

Deteksi Berita Palsu menggunakan Algoritma Random Forest

Fake News Detection using the Random Forest Algorithm

¹Rahmat Dipo Setyadin, ²Reza Handaru Winasis*, ³Gandung Triyono

^{1,2,3}Sistem Informasi, Magister Ilmu Komputer, Universitas Budi Luhur
Jakarta, Indonesia

*e-mail: ¹2311600965@student.budiluhur.ac.id,

²2311600213@student.budiluhur.ac.id, ³gandung.triyono@budiluhur.ac.id

(received: 20 January 2025, revised: 14 March 2025, accepted: 14 March 2025)

Abstrak

Deteksi berita hoaks menjadi semakin penting di era digital, di mana informasi yang tidak benar dapat menyebar dengan cepat dan mempengaruhi opini publik secara luas. Penyebaran berita hoaks dapat menyebabkan ketidakpercayaan terhadap media, kerugian ekonomi, dan bahkan konflik sosial. Penelitian ini bertujuan untuk mengembangkan sistem deteksi berita hoaks yang efektif menggunakan pendekatan algoritma Random Forest. Dataset yang digunakan dalam penelitian ini terdiri dari berita yang dikumpulkan dari website Kominfo, yang mencakup atribut seperti judul, deskripsi, penulis, tanggal, kategori, halaman, alamat url berita dan alamat url gambar. Proses preprocessing teks meliputi tokenisasi, penghapusan kata umum (stop words), normalisasi teks, dan ekstraksi fitur menggunakan Term Frequency dan Inverse Document Frequency untuk menghasilkan representasi numerik dari teks. Model Random Forest. Evaluasi kinerja model dilakukan menggunakan metrik akurasi, precision, recall, dan F1-score untuk menilai efektivitas sistem dalam mendeteksi berita hoaks. Hasil penelitian menunjukkan evaluasi model menggunakan k-fold cross-validation dengan k=5 menghasilkan hasil yang sangat baik, dengan nilai rata-rata akurasi yang tinggi untuk masing-masing model, yaitu Random Forest (0.9890).

Kata kunci: deteksi berita hoaks, random forest, TF-IDF, k-fold cross validation

Abstract

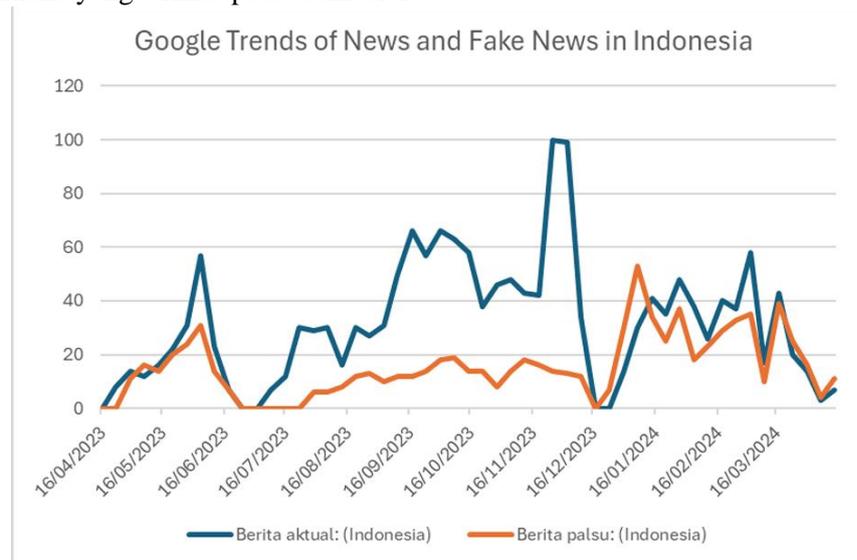
Detecting fake news has become increasingly important in the digital era, where false information can spread rapidly and significantly influence public opinion. The dissemination of fake news can lead to public distrust in the media, economic losses, and even social conflict. This study aims to develop an effective fake news detection system using the Random Forest algorithm approach. The dataset used in this research was collected from the official Kominfo website and includes attributes such as title, description, author, date, category, page, news URL, and image URL. The text preprocessing process involves tokenization, stop word removal, text normalization, and feature extraction using Term Frequency and Inverse Document Frequency (TF-IDF) to generate numerical representations of the textual data. The Random Forest model was evaluated using accuracy, precision, recall, and F1-score metrics to assess its effectiveness in detecting fake news. The results show that the model performed exceptionally well, with k-fold cross-validation (k=5) yielding high average accuracy—Random Forest achieved an accuracy of 0.9890.

Keywords: fake news detection, random forest, TF-IDF, k-fold cross validation

1 Pendahuluan

Pada era digital saat ini, penyebaran informasi terjadi dengan sangat cepat melalui berbagai platform seperti media sosial, situs berita, dan aplikasi pesan instan. Kemajuan teknologi telah memungkinkan penyebaran berita dalam hitungan detik ke berbagai penjuru dunia, namun hal ini juga meningkatkan risiko penyebaran informasi yang tidak akurat atau menyesatkan. Salah satu tantangan besar yang dihadapi saat ini adalah maraknya berita palsu atau hoaks. Berita hoaks adalah informasi yang disampaikan dengan tidak akurat atau bohong. Dalam satu tahun terakhir, istilah "fake news" sering digunakan sebagai label untuk publikasi atau pernyataan yang mengandung fakta yang salah atau sangat distorsi dengan tujuan memanipulasi opini atau persepsi orang [1]. Informasi ini dengan

cepat menyebar melalui jaringan sosial, di mana pengguna sering tidak meneliti kebenaran dari artikel tersebut, sebagaimana yang terlihat pada Gambar 1.



