KLASIFIKASI *HOAX* DAN FAKTA MENGGUNAKAN ALGORITMA *SHALLOW NEURAL NETWORK* PADA BERITA POLITIK PEMILIHAN PRESIDEN INDONESIA 2024

Mohamad Frananda Adiezwara Ramadhan, Iwan Rizal Setiawan, Asriyanik
Teknik Informatika, Universitas Muhammadiyah Sukabumi
Jl. R. Syamsudin, S.H. No. 50, Cikole, Kec. Cikole, Kota Sukabumi, Jawa Barat 43113
mohamadfrananda031@ummi.ac.id

ABSTRAK

Dalam beberapa tahun terakhir, isu *hoaks* telah menjadi ancaman serius terhadap integritas informasi dan proses demokrasi. Pada tahun 2023, Kementerian Komunikasi dan Informatika (Kominfo) mencatat adanya peningkatan signifikan dalam penanganan isu hoaks, dengan 1.615 konten teridentifikasi. Menjelang Pilpres 2024, diperkirakan penyebaran berita *hoaks* akan meningkat, yang berpotensi mengancam kualitas proses demokrasi di Indonesia. Berbagai penelitian telah dilakukan untuk mengatasi masalah ini menggunakan berbagai metode klasifikasi, termasuk *Naïve Bayes* dan SVM. Namun, penelitian ini berfokus pada penggunaan *Shallow Neural Network*, yang diharapkan mampu menangkap pola non-linear dan mengolah berbagai jenis fitur dari data teks. Penelitian ini bertujuan untuk menguji efektivitas *Shallow Neural Network* dalam klasifikasi berita *hoaks* dan fakta terkait pemilihan presiden Indonesia 2024. Metodologi yang digunakan dalam penelitian ini adalah SEMMA, yang meliputi tahapan *Sample*, *Explore*, *Modify*, Model, dan *Assess*. Hasil evaluasi menunjukkan bahwa model klasifikasi yang dikembangkan memiliki performa yang baik dalam membedakan berita *valid* dan *hoaks*, dengan akurasi sebesar 93%. Nilai presisi, *recall*, dan *F1 score* yang tinggi untuk kedua kelas menunjukkan bahwa model ini dapat diandalkan untuk tugas klasifikasi berita, dengan keseimbangan yang baik antara presisi dan *recall*, sehingga mampu meminimalkan kesalahan dalam mendeteksi berita *valid* maupun *hoaks*.

Kata kunci : *Pilpres 2024, Shallow Neural Network, Hoaks, Klasifikasi*

1. PENDAHULUAN

Dalam beberapa tahun terakhir, isu *hoaks* telah menjadi ancaman serius terhadap integritas informasi dan proses demokrasi. Pada tahun 2023, Kementerian Komunikasi dan Informatika (Kominfo) mengidentifikasi 1.615 konten *hoaks* di berbagai sektor, termasuk kesehatan, kebijakan pemerintah, dan politik. Peningkatan ini menunjukkan urgensi untuk meningkatkan kemampuan deteksi dan pencegahan penyebaran informasi palsu, terutama dalam konteks politik yang mempengaruhi proses demokrasi [1]

Menjelang Pemilu 2024, diperkirakan akan terjadi lonjakan penyebaran berita *hoaks* yang dapat mengancam kualitas proses demokrasi di Indonesia. Pada awal tahun 2024, Menkominfo Budi Arie Setiadi melaporkan bahwa terdapat 203 isu *hoaks* terkait Pemilu, dengan total 2.882 konten yang tersebar di berbagai platform digital seperti Facebook, Twitter, Instagram, TikTok, Snack Video, dan YouTube. Banyaknya konten yang diidentifikasi dan diambil tindakan oleh Kominfo menyoroti perlunya langkah efektif untuk mengatasi penyebaran *hoaks* dalam konteks politik guna menjaga integritas informasi dan proses demokrasi [2].

[3] dalam penelitian berjudul "*Malware Classification Based on Shallow Neural Network*" menemukan bahwa algoritma *Shallow Neural Network* memiliki tingkat akurasi hingga 98% dalam mengklasifikasikan *malware*. Metode ini menunjukkan potensi besar dalam mendeteksi *hoaks* dengan akurasi tinggi dan waktu pemrosesan yang cepat, menjadikannya pilihan yang ideal untuk penelitian ini.

Berbagai penelitian telah dilakukan untuk mengatasi masalah ini dengan berbagai metode klasifikasi. Salah satu penelitian menonjol adalah oleh [4] yang menggunakan algoritma *Naïve Bayes* untuk mengklasifikasikan berita hoaks selama kampanye pemilu. Algoritma ini, dalam kombinasi dengan metode *Knowledge Discovery in Database* (KDD) dan alat pengujian RapidMiner, menunjukkan akurasi tinggi sebesar 89,54%, dengan presisi 86,44% dan *recall* 82,00%. Metode ini efektif dalam mengelola *dataset* besar dan mengatasi tantangan dalam membedakan berita *hoaks* dari fakta [4].

Penelitian lain yang relevan dilakukan oleh [5] tidak hanya menggunakan *Naïve Bayes*, tetapi juga *Support Vector Machine* (SVM) untuk membandingkan kinerja kedua algoritma tersebut. Hasil penelitian menunjukkan bahwa *Naïve Bayes* mencapai akurasi 97%, dengan presisi 94%, *recall* 100%, dan F1-score 97%. Sementara itu, SVM menunjukkan akurasi 95%, dengan presisi 94%, *recall* 97%, dan F1-score 95%. Hasil ini menunjukkan bahwa kedua algoritma memiliki kinerja yang baik dalam mengklasifikasikan berita *hoaks* terkait pemilu.

