DETEKSI BERITA HOAX DENGAN PENDEKATAN LEXICON BASED DAN LSTM

THESIS

Oleh :
EDWIN HARI AGUS PRASTYO
NIM. 220605220005

PROGRAM STUDI MAGISTER INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2024

DETEKSI BERITA HOAX DENGAN PENDEKATAN LEXICON BASED DAN LSTM

THESIS

Diajukan kepada:
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Magister Komputer (M.Kom)

Oleh :

EDWIN HARI AGUS PRASTYO

NIM. 220605220005

PROGRAM STUDI MAGISTER INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2024

DETEKSI BERITA HOAX DENGAN PENDEKATAN LEXICON BASED DAN LSTM

THESIS

Oleh :
EDWIN HARI AGUS PRASTYO
NIM. 220605220005

Telah diperiksa dan disetujui untuk di uji:
Tanggal: 12 November 2024

Pembimbing I,	Pembimbing II,
Dr. M. Faisal, M.T	Prof. Dr. Suhartono, M.Kom
NIP. 19740510 200501 1 007	NIP. 19680519 200312 1 001

Mengetahui,
Ketua Program Studi Magister Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

DETEKSI BERITA HOAX DENGAN PENDEKATAN LEXICON BASED DAN LSTM

THESIS

Oleh :
EDWIN HARI AGUS PRASTYO
NIM. 220605220005

Telah Dipertahankan di Depan Dewan Penguji Thesis
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Magister Komputer (M.Kom)
Tanggal: 12 November 2024

Susunan Dewan Penguji		**Tanda Tangan**
Penguji I	: Dr. M. Ainul Yaqin, M.Kom NIP. 197610132006041004	( )
Penguji II	: Dr. Totok Chamidy, M.Kom NIP. 19691222 200604 1 001	( )
Pembimbing I	: Dr. M. Faisal, M.T NIP. 19740510 200501 1 007	( )
Pembimbing II	: Prof. Dr. Suhartono, M.Kom NIP. 19680519 200312 1 001	( )

Mengetahui dan Mengesahkan,
Ketua Program Studi Magister Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama	: EDWIN HARI AGUS PRASTYO
NIM	: 220605220005
Program Studi	: Magister Informatika
Fakultas	: Sains dan Teknologi
Judul Tesis	: DETEKSI BERITA HOAX DENGAN PENDEKATAN LEXICON BASED DAN LSTM

Menyatakan dengan sebenarnya bahwa Thesis yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan Thesis ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 12 November 2024

Yang membuat pernyataan,

EDWIN HARI AGUS PRASTYO

NIM. 220605220005

MOTTO

“Meniti Jalan Pengetahuan, Menuju Kebenaran″

KATA PENGANTAR

Puji syukur kepada Allah SWT yang telah melimpahkan karunia, rahmat, dan hidayah-Nya sehingga penulis dapat menyelesaikan laporan Thesis yang berjudul **"DETEKSI BERITA HOAX DENGAN PENDEKATAN LEXICON BASED DAN LSTM"** Laporan Thesis ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Magister Komputer di Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Penulis menyadari bahwa selama menyelesaikan tesis ini, banyak sekali bantuan, dukungan, dan saran dari semua pihak, baik secara langsung maupun tidak langsung, selama proses penyusunan hingga selesai. Oleh karena itu, penulis ingin mengucapkan terima kasih dan hormat yang sebesar-besarnya kepada:

1. **Bapak Dr. Muhammad Faisal, M.T.** selaku Dosen Pembimbing I dan **Bapak Prof. Dr. Suhartono, M.Kom.** selaku Dosen Pembimbing II atas motivasi, arahan, serta pengalaman berharga yang diberikan kepada penulis.
2. Seluruh dosen Program Studi Magister Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang yang telah dengan tiada lelah memberikan ilmunya kepada penulis.
3. **Kedua orang tua** (Bapak Eko Suharsono dan Ibu Sri Narmi) serta keluarga besar yang telah memberikan dukungan, perhatian, nasihat, motivasi, dan doa yang tiada henti kepada penulis untuk menyelesaikan tesis ini.
4. Seluruh rekan mahasiswa angkatan ke-7 Program Studi Magister Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang yang telah memberikan banyak pengalaman, saran, dan kritik yang membangun pada tesis ini.
5. Semua pihak yang tidak dapat penulis sebutkan satu per satu, yang telah membantu dalam menyelesaikan tesis ini.

Harapan dan doa penulis, semoga semua kebaikan dan jasa dari pihak-pihak yang telah membantu dalam penyelesaian tesis ini diterima oleh Allah SWT, serta mendapat balasan yang lebih baik dan berlipat ganda. Penulis menyadari bahwa

tesis ini masih jauh dari sempurna, baik dari segi materi maupun penyajian. Oleh karena itu, saran dan kritik yang membangun sangat diharapkan demi penyempurnaan tesis ini. Penulis berharap tesis ini dapat memberikan manfaat dan menambah wawasan bagi semua pihak.

Malang, 01 November 2024
Penulis,

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Edwin, 2024. **DETEKSI BERITA HOAX DENGAN PENDEKATAN LEXICON BASED DAN LSTM.** Thesis, Program Studi Magister Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Mualana Malik Ibrahim Malang. Pembimbing: (I) Dr. Muhammad Faisal, M.T. (II) Prof. Dr. Suhartono, M.Kom**.**

Kata Kunci: Deteksi Hoaks, Leksikon, LSTM, Pemrosesan Bahasa Alami

Di era digital, penyebaran berita hoaks menjadi isu yang signifikan, dengan dampak merugikan pada sosial, ekonomi, dan politik. Penelitian ini bertujuan untuk mengembangkan model deteksi berita hoaks menggunakan pendekatan hybrid berbasis leksikon dan Long Short-Term Memory (LSTM). Metode ini menggabungkan kekuatan leksikon dalam mengidentifikasi kata-kata kunci dengan kemampuan LSTM dalam menganalisis konteks dan pola teks secara mendalam. Hasil penelitian menunjukkan pendekatan hybrid ini efektif, dengan akurasi deteksi tertinggi mencapai 99%, terutama pada kategori berita politik. Pengaruh pengaturan hyperparameter, seperti "Max Features" 15,000 dan "Batch Size" 64, serta penggunaan optimizer ADAM, memberikan keseimbangan antara akurasi (0,9900) dan efisiensi waktu (209,24 detik). Namun, performa menurun pada berita olahraga, menyoroti pentingnya leksikon yang spesifik untuk tiap jenis konten. Penelitian ini memberikan kontribusi penting pada literatur deteksi hoaks dan merekomendasikan pengembangan aplikasi berbasis model ini untuk implementasi praktis, serta perluasan leksikon untuk meningkatkan akurasi lintas kategori konten.

ABSTRACT

Edwin, 2024. **HOAX NEWS DETECTION WITH LEXICON BASED APPROACH AND LSTM**. Thesis, Master of Informatics Study Programme, Faculty of Science and Technology, State Islamic University Mualana Malik Ibrahim Malang. Supervisor: (I) Dr Muhammad Faisal, M.T. (II) Prof. Dr Suhartono, M.Kom.

Keywords: Hoax Detection, Lexicon, LSTM, Natural Language Processing

In the digital era, the spread of fake news has become a significant issue, with adverse social, economic, and political impacts. This research aims to develop a hoax news detection model using a hybrid approach based on lexicon and Long Short-Term Memory (LSTM). This method combines the power of lexicon in identifying key words with the ability of LSTM in analysing text context and patterns in depth. The results show that this hybrid approach is effective, with the highest detection accuracy reaching 99%, especially in the political news category. The effect of hyperparameter settings, such as 'Max Features' 15,000 and 'Batch Size' 64, as well as the use of the ADAM optimiser, provides a balance between accuracy (0.9900) and time efficiency (209.24 seconds). However, performance degraded for sports news, highlighting the importance of a content-specific lexicon. This research makes an important contribution to the hoax detection literature and recommends developing applications based on this model for practical implementation, as well as expanding the lexicon to improve accuracy across content categories.

