameter internal *XGBoost* yang menunjukkan seberapa besar kontribusi fitur dalam mengurangi *loss function* [21]. Parameter ini tidak berlaku untuk *Random Forest*, yang menggunakan mekanisme *bagging* dan *voting* [10], [15]. Tabel 3.1 juga menunjukkan perbedaan fitur yang dianggap paling dominan oleh kedua model.

Selanjutnya, penelitian ini menghasilkan perbandingan kinerja dua model klasifikasi *ensemble tree* (*XGBoost* dan *Random Forest*) untuk klasifikasi surat masuk pemerintah daerah. Kinerja model diukur menggunakan empat indikator evaluasi, yaitu *Accuracy, Precision, Recall,* dan *F1-Score*. Hasil perbandingan kinerja kedua model pada keempat rasio *split data* tersebut disajikan secara rinci pada Tabel 1. Model yang secara konsisten menghasilkan nilai *F1-Score* dan *Accuracy* tertinggi dianggap sebagai model dengan performa terbaik untuk klasifikasi.

DOI: https://doi.org/10.31004/riggs.v4i4.5044
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

Tabel 2. Hasil evaluasi model *XGBoost* dan *Random Forest*

		XGBoost				Random Forest (RF)			
No	Split Data	Accuracy XGBoost (%)	Precision XGBoost (%)	Recall XGBoost (%)	F1-Score XGBoost (%)	Accuracy RF (%)	Precision RF (%)	Recall RF (%)	F1-Score RF (%)
1	70 : 30	81,00	81,91	81,00	81,16	76,58	77,39	76,58	76,64
2	**80 : 20**	**81, 87**	**82,66**	**81,87**	**82,00**	**76,00**	**77,17**	**76,00**	**76,03**
3	85 : 15	81,83	82,55	81,83	81,94	75,83	76,67	75,83	75,66
4	90 : 10	82,25	82,77	82,25	82,31	75,25	76,18	75,25	75,26

Berdasarkan hasil pada Tabel 2, rasio 80:20 menunjukkan performansi paling optimal pada kedua algoritma. Pada rasio ini, *XGBoost* memperoleh *Accuracy* 81,87% dan *F1-Score* sebesar 82.00%, sedangkan *Random Forest* juga menunjukkan performa yang relatif lebih baik dibandingkan rasio lain dengan *Accuracy* 76,00% dan *F1-Score* sebesar 76.03%. Hal ini mengindikasikan bahwa rasio pembagian 80:20 memberikan keseimbangan yang lebih baik antara jumlah data pelatihan dan pengujian, serta mampu menghasilkan model yang lebih stabil pada kedua algoritma yang diuji.Selain itu, jika dibandingkan secara keseluruhan, algoritma *XGBoost* menunjukkan performansi yang lebih unggul dibandingkan *Random Forest*.

3.2 Diskusi

Berdasarkan hasil pada Tabel 2, rasio *split data* 80:20 dipilih sebagai rasio perbandingan yang paling optimal. Meskipun *XGBoost* mencapai *F1-Score* dan *Accuracy* puncaknya pada rasio 90:10 (*F1-Score* 82,31%; *Accuracy* 82,25%), rasio 80:20 (*F1-Score* 82,00%; *Accuracy* 81,87%) menunjukkan kinerja yang hampir identik sekaligus memberikan titik perbandingan yang lebih stabil dan adil untuk kedua model. Pada rasio 80:20 ini, model *XGBoost* (*accuracy* 81,87%, *F1-Score* 82,00%) menunjukkan kinerja yang lebih unggul secara signifikan dibandingkan *Random Forest* (*accuracy* 76,00%, *F1-Score* 76,03%).

Keunggulan *XGBoost* ini dapat diatribusikan pada arsitektur *boosting* yang diusungnya. Berbeda dengan *Random Forest* yang membangun pohon secara independen dan menggabungkan hasilnya melalui *voting* [10], *XGBoost* membangun model secara sekuensial (bertahap). Setiap pohon baru dilatih secara spesifik untuk mengoreksi kesalahan (residual) dari pohon-pohon sebelumnya [11], [16]. Pendekatan iteratif ini terbukti lebih efektif dalam menangani pola data tekstual berdimensi tinggi yang dihasilkan oleh *TF-IDF*.

Hingga saat ini, penelitian yang secara spesifik menerapkan *machine learning* untuk klasifikasi arsip *e-government* di Indonesia masih terbatas. Temuan ini menjadi sangat relevan jika dibandingkan dengan penelitian oleh Damaryanti dan Supriyanto [20] yang juga mengoptimasi sistem kearsipan Srikandi menggunakan *XGBoost* untuk membedakan arsip "Musnah" dan "Permanen". Menariknya, pada skenario rasio 80:20, *Accuracy XGBoost* dalam penelitian ini (81,87%) berhasil melampaui *Accuracy* (77%) yang dilaporkan dalam studi [20] tersebut.