Penelitian ini berbeda dengan penelitian sebelumnya karena menggunakan *shallow neural network*, yang diharapkan memberikan kelebihan dalam beberapa aspek. Pertama, *shallow neural network* mampu menangkap pola non-linear yang mungkin terlewat oleh algoritma *Naïve Bayes* dan

SVM. Kedua, *shallow neural network* memiliki fleksibilitas dalam menangani berbagai jenis fitur dari data teks, termasuk fitur yang *kompleks* dan berinteraksi satu sama lain. Ketiga, dengan kemampuan generalisasi yang lebih baik, *shallow neural network* dapat meningkatkan akurasi dalam mendeteksi berita *hoaks* dibandingkan dengan metode sebelumnya.

2. TINJAUAN PUSTAKA

Setelah penulis menelaah beberapa penelitian, terdapat sejumlah studi yang memiliki kaitan dengan penelitian ini.

Studi oleh [6] membahas penyebaran berita *hoax* selama Pemilihan Umum Presiden 2019 melalui media daring dan media sosial. Mereka menganalisis tren penyebaran berita palsu serta dampaknya terhadap opini publik dan proses demokrasi. Penelitian ini memberikan wawasan penting tentang strategi yang diperlukan untuk menanggulangi penyebaran berita palsu di masa mendatang.

Selanjutnya, Studi oleh [7] Penelitian ini mengevaluasi kemampuan algoritma *Random Forest* dalam mendeteksi berita palsu pada pemilu 2024, menggunakan *dataset* berisi 859 *record*. Proses yang dilakukan meliputi *cleaning*, *tokenisasi*, dan *stemming*, dengan hasil akurasi sebesar 84,88%.

Selanjutnya, Studi oleh [8] Penelitian ini mengklasifikasikan sentimen publik tentang pemilu menggunakan 1064 artikel berita online dengan algoritma *Naive Bayes*, *Random Forest*, dan *Support Vector Machine*. Model diuji dengan dan tanpa SMOTE, menghasilkan akurasi tertinggi 92,05% oleh *Support Vector Machine* tanpa SMOTE.

2.1. Berita *Hoax* Politik

Berita politik memainkan peran krusial dalam membentuk opini publik dan kebijakan. Di era informasi saat ini, kecepatan dan volume informasi yang beredar dapat memicu penyebaran *hoax* yang mempengaruhi proses demokrasi dan kepercayaan publik [9] Memahami bagaimana berita politik diproduksi, disebarkan, dan diterima publik sangat penting untuk memastikan integritas informasi. Berita palsu tidak hanya mempengaruhi opini publik tetapi juga melayani kepentingan individu atau kelompok tertentu, sering kali dengan menarik lebih banyak pengunjung ke situs *web* yang diiklankan. Korban berita palsu dapat mengalami kerugian materi dan moral, seperti reputasi yang rusak, serta potensi permusuhan yang dapat membahayakan mereka.

2.2. *Artificial Intelligence* dan *Machine Learning*

Kecerdasan Buatan (AI) dan Pembelajaran Mesin (ML) telah mengubah cara kita memproses informasi dan membuat keputusan. AI mengacu pada simulasi kecerdasan manusia dalam mesin yang dirancang untuk berpikir dan bertindak seperti manusia. Di sisi lain, ML adalah cabang AI yang berfokus pada pengembangan algoritma yang memungkinkan mesin belajar dari data dan meningkatkan kinerjanya secara otomatis [10]. Teknologi ini memainkan peran penting dalam mengidentifikasi pola dalam data besar, termasuk mendeteksi hoax dalam berita politik.

2.3. *Natural Language Processing*

NLP memungkinkan mesin memahami dan memproses bahasa manusia, menjadi kunci dalam analisis teks dan deteksi *hoaks* [11] Melalui teknik NLP, sistem dapat menganalisis struktur dan makna teks untuk mengidentifikasi ciri-ciri informasi palsu atau menyesatkan, memungkinkan pengembangan algoritma canggih untuk menyaring data teks besar dan membedakan konten fakta dari hoaks dengan akurasi tinggi. NLP berperan penting dalam meningkatkan efisiensi dan efektivitas deteksi *hoaks*, berkontribusi pada upaya memerangi misinformasi dan menjaga integritas informasi di ruang digital**.**

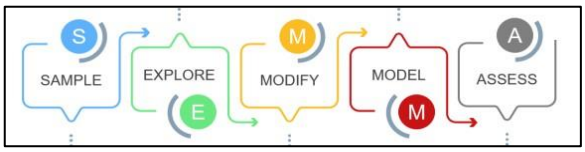
2.4. *Shallow Neural Network*

Shallow Neural Networks (SNN) menawarkan pendekatan yang lebih sederhana dibandingkan *deep neural networks* untuk klasifikasi teks, termasuk deteksi *hoaks* [12]. Meskipun memiliki lebih sedikit lapisan, SNN dapat efektif dengan memanfaatkan fitur teks yang menunjukkan hoaks atau informasi palsu. Arsitektur yang lebih dangkal membuat SNN lebih mudah diinterpretasikan dan diimplementasikan, menjadikannya pilihan menarik untuk aplikasi deteksi hoaks di mana interpretasi hasil penting. Potensi SNN untuk memberikan performa memadai dengan kompleksitas lebih rendah menunjukkan kemampuannya dalam mengembangkan sistem deteksi *hoaks* yang efisien dan mudah dipahami.