الملخص

،أطروحة .**lstm** إدوين، 2024. **الكشف عن الأخبار الخادعة باستخدام النهج القائم على المعجم و** برنامج دراسة الماجستير في المعلوماتية، كلية العلوم والتكنولوجيا، الجامعة الإسلامية الحكومية مولانا مالك .إبراهيم مالانج. المشرف: (أولاً) الدكتور محمد فيصل، م. ت. (ثانيًا) الأستاذ الدكتور سوهارتونو، م. كوم

.في العصر الرقمي، أصبح انتشار الأخبار الكاذبة مشكلة كبيرة لها آثار اجتماعية واقتصادية وسياسية سلبية يهدف هذا البحث إلى تطوير نموذج للكشف عن الأخبار الكاذبة باستخدام نهج هجين يعتمد على المعجم وتجمع هذه الطريقة بين قوة المعجم في تحديد الكلمات الرئيسية .(LSTM) والذاكرة قصيرة المدى الطويلة ،وقدرة الذاكرة طويلة المدى على تحليل سياق النص وأنماطه بعمق. تُظهر النتائج أن هذا النهج الهجين فعّال حيث وصلت أعلى دقة في الكشف إلى 99%، خاصة في فئة الأخبار السياسية. ويوفر تأثير إعدادات المعلمة الفائقة، مثل "الحد الأقصى للميزات" 15000 و"حجم الدُفعات" 64، بالإضافة إلى استخدام مُحسِّن توازنًا بين الدقة (0.9900) وكفاءة الوقت (209.24 ثانية). ومع ذلك، تدهور الأداء بالنسبة ،ADAM للأخبار الرياضية، مما يسلط الضوء على أهمية وجود معجم خاص بالمحتوى. يقدم هذا البحث مساهمة ،مهمة في أدبيات الكشف عن الخدع ويوصي بتطوير تطبيقات تستند إلى هذا النموذج للتطبيق العملي .بالإضافة إلى توسيع المعجم لتحسين الدقة عبر فئات المحتوى

معالجة اللغة الطبيعية ،LSTM ،**الكلمات المفتاحية**: كشف الخداع، معجم، معجم

BAB I
PENDAHULUAN

1.1.Latar Belakang

Di era digital saat ini, informasi dengan mudah tersebar melalui internet di situs berita online maupun di media sosial. Berita hoaks menjadi masalah global, termasuk di Indonesia, dengan dampak seperti konflik sosial, keresahan publik, hingga ancaman keselamatan. Data Kominfo mencatat 11.357 isu hoaks tersebar antara Agustus 2018 dan Maret 2023 (Kominfo, t.t.), menunjukkan bahwa hoaks masih menjadi ancaman serius. Dampaknya meliputi penurunan kepercayaan pada pemerintah dan media, polarisasi politik, serta penyebaran informasi keliru terkait kesehatan. Selain keresahan sosial, hoaks juga menyebabkan kerugian ekonomi dan mengancam keamanan.Oleh karena itu, diperlukan langkah-langkah strategis untuk mengatasi penyebaran berita palsu ini, seperti literasi digital yang lebih kuat dan regulasi yang lebih tegas. Dalam konteks penyebaran berita hoax di Indonesia melalui situs berita online, terdapat beberapa faktor yang mempengaruhinya. Faktor-faktor tersebut antara lain mudahnya akses internet, rendahnya literasi digital di masyarakat, dan kebiasaan menyebarkan informasi tanpa verifikasi terlebih dahulu. Situasi ini menciptakan lingkungan di mana berita hoax dapat dengan mudah menyebar dan berdampak negatif signifikan. Penelitian tentang berita hoax semakin relevan dalam konteks modern.

Studi-studi terkait telah menyoroti pentingnya deteksi berita hoax dengan pendekatan berbasis teknologi dan juga berlandaskan pada nilai-nilai agama, seperti yang dijelaskan dalam Al Quran dan Ayat yang relevan dalam Al-Qur'an terkait dengan **berita bohong** atau **hoax** dan bagaimana hal ini digunakan oleh orang-orang munafik untuk tujuan buruk dapat ditemukan dalam **Surah An-Nur (24:11-12)**. Ayat ini menceritakan peristiwa fitnah yang terjadi pada masa Nabi Muhammad SAW, yang dikenal sebagai **Peristiwa Haditsul Ifk** (kisah dusta), di mana orang-orang munafik menyebarkan berita bohong tentang Aisyah RA, istri Nabi Muhammad SAW.

Surah An-Nur (24:11):

إِنَّ الَّذِينَ جَاءُوا بِالْإِفْكِ عُصْبَةٌ مِّنكُمْ ۚ لَا تَحْسَبُوهُ شَرًّا لَّكُم بَلْ هُوَ خَيْرٌ لَّكُمْ ۚ لِكُلِّ امْرِئٍ مِّنْهُم مَّا اكْتَسَبَ مِنَ الْإِثْمِ ۚ وَالَّذِي تَوَلَّىٰ كِبْرَهُ مِنْهُمْ لَهُ عَذَابٌ عَظِيمٌ

Artinya : *"Sesungguhnya* orang-orang yang membawa berita bohong itu adalah dari golongan kamu juga. Janganlah kamu kira bahwa berita bohong itu buruk bagi kamu, bahkan itu adalah baik bagi kamu. Tiap-tiap seseorang dari mereka mendapat balasan dari dosa yang dikerjakannya, dan siapa di antara mereka yang mengambil bagian terbesar dalam penyiaran berita bohong itu, baginya azab yang besar.*"*

Surah An-Nur (24:12):

لَّوْلَا إِذْ سَمِعْتُمُوهُ ظَنَّ الْمُؤْمِنُونَ وَالْمُؤْمِنَاتُ بِأَنفُسِهِمْ خَيْرًا وَقَالُوا هَٰذَا إِفْكٌ مُّبِينٌ

Artinya: *"Mengapa di waktu kamu mendengar berita bohong itu orang-orang mukminin dan mukminat tidak bersangka baik terhadap diri mereka sendiri, dan (mengapa tidak) berkata: 'Ini adalah suatu berita bohong yang nyata'?"*

Relevansi dengan Konsep Hoax, Ayat ini menunjukkan bahwa dalam Islam, menyebarkan berita bohong (hoax) adalah perbuatan tercela yang identik dengan sifat orang munafik. Mereka menggunakan kebohongan sebagai alat untuk merusak kehormatan, menciptakan fitnah, dan melancarkan niat

buruk terhadap orang lain. Oleh karena itu, umat Muslim diingatkan untuk berhati-hati terhadap informasi yang beredar dan selalu melakukan verifikasi sebelum mempercayai atau menyebarkannya.

Dalam konteks modern, di mana hoax dan berita palsu sering beredar, terutama di media sosial, ayat ini sangat relevan. Prinsip yang dapat dipetik adalah pentingnya *tabayyun* (verifikasi) sebelum menyebarkan berita serta menjaga kejujuran dan kehati-hatian dalam berkomunikasi, agar tidak terjerumus dalam perilaku yang menyerupai orang munafik yang menyebarkan fitnah

Penyebaran informasi palsu di situs berita dapat dijelaskan oleh berbagai teori terkait. Awalnya, teori propaganda menganggap berita hoax sebagai jenis propaganda yang dirancang untuk mempengaruhi opini publik. Berita hoax sering digunakan untuk menyebarkan alur cerita yang menjunjung tinggi agenda tertentu, apakah itu politik, sosial, atau ekonomi. Selain itu, pentingnya teori media di ranah situs berita sangat penting karena peran penting yang dimainkan platform berita online dalam menyebarkan berita hoax. Sifat yang cepat dan mudah diakses dari platform ini memfasilitasi penyebaran berita palsu yang cepat di antara audiens. Selain itu, elemen algoritmik di situs berita dapat mempercepat proliferasi berita hoax dengan menyajikan berita kontroversial atau merangsang kepada pengguna. Dalam teori psikologi kognitif telah membuat dampak besar. Metode yang lebih efektif dalam deteksi berita hoax telah ditunjukkan melalui penggunaan metode pembelajaran yang melibatkan teknik-teknik pembelajaran mesin. Dalam

konteks ini, metode pembelajaran yang diawasi (supervised learning) telah terbukti lebih unggul dalam mendeteksi berita hoax dibandingkan dengan pendekatan pembelajaran yang tidak diawasi (unsupervised learning) (Dong dkk., 2020).

Penelitian oleh (Shu dkk., 2020) mengatasi masalah deteksi berita hoax dengan memanfaatkan informasi jaringan sosial untuk meningkatkan akurasi deteksi. Mereka mengembangkan model yang memadukan analisis konten dengan pola penyebaran berita di media sosial disebut juga dengan basis Lexicon dengan mendeteksi pola-pola yang terindikasi Hoax. Hasil penelitian menunjukkan bahwa memanfaatkan informasi jaringan sosial dapat secara signifikan meningkatkan kinerja deteksi hoax dibandingkan dengan metode berbasis konten atau sentimen analisis, (Balshetwar dkk., 2023) model ini memerlukan akses ke data jaringan sosial, memerlukan komputasi yang tinggi dan waktu pelatihan yang cukup, yang mungkin tidak selalu tersedia atau mudah diakses.

Seperti di Penelitian oleh (Zhi dkk., 2021) mengatasi tantangan dalam mendeteksi berita hoax dengan menggunakan arsitektur transformer yang lebih canggih. Mereka mengusulkan model FakeBERT yang menggunakan BERT dan LSTM yang telah diadaptasi untuk tugas deteksi hoax.

