Journal of Digital Business and Technology Innovation (DBESTI)

https://journal.nurulfikri.ac.id/index.php/DBESTI
P-ISSN: 3047-1028 E-ISSN: 3032-775X

PENINGKATAN AKURASI DETEKSI BERITA PALSU MELALUI OPTIMASI PARAMETER MODEL RANDOM FOREST

Aulia Rohmatul Hasanah[1], Tifanny Nabarian[2]

[1,2] Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri
Jakarta Selatan, DKI Jakarta, Indonesia 12640
auli21244ti@student.nurulfikri.ac.id, nabarian@nurulfikri.ac.id

Abstract

This study aims to evaluate the effect of parameter optimization on the accuracy of the Random Forest algorithm in detecting fake news. The dataset used is secondary data obtained from Kaggle, consisting of over 23.000 text entries labeled as real or fake. This research applies a quantitative method by comparing the model's performance before and after parameter tuning. The text preprocessing stages include case folding, removal of special character, stopwords, and stemming using the Sastrawi library. Parameter optimization is performance using GridSearchCV on key hyperparameters n_estimators, max_depth, and min_samples_split. Model performance is evaluated using accuracy, precision, recall, and F1-score metrics. The results indicate that optimized model achieved an accuracy improvement from 78.5% to 82%, along with increased precision, recall, and F1-score values. These findings suggest that parameter tuning in Random Forest significantly enhances classification performance in the context of fake news detection. This research is expected to contributed to the development of more accurate and efficient automated fake news detection systems. Future studies are recommended to explore other optimization techniques, such as RandomizedSearchCV or Bayesian Optimization, and apply the same evaluation approach to different machine learning algorithms.

Keywords: *Fake news detection, GridSearchCV, Parameter optimization, Random Forest, Text classification.*

Abstrak

Penelitian ini bertujuan untuk mengevaluasi pengaruh optimasi parameter terhadap akurasi algoritma Random Forest dalam mendeteksi berita palsu. Data yang digunakan merupakan *dataset* sekunder dari *platform* Kaggle, berisi lebih dari 23.000 entri teks dengan label klasifikasi *real* dan *fake*. Metode yang digunakan dalam penelitian ini adalah kuantitatif dengan pendekatan perbandingan performa model sebelum dan sesudah *tuning* parameter. Proses praproses teks dilakukan melalui tahapan *case folding*, penghapusan karakter khusus dan *stopwords*, serta *stemming* menggunakan pustaka Sastrawi. Optimasi parameter dilakukan menggunakan teknik GridSearchCV terhadap parameter *n_estimators, max_depth*, dan *min_samples_split*. Evaluasi model dilakukan dengan menggunakan metrik akurasi, presisi, *recall*, dan skor F1. Hasil pengujian menunjukkan bahwa model yang telah dioptimasi mengalami peningkatan akurasi dari 78,5% menjadi 82%, serta peningkatan nilai presisi, *recall*, dan skor F1. Temuan ini mengindikasikan bahwa *tuning* parameter pada algoritma Random Forest mampu meningkatkan performa klasifikasi dalam konteks deteksi berita palsu. Penelitian ini diharapkan dapat menjadi dasar pengembangan sistem deteksi otomatis yang lebih akurat dan efisien. Penelitian selanjutnya disarankan untuk mengeksplorasi metode *tuning* lain seperti RandomizedSearchCV atau Bayesian Optimization serta menerapkan pendekatan yang sama pada algoritma pembelajaran mesin lainnya.

Kata kunci: Deteksi berita palsu, GridSearchCV, Klasifikasi teks, Optimasi parameter, Random Forest.

1. PENDAHULUAN

Penggunaan media sosial dan *platform* digital yang meningkatkan di Indonesia menyebabkan peningkatan penyebaran berita palsu atau hoaks yang menyebabkan masalah sosial dan politik [1]. Berita palsu dapat merugikan masyarakat, membingungkan orang, dan merusak kepercayaan mereka. Oleh karena itu, penelitian ini berkonsentrasi pada masalah klasifikasi berita palsu di bidang informatika [2]. Meskipun kemajuan teknologi digital telah memungkinkan penyebaran informasi secara masif dan cepat, hal ini juga memungkinkan penyebaran informasi yang menyesatkan, diperparah oleh tingkat literasi digital yang rendah dan tingginya keterpaparan terhadap media sosial tanpa sistem verifikasi fakta yang memadai.