3. METODE PENELITIAN

Dalam proses pengembangan *Deep Learning* algoritma *shallow neural network*, metode SEMMA digunakan sebagai kerangka kerja yang efektif untuk proyek-proyek data mining. Pendekatan ini menawarkan panduan yang jelas dalam pembuatan model *data mining*, dengan fokus khusus pada strukturisasi pengembangan model [13].

Penelitian ini menggunakan metodologi SEMMA, sehingga tahapan penelitian disesuaikan dengan kerangka kerja SEMMA, yang merupakan akronim dari *Sample*, *Explore*, *Modify*, *Model*, dan Assess. Metodologi SEMMA dirancang khusus untuk memandu proses analisis data, terutama dalam pembelajaran mesin dan *data mining*. Berikut adalah tahapan penelitian yang diadaptasi menggunakan metodologi SEMMA:



Gambar 1. Metodologi semma

Pada Gambar 1. Diatas Kerangka kerja SEMMA (*Sample*, *Explore*, *Modify*, Model, *Assess*) adalah metodologi yang dirancang untuk memandu proses data mining dan pembelajaran mesin dengan memberikan struktur yang jelas dan terarah dalam pengembangan model. Dalam penelitian ini, SEMMA digunakan untuk mengembangkan *shallow neural network* yang bertujuan mengklasifikasikan berita *hoax* dan *factual* terkait pemilihan presiden Indonesia 2024. Tahapan pertama dalam metodologi ini adalah *Sample*. Pada tahap ini, data yang representatif dikumpulkan dari berbagai sumber berita terpercaya. Data yang terkumpul kemudian dipra-proses untuk menghilangkan duplikasi dan entri yang tidak relevan, serta dibagi menjadi *dataset* pelatihan dan pengujian untuk memastikan model yang dihasilkan memiliki validitas yang tinggi.

Setelah pengambilan sampel, tahap berikutnya adalah *Explore*. Pada tahap ini, eksplorasi data dilakukan untuk memahami karakteristik utama dari *dataset* yang dikumpulkan. Analisis deskriptif diterapkan untuk mengidentifikasi pola, distribusi, dan anomali dalam data. Berbagai teknik visualisasi seperti grafik batang, histogram, dan word clouds digunakan untuk mengeksplorasi distribusi kata dalam teks berita. Tahap ini juga melibatkan identifikasi fitur penting yang berpotensi mempengaruhi klasifikasi berita *hoax* dan *factual*, sehingga dapat memberikan panduan dalam proses modifikasi data selanjutnya.

Tahap ketiga adalah *Modify*, di mana data yang telah dieksplorasi dimodifikasi dan dipersiapkan untuk pemodelan. Pada tahap ini, pembersihan teks dilakukan, termasuk penghapusan *stop words*, stemming, dan lemmatization untuk menyederhanakan teks. Data teks kemudian dikonversi menjadi representasi numerik menggunakan teknik seperti *tokenizer* dan *sequence and padding*. Selain itu, seleksi fitur dilakukan untuk memilih fitur yang paling relevan dan mengurangi dimensi data jika diperlukan, memastikan bahwa model yang dibangun efisien dan efektif.

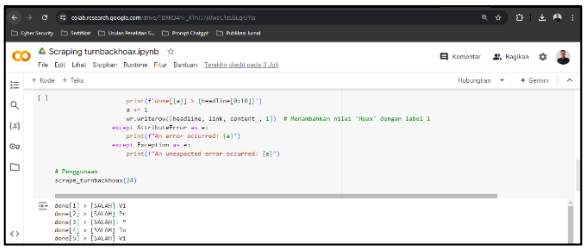
Tahap keempat dan kelima dalam kerangka SEMMA adalah *Model* dan *Assess*. Pada tahap *Model*, *shallow neural network* dikembangkan dan dilatih menggunakan *dataset* yang telah dipersiapkan. Penentuan arsitektur jaringan termasuk jumlah lapisan, neuron, dan fungsi aktivasi dilakukan dengan cermat. Model dilatih menggunakan *dataset* pelatihan, dan validasi silang digunakan untuk mengoptimalkan hyperparameters. Tahap akhir, ***Assess***, melibatkan evaluasi kinerja model dengan metrik evaluasi seperti akurasi, presisi, *recall*, dan *F1-score*. Hasil model dibandingkan dengan baseline sederhana atau model lain yang telah ada untuk menilai peningkatan kinerja. Analisis error dilakukan untuk mengidentifikasi area yang memerlukan perbaikan dan mengoptimalkan model lebih lanjut, memastikan bahwa model yang dihasilkan memiliki akurasi dan efektivitas yang tinggi dalam klasifikasi berita *hoax* dan *factual*.

4. HASIL DAN PEMBAHASAN

Penelitian bertujuan untuk mengklasifikasikan berita hoaks dalam pemilihan presiden Indonesia tahun 2024 menggunakan metodologi SEMMA, yang meliputi tahapan *Sample*, *Explore*, *Modify*, *Model*, dan *Assess*.