Penelitian sebelumnya telah menunjukkan potensi, namun masih menghadapi kendala dalam hal akurasi dan kecepatan deteksi. Pendekatan berbasis leksikon dan LSTM menawarkan peluang untuk mengatasi beberapa kelemahan ini. Deteksi hoax dengan metode berbasis leksikon dan Long Short-

Term Memory (LSTM) sangat penting karena kedua pendekatan ini saling melengkapi dalam mengidentifikasi dan menganalisis berita palsu. Metode berbasis leksikon akurat dalam mendeteksi kata-kata atau frasa spesifik yang sering muncul dalam berita hoax, sehingga memungkinkan identifikasi cepat terhadap potensi misinformasi. Di sisi lain, LSTM mampu menangkap konteks dan pola dalam teks secara lebih mendalam dan berurutan, yang sering kali tidak dapat dikenali oleh metode berbasis leksikon saja. Kombinasi ini dapat meningkatkan akurasi dan efisiensi dalam mendeteksi berita hoax, yang sangat diperlukan mengingat volume besar informasi yang tersebar di platform media sosial (Agrawal dkk., 2021) (Khanam dkk., 2021);

Penelitian berikutnya, seperti yang dilakukan oleh Kurniawan dan Mustikasari (Kurniawan & Mustikasari, 2021)dalam "Implementasi Deep Learning Menggunakan Metode CNN dan LSTM untuk Menentukan Berita Palsu dalam Bahasa Indonesia," telah menunjukkan bahwa metode CNN dan LSTM berhasil diterapkan untuk menentukan berita fakta dan berita palsu dalam bahasa Indonesia dengan akurasi mencapai 88% untuk CNN dan 84% untuk LSTM.

Dari penelitian sebelumnya penelitian itu belum mencapai performa akutasi yang sempurna, karena itu penelitian ini bertujuan mengembangkan model deteksi berita hoaks yang akurat, konsisten, cepat, dan adaptif terhadap perubahan pola dengan mengintegrasikan metode berbasis leksikon dan LSTM. Metode leksikon membantu mengidentifikasi kata atau frasa khas berita hoaks, sementara LSTM menangkap konteks dan pola teks secara

mendalam. Maka Peneliti mengambil judul Deteksi Berita Hoax dengan pendekatan Lexicon Based dan LSTM, peneliti juga akan melakukan serangkaian eksperimen pada model untuk mengoptimalkan kinerjanya, sehingga menghasilkan akurasi yang lebih tinggi dalam mendeteksi berita hoaks. Dengan pendekatan ini, model diharapkan menjadi langkah strategis dalam melawan disinformasi yang semakin kompleks di era digital.

1.2.Rumusan Masalah

Bagaimana mencegah penyebaran informasi yang tidak benar dengan pendekatan lexicon based dan LSTM?

1.3.Tujuan Penelitian

Berdasarkan penjelasan latar belakang serta rumusan masalah di atas adalah mendeteksi informasi yang tidak benar atau berita hoax dengan pendekatan lexicon based dan LSTM.

1.4.Batasan Masalah

Ruang lingkup permasalahan dalam penelitian ini akan dibatasi sebagai berikut:.

1. **Dataset**: Dataset yang digunakan terbatas pada berita berbahasa Indonesia yang telah diklasifikasikan sebagai hoaks atau non-hoaks dalam rentang waktu 2019-2023. Label hoaks diambil dari TurnbackHoax.id, sedangkan label non-hoaks diperoleh dari situs berita Detik dan Kompas
2. **Data Heterogen**: Data berasal dari satu jenis sumber platform situs berita, yaitu situs berita, dalam topik bidang politik dan olahraga.

1.5.Manfaat penelitian:

Output penelitian ini diharapkan bermanfaat sebagai berikut:

1. **Pengembangan Layanan Berita Publik**: Memberikan kontribusi bagi peningkatan sistem deteksi berita hoax di lingkungan KOMDIGI.
2. **Masukan untuk Kebijakan Keterbukaan Informasi**: Menyediakan rekomendasi bagi pemerintah dalam merumuskan kebijakan keterbukaan informasi dan pengendalian berita hoax.
3. **Edukasi Masyarakat**: Mendukung literasi masyarakat dalam membedakan berita hoax melalui sistem verifikasi yang lebih akurat.
4. **Pengembangan Standar Verifikasi Informasi**: Menjadi acuan bagi standar verifikasi informasi untuk institusi berita dan pemerintahan.
5. **Dasar Riset Lanjutan**: Menyediakan data dan temuan untuk riset lebih lanjut dalam pengembangan AI dan deteksi hoax berbasis pemrosesan bahasa alami khusunya Bahasa indonesia.

1.6.Lingkup penelitian:

Penelitian ini berfokus pada analisis deteksi berita hoax dengan menggunakan pendekatan lexicon based dan LSTM. Data yang digunakan dalam penelitian ini adalah teks berita dari berbagai sumber online. Penelitian ini tidak mencakup analisis deteksi berita hoax dengan menggunakan pendekatan lain, seperti analisis visual atau analisis audio.

BAB II
STUDI PUSTAKA

2.1 Penelitian Terkait

Dalam bagian ini, literatur ilmiah yang berkaitan dengan identifikasi informasi palsu melalui metodologi Berbasis Lexicon dan LSTM akan diperiksa secara menyeluruh. Tujuan dari tinjauan ini adalah untuk menawarkan analisis komprehensif tentang kemajuan dan keterbatasan teknik masa lalu yang digunakan untuk memerangi penyebaran informasi palsu di era komunikasi digital.

Penelitian yang dilakukan oleh Shrutika Jadhav dan Sudeep Thepade (Jadhav & Thepade, 2019) pada tahun 2019 bertujuan untuk mendeteksi dan mengklasifikasikan berita palsu menggunakan model Deep Structured Semantic Models (DSSM) yang ditingkatkan serta model Recurrent Neural Network (RNN). Tantangan utama dalam penelitian ini adalah kompleksitas dalam mendeteksi berita palsu yang memerlukan pengumpulan dan perbandingan informasi berita secara akurat. Mereka mencatat bahwa pengklasifikasi dasar kurang mampu menangkap semantik yang diperlukan untuk identifikasi berita palsu, ditambah dengan ketidakcukupan data berita palsu yang menghambat peningkatan deteksi. Metodologi penelitian mereka mencakup penggunaan model Naive Bayes, Clustering, dan pendekatan gabungan Pohon Keputusan untuk deteksi, serta model pembelajaran mendalam yang memanfaatkan kumpulan data visual dan tekstual.

Studi yang dilakukan oleh (Meel & Vishwakarma, 2021) Priyanka Meel dan Dinesh Kumar pada tahun 2021 bertujuan untuk mengembangkan ConvNet semi-diawasi untuk deteksi berita palsu dengan akurasi tinggi. Tantangan utama

dalam bidang ini termasuk informasi yang menyesatkan di media sosial serta kesulitan dalam membedakan pendapat dari berita factual. Pendekatan mereka melibatkan metode berbasis fitur yang menganalisis difusi berita palsu, perilaku sosial, dan gaya penulisan. Metodologi penelitian ini menggabungkan kerangka kerja ConvNet yang mengintegrasikan informasi linguistik, serta menerapkan analisis teks dan gambar untuk deteksi berita palsu multimodal. Hasil penelitian menunjukkan bahwa model mereka mencapai akurasi klasifikasi berita palsu sebesar 95,45% dengan menggunakan artikel berlabel 50%. Studi ini juga menyoroti pentingnya peningkatan data berlabel untuk stabilitas kinerja model. Meskipun demikian, keterbatasan termasuk kurangnya analisis multimedia dan penilaian kredibilitas sumber. Dataset yang digunakan mencakup 3988 entri dalam kategori Deteksi Berita Palsu dan 20.700 entri dalam kategori Data Berita Palsu.

Studi yang dilakukan oleh (Nasir dkk., 2021) Jamal Nasir, Osama Subhani Khan, dan Iraklis Varlamis pada tahun 2021 bertujuan untuk mengembangkan model pembelajaran mendalam hibrida untuk klasifikasi berita palsu. Tantangan utama dalam bidang ini meliputi informasi yang salah, disinformation, serta penyebaran berita yang cepat melalui media sosial yang mempersulit deteksi berita palsu. Pendekatan mereka menggabungkan CNN dan RNN dalam kerangka belajar Struktur Tingkat Diskursus Hierarkis dari Berita Palsu (HDSF). Studi ini menggeneralisasi model percobaan di seluruh kumpulan data yang berbeda dengan hasil yang menjanjikan, meskipun menghadapi tantangan seperti fluktuasi dalam kerugian dan generalisasi yang buruk pada dataset FA-KES. Dataset yang digunakan meliputi FEVER, ISOT, dan FA-KES untuk pengujian dan validasi.

Studi yang dilakukan oleh (Samadi dkk., 2021) Mohammadreza Samadi, Maryam Mousavian, dan Saideh Momtazi pada tahun 2021 bertujuan untuk membandingkan pengklasifikasi saraf dan model pra-terlatih untuk deteksi berita palsu. Tantangan utama dalam penelitian ini adalah kompleksitas deteksi berita palsu yang membutuhkan model canggih untuk klasifikasi yang akurat. Mereka memanfaatkan metadata dan analisis tekstual untuk meningkatkan akurasi deteksi berita palsu, menyadari bahwa individu biasa seringkali kesulitan dalam membedakan berita palsu tanpa bantuan informasi tambahan. Metodologi penelitian mereka mencakup penggunaan metode pembelajaran yang diawasi dan tidak diawasi dengan teknik penyematan kata, serta model pembelajaran mesin ensemble dengan pemilihan fitur untuk mencapai akurasi optimal. Hasil penelitian menunjukkan bahwa model yang diusulkan mengungguli model yang ada dalam akurasi klasifikasi pada kumpulan data LIAR, ISOT, dan COVID-19. Mereka menemukan bahwa penggunaan CNN dengan Funnel Transformer menghasilkan peningkatan yang signifikan dalam dataset LIAR, sementara penyematan RobertA dan Funnel Transformer memberikan hasil terbaik untuk dataset ISOT dan COVID-19 peningkatan akurasi 7% dan 0,1%.