Diterima 14 Juli 2025, Direvisi 24 November 2026, Diterima untuk publikasi 24 November 2026

Salah satu pendekatan yang digunakan untuk mendeteksi hoaks adalah pemanfaatan *machine learning*. Random Forest dikenal sebagai algoritma berbasis *ensemble learning* yang mampu memberikan akurasi tinggi dalam klasifikasi, namun performanya sangat bergantung pada pemilihan parameter yang optimal [3]. Oleh karena itu, fokus penelitian ini adalah untuk meningkatkan kinerja deteksi berita palsu Random Forest dengan mengoptimalkan parameternya. Namun, pemilihan dan pengaturan parameter optimal (*hyperparameter*), termasuk jumlah pohon (*n_estimators*), kedalaman pohon (*max_depth*), dan kriteria pemisahan (*criterion*) sangat memengaruhi kinerja algoritma [4].

Random Forest adalah salah satu algoritma pembelajaran mesin yang dianggap efektif karena mampu bekerja dengan baik pada data yang sangat besar dan tidak linear. Tantangan utamanya terletak pada bagaimana memilih konfigurasi parameter yang tepat agar model dapat bekerja dengan maksimal. Dengan demikian, pengaturan parameter atau pengaturan *hyperparameter* menjadi salah satu metode penting untuk meningkatkan kualitas klasifikasi model.

Machine learning adalah metode yang dapat digunakan untuk mendeteksi berita palsu secara otomatis. Karena keunggulannya dalam menangani data berukuran besar, mengurangi risiko *overfitting*, dan memberikan performa klasifikasi yang baik [5]. Oleh karena itu, pendekatan evaluasi yang lebih mendalam diperlukan untuk mengetahui seberapa besar peningkatan kinerja model yang dapat dicapai melalui optimasi parameter tersebut.

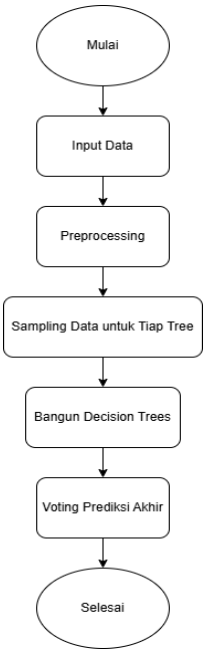
Tujuan dari penelitian ini adalah untuk mengetahui seberapa besar pengaruh optimasi parameter terhadap akurasi deteksi berita palsu menggunakan algoritma Random Forest. Dengan menggunakan metode *hyperparameter tuning* berbasis GridSearchCV, penelitian ini menguji seberapa baik model bekerja dengan parameter *default*. Penelitian ini hanya memanfaatkan algoritma Random Forest sebagai model klasifikasi dan menggunakan *dataset* yang diperoleh dari *platform* Kaggle. Proses optimasi hanya berfokus pada metode GridSearchCV dan penelitian ini tidak membahas pengembangan aplikasi atau penerapan sistem deteksi berita palsu secara langsung.

Studi Anosa Putri Ruise sebelumnya menunjukkan bahwa algoritma Support Vector Machine mencapai akurasi 98% dalam klasifikasi berita palsu [6], sementara studi oleh Fani Prasetya dan Ferdiansyah menggunakan Naïve Bayes untuk mendeteksi hoaks COVID-19 dengan akurasi 86,3% [7], dan studi Aziz Musthafa menggunakan Naïve Bayes untuk klasifikasi berita palsu pemilu dengan akurasi 91% [8]. Meskipun Random Forest dikenal memiliki performa yang kompetitif dan kemampuan untuk mengurangi *overfitting*, masih sedikit penentu yang menggunakannya.