4.1. *Sample*

Tahap ini melibatkan pemilihan dan pengumpulan *dataset* yang digunakan dalam penelitian. *Dataset* utama berasal dari Kaggle, terdiri dari 27 *dataset* berita hoaks politik dan fakta, ditambah dengan 2500 data berita politik fakta dan *hoaks* terkait pemilihan presiden Indonesia 2024, yang diambil dari sumber berita terpercaya seperti CNN, Kompas, Tempo, Kominfo, dan Turnbackhoax. Selain itu, data tambahan dikumpulkan melalui teknik *scraping* dari situs web Kementerian Komunikasi dan Informatika (Kominfo) untuk memperkaya *dataset* dengan informasi terbaru dan relevan**.**



Gambar 2. Program *scraping*

	text_news	hoax
0	Hasil periksa fakta Imroatul Unggahan video de...	1
1	Hasil periksa fakta Luthfiyah Gambar rekor di ...	1
2	Hasil periksa fakta Moch. Marcellodiansyah Mel...	1
3	Hasil periksa fakta Siti Lailatul Fitriyah Ung...	1
4	Hasil periksa fakta Siti Lailatul Fitriyah Vid...	1

Gambar 3. *Sample dataset*

Pada Gambar 3. diatas data diperoleh melalui proses *web scraping* menggunakan t*ools* Beautiful Soup dan disimpan dalam file CSV yang terletak di Google Drive. Kemudian, pustaka pandas di Python yang dijalankan di Google Colaboratory digunakan untuk membaca file CSV tersebut. Dengan menggunakan metode pd.read_csv, data dari file CSV dimuat ke dalam sebuah *DataFrame* pandas, yang secara default memisahkan kolom dengan koma (,). Setelah data dimasukkan ke dalam *DataFrame* bernama df, metode df.head() digunakan untuk menampilkan lima baris pertama dari *DataFrame* tersebut, sehingga pengguna dapat melihat cuplikan awal data yang telah dibaca.

Dataset ini terdiri dari empat kolom: Judul, Link, Text_news, dan *Hoax*. Dua atribut utama yang digunakan untuk klasifikasi teks adalah Text_news dan *Hoax*. Atribut Text_news berisi teks berita atau

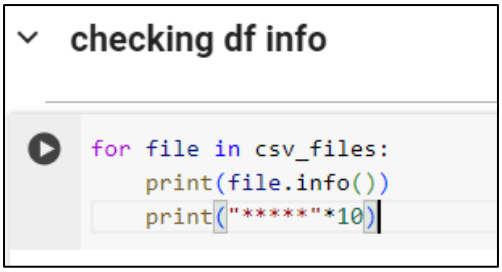
pernyataan yang dapat berupa judul berita, kutipan, atau pernyataan lengkap yang dianalisis untuk menentukan kebenarannya, dengan tipe data teks/string. Atribut *Hoax* berisi label yang menunjukkan apakah teks yang diberikan merupakan hoax atau tidak, digunakan sebagai target dalam proses klasifikasi. Tipe data *Hoax* adalah numerik/biner, dengan nilai 0 untuk "bukan *hoax*" dan 1 untuk "*hoax*". Atribut-atribut ini digunakan bersama-sama dalam proses klasifikasi teks untuk membangun model yang dapat secara otomatis membedakan antara informasi yang benar dan yang menyesatkan, yang sangat berguna dalam konteks media sosial, portal berita, dan platform digital lainnya.

4.2. *Explore*

Pada tahap eksplorasi, *dataset* dianalisis untuk memahami karakteristik dan strukturnya, termasuk distribusi kelas (*hoax* vs. fakta), frekuensi kata, dan pola umum. Visualisasi data seperti word clouds dan distribusi frekuensi kata membantu dalam proses ini.

Gambar 4. Visualisasi data wordcloud

Gambar 4. tersebut adalah word cloud yang menampilkan kata-kata paling sering muncul dalam *dataset* berita politik. Kata-kata seperti "Ketua", "Umum", "saat", "ini", dan "partai" menonjol dengan ukuran besar, menunjukkan frekuensi kemunculannya yang tinggi. Kata-kata lain seperti "politik", "Jakarta", "Anies Baswedan", "Joko Widodo", dan "Presiden" juga terlihat menonjol, mencerminkan topik dan tokoh yang sering dibahas. Word cloud ini memberikan gambaran visual tentang frekuensi dan pentingnya kata-kata dalam konteks politik, menyoroti topik, peristiwa, dan individu yang sering dibicarakan.