Penelitian yang dilakukan oleh (Azer dkk., 2021) Marina Azer, Mohamed Taha, Hala Zayed, dan Mahmud Gadallah bertujuan untuk mengembangkan sistem pembelajaran mesin yang dapat mendeteksi berita palsu di Twitter dengan memperkenalkan fitur baru untuk meningkatkan kinerja pengklasifikasi. Tantangan utama dalam penelitian ini adalah kumpulan data terbatas dengan fitur tweet yang menghambat akurasi deteksi kredibilitas, termasuk bahasa informal, kata-kata non-

bahasa Inggris, dan panjang tweet yang berbeda-beda. Metodologi penelitian mencakup penggunaan pengklasifikasi pembelajaran mesin yang diawasi seperti Naïve Bayes dan Random Forest, dengan memanfaatkan konten baru dan fitur berbasis pengguna untuk klasifikasi. Ekstraksi fitur dilakukan menggunakan metode k-best untuk memilih fitur yang optimal, yang kemudian digunakan dalam pengumpulan data, pra-pemprosesan, ekstraksi fitur, pelatihan, dan pengujian model. Hasil penelitian menunjukkan bahwa kombinasi fitur berbasis konten dan berbasis pengguna menghasilkan kinerja terbaik akurasi 82.2%, di mana fitur berbasis pengguna terbukti lebih unggul dalam deteksi kredibilitas. Pengklasifikasi Random Forest mencapai akurasi tertinggi menggunakan fitur gabungan. Studi ini menggunakan dataset Pheme yang terdiri dari 5.802 tweet yang telah diannotasi secara manual, dengan 3.830 tweet kredibel dan 1.792 tweet tidak kredibel.

Penelitian yang dilakukan oleh (Santoso dkk., 2020) Penelitian oleh (Penulis, Tahun) berjudul "Klasifikasi hoax dan analisis sentimen berita Indonesia menggunakan optimasi Naive Bayes" berfokus pada penyelesaian masalah penyebaran berita hoax di media sosial, yang dapat memprovokasi pembaca dan menyebarkan informasi yang salah. Penelitian ini menggunakan metode pencarian, cuplikan, dan kesamaan kosinus untuk mendeteksi berita hoax, serta Naive Bayes (NB) yang dioptimalkan dengan Particle Swarm Optimization (PSO) untuk analisis sentimen. Temuan penelitian menunjukkan sistem ini efektif dalam mendeteksi berita hoax, dengan akurasi rata-rata 77% pada beberapa sampel individu. Hasil ini menunjukkan bahwa metode yang diusulkan mampu mengidentifikasi berita hoax dengan baik, meskipun ada variasi akurasi di berbagai sampel.

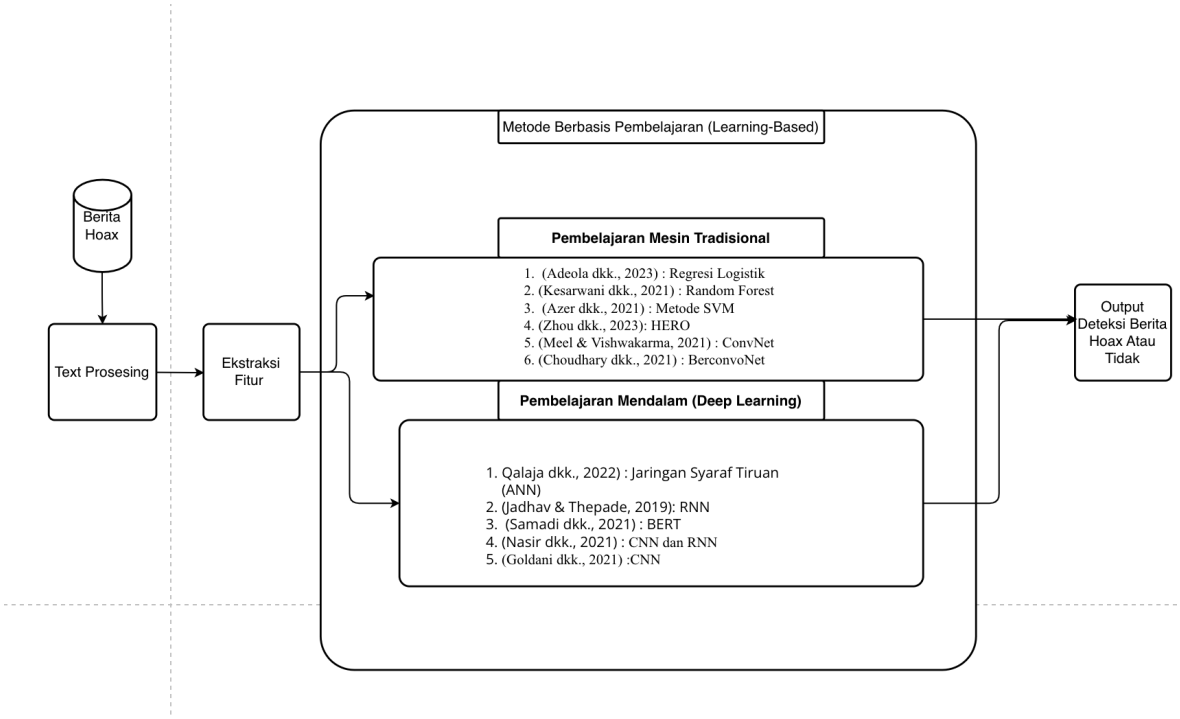
Studi yang dilakukan oleh Xinyi Zhou, Jiayu Li, Qinzhou Li, dan Reza Zafarani (Zhou dkk., 2023) bertujuan untuk meningkatkan akurasi dalam deteksi berita palsu dengan memanfaatkan pendekatan linguistik dan pembelajaran mesin. Mereka menggabungkan berbagai metode seperti analisis gaya linguistik untuk membedakan antara berita palsu dan berita yang benar. Melalui penggunaan dataset seperti Pemulihan dan MM-COVID, mereka mengembangkan model hierarchical recursive neural network (HERO) yang mengintegrasikan informasi multimodal dan konteks sosial untuk meningkatkan prediksi berita palsu. Metodologi penelitian mereka melibatkan ekstraksi fitur menggunakan jaringan saraf rekursif dan pendekatan berbasis konten yang memungkinkan mereka untuk membangun model yang mampu memprediksi dengan akurat berita sebagai palsu atau benar. menunjukkan akurasi tinggi dalam memprediksi berita palsu dengan skor AUC mulai dari 0,87.

Keterbaruan yang diberikan oleh peneliti dalam penelitian ini terletak pada penggabungan metode berbasis leksikon dan Long Short-Term Memory (LSTM) untuk deteksi dan klasifikasi berita hoax yang lebih akurat dan efisien. Pendekatan ini merupakan inovasi karena memanfaatkan keunggulan dari kedua metode: leksikon untuk identifikasi cepat kata-kata atau frasa yang sering muncul dalam berita hoax, dan LSTM untuk menangkap konteks serta pola kalimat secara lebih mendalam dan berurutan, sehingga lebih efektif dalam klasifikasi berita sebagai hoax atau tidak. Selain itu, peneliti juga menerapkan hyperparameter-tuning pada model LSTM, yang memungkinkan penentuan parameter optimal guna meningkatkan performa model dalam mendeteksi dan mengklasifikasikan hoax.

Fokus penelitian pada berita hoax yang beredar di Indonesia dalam tiga tahun terakhir semakin memperkuat kontribusi spesifik terhadap konteks lokal, yang belum banyak dieksplorasi secara komprehensif dalam penelitian sebelumnya. Dengan kombinasi ini, diharapkan hasil penelitian dapat membantu platform digital dan lembaga terkait dalam mengklasifikasikan berita hoax secara cepat dan akurat.

2.2 Kerangka Teori

Kerangka teori dalam penelitian ini didasarkan pada penelitian sebelumnya atau studi yang relevan yang dikutip sebagai titik acuan. Berikut kerangka teori dalam penelitian ini dapat dilihat pada Gambar 2.1.