Hasil penelitian terdahulu menunjukkan bahwa masing-masing algoritma memiliki tingkat kinerja yang berbeda tergantung pada konteks dan data yang digunakan. Namun, belum banyak penelitian yang secara khusus melakukan eksperimen optimasi parameter terhadap model Random Forest untuk deteksi berita palsu. Oleh karena itu, penelitian ini diharapkan dapat membantu mengembangkan sistem deteksi berita palsu berbasis *machine learning* yang lebih baik dan efisien melalui proses *tuning* parameter yang terarah.

2. METODE PENELITIAN

Penelitian ini menggunakan metode kuantitatif tujuan penelitian ini adalah untuk mengetahui bagaimana pengoptimalan parameter berdampak pada akurasi algoritma Random Forest untuk mendeteksi berita palsu. Desain penelitian berbasis eksperimen model, di mana kinerja model dengan parameter *default* dibandingkan dengan model hasil optimasi menggunakan GridSearchCV.



Gambar 1. *Flowchart* Random Forest

Random Forest Algoritma adalah metode pembelajaran mesin yang banyak digunakan karena sangat akurat, fleksibel, dan dapat digunakan untuk berbagai jenis data untuk tugas klasifikasi dan regresi [9]. Algoritma ini menggunakan teknik *bagging* yaitu pengambilan sampel acak dari data pelatihan untuk membangun berbagai pohon keputusan (*decision trees*). Setiap pohon membuat prediksi sendiri kemudian semua prediksi digabungkan untuk menghasilkan hasil akhir yang didasarkan pada mayoritas suara atau rata-rata prediksi. Prediksi akhir dari model Random Forest dirumuskan secara matematis sebagai berikut:

$$\boldsymbol{H(x) = \frac{1}{T} \sum_{t=1}^{T} h_t(x)} \quad (1)$$

Di mana $H(x)$ adalah prediksi akhir model, $h_t(x)$ prediksi dari pohon ke-t, dan T jumlah semua pohon dalam kelompok. Untuk meningkatkan akurasi dan stabilitas prediksi, rumus ini mencerminkan metode kolektif Random Forest.

Alur kerja algoritma Random Forest untuk mengklasifikasikan data, termasuk deteksi berita palsu digambarkan pada Gambar 1. Proses dimulai dengan input data dan prapemrosesan teks. Kemudian, beberapa pohon keputusan dilatih tentang kumpulan data yang diambil secara acak. Setiap pohon memberikan prediksi sendiri dan hasilnya digabung untuk menentukan kelas akhir.

2.1 Metode Pengumpulan Data, Instrumen Penelitian, dan Metode Pengujian

Data sekunder yang digunakan dalam penelitian ini berasal dari Kaggle yaitu *dataset* FakeNewsNet, yang berjumlah dari 23.197 fitur-fitur. *Dataset* ini diberi label *real* dan *fake* yang memiliki struktur teks panjang judul dan isi artikel serta label klasifikasi sebagai target.

Kompatibilitas dengan berbagai pustaka Python yang penting untuk pengembangan, pelatihan, dan evaluasi model deteksi berita palsu mendukung penelitian ini. Pandas adalah pustaka utama untuk manipulasi data seperti membaca *dataset*, membersihkan data, dan mengelola struktur data dalam bentuk *DataFrame* [10]. Pustaka Scikit-learn digunakan dalam proses pembangunan model pembelajaran mesin, termasuk Random Forest dengan metode GridSearchCV juga digunakan untuk pemisahan data, praproses fitur, dan optimasi *hyperparameter* [11]. NLTK (Natural Language Toolkit) digunakan untuk tokenisasi, penghapusan karakter tertentu, *case folding*, dan penghapusan *stopword* dalam pemrosesan teks [12]. Namun, pustaka Sastrawi digunakan secara khusus untuk melakukan *stemming* yang mengubah kata turunan menjadi bentuk dasar yang didasarkan pada aturan linguistik [13]. Untuk menampilkan data dan hasil evaluasi model digunakan pustaka Matplotlib dan Seaborn. Matplotlib untuk membuat grafik seperti grafik garis dan batang [14], sedangkan Seaborn digunakan untuk membuat visualisasi statistik yang lebih menarik dan kompleks, seperti *Confusion Matrix* sampai dapat memahami hasil penelitian ini [15].