Gambar 5. Program mengecek informasi *dataset*

Pada gambar 5. adalah sebuah program loop yang digunakan untuk mencetak informasi mengenai setiap file CSV dalam daftar csv_files. Pada setiap iterasi loop, fungsi info() dipanggil pada objek file, yang kemungkinan merupakan *DataFrame* dari pustaka pandas, untuk menampilkan ringkasan informasi mengenai DataFrame tersebut, termasuk jumlah baris dan kolom, tipe data setiap kolom, serta jumlah nilai non-null di setiap kolom. Setelah itu, string "*****" diulang sebanyak 10 kali untuk mencetak garis pemisah, memberikan visualisasi yang jelas antara informasi dari satu file CSV dengan file CSV berikutnya. Tujuan dari kode ini adalah untuk memberikan gambaran umum dan terstruktur mengenai konten dari setiap file CSV dalam daftar tersebut.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2407 entries, 0 to 2406
Data columns (total 2 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   text_news  2407 non-null   object
 1   hoax       2407 non-null   int64
dtypes: int64(1), object(1)
memory usage: 37.7+ KB
None
**************************************************
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27447 entries, 0 to 27446
Data columns (total 2 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   text_news  27447 non-null  object
 1   hoax       27447 non-null  int64
dtypes: int64(1), object(1)
memory usage: 429.0+ KB
None
**************************************************
```

Gambar 6. Informasi *dataset*

Pada Gambar 6 *output* menunjukkan informasi tentang dua *DataFrame*, masing-masing memiliki dua kolom: 'text_news' dan '*hoax*'. *DataFrame* pertama berisi 2407 entri, dengan semua nilai di kolom 'text_news' dan '*hoax*' non-null. Kolom '*hoax*' bertipe int64, sedangkan 'text_news' bertipe *object*, dengan penggunaan memori sekitar 37.7 KB. *DataFrame* kedua berisi 27447 entri, juga dengan semua nilai di kolom 'text_news' dan 'hoax' non-null. Kolom '*hoax*' bertipe int64 dan 'text_news' bertipe *object*, dengan penggunaan memori sekitar 429.0 KB. Garis pemisah di antara informasi ini membantu membedakan detail masing-masing *DataFrame*.

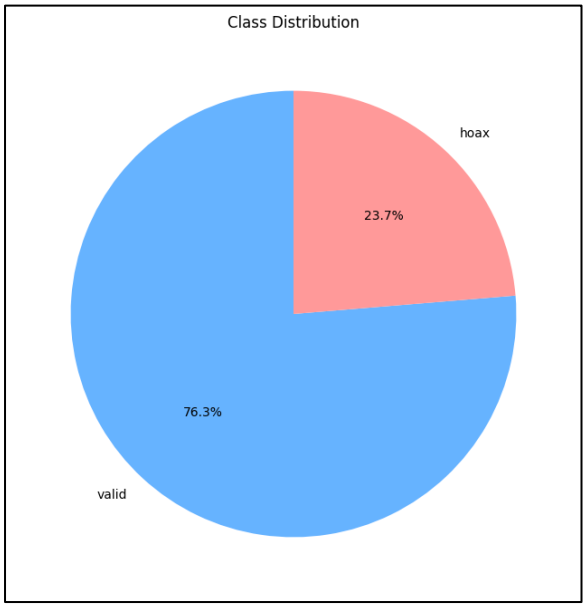
4.3. *Modify*

Sebelum data digunakan untuk melatih model, tahap *preprocessing* diperlukan, termasuk pembersihan data dari noise, normalisasi teks (mengubah semua teks menjadi huruf kecil, menghapus tanda baca, dan menghilangkan *stop words*), serta tokenisasi untuk mengubah teks menjadi format yang dapat diproses oleh model ML dan menangani ketidakseimbangan data dengan *undersampling* pada *dataset* berita politik.

```
# undersampling valid label
from sklearn.utils import resample

label_counts = df['hoax'].value_counts()

minority_class = label_counts.idxmin()
majority_class = label_counts.idxmax()

hoax_data = df[df['hoax'] == minority_class]
valid_data = df[df['hoax'] == majority_class]

valid_downsampled = resample(valid_data, replace=False, n_samples=label_counts[minority_class], random_state=42)

balanced_data = pd.concat([hoax_data, valid_downsampled])

balanced_data = balanced_data.sample(frac=1, random_state=42).reset_index(drop=True)

balanced_label_counts = balanced_data['hoax'].value_counts()
```

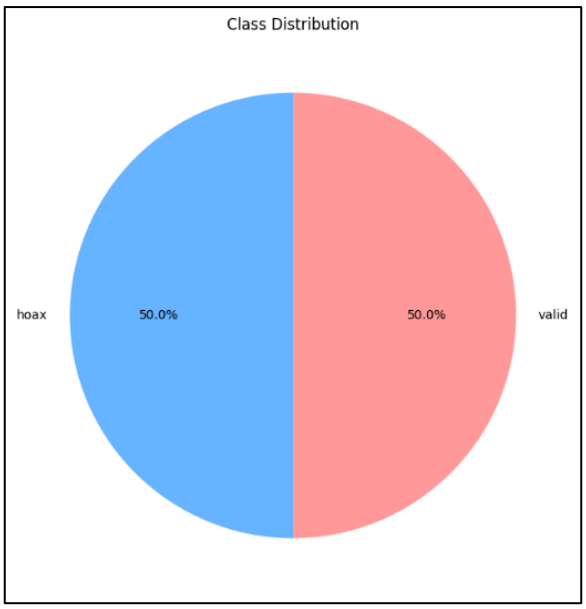
Gambar 7. Program *undersampling dataset*

Program pada Gambar 7. melakukan *undersampling* pada data untuk menyeimbangkan jumlah label antara dua kelas ('*hoax*' dan '*valid*') dalam *dataset*. Pertama, program menghitung jumlah setiap label di kolom '*hoax*' dan mengidentifikasi kelas minoritas (kelas dengan jumlah sampel lebih sedikit) dan kelas mayoritas (kelas dengan jumlah sampel lebih banyak). Data dengan label minoritas disimpan dalam variabel *hoax*_data dan data dengan label mayoritas disimpan dalam valid_data. Kemudian, kelas mayoritas di-resample (undersample) menggunakan fungsi resample dari scikit-learn untuk mendapatkan jumlah sampel yang sama dengan kelas minoritas. Data yang sudah di-*resample* ini digabungkan dengan data kelas minoritas untuk membentuk *dataset* seimbang yang baru, balanced_data. Data ini kemudian diacak (shuffle) dan indeksnya direset. Akhirnya, program menghitung dan menampilkan jumlah setiap label dalam *dataset* yang sudah seimbang.



Gambar 8. *Dataset* sebelum *undersampling*

Gambar 9 menunjukkan distribusi kelas dalam bentuk diagram lingkaran (*pie chart*) yang terbagi rata antara dua kategori: "*hoax*" dan "*valid*". Masing-masing kategori memiliki porsi yang sama, yaitu 50%. Hal ini diindikasikan dengan warna biru untuk kategori "*hoax*" dan warna merah muda untuk kategori "*valid*". Diagram ini menunjukkan bahwa data yang dianalisis memiliki distribusi yang seimbang antara kedua kelas tersebut.



Gambar 9. Distribusi kelas setelah *undersampling*

4.4. Model

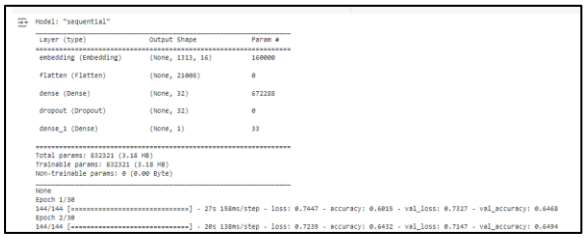
Setelah data siap, tahap selanjutnya adalah pemodelan. Penelitian ini menggunakan model *deep learning* untuk klasifikasi teks. *Model Shallow neural network* dikembangkan dengan TensorFlow dan Keras, dengan arsitektur yang dirancang untuk mengoptimalkan akurasi dalam mengklasifikasikan berita sebagai *hoax* atau fakta.