Gambar 1. Trend fake news di Indonesia [2]

Penyebaran berita hoaks sering terjadi tanpa verifikasi fakta oleh pengguna, memicu ketegangan sosial dan mengganggu kepercayaan publik terhadap media yang sah. Dengan berkembangnya platform media yang memungkinkan berbagi informasi dengan cepat, kebutuhan akan metode deteksi yang efisien menjadi semakin mendesak. Dalam beberapa tahun terakhir, berbagai platform media memungkinkan pengumpulan informasi dengan mudah dan berbagi informasi dengan cepat [3]. Ketersediaan data tidak autentik di platform media telah menarik perhatian luas di kalangan peneliti dan menjadi tempat penyebaran berita hoaks. Oleh karena itu, penyebaran berita hoaks menjadi masalah yang mengganggu secara global, dan banyak negara mulai mengkriminalisasi penciptaan dan distribusi informasi hoaks secara *online* [4]. Masyarakat kini lebih sering mencari berita melalui media sosial. Masyarakat perlu menggali lebih banyak informasi dan data untuk meningkatkan keakuratan deteksi. Selain itu, media sosial semakin berorientasi pada platform utama [5].

Metode pembelajaran mesin memungkinkan deteksi berita hoaks secara otomatis dengan menganalisis pola dan karakteristik tertentu dalam teks berita. Namun, metode tradisional yang telah ada sebelumnya memiliki beberapa kekurangan, seperti keterbatasan waktu dan sumber daya yang dibutuhkan untuk verifikasi secara manual. Hal ini mendorong perlunya pengembangan metode yang lebih efektif dan efisien, seperti algoritma pembelajaran mesin yang mampu mengatasi keterbatasan tersebut. Dengan pendekatan ini, diharapkan dapat diatasi beberapa kelemahan dari metode tradisional dalam mendeteksi berita hoaks, serta mempercepat proses verifikasi yang biasanya memakan waktu lama.

Penelitian sebelumnya membahas tentang klasifikasi *fake news* menggunakan *random forest* dan *decision tree* [6]. Penelitian ini mencoba untuk meningkatkan hasil akurasi klasifikasi berita hoaks menggunakan ekstraksi fitur TF-IDF, mengekstrak kata-kata penting dari artikel berita hoaks menggunakan dua pengklasifikasi berbeda (*random forest* dan *decision tree*), dan meningkatkan hasil akurasi.

Penelitian lainnya membahas tentang deteksi *Fake News* di *Arabic Corpus* menggunakan *Hunter Prey Optimization* dengan *Hybrid Deep Learning* [7]. Penelitian ini mengembangkan optimasi hunter-prey baru dengan model deteksi berita hoaks berbasis *hybrid deep learning* (HPOHDL-FND) pada korpus bahasa Arab. Teknik HPOHDL-FND yang disajikan bertujuan untuk mengenali dan mengklasifikasikan berita hoaks secara efektif pada korpus bahasa Arab.

2 Tinjauan Literatur

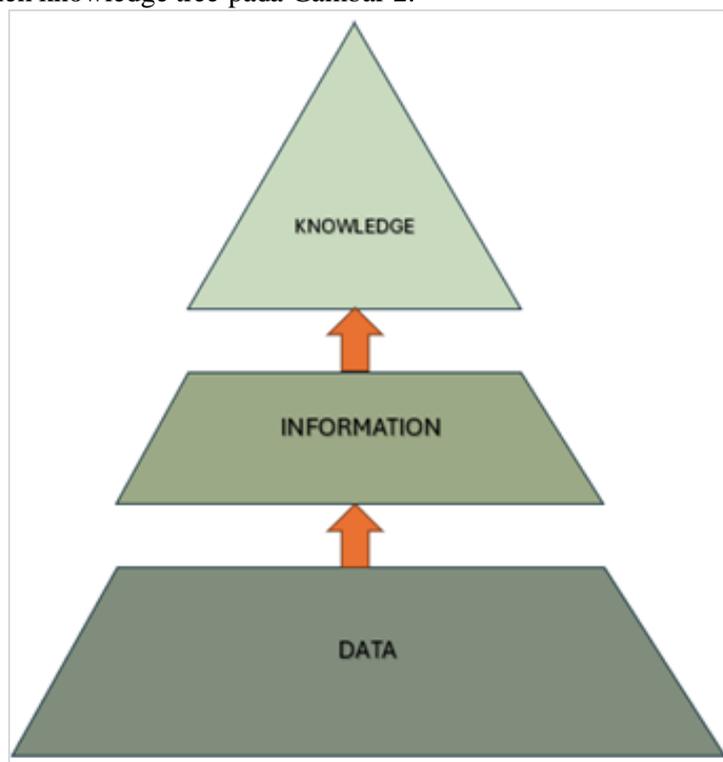
Dalam penelitian ini diambil beberapa rujukan penelitian terdahulu sebagai referensi dalam penulisan, berikut ini adalah beberapa penelitian terdahulu yang terkait topik yang dibahas pada penelitian ini. Dalam penelitian yang berjudul "Klasifikasi Berita Palsu Menggunakan Random Forest

dan Decision Tree (J48)" oleh peneliti [6], berfokus pada fenomena berkembangnya berita palsu, khususnya di ranah politik, yang dipicu oleh penggunaan luas internet dan media sosial. Penelitian ini menggunakan data dari berbagai sumber seperti halaman web agensi berita, Twitter, Facebook, dan Instagram, dengan tujuan utama untuk mendeteksi berita palsu menggunakan algoritma Random Forest dan Decision Tree (J48).

Sementara itu, penelitian oleh [7] dengan judul "Optimisasi Hunter Prey dengan Pembelajaran Mendalam Hibrida untuk Deteksi Berita Palsu dalam Korpus Arab" mengatasi masalah penyebaran berita palsu yang mempengaruhi masyarakat berbahasa Arab, menggunakan data dari Covid19Fakes dan dataset satiris. Tujuan dari studi ini adalah untuk mengembangkan model yang efektif dalam mendeteksi dan mengklasifikasikan berita palsu, dengan menggunakan teknik pembelajaran mendalam dan optimisasi Hunter Prey, dikombinasikan dengan model Long Short-Term Memory Recurrent Neural Network (LSTM-RNN).