Data diproses dalam lingkungan Google Colab yang memberikan akses GPU untuk meningkatkan efisiensi pemrosesan. Metode pengujian dengan membagi data menjadi 80% untuk pelatihan (*training*) dan 20% untuk pengujian (*testing*) untuk menguji model algoritma Random Forest untuk mengetahui seberapa baik model mendeteksi berita palsu, evaluasi kinerja dilakukan menggunakan metrik akurasi, presisi, *recall*, dan skor F1. Untuk mengevaluasi kemampuan model klasifikasi untuk mengklasifikasikan data dengan benar, terutama untuk klasifikasi biner seperti deteksi berita palsu. Akurasi dihitung dengan menggunakan rumus berikut untuk menghitung proporsi prediksi yang benar terhadap seluruh prediksi yang dibuat:

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{TP+TN}}{\boldsymbol{TP+TN+FP+FN}} \quad (2)$$

Di mana TP (*True Positive*) menunjukkan jumlah prediksi benar untuk kelas positif, TN (*True Negative*) menunjukkan jumlah prediksi benar untuk kelas negatif, FP (*False Positive*) menunjukkan kesalahan prediksi positif, dan FN (*False Negative*) menunjukkan kesalahan prediksi negatif. Untuk mengetahui seberapa akurat model memberikan prediksi positif. Presisi melakukan evaluasi dengan rumus berikut:

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}} \quad (3)$$

Nilai presisi yang tinggi menunjukkan bahwa sebagian besar prediksi positif memang benar. Kemampuan model untuk mengumpulkan seluruh data yang benar-benar termasuk kelas positif diukur dengan *recall* yang dirumuskan sebagai:

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} \quad (4)$$

Recall yang tinggi menunjukkan bahwa model menemukan sebagian besar data positif yang sebenarnya dengan sukses. Terakhir, skor F1 yaitu rata-rata harmonik dari *recall* dan presisi berguna ketika keduanya harus seimbang terutama pada data yang tidak seimbang. Rumus ini adalah:

$$\mathbf{F1-Score} = \mathbf{2} \times \frac{\boldsymbol{Precision \times Recall}}{\boldsymbol{Precision+Recall}} \quad (5)$$

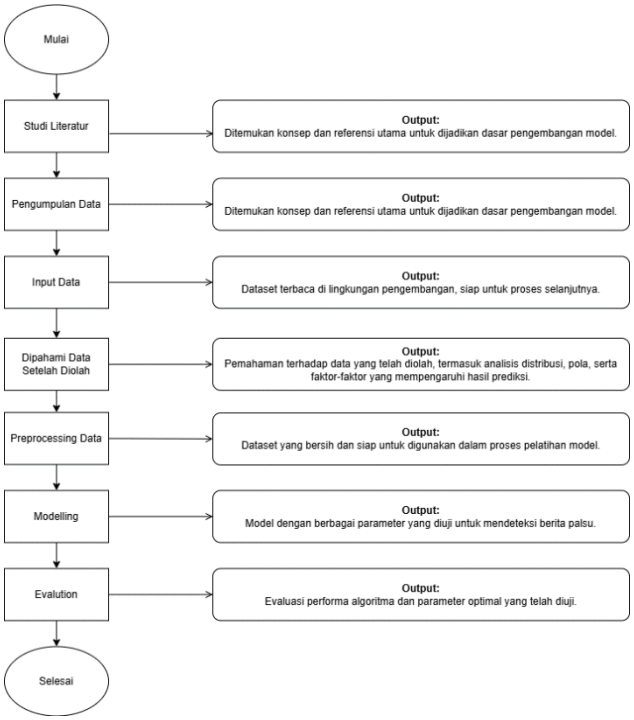
Tujuan penggunaan keempat metrik ini yaitu untuk memberikan evaluasi menyeluruh terhadap kemampuan model untuk secara efektif mengklasifikasikan berita palsu.