```
import tensorflow as tf
from tensorflow.keras.layers import Embedding, Dense, Dropout, Flatten
from tensorflow.keras.models import Sequential
from tensorflow.keras.callbacks import Callback
from tensorflow.keras.utils import plot_model

# Model definition
model = Sequential([
    Embedding(vocab_size, embedding_dim, input_length=max_length),
    Flatten(),
    Dense(32, activation='relu', kernel_regularizer=tf.keras.regularizers.l2(0.001)),
    Dropout(0.5),
    Dense(1, activation='sigmoid')
])
```

Gambar 10.Program *modelling*

Pada gambar 10. program ini mendefinisikan dan melatih model jaringan saraf sederhana (*Shallow Neural Network*) untuk klasifikasi teks biner menggunakan TensorFlow dan Keras. Model ini dimulai dengan lapisan *Embedding* untuk representasi vektor kata, diikuti dengan lapisan Flatten untuk meratakan input. Selanjutnya, terdapat dua lapisan Dense, lapisan pertama dengan 32 unit dan aktivasi ReLU serta regularisasi L2, dan lapisan kedua dengan satu unit dan aktivasi sigmoid untuk keluaran biner. Dropout digunakan untuk mencegah *overfitting*. Model ini dikompilasi menggunakan binary crossentropy sebagai fungsi loss, optimizer Adam dengan learning rate 0.00001, dan metrik akurasi. Callback khusus dibuat untuk menghentikan pelatihan dini jika akurasi pelatihan dan validasi melebihi 95%.

Model kemudian dilatih menggunakan data yang telah diproses dengan padding, selama maksimal 30 *epoch*, dengan ukuran batch 64, dan validasi dilakukan menggunakan data validasi yang terpisah. Diagram arsitektur model juga dihasilkan dan disimpan dalam file 'model_no8_plot.png'.



Gambar 11. *Output* arsitektur model

Pada gambar 11 tersebut menunjukkan *output* arsitektur dan hasil pelatihan dari sebuah *shallow neural network*. Model ini bernama "*sequential*" dan terdiri dari beberapa lapisan: lapisan embedding dengan output shape (None, 100, 50) dan 1600000 parameter, lapisan flatten dengan output shape (None, 5000), lapisan dense dengan 6732368 parameter, lapisan dropout dengan output shape (None, 32) dan 0 parameter, serta lapisan dense terakhir dengan output shape (None, 1) dan 33 parameter. Total parameter dari model ini adalah 8332701, yang semuanya adalah parameter yang dapat dilatih. Selanjutnya, terdapat log dari proses pelatihan selama 2 *epoch*, yang menunjukkan metrik akurasi dan loss untuk data training dan validation. Pada *epoch* pertama, akurasi training adalah 0.7437 dan validation adalah 0.8047, dengan loss masing-masing 5.7087 dan 5.7697. Pada *epoch* kedua, akurasi training meningkat menjadi 0.7583 dan validation menjadi 0.8086, dengan loss masing-masing 5.7198 dan 5.7421. Log ini memberikan gambaran bahwa model mengalami peningkatan performa seiring dengan bertambahnya *epoch*.

4.5. *Assess*

Setelah model dilatih, tahap evaluasi dilakukan untuk menilai kinerjanya. Metrik seperti akurasi, presisi, *recall*, dan F1-*score* digunakan untuk mengukur kemampuan model dalam mengidentifikasi berita hoax dan fakta. Berdasarkan hasil evaluasi, model dapat disempurnakan lebih lanjut untuk meningkatkan kinerjanya. Berikut adalah informasi yang diubah ke dalam bentuk tabel, disertai dengan pembagian data menjadi tiga set: pelatihan, validasi, dan pengujian.

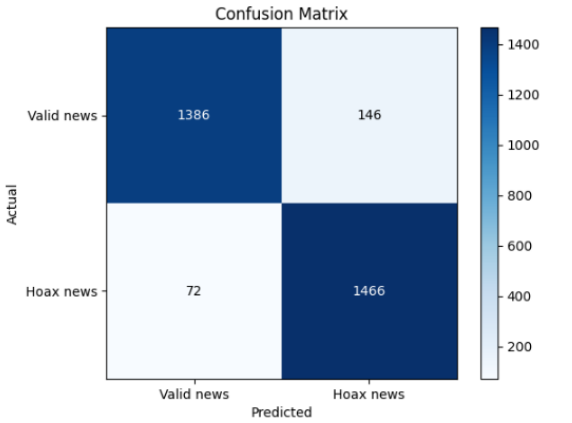
Tabel 1. Pembagian Data

Set Data	Persentase	Jumlah Data
Pelatihan	60%	9207
Validasi	20%	3069
Pengujian	20%	3070