2.1 Data Mining

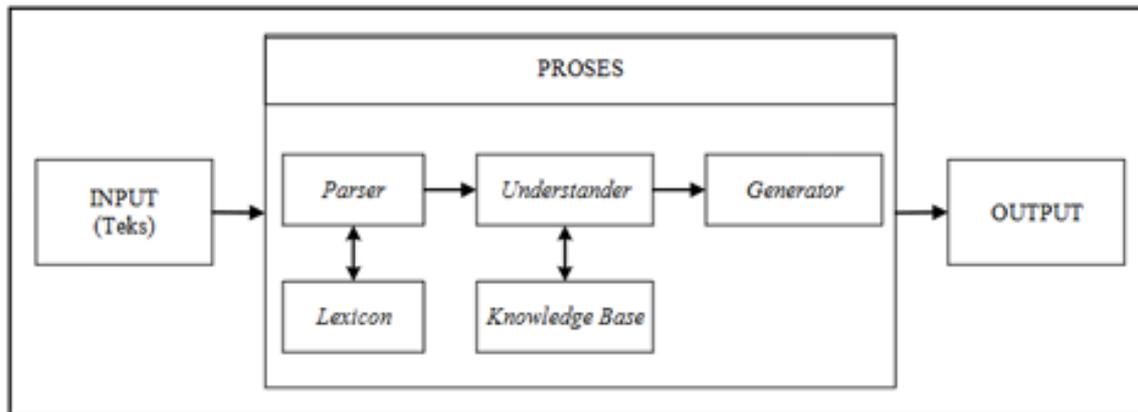
Data mining adalah suatu proses menemukan korelasi, pola, dan tren baru yang bermakna dengan memilah-milah data dalam jumlah besar yang tersimpan dalam repositori, menggunakan teknologi pengenalan pola serta teknik statistik dan matematika [8]. Dengan demikian, data mining dapat didefinisikan sebagai proses menganalisis data dalam jumlah besar untuk mengidentifikasi pola dan pengetahuan untuk memecahkan masalah melalui analisis data. Sebagai hasilnya, data mining telah digunakan di berbagai bidang termasuk perawatan kesehatan, bisnis, pendidikan, manufaktur, dan lain-lain. Data mining digunakan untuk mengekstrak informasi ini dari data yang diberikan seperti yang ditunjukkan oleh knowledge tree pada Gambar 2.



Gambar 2. Knowledge tree data mining [8]

2.2 Natural Language Processing

Bahasa adalah mode komunikasi utama pertama dari sistem cerdas manusia yang NLP memainkan peran banyak aspek bidang yang terkait dengan perhitungan properti. NLP adalah cara untuk mengeksplorasi penelitian dan aplikasi yang menggambarkan cara menggunakan sistem (terutama komputer) untuk memahami, mengidentifikasi dan mengoperasikan bahasa alami [9]. Di era informasi, Natural Language Processing (NLP) menciptakan permintaan di berbagai bidang aplikasi. Pemrosesan bahasa alami telah direncanakan sebagai area kerja sejak 1950-an untuk memberikan pengetahuan penting kepada non-programmer sistem komputer [10].



Gambar 3. Elemen utama dalam neural language processing [11]

Pada Gambar 3 yaitu elemen utama dalam natural language processing, adapun elemen detailnya akan dijelaskan dibawah ini [11]:

2.2.1. Parser

Parser adalah bagian dari program yang mem-parsing input. Setiap kata dan bagian pidatonya diidentifikasi. Ada tiga jenis parser: parser state machine, Parser Context-Free Recursive-Descent, dan Parser Noise-Disposal. Parser State Machine menggunakan keadaan kalimat yang sebenarnya untuk memprediksi jenis kata mana yang akan cocok. Parser Context-Free Recursive-Descent menggunakan aturan produksi untuk menganalisa sebuah kalimat. Parser Noise-Disposal merupakan jenis parser yang sangat sering digunakan dalam aplikasi berbasis basis data.

2.2.2. Lexicon

Parser bekerja dengan lexicon untuk menganalisis sintaks. Lexicon berisi semua kata yang dikenali oleh program. Lexicon juga dapat membentuk bagian audio, termasuk ejaan yang benar dari setiap kata. Dalam implementasi itu, parser adalah alat pencocokan pola. Setelah kata-kata diidentifikasi, parser melacak lexicon dan membandingkan setiap kata yang dimasukkan dengan semua kata yang disimpan dalam basis data.

2.2.3. Understander dan Knowledge Base

Understander bekerja sama dengan knowledge base untuk menentukan makna kalimat. Tugas understander adalah untuk merakit struktur data yang bermakna, memahami kalimat yang dimasukkan, dan menyimpannya dalam memori. Untuk menentukan apa arti kalimat yang dimasukkan, sistem perlu mengetahui kata-kata dan cara menyimpannya untuk membuat pernyataan yang bermakna. Tujuan dari understander adalah menggunakan output dari parser untuk mereferensikan knowledge base. Jika kalimat yang dimasukkan adalah pernyataan, understander melihat kata atau kalimat dasar pengetahuan untuk menentukan maknanya.

2.2.4. Generator

Generator adalah input yang dipahami untuk menghasilkan output yang bermanfaat. Struktur data yang dibuat oleh understander kemudian memulai tindakan atau langkah selanjutnya. Dalam bentuknya yang paling sederhana, generator bahasa alami memasukkan output dari jawaban standar yang diberikan kepada pengguna sebelumnya, berdasarkan makna yang dirangkum dari input tersebut.

2.3 TF-IDF

Pengklasifikasi pembelajaran mesin yang diawasi membutuhkan data tekstual dalam bentuk vector untuk dilatih. Fitur-fitur tekstual diubah menjadi bentuk vektor menggunakan teknik TF dan TF-IDF. Teknik ekstraksi fitur tidak hanya mengubah fitur tekstual menjadi bentuk vektor, tetapi juga membantu menemukan fitur signifikan yang diperlukan untuk membuat prediksi. Untuk sebagian besar fitur tidak berkontribusi pada prediksi kelas target [12].