2.2 Tahapan Penelitian

Tahapan-tahapan dalam penelitian ini disusun secara sistematis pada Gambar 2 sebagai berikut:

1. Studi Literatur: dilakukan penelusuran terhadap jurnal, artikel ilmiah, dan sumber relevan yang membahas deteksi berita palsu, algoritma Random Forest serta metode optimasi parameter.
2. Pengumpulan *Dataset*: *Dataset* diunduh dari Kaggle dan disimpan dalam format CSV.
3. Input Data: *Dataset* diimpor dan dibaca ke dalam lingkungan Google Colab menggunakan pustaka Python seperti Pandas.
4. Pemahaman Data Setelah Diolah: Analisis awal dilakukan untuk memahami distribusi label, struktur data, serta pola-pola yang muncul melalui visualisasi dan deskripsi statistik.
5. *Preprocessing* Data: Dilakukan proses pembersihan data seperti *case folding*, penghapusan *stopwords*, *tokenization*, dan transformasi teks menjadi numerik menggunakan TF-IDF.
6. *Modeling*: Pembangunan model dilakukan menggunakan algoritma Random Forest. Model dilatih menggunakan data *training* dengan parameter *default* dan parameter hasil optimasi melalui GridSearchCV.
7. *Evaluation*: Model yang telah dilatih kemudian dievaluasi menggunakan data *testing*. Metrik yang

digunakan meliputi akurasi, presisi, *recall*, dan skor F1 untuk menilai performa model.



Gambar 2. Tahapan Penelitian

3. HASIL DAN PEMBAHASAN

Penelitian ini memeriksa bagaimana pengoptimalan parameter algoritma Random Forest meningkatkan akurasi deteksi berita palsu. Ini dilakukan dengan membandingkan kinerja model Random Forest sebelum dan sesudah optimasi dengan GridSearchCV.

3.1 Hasil Evaluasi Model

Untuk meningkatkan kualitas input yang akan digunakan dalam proses klasifikasi teks di kolom judul dibersihkan selama tahap prapemrosesan data. *Preprocessing* meliputi normalisasi huruf menjadi huruf kecil (*lowercasing*), penghapusan *stopwords* dan pustaka NLTK, dan proses *stemming* menggunakan *library* Sastrawi yang mengubah kata menjadi bentuk dasarnya.

	title	title_cleaned
0	Kandi Burruss Explodes Over Rape Accusation on...	kandi burruss explodes over rape accusation on...
1	People's Choice Awards 2018: The best red carp...	peoples choice awards the best red carpet looks
2	Sophia Bush Sends Sweet Birthday Message to 'O...	sophia bush sends sweet birthday message to on...
3	Colombian singer Maluma sparks rumours of inap...	colombian singer maluma sparks rumours of inap...
4	Gossip Girl 10 Years Later: How Upper East Sid...	gossip girl years later how upper east siders ...

Gambar 3. Hasil *Preprocessing*

Hasil akhir dari semua langkah tersebut pada Gambar 3, teks yang lebih bersih dan seragam disimpan dalam kolom baru yaitu *title_cleaned*. Prosedur ini dilaksanakan dengan menggunakan fungsi *clean_text* dan bilah kemajuan *tqdm* digunakan untuk melacak kemajuan prapemrosesan di seluruh data.

Ketika data dibagi menjadi set pelatihan dan pengujian, *sampling stratified* digunakan untuk memastikan bahwa persentase distribusi label asli tetap konstan antara kedua set data yang membuat hasil evaluasi model lebih adil dan representatif. Tanpa *sampling stratified*, model mungkin dilatih pada data yang secara tidak akurat menunjukkan distribusi membuat metode ini sangat penting untuk *dataset* yang menunjukkan ketidakseimbangan kelas pada Gambar 4 di bawah.

```
[ ] #Pembagian dataset menjadi set pelatihan dan pengujian
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.8, stratify=y, random_state=42)

    print("Ukuran set pelatihan:", X_train.shape)
    print("Ukuran set pengujian:", X_test.shape)

→ Ukuran set pelatihan: (4573, 1)
  Ukuran set pengujian: (18293, 1)
```