Gambar 2. 1 Kerangka Teori

Tabel 2. 1 Penelitian Terdahulu

No	Peneliti	Problem	Methods Used	Text Processing	Feature Extraction	Persamaan	Perbedaan	Rencana Penelitian
1	Jadhav & Thepade, 2019	Deteksi dan klasifikasi berita palsu menggunakan AI dan NLP di jejaring sosial online	DSSM dan RNN	Tokenisasi dan stemming, hashing kata, proyeksi non-linier multi-layer	Transformasi TF-IDF, penghitungan vektorisasi	Text Processing menggunakan tokenisasi, stemming, pembuatan fitur semantik	Metode LSTM dan Dataset Berita Online	Kombinasi Fitur Lexicon Based sebagai Input LSTM, untuk Preprocessing Data dan untuk Interpretabilitas Model ke LSTM
2	Meel & Vishwakarma, 2021	Pengembangan kerangka kerja semi-diawasi dengan CNN berbasis leksikon untuk mendeteksi berita palsu	CNN (ConvNet)	Penambahan Lapisan Padat untuk Klasifikasi, Self-Ensembling	Fitur berbasis konten: kata ganti, kata positif/negatif, URL/hashtag	Fitur berbasis konten	Kerangka LSTM	Kombinasi Fitur Lexicon Based sebagai Input LSTM, untuk Preprocessing Data dan untuk Interpretabilitas Model ke LSTM

3	Samadi dkk., 2021	Klasifikasi artikel berita sebagai palsu atau nyata menggunakan kerangka BerconvoNet	BERT	Blok Penyematan Berita (NEB), Blok Fitur Multi-Skala (MSFB)	Lapisan Konvolusi MSFB, Max Pooling, dropout	Tidak menggunakan Lexicon Based dan LSTM	Tidak menggunakan Lexicon Based dan LSTM	Kombinasi Fitur Lexicon Based sebagai Input LSTM, untuk Preprocessing Data dan untuk Interpretabilitas Model ke LSTM
4	Nasir dkk., 2021	Deteksi berita hoax dengan model Lexicon Based	Lexicon Based	Format numerik menggunakan penyematan kata seperti GloVe	Penyematan kata dengan GloVe	Metode LSTM dan dataset berita online	Metode LSTM dan Dataset Berita Online sendiri	Kombinasi Fitur Lexicon Based sebagai Input LSTM, untuk Preprocessing Data dan untuk Interpretabilitas Model ke LSTM
5	Azer dkk., 2021	Meningkatkan kinerja pengklasifikasi hoax data di Twitter	Naïve Bayes dan SVM	Pembersihan Teks, Tokenisasi, Stemming	Fitur non-leksikal, fitur semantik, fitur gaya	Lexicon Based, Pembersihan Teks, Tokenisasi, Stemming	Objek data sumber berita online	Kombinasi Fitur Lexicon Based sebagai Input LSTM, untuk Preprocessing

								Data dan untuk Interpretabilitas Model ke LSTM
6	Qalaja dkk., 2022	Mendeteksi berita palsu terkait COVID-19 di Twitter	ANN	Fitur stylometric, Bag-of-Words, TF-IDF	Redundansi minimum dan relevansi maksimum	Objek data sumber berita online	Objek data sumber berita online	Kombinasi Fitur Lexicon Based sebagai Input LSTM, untuk Preprocessing Data dan untuk Interpretabilitas Model ke LSTM
7	Zhou dkk., 2023	Deteksi berita palsu menggunakan HERO untuk mempelajari gaya linguistik dokumen berita	HERO (Hierarchical Recursive Neural Network)	Pohon linguistik hierarkis diintegrasikan dengan jaringan saraf	Linguistik hierarkis, penyematan kata, bi-GRU	Menggunakan Teknik Lexicon Based	Metode LSTM dan Dataset Berita Online sendiri	Kombinasi Fitur Lexicon Based sebagai Input LSTM, untuk Preprocessing Data dan untuk Interpretabilitas Model ke LSTM

Dari hasil rujukan penelitian sebelumnya yang telah diuraikan di atas, digunakan Algoritma Lexicon-Based Long Short-Term Memory (LSTM) sebagai alat bantu analisis dalam penelitian ini dengan menggunakan bahasa pemrograman Python dan engine Google Colab.

2.3 Model lexicon-based

Metodologi berbasis leksikon biasanya digunakan dalam model analisis sentimen dalam studi penelitian untuk mengkategorikan sentimen dengan memanfaatkan leksikon atau korpus yang berisi kata-kata bahasa berbobot sebagai sumber daya linguistik (Ni Made Ayu Juli Astari dkk., 2020)

Model lexicon-based adalah metode yang menggunakan daftar kata (lexicon) yang telah ditentukan sebelumnya untuk mendeteksi hoax. Model ini bergantung pada pola kata-kata dan frasa tertentu yang sering muncul dalam berita hoax. Tahapan dalam perancangan model lexicon-based meliputi membandingkan setiap kata dalam teks berita dengan daftar lexicon, menghitung skor berdasarkan kemunculan kata-kata dalam lexicon, dan menentukan apakah sebuah berita termasuk hoax berdasarkan skor . Fitur-fitur leksikal ini kemudian diolah dengan menggunakan machine learning untuk mengklasifikasikan berita hoax dan berita asli. Berikut fitur dan penggunaannya dalam kombinasi LSTM:

1. **Fitur Lexicon Based sebagai Input LSTM**

Integrasi fitur berbasis leksikon yang diekstrak dari teks berita menggunakan metode seperti SVM atau Naive Bayes sebagai input untuk model LSTM menawarkan beberapa keuntungan. Pendekatan ini menggabungkan kekuatan teknik berbasis leksikon dalam menangkap fitur leksikal dengan kemampuan

LSTM untuk mempelajari urutan data temporal. Pendekatan ini sangat cocok untuk data teks berita terstruktur dengan volume yang besar, sehingga meningkatkan efisiensi dan akurasi proses deteksi hoaks (Saleh & Maryam, 2019).

Contoh :

Teks Berita: "Vaksin COVID-19 mengandung mikrochip untuk mengontrol manusia, dan semua yang sudah divaksin akan dipantau melalui jaringan 5G."

a. Input (Fitur Lexicon Based)

   Ekstraksi Fitur Leksikal: Menggunakan metode berbasis leksikon seperti Support Vector Machine (SVM) atau Naive Bayes, kata-kata yang sering muncul dalam berita hoax diidentifikasi. Dalam hal ini, kata-kata seperti "mikrochip", "kontrol", "pantau", dan "5G" dianggap sebagai kata-kata berisiko tinggi yang terkait dengan berita hoax tentang teori konspirasi.

   Vektor Fitur: Teks berita diubah menjadi vektor fitur berbasis kata-kata kunci leksikal yang sebelumnya telah diidentifikasi:

   - [1 (mikrochip), 1 (kontrol), 1 (pantau), 1 (5G)]

   Setiap angka "1" menunjukkan kata yang terkait dengan hoax dalam berita tersebut.

b. Proses (Model LSTM)

   Input ke LSTM: Setelah fitur leksikal diidentifikasi, vektor yang terbentuk ini dimasukkan ke dalam model Long Short-Term Memory (LSTM). LSTM, yang unggul dalam menangkap konteks dan pola temporal dalam teks, akan menganalisis urutan kemunculan kata-kata tersebut dalam berita.

LSTM akan mempelajari bagaimana kata-kata seperti "mikrochip" dan "5G" sering kali muncul dalam konteks berita hoax mengenai vaksin.

Pembelajaran Konteks: LSTM akan memperhitungkan pola sekuensial dari teks tersebut, sehingga memahami bahwa kata "mikrochip" sering kali muncul bersama dengan teori konspirasi tentang "5G" dalam berita hoax. LSTM belajar bahwa pola ini sangat khas dalam berita hoax yang terkait dengan COVID-19.

c. Output (Prediksi)

Keluaran dari LSTM: Setelah model LSTM menganalisis urutan kata dan pola teks, ia akan mengeluarkan prediksi apakah berita tersebut merupakan hoax atau tidak. Berdasarkan analisis, model akan memberikan skor prediksi tinggi bahwa berita tersebut adalah hoax, misalnya output "1" yang berarti **hoax**, atau probabilitas tinggi (contoh: 0,92/1) bahwa berita tersebut adalah berita palsu.

Prediksi Akhir: Berdasarkan analisis gabungan leksikon dan pola LSTM, berita ini akan diklasifikasikan sebagai **hoax** karena mengandung kata-kata dan pola yang sering ditemukan dalam berita palsu terkait teori konspirasi vaksin.

2. Fitur Lexicon Based untuk Preprocessing Data

Pemanfaatan fitur berbasis leksikon dalam pra pemrosesan data untuk model LSTM telah terbukti dapat meningkatkan kualitas data dan kesesuaian data untuk data tekstual yang berisik atau tidak. Dengan menggabungkan metode berbasis leksikon untuk mengekstrak fitur leksikal dari teks berita, data dapat secara efektif

dibersihkan dan dipra-proses sebelum dimasukkan ke dalam model LSTM, sehingga meningkatkan kualitas keseluruhan data masukan (Yang dkk., 2020).

Contoh Teks Berita Hoax: "Vaksin COVID-19 dibuat untuk mengontrol populasi dunia melalui mikrochip yang diaktifkan dengan jaringan 5G."

a. Data Awal (Teks Berita)

   "Vaksin COVID-19 dibuat untuk mengontrol populasi dunia melalui mikrochip yang diaktifkan dengan jaringan 5G."

b. Pra pemprosesan dengan Fitur Berbasis Leksikon

   Identifikasi Kata Kunci: Dengan menggunakan leksikon yang berisi daftar kata yang sering muncul dalam berita hoax (misalnya "mikrochip", "kontrol", "5G"), sistem akan menandai kata-kata yang berpotensi terkait dengan hoax.

   Pembersihan Data: Kata-kata umum seperti "untuk", "yang", "dengan" (stop words) akan dihapus karena tidak memberikan kontribusi yang signifikan dalam proses analisis.

   Ekstraksi Fitur Leksikal: Dari teks tersebut, kata-kata penting yang terkait dengan hoax seperti "vaksin", "mikrochip", "kontrol", dan "5G" akan diekstraksi.