Secara keseluruhan, dibandingkan dengan penelitian terdahulu [20], hasil ini memperkuat bukti bahwa arsitektur *gradient boosting* (*XGBoost*) memiliki kemampuan klasifikasi yang sangat andal untuk konteks data tekstual surat masuk pemerintah. Temuan ini memberikan dasar pertimbangan teknis yang kuat untuk implementasi model *XGBoost* dalam sistem *e-office* [8] atau Srikandi [20] guna meningkatkan efisiensi administrasi dan mendukung transformasi digital di lingkungan pemerintahan.

4. Kesimpulan

Penelitian ini bertujuan untuk membandingkan performa algoritma *Random Forest (RF)* dan *Extreme Gradient Boosting (XGBoost)* dalam melakukan klasifikasi surat masuk pemerintah daerah. Setelah menguji beberapa skenario rasio *split data*, rasio 80:20 dipilih sebagai titik perbandingan yang paling optimal dan adil untuk kedua model. Berdasarkan hasil pengujian pada rasio tersebut, model *XGBoost* menunjukkan performa yang lebih unggul secara signifikan (*Accuracy* 81,87%; *F1-Score* 82,00%) dibandingkan dengan *Random Forest* (*Accuracy* 76,00%;

DOI: https://doi.org/10.31004/riggs.v4i4.5044
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

F1-Score 76,03%). Hasil penelitian ini menunjukkan bahwa arsitektur *boosting* pada *XGBoost*, yang membangun model secara bertahap untuk memperbaiki kesalahan, lebih efektif dalam menangkap pola data tekstual (hasil *TF-IDF*) dibandingkan pendekatan *bagging* pada *Random Forest*. Ke depan, penelitian ini dapat dikembangkan dengan menggunakan *dataset* yang lebih besar untuk meningkatkan generalisasi model. Selain itu, dapat dilakukan eksplorasi teknik ekstraksi fitur yang lebih canggih, seperti model *transformer*, untuk melihat potensi peningkatan akurasi. Hasil penelitian ini dapat dijadikan dasar teknis bagi instansi pemerintah dalam memilih algoritma *machine learning* yang paling efisien untuk sistem klasifikasi surat otomatis, serta mendukung pengembangan aplikasi *e-office* dan *e-Government* yang lebih akurat dan andal.

Referensi

[1] D. Dwiyanto, "Dasar Hukum Bagi E-Goverment di Indonesia: Studi Pemetaan Hukum Pada Pemerintah Daerah," *J. Penelit. Huk.*, vol. 2, no. 8.5.2017, pp. 2003–2005, 2022.

[2] N. Irma, B. Ginting, Agusmidah, and J. Leviza, "Penerapan E-Government dalam Penyelenggaraan Pemerintahan di Kota Binjai," *Locus J. Acad. Lit. Rev.*, vol. 2, no. 6, pp. 454–466, 2023, [Online]. Available: https://jurnal.locusmedia.id/index.php/jalr/article/view/168

[3] C. P. Matau, S. Yohanes, and H. R. Udju, "Implementaasi E-Goverment Bagi Keterbukaan Informasi Publik Melalui Website Pemerintah Kabupaten Manggarai," *Petitum Law J.*, vol. 2, no. 1, pp. 177–188, 2025.

[4] K. P. Harmandini and K. M. L, "Analysis of TF-IDF and TF-RF Feature Extraction on Product Review Sentiment," *J. dan Penelit. Tek. Inform.*, vol. 8, no. 2, pp. 929–937, 2024, doi: 10.33395/sinkron.v8i2.13376.

[5] D. A. Suci *et al*., "Pengelolaan Administrasi Surat Menyurat Menggunakan Tata Naskah Dinas Berbasis Elektronik Di Balai Prasarana Permukiman Wilayah Gorontalo," *J. Ilmu Pemerintah. dan Adm. Publik*, vol. 2, pp. 1–9, 2024.

[6] M.- Azdiansyah and N. Chalik Azhar, "Pengembangan Sistem Informasi Surat Masuk dan Keluar Berbasis Web pada Instansi Pemerintah dengan Evaluasi UAT," *J. Manaj. Sist. Inf. dan Teknol.*, vol. 15, no. 1, p. 82, 2025, doi: 10.36448/expert.v15i1.4300.

[7] H. Hermawan, E. A. Muhtar, and Milwan, "Pengelolaan Administrasi Umum Pemerintahan Desa Dalam Perspektif Implementasi Kebijakan Publik di Desa Rian Dan Desa Kapuak," *J. Adm. Reform*, vol. 10, no. 1, pp. 52–69, 2022.

[8] A. Abidin and R. Hayati, "Implementasi Aplikasi Elektronik (E-Office) Dilihat Dari Aspek Sumber Daya Pada Sekretariat Daerah Kabupaten Tabalong," *J. Adm. Publik Adm. Bisnis*, vol. 7, pp. 330–346, 2024.