Gambar 4. *Stratified Sampling*

Proses optimasi parameter (*hyperparameter tuning*) pada algoritma Random Forest menggunakan GridSearchCV yang ada di Gambar 5 berikut.

```
#Grid Search untuk Random Forest
param_grid = {
    'n_estimators': [50, 100, 200],
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 5, 10]
}

grid_search_rf = GridSearchCV(estimator=rf_model, param_grid=param_grid, cv=5, scoring='f1', n_jobs=-1, verbose=2)
grid_search_rf.fit(X_train, y_train)

print("Hasil Grid Search Random Forest:")
print(grid_search_rf.best_params_)

→ Fitting 5 folds for each of 36 candidates, totalling 180 fits
  Hasil Grid Search Random Forest:
  {'max_depth': 10, 'min_samples_split': 10, 'n_estimators': 200}
```

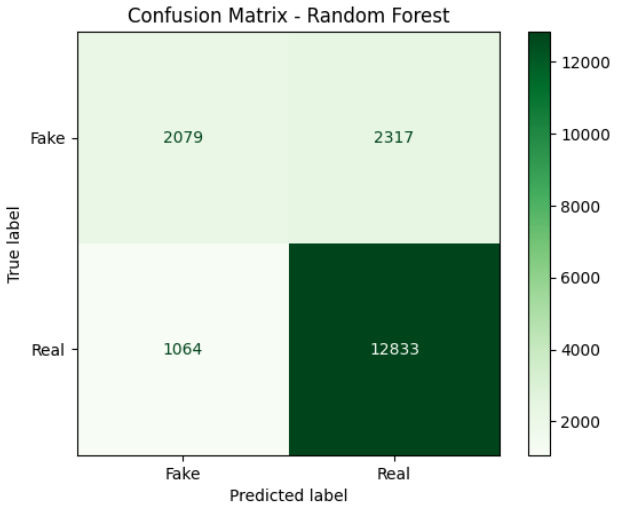
Gambar 5. GridSearchCV

Dalam proses ini, kombinasi parameter '*n_estimators'*, dan *'max_depth'*, dan *'min_samples_split'* dinilai melalui *cross validation* lima kali menggunakan skor metrik F1. Hasil konfigurasi terbaik yaitu *'n_estimators*=200', '*max_depth*=10', dan '*min_samples*=10' yang digunakan untuk meningkatkan performa model dalam mendeteksi berita palsu. Setelah proses pelatihan dan pengujian selesai, evaluasi performa model dengan parameter hasil optimasi dan parameter *default* dikumpulkan. Untuk menilai kemampuan model untuk membedakan antara berita asli dan palsu, metrik akurasi, presisi, *recall*, dan skor F1 digunakan.

Tabel 1. Hasil Evaluasi Kinerja Model Random Forest

Parameter	Akurasi	Presisi	*Recall*	Skor F1
Default (Sebelum *Tuning*)	78,50%	79%	78%	78,5%
Optimal (Sesudah *Tuning*)	82,00%	83%	81%	82%

Tabel 1 menunjukkan peningkatan akurasi model sebesar 3,5% setelah melakukan optimasi parameter menggunakan GridSearchCV. Selain itu, nilai presisi, *recall*, dan skor F1 juga meningkat. Ini menunjukkan bahwa kinerja algoritma untuk mengklasifikasikan berita palsu dapat secara signifikan ditingkatkan dengan mengatur parameternya.



Gambar 6. *Confusion Matrix* Random Forest

Confusion Matrix Random Forest (Gambar 6) menunjukkan hasil klasifikasi yang paling seimbang dan ideal. 2.317 sampel secara keliru diidentifikasikan sebagai "*Real*" (*False Positive*) menggunakan data kelas "*Fake*", sedangkan 2.079 sampel diklasifikasikan dengan benar (*True Negative*) untuk data kelas "*Real*", sedangkan hanya 1.064 sampel yang salah diklasifikasikan sebagai "*Fake*" (*False Negative*). Grafik ini mendukung kesimpulan bahwa Random Forest adalah model yang paling akurat dan memiliki skor F1 tinggi dalam mengidentifikasi berita palsu dari kumpulan data karena jumlah kesalahan prediksi yang rendah dan keseimbangan tinggi antara positif dan negatif nyata.