   Hasil Pra pemprosesan:

   ["vaksin", "kontrol", "mikrochip", "5G"]

c. Input ke Model LSTM

Setelah pembersihan dan ekstraksi fitur leksikal, hasil pra pemprosesan ini akan digunakan sebagai input ke dalam model LSTM, di mana model akan menganalisis hubungan dan pola kata-kata tersebut dalam konteks berita hoax.

3. Fitur Lexicon Based untuk Interpretabilitas Model:

Dalam penelitian yang dilakukan oleh (Anisa dkk., 2023) fitur Lexicon Based digunakan untuk meningkatkan interpretabilitas model deteksi berita hoax berbasis LSTM. Metode yang digunakan adalah dengan menggunakan fitur leksikal yang diekstrak dari teks berita untuk membantu interpretasi hasil prediksi model LSTM. Kelebihan dari penggunaan fitur Lexicon Based ini adalah meningkatkan interpretabilitas model, sehingga memudahkan untuk memahami alasan prediksi yang dilakukan oleh model. Hal ini sangat cocok untuk aplikasi yang membutuhkan interpretabilitas model yang tinggi, karena fitur Lexicon Based dapat memberikan pemahaman yang lebih dalam terhadap proses prediksi yang dilakukan oleh model LSTM.

Contoh:

a. Input:

Data Teks Berita: Teks berita yang akan dianalisis. Misalnya, berita berjudul: *"Vaksin COVID-19 adalah bagian dari konspirasi global untuk mengendalikan populasi melalui 5G."*

Fitur Leksikal: Fitur-fitur berbasis leksikon diambil dari teks berita ini, yaitu kata-kata kunci yang sering muncul dalam berita hoax, seperti

"konspirasi", *"mengendalikan"*, dan *"5G"*. Kata-kata ini merupakan indikator kuat berita hoax berdasarkan pola data sebelumnya.

b. Proses:

Ekstraksi Fitur Leksikal: Fitur-fitur leksikon diidentifikasi dari teks berita menggunakan metode seperti SVM atau Naive Bayes untuk mendeteksi kata-kata yang berkaitan dengan hoax.

- Contoh vektor fitur untuk teks berita di atas: [1 (konspirasi), 1 (mengendalikan), 1 (5G)].

Input ke LSTM: Vektor fitur leksikal ini kemudian dimasukkan ke dalam model LSTM. Model LSTM akan memproses urutan teks berita dan menganalisis konteks serta pola penggunaan kata-kata tersebut secara mendalam.

Pembelajaran Konteks: LSTM memahami bagaimana kata-kata seperti *"konspirasi"* dan *"5G"* biasanya muncul dalam berita hoax dan bagaimana urutan kata tersebut membentuk pola yang mencurigakan.

c. Output:

Prediksi Model: Berdasarkan analisis LSTM, model mengeluarkan prediksi bahwa berita tersebut adalah **hoax**.

Interpretability: Fitur Lexicon Based menjelaskan mengapa model memprediksi bahwa berita tersebut adalah hoax. Kata-kata kunci seperti *"konspirasi"*, *"mengendalikan"*, dan *"5G"* ditampilkan sebagai alasan utama model mendeteksi berita ini sebagai hoax.

2.4 LSTM

Jaringan Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997) adalah jenis khusus jaringan saraf berulang (RNN) (Chollet, 2018) yang dirancang untuk mengatasi masalah gradien lenyap pada RNN (Senhaji et al., 2021). LSTM unggul dalam menangani celah panjang dengan tingkat ketahanan tertentu, membedakannya dari RNN lainnya, model Markov tersembunyi, dan pendekatan pembelajaran urutan lainnya. Kemampuan utama LSTM terletak pada mempertahankan memori jangka pendek selama beberapa langkah waktu, sehingga disebut "memori jangka pendek yang panjang." Model ini banyak diterapkan dalam klasifikasi, pemrosesan data, dan analisis prediktif berbasis data deret waktu, termasuk pengenalan tulisan tangan, suara, penerjemahan mesin, deteksi aktivitas bicara, kontrol robot, video game, dan perawatan kesehatan.

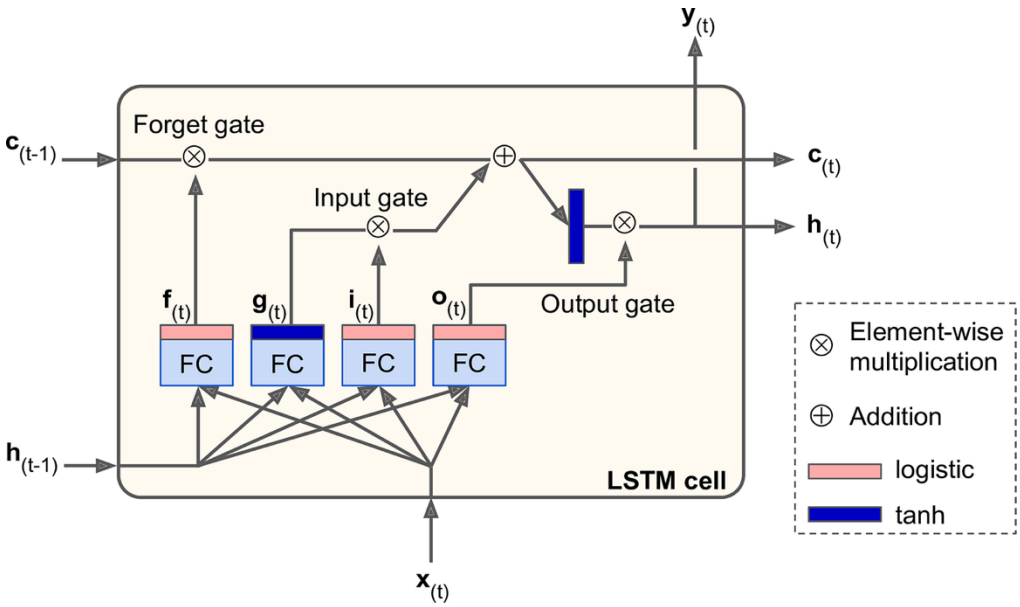
Pada model LSTM (Long Short-Term Memory), terdapat beberapa parameter yang berperan penting dalam proses pelatihan dan prediksi.

Tabel 2. 2 Parameter LSTM

Variabel	**Keterangan**
max_features	Jumlah maksimum fitur yang akan diproses oleh Tokenizer pada teks sebelum dilakukan padding.
max_len	Panjang maksimum dari urutan token setelah dilakukan tokenisasi dan padding.
embedding_dim	Dimensi embedding yang digunakan pada layer Embedding dalam model LSTM.
epochs	Jumlah iterasi pada saat melatih model LSTM.

batch_size	Jumlah sampel data yang digunakan dalam satu iterasi saat melatih model.
threshold	Ambang batas untuk menentukan prediksi (Hoax atau Non-Hoax) dari probabilitas yang diperoleh dari model. Nilai ini bisa disesuaikan.
loss_function	Fungsi kerugian yang digunakan saat melatih model LSTM.
optimizer	Algoritma optimisasi yang digunakan saat melatih model LSTM.
dropout	Nilai dropout yang digunakan pada layer SpatialDropout1D dalam model LSTM.
recurrent_dropout	Nilai dropout yang digunakan pada layer LSTM dalam model LSTM.

LSTM sendiri dibangun bertujuan untuk memecahkan masalah *vanishing gradient* pada RNN (Recurrent *Neural Network*) yang dimana saat memproses data sekuensial yang panjang, kemiringan fungsi kerugian turun secara eksponensial (Kim & Kang, 2020). Arsitektur umum LSTM dapat dilihat pada Gambar 2.2.



Gambar 2. 2 Arsitektur LSTM (Géron, 2018)

Berdasarkan Gambar 2.2, Arsitektur LSTM terdiri dari sel-sel memori yang disebut sel. Dalam setiap langkah waktu (*time step*), dua state diteruskan ke sel berikutnya, yaitu *cell state* dan *hidden state*. *Cell state* adalah bagian utama yang mengalirkan data sepanjang jaringan, memungkinkan data untuk mengalir maju dengan sedikit perubahan. Meskipun demikian, beberapa transformasi *linier* dapat terjadi pada *cell state*. Data dapat ditambahkan atau dihapus dari *cell state* melalui penggunaan gerbang sigmoid. Gerbang tersebut memiliki struktur yang mirip dengan lapisan atau serangkaian operasi matriks, yang mengandung bobot individu yang berbeda untuk melakukan transformasi pada data

Gambar 2.2 menggambarkan standarisasi dari arsitektur metode LSTM yang dibagi menjadi empat komponen yaitu, *Input Gate (*i) untuk mengontrol arus input yang masuk ke *neuron*, *Forget Gate* (f) untuk memasukkan *neuron*

ke kondisi reset saat ini. keadaan, *Output Gate* (o) untuk mengontrol efek aktivasi *neuron* pada *neuron* lain dan sel memori (c).