Hal ini dapat dicapai dengan maksud bahwa suatu istilah akan tampak lebih jauh dalam dokumen yang panjang daripada dokumen yang pendek karena setiap dokumen memiliki varian yang berbeda. Seperti mode tentang standardisasi:

$$TF(t) = \frac{\text{no.of time term } t \text{ shows in a document}}{\text{(total no.of terms inside document)}} \quad (1)$$

Persamaan (1) digunakan untuk menghitung Term Frequency atau frekuensi istilah dalam sebuah dokumen. TF(t) merujuk pada nilai frekuensi suatu istilah tertentu, yang menunjukkan seberapa sering istilah tersebut muncul dalam dokumen yang sedang dianalisis. Untuk menghitung nilai ini, kita perlu mengetahui dua komponen penting: pertama, jumlah kemunculan istilah t dalam dokumen, yang merupakan total berapa kali istilah tersebut muncul. Misalnya, jika istilah "AI" muncul 5 kali dalam sebuah artikel, maka nilai ini adalah 5. Kedua, kita juga perlu mengetahui total jumlah semua istilah (kata) yang terdapat dalam dokumen tersebut, termasuk pengulangan dari istilah yang sama. Sebagai contoh, jika dokumen terdiri dari 100 kata, maka total jumlah kata adalah 100. Dengan membagi jumlah kemunculan istilah dengan total jumlah kata dalam dokumen, kita mendapatkan proporsi atau persentase seberapa besar kontribusi istilah tersebut terhadap keseluruhan dokumen. Nilai TF yang lebih tinggi menunjukkan bahwa istilah tersebut lebih relevan atau dominan dalam konteks dokumen. Sebagai ilustrasi, jika sebuah dokumen memiliki 200 kata dan istilah "data" muncul 10 kali, maka perhitungan TF untuk "data" adalah 0,05 atau 5%. Ini berarti bahwa istilah "data" menyumbang 5% dari total kata dalam dokumen tersebut.

Frekuensi term sering dibagi dengan panjang dokumen (jumlah total term dalam dokumen). IDF yaitu Inverse Document Frequency digunakan untuk menemukan seberapa besar sebuah term signifikan dalam teks. Setiap istilah diukur secara sama ketika TF dihitung. Namun demikian, diakui bahwa istilah-istilah yang meyakinkan, seperti "adalah", "dari", dan "bahwa", dapat muncul lebih banyak. Oleh karena itu, istilah yang sering muncul perlu ditimbang sebagai istilah luar biasa, dengan menghitung sebagai berikut:

$$IDF(t) = \log(e) \frac{\text{no. of time term } t \text{ shows in a document}}{\text{(total no. of terms inside document)}} \quad (2)$$

Pada persamaan (2) menggunakan logaritma natural untuk mengubah rasio kemunculan suatu term dalam dokumen ke dalam skala logaritmik yang membantu mereduksi perbedaan nilai antar term dengan frekuensi yang sangat bervariasi. Sebagai contoh, jika kata "data" muncul 5 kali dalam dokumen yang terdiri dari 100 kata, maka frekuensinya adalah $5/100 = 0,05$. Dengan menerapkan fungsi logaritma natural, didapatkan nilai $\log(0,05)$ yang sekitar -3,0, mengindikasikan bahwa kata tersebut muncul relatif jarang.

2.4 Random Forest

Random Forest adalah pengklasifikasi yang serbaguna dan dapat digunakan untuk mendeteksi berita hoaks. Algoritma ini pertama-tama menganalisis konten artikel berita, kemudian memperhatikan gaya penulisan, nada, dan sumber artikel tersebut. Setelah mengevaluasi semua fitur ini, algoritma mengkombinasikan hasilnya untuk memprediksi apakah artikel berita tersebut hoaks atau non-hoaks. Random Forest juga dapat mendeteksi bias dalam data dengan memeriksa titik data yang digunakan untuk pohon keputusan. Selain itu, Random Forest merupakan metode yang cepat, efisien, dan sederhana untuk mengidentifikasi berita hoaks. Algoritma ini juga tidak terlalu rentan terhadap overfitting, sehingga dapat menggeneralisasi dengan baik pada data yang belum pernah dilihat. Hal ini membuat Random Forest menjadi alat yang efektif untuk mendeteksi berita hoaks [13].

Menurut [14] pengklasifikasi ensemble ini membangun beberapa pohon keputusan dan menggabungkannya untuk mendapatkan hasil terbaik. Untuk pembelajaran pohon, terutama menggunakan agregasi bootstrap atau bagging. Untuk data yang diberikan, $X = \{x_1, x_2, x_3, \dots, x_n\}$ dengan respon $Y = \{y_1, y_2, y_3, \dots, y_n\}$ yang mengulangi bagging dari $b = 1$ sampai B. Sampel tak terlihat x' dibuat dengan merata-ratakan prediksi $\sum_{b=1}^B fb(x')$ dari setiap pohon individu pada x' :

$$j = \frac{1}{B} \sum_{b=1}^B fb(x') \quad (3)$$

Persamaan (3) menggambarkan cara kerja prediksi pada algoritma Random Forest untuk regresi, di mana $fb(x')$ adalah prediksi yang dihasilkan oleh pohon ke-(b) untuk data input (x'), dan (B) adalah jumlah total pohon dalam hutan. Dengan mengambil rata-rata dari semua prediksi tersebut, model menghasilkan prediksi akhir (j) yang cenderung lebih stabil dan kurang rentan terhadap noise karena kesalahan individual masing-masing pohon dapat saling menetralkan. Sebagai contoh, jika terdapat tiga pohon yang menghasilkan prediksi 2.5, 3.0, dan 3.5 untuk suatu input, maka prediksi

akhir model adalah $\frac{2.5+3.0+3.5}{3} = 3,0$, yang menunjukkan pendekatan ensemble untuk meningkatkan akurasi dan generalisasi model.

Ketidakpastian prediksi pada pohon ini dibuat melalui deviasi standarnya.

$$\sigma = \sqrt{\frac{\sum_{b=1}^B (fb(x') - f)^2}{B-1}} \quad (4)$$

Persamaan (4) menghitung deviasi standar dari prediksi individu yang dihasilkan oleh setiap pohon dalam Random Forest untuk suatu input (x'). Di sini, $fb(x')$ adalah prediksi yang dihasilkan oleh pohon ke-(b) dan f merupakan rata-rata prediksi seluruh pohon, yaitu hasil agregasi ensemble. Dengan menghitung selisih tiap prediksi dari nilai rata-rata, dikuadratkan selisih tersebut, kemudian menjumlahkannya dan membaginya dengan $B-1$ (jumlah pohon dikurangi satu) untuk mendapatkan varians sampel, akar dari nilai tersebut memberikan standar deviasi. Nilai σ ini penting karena menggambarkan seberapa besar variasi atau ketidakpastian prediksi antar pohon, sehingga memberikan gambaran mengenai keandalan prediksi akhir model.