3.2 Pembahasan

Hasil evaluasi menunjukkan bahwa metrik evaluasi model secara keseluruhan dapat ditingkatkan dengan mengoptimalkan parameter Random Forest. Hal ini sesuai dengan teori dan penelitian sebelumnya yang menyatakan bahwa pengaturan *hyperparameter* dapat mengurangi *overfitting*, mengubah kompleksitas model, dan mengarahkan proses pelatihan ke solusi klasifikasi yang lebih baik. Peningkatan skor F1 menjadi sinyal penting bahwa model akurat dan mampu mengimbangi *recall* dan presisi.

Selain itu, penelitian ini menunjukkan bahwa Random Forest adalah algoritma yang kuat untuk klasifikasi teks, tetapi kinerjanya sangat bergantung pada pengaturan parameter yang tepat. Oleh karena itu, proses penyesuaian tidak hanya merupakan proses tambahan, tetapi sangat penting untuk penerapan model pembelajaran mesin untuk mendeteksi berita palsu.

4. KESIMPULAN

Studi ini menunjukkan bahwa peningkatan akurasi deteksi berita palsu dipengaruhi secara signifikan oleh proses optimasi parameter pada algoritma Random Forest. Hasil pengujian model sebelum dan sesudah penyesuaian menunjukkan peningkatan nilai akurasi sebesar 3,5% serta peningkatan metrik presisi, *recall*, dan skor F1. Hal ini menunjukkan bahwa terutama dalam klasifikasi teks konfigurasi parameter yang tepat dapat meningkatkan kinerja model secara keseluruhan.

Hasilnya juga menunjukkan bahwa penyesuaian *hyperparameter* GridSearchCV adalah komponen penting dalam membangun model pembelajaran mesin yang ideal, bukan hanya sebagai pelengkap. Random Forest terbukti menjadi algoritma yang andal untuk meminimalkan kesalahan klasifikasi dan menjaga keseimbangan antara *recall* dan presisi karena memiliki konfigurasi terbaik untuk nilai *n_estimators, max_depth*, dan *min_samples_split*.

Penelitian ini dapat diperluas untuk pengembangan lebih lanjut dengan membandingkan metode optimasi lainnya, seperti RandomizedSearchCV atau Bayesian Optimization, dan menggunakan *dataset* multibahasa untuk menerapkan model lebih luas. Selain itu, metode yang sama dapat digunakan untuk melakukan pengujian pada algoritma lain untuk mengetahui seberapa efektif pengaturan parameter pada berbagai algoritma untuk mendeteksi berita palsu pada sistem nyata atau alat verifikasi untuk menghentikan penyebaran hoaks.

Ucapan Terima Kasih

Penulis A.R.H. menyampaikan penghargaan dan terima kasih yang sebesar-besarnya kepada Ketua Sekolah Tinggi Teknologi Terpadu Nurul Fikri, Ketua Program Studi Teknik Informatika, Dosen Pembimbing Akademik, Dosen Pembimbing Tugas Akhir serta Dosen Penguji atas arahan, waktu dan masukan yang sangat berharga selama proses perkuliahan hingga penyusunan penelitian ini. Penulis juga mengucapkan terima kasih kepada seluruh jajaran dosen dan staf di Sekolah Tinggi Teknologi Terpadu Nurul Fikri yang memberikan ilmu, dukungan, dan fasilitas yang mendukung kelancaran proses studi. Tidak lupa, penulis juga berterima kasih kepada rekan-rekan mahasiswa Teknik Informatika atas semangat, kolaborasi, dan kebersamaan yang telah menjadi bagian penting dalam perjalanan akademik ini.

DAFTAR PUSTAKA

[1] R. D. Setyadin, R. H. Winasis, and G. Triyono, "Deteksi Berita Palsu menggunakan Algoritma Random Forest," Jakarta, Mar. 2025. [Online]. Available: http://sistemasi.ftik.unisi.ac.id

[2] S. Nurohanisah, R. Astuti, and F. M. Basysyar, "DETEKSI BERITA PALSU MENGGUNAKAN ALGORITMA RANDOM FOREST," Cirebon, Feb. 2024.