Keberadaan *Forget Gate* membantu menentukan derajat lupanya aliran informasi sebelum sel saat ini. Persamaan *Forget Gate* ditunjukkan pada persamaan (2.1).

$$f_t = \sigma\big(W_f.[h_{t-1} \cdot x_t] + b_f\big) \tag{2.1}$$

Komponen pada persamaan (2.1) adalah f_t nilai dari *forget gate*, σ adalah fungsi sigmoid, yang mengubah nilai input menjadi rentang dari 0 hingga 1. Fungsi sigmoid digunakan di sini untuk memastikan bahwa nilai f_t akan berada di antara 0 dan 1, sesuai dengan rasio bagian-bagian elemen seluler negara harus dilupakan atau dilestarikan. W_f adalah matriks bobot yang digunakan untuk mengalikan status masukan dan *hidden state*. h_{t-1} adalah *hidden state* dari sel LSTM pada *timestep* sebelumnya. $x_t$adalah input pada timestep saat ini. b_f adalah bias.

Fungsi *Input Gate* adalah untuk menentukan besarnya arus informasi yang ditambahkan pada arus informasi. Persamaan *Input Gate* disajikan pada persamaan (2.2) dan (2.3):

$$i_t = \sigma(W_i \cdot [h_{t-1} \cdot x_t] + b_i) \tag{2.2}$$
$$C_t = \tanh\,(W_c \cdot [h_{t-1} \cdot x_t] + b_c) \tag{2.3}$$

Setelah informasi melewati input *gate* dan *forget gate*, LSTM memperbarui status sel untuk menghitung keluaran sel LSTM saat ini dan meneruskannya ke sel LSTM berikutnya. Persamaan disajikan pada persamaan (2.4):

$$C_t = f_t * C_{t-1} + i_t * C_t \tag{2.4}$$

Output Gate menggabungkan masukan saat ini dan keadaan sel untuk menentukan keluaran sel LSTM saat ini. Persamaan *output gate* disajikan pada persamaan (2.5) dan (2.6):

$$O_t = \sigma(W_o \cdot [h_{t-1} \cdot x_t] + b_0) \quad (2.5)$$

$$h_t = O_t * \tanh(C_t) \quad (2.6)$$

Dimana x_t mewakili variabel *input* pada langkah waktu saat ini, $h_t$adalah output dari sel sebelumnya, C_{t-1} adalah keadaan sel sebelumnya yang memberikan informasi masa lalu. Dalam fungsi sigmoid logistik σ dan tanh pada gate input, forget gate, dan output, parameter ini digunakan dengan matriks bobot dan vektor bias. (Vu dkk., 2021)

Fungsi *sigmoid* diterapkan dalam pembelajaran mesin untuk *logistic regression* dan penerapan jaringan saraf. Nilai keluaran dari fungsi *sigmoid* berkisar antara 0 hingga 1. Persamaan yang digunakan dalam fungsi *sigmoid* ditunjukkan pada persamaan (2.7).

Dengan e adalah bilangan Euler yang merupakan konstanta matematika yang mendekati 2,71828 (Nugraha dkk., 2023)

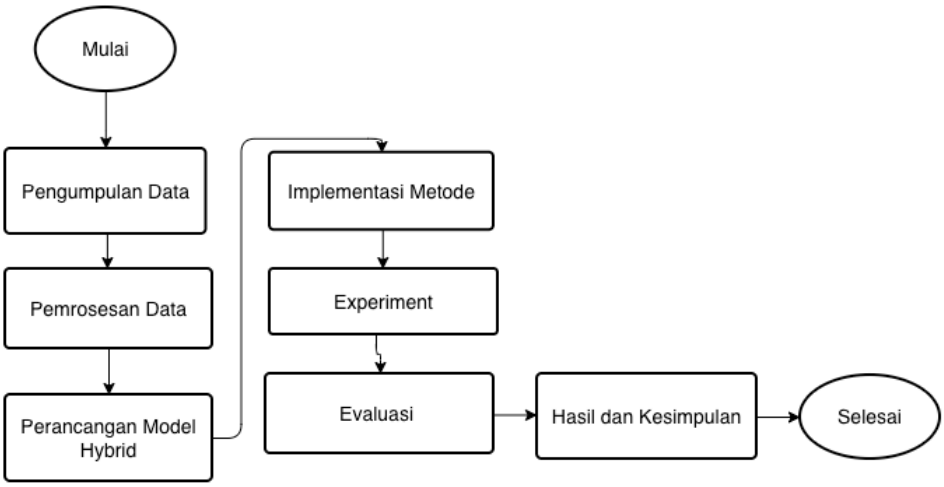
$$f(x) = \frac{1}{1 \div e^{-x}} \quad (2.7)$$

BAB III
METODOLOGI PENELITIAN

Bab ini akan membahas tentang analisis sistem dan perancangan penelitian ini. Beberapa tahapan yang akan dibahas termasuk tahapan penelitian yang dilakukan, persyaratan yang akan dibuat, dan penyelesaian masalah untuk pengembangan sistem deteksi hoax berita yang menggunakan pendekatan lexicon-based dan *Long Short- Term Memory* (LSTM).

3.1. Desain Penelitian

Dalam penelitian ini, untuk memudahkan pelaksanaannya, langkah-langkah yang akan diambil akan dijelaskan secara rinci. Gambar 3.1 memperlihatkan alur penelitian ini.



Gambar 3. 1 Alur Penelitian

Penelitian ini menggunakan metode eksperimental. Peneliti akan mengembangkan dua model deteksi berita hoax pendekatan berbasis lexicon dan LSTM. Kedua model tersebut akan dikombinasikan dan dibandingkan untuk menilai akurasinya dalam mendeteksi berita hoax.

3.2. Pengumpulan Data

Sumber data untuk penelitian ini adalah teks berita dari situs berita online. Dengan menggunakan teknik Crawling. Teknik Crawling adalah metode pengumpulan data secara otomatis dari situs web dengan menggunakan program khusus yang disebut web crawler atau spider menggunakan Python penggunaan web crawler atau spider, yang dirancang untuk menavigasi web secara sistematis, Proses crawling melibatkan beberapa tahapan penting (Prastyo, 2021):

1. Inisialisasi: Web crawler memulai dari satu atau lebih URL yang dikenal sebagai seed. Ini adalah titik awal dari mana crawler mulai menjelajahi web (Neelakandan dkk., 2022).
2. Pengambilan Halaman: Crawler mengunduh halaman web dari seed URL dan memeriksa konten halaman tersebut (Wang dkk., 2023).
3. Ekstraksi Tautan: Dalam proses ini, crawler mengidentifikasi semua tautan (hyperlink) dalam halaman web yang diunduh dan menambahkannya ke daftar URL yang akan dikunjungi berikutnya (K dkk., 2023)
4. Pengindeksan Konten: Informasi yang relevan dari halaman yang diunduh diekstraksi dan diindeks. Ini bisa termasuk teks, gambar, metadata, dan data lain yang dapat diakses (Rajiv & Navaneethan, 2022)

5. Pemrosesan Berulang: Proses ini diulang untuk setiap URL baru yang ditemukan, memungkinkan crawler untuk menjelajahi situs web secara luas (Kumar dkk., 2022).
6. Simpan CSV Data Menggunakan Python Scraping: Data yang dikumpulkan oleh crawler dikelola dan disimpan dalam format CSV.

Source Code Scraping Data

```
# Detik
import requests
from bs4 import BeautifulSoup
import csv
# Inisialisasi variabel
url_utama = 'https://sport.detik.com/' # URL utama
halaman_ke = 200 # Mulai dari halaman ke-1
# Definisikan nama file CSV dan nama kolom
nama_file_csv = 'detik_test2024.csv'
nama_kolom = ['Judul', 'Tanggal', 'Link']
# Membuka file CSV dalam mode tulis dan menulis header
with open(nama_file_csv, mode='w', newline='', encoding='utf-8')
as file:
penulis_csv = csv.writer(file)
penulis_csv.writerow(nama_kolom)
while True:
# Buat URL untuk halaman saat ini
url = f'{url_utama}indeks/{halaman_ke}/'
respons = requests.get(url)
if respons.status_code == 200:
# Parsing konten HTML
soup = BeautifulSoup(respons.content, 'html.parser')
# Cari dan ekstrak elemen-elemen yang diperlukan
elemen_utama = soup.find(class_="grid-row list-content")
elemen_judul = elemen_utama.find_all('h3',
class_='media__title')
daftar_judul = [judul.text.strip() for judul in elemen_judul]
elemen_tanggal = elemen_utama.find_all('div',
class_="media__date")
daftar_tanggal = [tanggal.text.strip() for tanggal in
elemen_tanggal]
elemen_link = elemen_utama.find_all('a', class_='media__link')
daftar_link = [link.get('href') for link in elemen_link]
# Jika tidak ada data, anggap halaman terakhir sudah tercapai
if halaman_ke == 21:
break
# Menulis data ke dalam file CSV
for judul, tanggal, link in zip(daftar_judul, daftar_tanggal,
daftar_link):
penulis_csv.writerow([judul, tanggal, link])
# Pindah ke halaman berikutnya
halaman_ke += 1
```

```
else:
print(f'Gagal mengakses halaman {halaman_ke}. Status kode:
{respons.status_code}')
break
print(f'Data telah disimpan dalam file {nama_file_csv}')
```

Source Code Scraping Data digunakan untuk melakukan web scraping dari situs Detik.com pada bagian olahraga, bertujuan mengumpulkan data berupa judul artikel, tanggal publikasi, dan tautan, yang kemudian disimpan dalam file CSV bernama detik_test2024.csv. Skrip dimulai dengan mendefinisikan URL dasar dan halaman indeks yang akan diakses, lalu membuka file CSV untuk menuliskan data dengan kolom "Title", "Date", dan "Link". Menggunakan pustaka requests dan BeautifulSoup, kode ini mengirim permintaan HTTP ke setiap halaman indeks, kemudian memproses konten HTML untuk menemukan elemen-elemen yang berisi informasi artikel (judul, tanggal, dan tautan). Data yang berhasil ditemukan ditambahkan ke file CSV, sedangkan jika halaman kosong atau tidak ditemukan, proses berhenti. Dengan mekanisme iteratif ini, skrip dapat menjelajahi banyak halaman sekaligus, memastikan data tersimpan secara terstruktur, Data yang dikumpulkan di tujukan di tabel 3.1 pengumpulan data.