2.5 K-Fold Cross Validation

Pentingnya evaluasi model tidak dapat ditekankan secara berlebihan dalam tugas pemodelan prediktif. Pilihan strategi evaluasi dapat secara signifikan mempengaruhi kinerja model yang dirasakan. Dalam penelitian ini, penerapan validasi silang k-fold dianggap tepat. Dataset dibagi menjadi tiga subset yang berbeda: (i) pelatihan, (ii) validasi, dan (iii) pengujian. Subset pelatihan secara khusus disisihkan untuk tujuan penyesuaian model. Subset validasi, di sisi lain, memainkan peran penting dalam penyetelan hiperparameter dan pemilihan model. Subset pengujian dicadangkan untuk evaluasi akhir, yang bertujuan untuk mengukur kapasitas model untuk generalisasi [15].

$$CV_k = \frac{1}{k} \sum_{i=1}^k \text{Metric}(M_i) \quad (5)$$

Dimana:

- CV_k adalah nilai *Cross-Validation* yang dihasilkan dari k fold.
- k adalah jumlah fold yang digunakan dalam *Cross-Validation*.
- $\text{Metric}(M_i)$ adalah metrik evaluasi (seperti akurasi, F1-score, dll.) yang dihitung pada fold ke-i menggunakan model M_i .

Pada persamaan (5) (terdapat asumsi bahwa indeks penjumlahan seharusnya berjalan dari $i=1$ ke k) menggambarkan proses evaluasi model menggunakan k-fold cross validation. Pada setiap lipatan (fold), model M_i diuji menggunakan metrik evaluasi tertentu misalnya akurasi, error, atau metrik lainnya yang menunjukkan performa model pada bagian data tersebut. Kemudian, nilai metrik dari setiap lipatan dijumlahkan dan dibagi dengan jumlah lipatan k untuk memperoleh rata-rata performa model secara keseluruhan. Rata-rata inilah yang memberikan gambaran lebih andal mengenai kemampuan generalisasi model terhadap data baru, karena dengan cara ini variabilitas yang muncul akibat pembagian data secara acak dapat dikurangi.

2.6 Confusion Matrix

Akurasi, Sensitivitas, Spesifisitas adalah metrik yang umum digunakan untuk memprediksi kemandirian model. Namun, tidak tepat untuk mengukur kemandirian model menggunakan matriks ini ketika dataset tidak seimbang dalam hal distribusi kelas dan model dapat memberikan akurasi yang tinggi dalam kasus seperti itu karena bias terhadap kelas mayoritas. Oleh karena itu, matriks kebingungan berguna dalam domain yang tidak seimbang dan memberikan wawasan yang lebih baik ke dalam model. Mari kita ambil contoh sebuah model yang memprediksi apakah sebuah artikel berita itu hoaks atau non-hoaks [16].

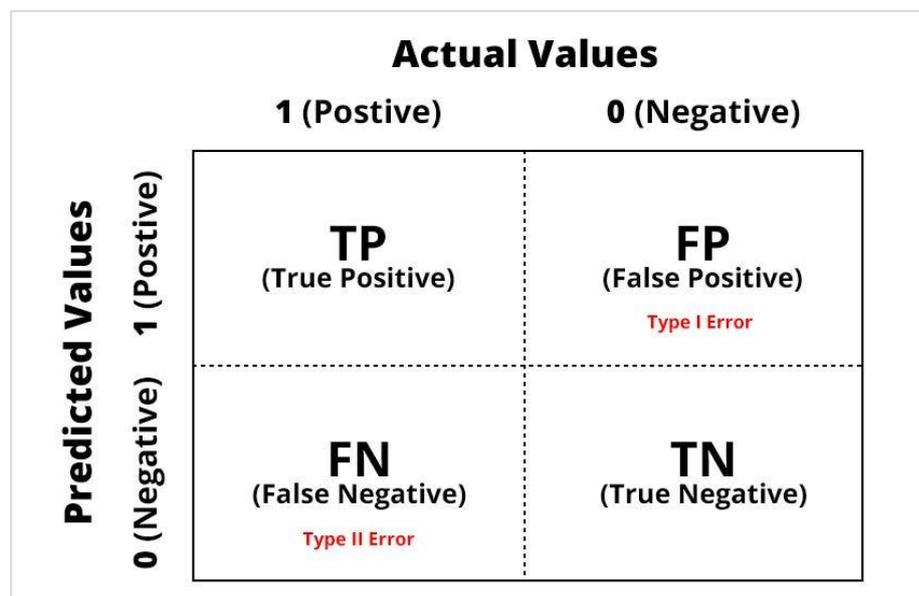
3 Metode Penelitian

Penelitian ini dilakukan dengan menggunakan pendekatan kuantitatif dan desain eksperimental untuk mengembangkan dan menguji model deteksi berita hoaks. Berikut adalah metode penelitian yang digunakan dalam penelitian ini

Populasi data dalam penelitian ini adalah semua artikel berita yang tersedia di website Kominfo, mencakup berbagai topik dan kategori. Sampel yang digunakan dalam penelitian ini adalah 12000 artikel berita yang terdiri dari 6000 berita hoaks dan 6000 berita non-hoaks. Artikel-artikel ini dipilih menggunakan teknik purposive sampling untuk memastikan relevansi dan representativitas data. Sumber Data dikumpulkan dari website Kominfo, yang menyediakan informasi terkait berita hoaks yang telah diverifikasi oleh pemerintah. Metode Pengumpulan Data pada penelitian ini menggunakan Web Scraping. Data dikumpulkan menggunakan teknik web scraping dengan alat seperti Scrapy dan BeautifulSoup. Proses scraping dilakukan untuk mengekstraksi informasi yang relevan dari halaman web Kominfo. Atribut dalam dataset tersebut adalah judul berita, deskripsi berita, penulis, tanggal berita, alamat url, kategori berita, alamat url gambar dan halaman. Selanjutnya dilakukan preprocessing dataset yaitu lowercase, remove special characters, remove stop words, stemming dan TF-IDF.