```
from sklearn.metrics import confusion_matrix
# Assuming y_pred_classes and y_test_nump are defined
conf_matrix = confusion_matrix(y_test_nump, y_pred_classes_row_array)
# Display the confusion matrix
print("Confusion Matrix:")
print(conf_matrix)
# trial: FP increase 50 >> 140 >> 200
```

Gambar 12. Program evaluasi model

Kode ini menghitung dan menampilkan matriks kebingungan (confusion matrix) menggunakan scikit-learn. Pertama, modul confusion_matrix diimpor dari sklearn.metrics. Dengan asumsi bahwa y_pred_classes_row_array (prediksi kelas) dan y_test_nump (label sebenarnya) sudah didefinisikan sebelumnya, fungsi confusion_matrix digunakan untuk menghitung matriks kebingungan, yang kemudian disimpan dalam variabel conf_matrix. Matriks kebingungan ini menunjukkan jumlah prediksi benar dan salah untuk masing-masing kelas dalam bentuk tabel. Setelah dihitung, matriks kebingungan dicetak ke layar dengan label "Confusion Matrix:". Catatan tambahan menunjukkan percobaan peningkatan *False Positives* (FP) dari 50 menjadi 140 dan kemudian menjadi 200



Gambar 13. *Output* evaluasi model

Gambar 13 menunjukkan matriks kebingungan (*confusion matrix*) dari sebuah model klasifikasi yang memprediksi apakah berita adalah *valid* atau *hoax*. Sumbu y menunjukkan kelas aktual (*Valid news* dan *Hoax news*), sedangkan sumbu x menunjukkan kelas prediksi (*Valid news* dan *Hoax news*). Dari matriks ini, kita bisa melihat bahwa model mengklasifikasikan 1386 berita *valid* sebagai *valid* (*True Positives*) dan 146 berita *valid* sebagai *hoax* (*False Negatives*). Sebaliknya, 1466 berita *hoax* diklasifikasikan sebagai *hoax* (*True Negatives*) dan 72 berita *hoax* diklasifikasikan sebagai *valid* (*False Positives*). Warna dan skala di sebelah kanan menunjukkan jumlah kasus dalam setiap sel matriks. Matriks ini menunjukkan bahwa model memiliki kinerja yang baik dengan kesalahan yang relatif sedikit.

Metrik evaluasi digunakan untuk menilai kinerja model berdasarkan hasil dari *confusion matrix*. Berikut adalah perhitungan untuk beberapa metrik utama yaitu

a. *Accuracy*, proporsi prediksi yang benar dari total prediksi. Rumusnya adalah

$$\frac{TP+TN}{TP+TN+FP+FN}$$

Dengan nilai :

$TP = 1466, TN = 1386, FP = 146, FN = 72$

Maka :

$$Akurasi = \frac{1466 + 1386}{1466 + 1386 + 146 + 72} = \frac{2852}{3070} = 0.93$$

Akurasi sebesar 93% menunjukkan bahwa model *shallow neural network* mampu membuat prediksi yang benar sebanyak 93% dari seluruh data yang diuji. Ini berarti dari 3070 instance (berita) yang diuji, model membuat 2852 prediksi yang benar.

b. Precision, mengukur proporsi prediksi positif yang benar. Presisi dihitung sebagai:

$$Presisi = \frac{TP}{TP+FP}$$

Untuk kelas "Hoax News" :

$$Presisi = \frac{1466}{1466+146} = 0.91$$

Untuk kelas "Valid News" :

$$Presisi = \frac{1386}{1386+72} = 0.95$$

Penjelasan :

Presisi untuk "*Hoax news*" sebesar 91% menunjukkan bahwa dari semua prediksi yang mengatakan berita adalah *hoax*, 91% di antaranya benar-benar *hoax*. Presisi untuk "*Valid news*" sebesar 95% menunjukkan bahwa dari semua prediksi yang mengatakan berita adalah *valid*, 95% di antaranya benar-benar *valid*.

c. Recall, mengukur proporsi kasus positif yang benar-benar terdeteksi. *Recall* dihitung sebagai:

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

Untuk kelas "*Hoax news*":

$$Recall = \frac{1466}{1466+72} = 0.95$$

Untuk kelas "Valid news":

$$Recall = \frac{1386}{1386 + 146} = 0.90$$

Penjelasan :

Recall untuk "Hoax news" sebesar 95% menunjukkan bahwa dari semua berita yang benar-benar *hoax*, 95% di antaranya terdeteksi dengan benar oleh model. *Recall* untuk "*Valid news*" sebesar 90% menunjukkan bahwa dari semua berita yang benar-benar *valid*, 90% di antaranya terdeteksi dengan benar oleh model.

Tabel 2. Evaluasi Model

	Precision	**Recall**	**F1-Score**	**Support**
0	0.95	0.90	0.93	1532
1	0.91	0.95	0.93	1538
Accuracy			0.93	3070
Macro AVG	0.93	0.93	0.93	3070
Weighted AVG	0.93	0.93	0.93	3070