[9] F. Rizky and A. Surahmat, "Sistem Otomatis Ringkasan Laporan Keuangan Berbasis PDF Menggunakan Metode NLP Transformer," *J. Ilm. Tek. Inform.*, vol. 14, no. 164, pp. 238–244, 2025.

[10] K. Rahayu, V. Fitria, D. Septhya, and ..., "Klasifikasi Teks untuk Mendeteksi Depresi dan Kecemasan pada Pengguna Twitter Berbasis Machine Learning," *Indones. J. Mach. Learn. Comput. Sci.*, vol. 3, no. October, pp. 108–114, 2023, [Online]. Available: https://journal.irpi.or.id/index.php/malcom/article/view/780

[11] K. A. Wardana and A. M. A. Rahim, "Analisis Perbandingan Algoritma XGBoost Dan Algoritma Random Forest Untuk Klasifikasi Data Kesehatan Mental," *J. Ilmu Komput. dan Pendidik.*, vol. 2, no. 5, pp. 808–818, 2024, [Online]. Available: https://www.kaggle.com/datasets/bhavikjikadara/mental-health-dataset

[12] M. S. Maulana, Y. Anshori, R. Azhar, R. Laila, and N. T. Lapatta, "Implementasi Pembobotan Tf-Idf Pada Chatbot Telegram Untuk Sistem Layanan Informasi," *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 10, no. 3, pp. 1869–1877, 2025, doi: 10.29100/jipi.v10i3.6314.

[13] D. Septiani and I. Isabela, "Analisis Term Frequency–Inverse Document Frequency (TF-IDF) dalam Temu Kembali Informasi pada Dokumen Teks," *J. Sist. dan Teknol. Inf. Indones.*, vol. 1, no. 2, pp. 81–88, 2022.

[14] I. S. Wibowo, A. Witanti, and I. Susilawati, "Keyword Extraction Judul Berita Online Di Indonesia Menggunakan Metode TF-IDF," *J. Tek. Inform. dan Sist. Inf.*, vol. 11, no. 1, pp. 99–111, 2024, [Online]. Available: http://jurnal.mdp.ac.id

[15] H. L. Wati, N. Anggraeni, S. Kolbiah, U. Hendar, and N. Agustina, "Perbandingan Algoritma Random Forest Dan XGBoost Untuk Analisis Sentimen Publik Terhadap Rumah Makan Payakumbuah," *J. Ilm. Nas. Ris. Apl. dan Tek. Inform.*, vol. 07, no. 01, pp. 64–71, 2025.

[16] M. Erkamim, S. Suswadi, M. Z. Subarkah, and E. Widarti, "Komparasi Algoritme Random Forest dan XGBoosting dalam Klasifikasi Performa UMKM," *J. Sist. Inf. Bisnis*, vol. 13, no. 2, pp. 127–134, 2023, doi: 10.21456/vol13iss2pp127-134.

[17] R. Yulianti, E. Aghnia Ilmani, M. Z. Waliulu, B. Sartono, and A. R. Firdawanti, "Perbandingan Algoritma Random Forest dan XGBoost dalam Klasifikasi Penerima Bantuan Pangan Non-Tunai (BPNT) di Provinsi Jawa Barat," *J. Ilm. Pendidik. Mat. Mat. dan Sains*, vol. 6, no. 1, pp. 21–32, 2025, [Online]. Available: http://lebesgue.lppmbinabangsa.id/index.php/home

[18] B. Juarto and Yulianto, "Indonesian News Classification Using IndoBert," *Int. J. Intell. Syst. Appl. Eng.*, vol. 11, no. 2, pp. 454–460, 2023.

[19] I. Hendrawan Rifky, E. Utami, and A. Hartanto Dwi, "Analisis Perbandingan Metode Tf-Idf dan Word2vec pada Klasifikasi Teks Sentimen Masyarakat Terhadap Produk Lokal di Indonesia," *Smart Comp*, vol. 11, no. 3, pp. 497–503, 2022, doi: 10.30591/smartcomp.v11i3.3902.

[20] F. Damaryanti and Aji Supriyanto, "Optimization of the SRIKANDI E-Government System Using XGBoost-Based Classification and One-Class SVM Anomaly DetectionType," *Inf. Technol. Int. J.*, vol. 3, no. 1, pp. 29–43, 2025, doi: 10.33005/itij.v3i1.50.

[21] M. B. Prayogi, F. Apriani, and Nirma, "Prediksi Angka Harapan Hidup Menggunakan Random Forest dan XGBoost Regression," *J. Inform. dan Teknol.*, vol. 2, no. 1, pp. 112–121, 2025.

[22] D. Iskandar and A. Kurniawati, "Analisis Perbandingan Teknik Word2vec dan Doc2vec dalam Mengukur Kemiripan Dokumen Menggunakan Cosine Similarity," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 12, no. 1, pp. 133–144, 2025, doi: 10.25126/jtiik.2025129143.

DOI: https://doi.org/10.31004/riggs.v4i4.5044
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)