Judul dan deskripsi berita dianalisis menggunakan teknik pemrosesan teks seperti *Term Frequency-Inverse Document Frequency (TF-IDF)*. *TF-IDF* menghitung seberapa penting kata-kata tertentu dalam sebuah dokumen dibandingkan dengan keseluruhan korpus. *TF-IDF* membantu mengidentifikasi kata-kata yang sering muncul dalam berita hoaks dibandingkan dengan berita non-hoaks, memberikan bobot lebih pada kata-kata yang signifikan untuk deteksi. Dataset lalu dibagi untuk training sebanyak 80% serta 20% sebagai data testing

Klasifikasi model pada penelitian ini menggunakan algoritma *Random Forest*. *Random Forest* mengolah data melalui berbagai *decision tree* untuk menghasilkan prediksi yang robust. Kinerja model klasifikasi diukur menggunakan metode Confusion matrix. Matrix ini bekerja dengan membandingkan hasil model klasifikasi dengan label yang seharusnya. Matrix dapat dilihat pada Gambar 4.



Gambar 4. Confusion matrix

True Positives (TP) merupakan jumlah data positif yang terdeteksi benar, sedangkan False Negatives (FN) merupakan jumlah data positif yang terdeteksi sebagai data negatif. True Negatives (TN) merupakan jumlah data negative yang terdeteksi secara benar, sedangkan False Positives merupakan data negatif yang terdeteksi sebagai data positif [16]. Untuk mengevaluasi efektivitas model, *accuracy*, *precision*, *recall*, dan *F1-score*, digunakan sebagai indikator evaluasi [17].

Persamaan (6) merupakan perhitungan *Accuracy* yang merupakan proporsi semua sampel yang diprediksi benar dari semua hasil yang diprediksi dengan cara membagi jumlah prediksi yang benar (TP+TN) dengan total seluruh prediksi (TP+TN+FP+FN), kita memperoleh proporsi prediksi yang akurat. Meskipun akurasi memberikan gambaran umum tentang performa model:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (6)$$

Persamaan (7) merupakan perhitungan *Precision* yang merupakan proporsi sampel positif yang diprediksi dengan benar sebagai positif dari semua sampel positif yang diprediksi. Cara perhitungannya adalah membagi jumlah true positives (*TP*) yaitu kasus di mana model dengan benar mengidentifikasi data sebagai positif—dengan jumlah total prediksi positif (*TP+FP*), di mana *FP* adalah false positives, yaitu kasus di mana data negatif salah diklasifikasikan sebagai positif. Dengan demikian, nilai *precision* menunjukkan proporsi prediksi positif yang benar; semakin tinggi nilai *precision*, semakin sedikit kesalahan dalam mengklasifikasikan data negatif sebagai positif, yang sangat penting terutama ketika konsekuensi false positive cukup signifikan:

$$Precision = \frac{TP}{TP+FP} \tag{7}$$

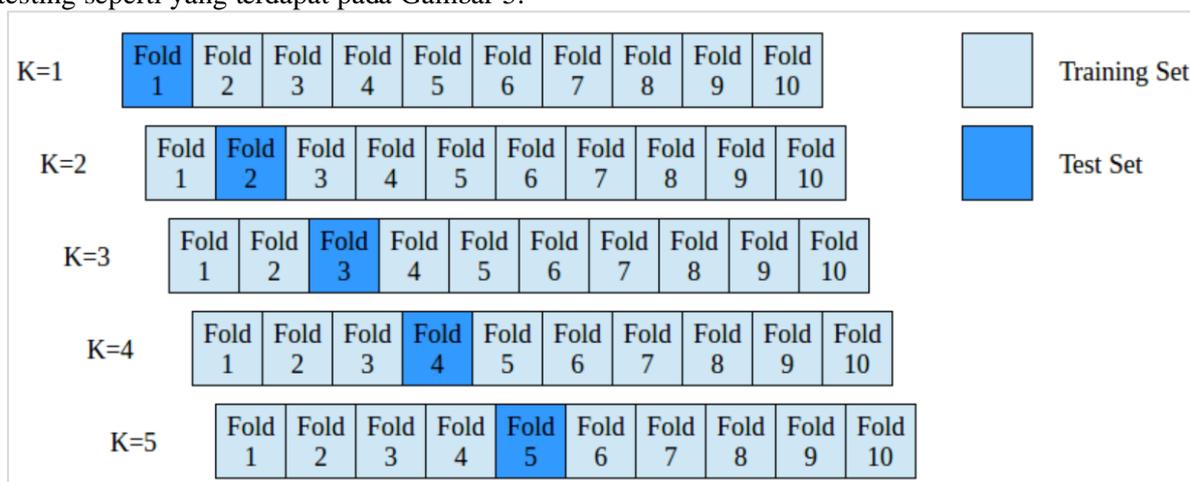
Persamaan (8) merupakan perhitungan *Recall* yang merupakan proporsi sampel positif yang diprediksi dengan benar sebagai positif dari semua sampel positif. Nilai *recall* nilai ini dihitung dengan membagi jumlah true positives (*TP*), yaitu kasus positif yang berhasil dideteksi dengan benar oleh model, dengan total jumlah kasus positif sebenarnya (*TP + FN*), di mana *FN* (false negatives) adalah kasus positif yang gagal terdeteksi oleh model. Semakin tinggi nilai *recall*, semakin baik model dalam mendeteksi data positif:

$$Recall = \frac{TP}{TP+FN} \tag{8}$$

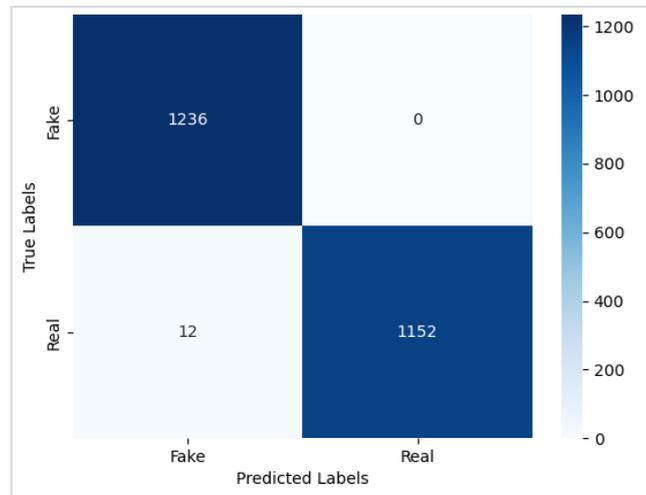
Persamaan (8) merupakan perhitungan *F1-Score* ketika *precision* dan *recall* dipertimbangkan secara bersamaan, keduanya mencapai nilai terbesar pada saat yang sama. Tugas ini meningkatkan tingkat penarikan kembali dengan alasan untuk memastikan tingkat akurasi. Persamaan ini mengukur kinerja model klasifikasi dengan menggabungkan *precision* dan *recall* menjadi satu metrik evaluasi yang menyajikan rata-rata harmonis dari kedua nilai tersebut. *Precision* sendiri mengukur ketepatan prediksi positif, sedangkan *recall* mengukur kemampuan model dalam menangkap semua kasus positif yang sebenarnya. Karena *F1-Score* merupakan rata-rata harmonis, nilai ini akan rendah jika salah satu dari *precision* atau *recall* rendah, sehingga hanya akan tinggi ketika kedua metrik tersebut sama-sama tinggi. Pendekatan ini sangat berguna terutama pada masalah dengan data yang tidak seimbang, karena memberikan gambaran yang lebih adil mengenai performa model dibandingkan dengan hanya menggunakan salah satu metrik saja:

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{9}$$

Pada bagian validasi penelitian ini menggunakan metode K-Fold Cross Validation dengan membagi dataset menjadi 2 bagian yaitu 80% sebagai data training dan 20% sebagai data testing. Proses ini dilakukan iterasi sebanyak 5 kali sehingga semua data mendapat bagian menjadi data testing seperti yang terdapat pada Gambar 5.



Gambar 5. K-fold cross validation



Gambar 7. Confusion matrix

Pada Gambar 7 dapat menghitung berbagai metrik evaluasi untuk menilai kinerja model lebih lanjut. Metrik yang digunakan mencakup Accuracy, Precision, Recall, dan F1-Score. Berikut nilai yang didapatkan :

$$Accuracy = \frac{1236 + 1152}{1236 + 1152 + 12 + 0} = 0.995$$

$$Precision = \frac{1236}{1236+12} = 0.990$$

$$Recall = \frac{1236}{1236 + 0} = 1.000$$

$$F1\ Score = 2 \times \frac{0.990 \times 1.000}{0.990 + 1.000} = 0.995$$

Secara keseluruhan, confusion matrix menunjukkan bahwa model ini memiliki performa yang sangat baik dalam mengklasifikasikan data ke dalam kelas yang benar, dengan sedikit kesalahan prediksi. Hasil ini mengindikasikan bahwa model dapat diandalkan untuk aplikasi klasifikasi yang lebih luas.

Aplikasi yang dikembangkan untuk penelitian ini menggunakan bahasa pemrograman python dengan framework flask, aplikasi ini bertujuan untuk mempermudah proses klasifikasi berita sebagai hoaks atau non-hoaks berdasarkan judul dan deskripsi yang diberikan. Prototype dibuat untuk pengujian data baru seperti pada Gambar 8

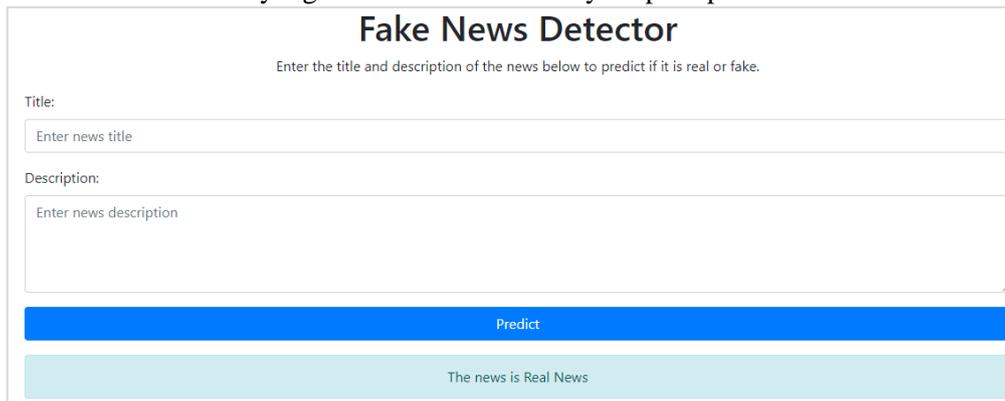
Fake News Detector
Enter the title and description of the news below to predict if it is real or fake.

Title:

Description:

Gambar 8. Tampilan antarmuka aplikasi

Pengguna aplikasi cukup menginput dua informasi utama, yaitu judul dan deskripsi berita yang ingin diklasifikasikan ke dalam *form input* yang disediakan oleh aplikasi. Setelah itu, aplikasi akan memproses input tersebut dan mengklasifikasikan berita tersebut sebagai hoaks atau non-hoaks menggunakan model klasifikasi yang telah dilatih sebelumnya seperti pada Gambar 9.



Gambar 9. Hasil klasifikasi berita

Aplikasi yang dikembangkan berhasil memberikan solusi praktis untuk mengklasifikasikan berita sebagai hoaks atau non-hoaks. Aplikasi ini mudah digunakan, bahkan oleh pengguna yang tidak memiliki latar belakang teknis. Dengan antarmuka yang sederhana, pengguna hanya perlu memasukkan judul dan deskripsi berita untuk mendapatkan hasil klasifikasi yang akurat.