[3] N. G. Ramadhan, F. D. Adhinata, A. J. T. Segara, and D. P. Rakhmadani, "Deteksi Berita Palsu Menggunakan Metode Random Forest dan Logistic Regression," *JURIKOM (Jurnal Riset Komputer)*, vol. 9, no. 2, p. 251, Apr. 2022, doi: 10.30865/jurikom.v9i2.3979.

[4] I. Adriansyah, M. D. Mahendra, E. Rasywir, and Y. Pratama, "Perbandingan Metode Random Forest Classifier dan SVM Pada Klasifikasi Kemampuan Level Beradaptasi Pembelajaran Jarak Jauh Siswa," *Bulletin of Informatics and Data Science*, vol. 1, no. 2, 2022, [Online]. Available: https://ejurnal.pdsi.or.id/index.php/bids/index

[5] E. P. Febtiawan, L. A. S. I. Akbar, and A. S. Rachman, "Forecasting Produksi Energi Photovoltaic Menggunakan Algoritma Random Forest Classification," *Journal of Information System Research (JOSH)*, vol. 5, no. 4, pp. 1053–1062, Jul. 2024, doi: 10.47065/josh.v5i4.5514.

[6] A. P. Ruise, A. S. Mashuri, M. Sulaiman, and F. Rahman, "Studi Komparasi Metode Svm, Logistic Regresion Dan Random Forest Clasifier Untuk Mengklasifikasi Fake News di Twitter," *J I M P - Jurnal Informatika Merdeka Pasuruan*, vol. 7, no. 2, p. 64, Sep. 2023, doi: 10.51213/jimp.v7i2.472.

[7] F. Prasetya and F. Ferdiansyah, "Analisis Data Mining Klasifikasi Berita Hoax COVID 19 Menggunakan Algoritma Naive Bayes," *Jurnal Sistem Komputer dan Informatika (JSON)*, vol. 4, no. 1, p. 132, Sep. 2022, doi: 10.30865/json.v4i1.4852.

[8] A. Musthafa, D. Muriyatmoko, S. Kamal Sholihin, P. Studi Teknik Informatika, F. Sains dan Teknologi, and U. Darussalam Gontor, "Deteksi Berita Salah Pada Pemilihan Umum Presiden 2024 Menggunakan Metode Naïve Bayes Berbasis Website".

[9] A. T. S. Al Azhima, D. Darmawan, N. F. Arief Hakim, I. Kustiawan, M. Al Qibtiya, and S. N. Syafei, "Hybrid Machine Learning Model Untuk Memprediksi Penyakit Jantung Pada Sistem Informasi Berbasis Dekstop," Jun. 2022.

[10] S. Mubarak, "Analisis Sentimen Review Aplikasi Mobile Legends," Jakarta, Jun. 2024.

[11] M. N. Fahmi, "Implementasi Machine Learning menggunakan Python Library: Scikit-Learn (Supervised dan Unsupervised Learning)," *Sains Data Jurnal Studi Matematika dan Teknologi*, vol. 1, no. 2, pp. 87–96, Dec. 2023, doi: 10.52620/sainsdata.v1i2.31.

[12] F. Wijaya and Muhammad, "Analisis Sentimen Hasil Crawling Pada Media Online Menggunakan Elasticsearch dan Algoritma Naive Bayes," *Repository STT Terpadu Nurul Fikri*, p. 15, Feb. 2020.

[13] R. Kaestria, F. E. Himmah, and R. Irawan, "Penerapan Matplotlib dalam Visualisasi Data untuk Analisis Hubungan Penggunaan Gadget dan Hasil Belajar," *Journal of Digital Business and Information Technology*, vol. 1, no. 1, pp. 29–39, Jun. 2024, doi: 10.23971/jobit.v1i1.204.

[14] K. Hermanto, D. Salim, B. Wu, R. O. Salim, and B. R. Gunadi, "Penggunaan Python Untuk Menganalisis Pola Penyebaran Covid-19 Di Masa Pandemi," *Journal of Student Development Information System (JoSDIS)*, vol. Volume 2 Nomor 2, Jul. 2022.

[15] Z. Putrama, "Implementasi Metode K-Medoid Dalam Penentuan Cluster Daerah Berdasarkan Tingkat Pendapatan Pajak di Kabupaten Pasaman Barat," Jakarta, Feb. 2024.