Data dibagi menjadi tiga set: pelatihan (60%), validasi (20%), dan pengujian (20%). Proses pembagian dimulai dengan memisahkan data menjadi set pelatihan (60%) dan set sementara (40%). Set sementara ini kemudian dibagi lagi menjadi set validasi dan set pengujian, masing-masing mengambil 50% dari set sementara (20% dari total data). Akhirnya, set pelatihan berukuran 9207, set validasi 3069, dan set pengujian 3070. Pembagian ini bertujuan untuk memastikan bahwa model dilatih dengan data yang cukup untuk mempelajari pola, divalidasi dengan data terpisah untuk menyesuaikan hiperparameter dan mencegah *overfitting*, serta diuji dengan data yang tidak pernah dilihat sebelumnya untuk memberikan evaluasi kinerja yang objektif dan akurat. Evaluasi kinerja model dilakukan dengan metrik seperti *precision*, *recall*, dan *F1-score* untuk kedua kelas (0 dan 1), serta akurasi keseluruhan. Hasil evaluasi menunjukkan *precision*, *recall*, dan *F1-score* yang seimbang antara kedua kelas, dengan nilai makro dan *weighted average* yang konsisten, menandakan model memiliki kinerja yang baik dalam mengklasifikasikan data sebagai *hoax* atau fakta.

5. KESIMPULAN DAN SARAN

Kesimpulannya, model *deep learning* dengan arsitektur *shallow neural network* yang dikembangkan mampu mengklasifikasikan berita sebagai *hoax* atau fakta dengan akurasi, *precision*, *recall*, dan *F1-score* yang seimbang antara kedua kelas, menunjukkan kinerja yang baik. Pembagian data yang tepat memastikan bahwa model diuji dengan data yang tidak digunakan selama pelatihan, memberikan evaluasi kinerja yang akurat. Saran untuk penelitian selanjutnya adalah mencoba arsitektur model yang lebih kompleks, seperti *deep neural networks* atau *transformer-based models*, untuk melihat apakah dapat meningkatkan kinerja lebih lanjut. Selain itu, penggunaan teknik augmentasi data atau metode pengurangan bias dapat dieksplorasi untuk memperbaiki hasil pada *dataset* yang mungkin tidak seimbang atau memiliki variasi yang lebih besar.

DAFTAR PUSTAKA

[1] B. H. K. Kominfo, "Hingga Akhir Tahun 2023, Kominfo Tangani 12.547 Isu Hoaks," Kominfo. Accessed: Jan. 25, 2024. [Online]. Available: https://www.kominfo.go.id/content/detail/53899/siaran-pers-no-02hmkominfo012024-tentang-hingga-akhir-tahun-2023-kominfo-tangani-12547-isu-hoaks/0/siaran_pers

[2] B. H. K. Kominfo, "Jaga Ruang Digital, Menkominfo: Kami Tangani 203 Isu Hoaks Pemilu 2024," Kominfo. Accessed: Apr. 01, 2024. [Online]. Available: https://www.kominfo.go.id/content/detail/53920/siaran-pers-no-03hmkominfo012024-tentang-jaga-ruang-digital-menkominfo-kami-tangani-203-isu-hoaks-pemilu-2024/0/siaran_pers

[3] P. Yang, H. Zhou, Y. Zhu, L. Liu, and L. Zhang,

"Malware classification based on shallow neural network," *Futur. Internet*, vol. 12, no. 12, pp. 1–17, 2020, doi: 10.3390/fi12120219.

[4] S. F. Yeremia, A. Voutama, and A. A. Hendriadi, "MENGGUNAKAN ALGORITMA NAÏVE BAYES," vol. 8, no. 2, pp. 2112–2115, 2024.

[5] B. Imran, M. N. Karim, and N. I. Ningsih, "Klasifikasi Berita Hoax Terkait Pemilihan Umum Presiden Republik Indonesia Tahun 2024 Menggunakan Naïve Bayes Dan Svm," *Din. Rekayasa*, vol. 20, no. 1, pp. 1–9, 2024, doi: 10.20884/1.dinarek.2024.20.1.27.

[6] E. A. Sosiawan and R. Wibowo, "Kontestasi Berita Hoax Pemilu Presiden Tahun 2019 di Media Daring dan Media Sosial," *J. Ilmu Komun.*, vol. 17, no. 2, p. 133, 2020, doi: 10.31315/jik.v17i2.3695.

[7] A. S. Nurhikam, R. Syaputra, S. Rohman, and ..., "Deteksi Berita Palsu Pada Pemilu 2024 Dengan Menggunakan Algoritma Random Forest," *DoubleClick J. Comput. Inf. Technol.*, vol. 7, no. 1, pp. 41–50, 2023, [Online]. Available: http://e-journal.unipma.ac.id/index.php/doubleclick/article/view/15456%0Ahttp://e-journal.unipma.ac.id/index.php/doubleclick/article/download/15456/5393

[8] A. Nasir, et, "ANALISIS SENTIMEN ARTIKEL BERITA PEMILU BERBASIS METODE KLASIFIKASI THESIS Oleh : FATHIR 200605220005 PROGRAM STUDI MAGISTER INFORMATIKA FAKULTAS SAINS DAN TEKNOLOGI UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG 2023," vol. 9, pp. 356–363, 2023.

[9] H. Allcott and M. Gentzkow, "Social Media and Fake News in the 2016 Election," vol. 31, no. 2, pp. 211–236, 2017.

[10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[11] D. Jurafsky and J. H. Martin, "Speech and language processing (draft)," *Chapter A Hidden Markov Model. (Draft Sept. 11, 2018). Retrieved March*, vol. 19, p. 2019, 2018.

[12] A. Géron, "Hands-on machine learning with Scikit-Learn," *Keras, TensorFlow Concepts, tools, Tech. to build Intell. Syst.*, vol. 1, 2019.

[13] Y. A. Suwitono and F. J. Kaunang, "Implementasi Algoritma Convolutional Neural Network (CNN) Untuk Klasifikasi Daun Dengan Metode Data Mining SEMMA Menggunakan Keras," *J. Komtika (Komputasi dan Inform.*, vol. 6, no. 2, pp. 109–121, 2022, doi: 10.31603/komtika.v6i2.8054.