5 Kesimpulan

Penelitian ini bertujuan untuk membangun dan mengimplementasikan sebuah model machine learning yang dapat mengklasifikasikan berita sebagai hoaks atau non-hoaks, dengan memanfaatkan judul dan deskripsi berita sebagai input. Berdasarkan hasil yang diperoleh, dapat disimpulkan bahwa algoritma Random Forest (RF) berhasil digunakan dalam penelitian ini. Evaluasi model menggunakan k-fold cross-validation dengan k=5 menghasilkan hasil yang sangat baik, dengan nilai akurasi yang tinggi untuk model Random Forest 0.989, nilai precision 0.990, nilai recall 1.000, dan nilai f1 score 0.995. Hasil ini menunjukkan bahwa model mampu mengklasifikasikan berita dengan akurasi tinggi dan performa yang konsisten baik pada data latih maupun uji. Aplikasi yang dikembangkan mampu menerima input berupa judul dan deskripsi. Evaluasi menggunakan confusion matrix menunjukkan bahwa model sangat efisien dalam mengklasifikasikan berita dengan sedikit kesalahan, dengan False Positive dan False Negative yang minim. Pengujian terhadap waktu proses juga menunjukkan bahwa aplikasi dapat mengklasifikasikan berita dalam waktu singkat meskipun dengan ukuran dataset yang besar.

Referensi

- [1] S. Sadiq, N. Wagner, M. L. Shyu, and D. Feaster, "High Dimensional Latent Space Variational AutoEncoders for Fake News Detection," *Proc. - 2nd Int. Conf. Multimed. Inf. Process. Retrieval, MIPR 2019*, pp. 437–442, 2019, doi: 10.1109/MIPR.2019.00088.
- [2] trends.google.com, "Fake News Trend," 2024. [https://trends.google.com/trends/explore?date=2023-04-16 2024-03-16&geo=ID&q=Berita aktual,Berita palsu&hl=id](https://trends.google.com/trends/explore?date=2023-04-16%2024-03-16&geo=ID&q=Berita%20aktual,Berita%20palsu&hl=id)
- [3] R. K. Kaliyar, A. Goswami, and P. Narang, "FakeBERT: Fake News Detection in Social Media with a BERT-based Deep Learning Approach," *Multimed. Tools Appl.*, vol. 80, no. 8, pp. 11765–11788, 2021, doi: 10.1007/s11042-020-10183-2.
- [4] E. Qawasmeh, M. Tawalbeh, and M. Abdullah, "Automatic Identification of Fake News using Deep Learning," *2019 6th Int. Conf. Soc. Networks Anal. Manag. Secur. SNAMS 2019*, pp. 383–388, 2019, doi: 10.1109/SNAMS.2019.8931873.
- [5] S. Lyu and D. C.-T. Lo, "Fake News Detection by Decision Tree," *IEEE SoutheastCon 2020*, pp. 430–435, 2020, doi: 10.1109/SoutheastCon44009.2020.9249688.
- [6] R. Jihad and S. A. Yousif, "Fake News Classification using Random Forest and Decision Tree <http://sistemasi.ftik.unisi.ac.id>

- (J48),” *Al-Nahrain J. Sci.*, vol. 23, no. 4, pp. 49–55, 2020, doi: 10.22401/anjs.23.4.09.
- [7] H. J. Alshahrani *et al.*, “Hunter Prey Optimization with Hybrid Deep Learning for Fake News Detection on Arabic Corpus,” *Comput. Mater. Contin.*, vol. 75, no. 2, pp. 4255–4272, 2023, doi: 10.32604/cmc.2023.034821.
- [8] S. Bachhety, R. Singhal, and R. Jain, “Intelligent Data Analysis with Data Mining: Theory and Applications,” *Intell. Data Anal. From Data Gather. to Data Compr.*, vol. 1, pp. 63–83, 2020, doi: 10.1002/9781119544487.ch4.
- [9] P. Mukherjee, S. Santra, S. Bhowmick, A. Paul, P. Chatterjee, and A. Deyasi, “Development of GUI for Text-to-Speech Recognition using Natural Language Processing,” *2018 2nd Int. Conf. Electron. Mater. Eng. Nano-Technology, IEMENTech 2018*, pp. 1–4, 2018, doi: 10.1109/IEMENTECH.2018.8465238.
- [10] S. K. A. Fahad and A. E. Yahya, “Inflectional Review of Deep Learning on Natural Language Processing,” *2018 Int. Conf. Smart Comput. Electron. Enterp. ICSCEE 2018*, no. D1, pp. 2018–2021, 2018, doi: 10.1109/ICSCEE.2018.8538416.
- [11] E. A. Lisangan, “Natural Language Processing dalam memperoleh Informasi Akademik Mahasiswa Universitas Atma Jaya Makassar,” *J. Temat.*, vol. 1, no. March 2013, pp. 1–9, 2015, doi: 2303-3878.
- [12] A. Yousaf *et al.*, “Emotion Recognition by Textual Tweets Classification using Voting Classifier (LR-SGD),” *IEEE Access*, vol. 9, pp. 6286–6295, 2021, doi: 10.1109/ACCESS.2020.3047831.
- [13] M. Tajrian, A. Rahman, M. A. Kabir, and M. R. Islam, “A Review of Methodologies for Fake News Analysis,” *IEEE Access*, vol. 11, no. June, pp. 73879–73893, 2023, doi: 10.1109/ACCESS.2023.3294989.
- [14] S. Mohan, C. Thirumalai, and G. Srivastava, “Effective Heart Disease Prediction using Hybrid Machine Learning Techniques,” *IEEE Access*, vol. 7, pp. 81542–81554, 2019, doi: 10.1109/ACCESS.2019.2923707.
- [15] K. Rajesh, A. Kumar, and R. Kadu, “Fraudulent News Detection using Machine Learning Approaches,” *2019 Glob. Conf. Adv. Technol. GCAT 2019*, pp. 1–5, 2019, doi: 10.1109/GCAT47503.2019.8978436.
- [16] M. Al Bataineh, D. I. A. Abdoun, H. Alnuaimi, Z. Al-Qudah, Z. Albataineh, and M. Al Ahmad, “Head Impact Detection using Machine Learning Algorithms,” *IEEE Access*, vol. 12, no. January, pp. 4938–4947, 2024, doi: 10.1109/ACCESS.2023.3349212.
- [17] D. Rohera *et al.*, “A Taxonomy of Fake News Classification Techniques: Survey and Implementation Aspects,” *IEEE Access*, vol. 10, pp. 30367–30394, 2022, doi: 10.1109/ACCESS.2022.3159651.