Tabel 3. 1 Pengumpulan Data

Situs	**Jumlah Data**	**Tahun**	**Label**
Turnbackhoax.id	5000	2019-2023	Hoax
Kompas.com	3000	2019-2023	Non-Hoax
Detik	2000	2019-2023	Non-Hoax

3.3. Preprocessing data

Preprocessing data merupakan langkah penting dalam mempersiapkan data teks untuk analisis lebih lanjut. Tujuannya adalah untuk meningkatkan struktur data dengan membersihkan dan mempersiapkan teks agar siap diproses pada tahapan selanjutnya (Zhu & Luo, 2023). Langkah-langkah dalam preprocessing teks meliputi penghapusan stop words, angka, tanda baca, tautan, spasi kosong, dan kata-kata non-relevan (Bouchiha dkk., 2022). Selain itu, teknik seperti stemming juga dapat digunakan untuk mengurangi kata-kata menjadi bentuk dasarnya (HaCohen-Kerner dkk., 2020).Teknik pra pemprosesan khusus diperlukan untuk mengkondisikan data teks secara efektif untuk digunakan dalam model pembelajaran mesin**.**

a. *Cleansing*

Proses penghapusan atau pembersihan pada data terjemahan yang berisi angka, delimeter seperti tanda koma (,), tanda petik (“) dan tanda titik (.) dan yang lainnya.

Tabel 3. 2 *Cleansing*

Teks Sebelum	Teks Sesudah
"Breaking News!!! Vaksin Covid-19 menyebabkan mutasi genetik pada manusia. Ini adalah konspirasi global untuk mengontrol populasi dunia!!! #Hoax #FakeNews"	Breaking News Vaksin Covid menyebabkan mutasi genetik pada manusia Ini adalah konspirasi global untuk mengontrol populasi dunia

b. *Case Folding*

Case folding adalah proses mengubah semua huruf dalam teks menjadi huruf kecil untuk memastikan keseragaman.

Tabel 3. 3 *Case Folding*

Teks Sebelum	Teks Sesudah
Breaking News Vaksin Covid menyebabkan mutasi genetik pada manusia Ini adalah konspirasi global untuk mengontrol populasi dunia	breaking news vaksin covid menyebabkan mutasi genetik pada manusia ini adalah konspirasi global untuk mengontrol populasi dunia

c. *Tokenizing*

Tokenizing adalah proses memecah teks menjadi unit-unit kata yang lebih kecil yang disebut token.

Tabel 3. 4 *Tokenizing*

Teks Sebelum	Teks Sesudah
breaking news vaksin covid menyebabkan mutasi genetik pada manusia ini adalah konspirasi global untuk mengontrol populasi dunia	["breaking", "news", "vaksin", "covid", "menyebabkan", "mutasi", "genetik", "pada", "manusia", "ini", "adalah", "konspirasi", "global", "untuk", "mengontrol", "populasi", "dunia"]

3.4. Perancangan Model Lexicon Based

Model lexicon based merupakan metode yang menggunakan daftar kata (lexicon) yang telah didefinisikan sebelumnya untuk mendeteksi hoax. Model ini mengandalkan pola kata-kata dan frasa tertentu yang sering muncul dalam berita hoax.

Tabel 3. 5 Proses Model Lexicon-Based untuk Deteksi Hoax

Langkah	Deskripsi	Input	Hasil
1. Pencocokan Kata	Memeriksa kata-kata dalam teks yang cocok dengan kata-kata dalam kamus hoax dan menghitung frekuensinya.	Teks: "Breaking News!!! Vaksin Covid-19 menyebabkan mutasi genetik pada manusia. Ini adalah konspirasi global untuk mengontrol populasi dunia!!!"	Kata yang cocok dan frekuensinya: "breaking" (1) "vaksin" (1) "genetik" (1) "mutasi" (1) "konspirasi" (1) "global" (1) "kontrol" (1) "populasi" (1)
2. Scoring	Menghitung skor berdasarkan frekuensi kata kunci yang cocok. Skor ini menunjukkan jumlah total kata kunci yang ada dalam teks.	Frekuensi Kata Kunci: "breaking" (1) "vaksin" (1) "genetik" (1) "mutasi" (1) "konspirasi" (1) "global" (1) "kontrol" (1) "populasi" (1)	Total Skor: 8
3. Analisis dan Penilaian	Menilai apakah teks termasuk hoax berdasarkan skor total dan frekuensi kata kunci. Menentukan apakah teks	Skor Total: 8 Ambang Batas: 5	"Teks ini mungkin hoax, karena mengandung banyak kata kunci: breaking, vaksin, genetik, mutasi, konspirasi, global, kontrol, populasi"

	melebihi ambang batas (threshold).		

3.5. Perancangan Model LSTM

Long Short-Term Memory (LSTM) dengan algoritma LSTM itu sendiri dengan pembacaan teks dalam satu arah Kita akan melakukan perhitungan LSTM dengan empat gerbang yang berbeda: forget gate, input gate, cell gate, dan output gate. Berikut Contoh prosesnya:

1. **Vektor Input**

   Vector input mengonversi teks berita menjadi representasi numerik. **Contoh Input**: Teks berita "Breaking News!!! Vaksin Covid-19 menyebabkan mutasi genetik pada manusia. Ini adalah konspirasi global untuk mengontrol populasi dunia!!!" diubah menjadi vektor input x_t.

   Misalkan vektor input x_t adalah [1, 0, 0, 0, 1] (contoh sederhana dengan 5 dimensi).

2. **Forget Gate (f_t)**

   Forget gate menentukan informasi mana dari cell state sebelumnya yang harus dilupakan.

   Contoh Input:

   - Hidden State (h_{t-1}): [0.5, 0.6]
   - Cell State (C_{t-1}): [0.2, 0.3]
   - Vektor Input (x_t): [1, 0, 0, 0, 1]

Bobot dan Bias:

- Bobot Forget Gate (W_f): $\begin{bmatrix} 0.2 & 0.4 & 0.3 & 0.5 & 0.6 \\ 0.1 & 0.3 & 0.2 & 0.4 & 0.5 \end{bmatrix}$
- Bias Forget Gate (b_f): [0.1, 0.2]

Proses:$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$

- Hitung:$[h_{t-1}, x_t] = [0.5,0.6,1,0,0,0,1] W_f \cdot [h_{t-1}, x_t] =$

  $\begin{bmatrix} 0.2 \cdot 0.5 + 0.4 \cdot 0.6 + 0.3 \cdot 1 + 0.5 \cdot 0 + 0.6 \cdot 0 + 0.3 \cdot 0 + 0.5 \cdot 1 \\ 0.1 \cdot 0.5 + 0.3 \cdot 0.6 + 0.2 \cdot 1 + 0.4 \cdot 0 + 0.5 \cdot 0 + 0.2 \cdot 0 + 0.3 \cdot 1 \end{bmatrix} =$

  $\begin{bmatrix} 0.1 + 0.24 + 0.3 + 0.5 \\ 0.05 + 0.18 + 0.2 + 0.3 \end{bmatrix} = \begin{bmatrix} 1.14 \\ 0.73 \end{bmatrix} f_t = \sigma(\begin{bmatrix} 1.14 + 0.1 \\ 0.73 + 0.2 \end{bmatrix}) =$

  $\sigma(\begin{bmatrix} 1.24 \\ 0.93 \end{bmatrix}) f_t = \begin{bmatrix} 0.776 \\ 0.717 \end{bmatrix}$ (menggunakan fungsi sigmoid)

3. Input Gate (i_t)

Menentukan informasi baru mana yang akan ditambahkan ke cell state.

Contoh Input:

- Hidden State (h_{t-1}): [0.5, 0.6]
- Vektor Input (x_t): [1, 0, 0, 0, 1]

Bobot dan Bias:

- Bobot Input Gate (W_i): $\begin{bmatrix} 0.3 & 0.2 & 0.4 & 0.1 & 0.5 \\ 0.6 & 0.7 & 0.8 & 0.3 & 0.2 \end{bmatrix}$
- Bias Input Gate (b_i): [0.1, 0.2]

Proses:$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$

- Hitung:$W_i \cdot [h_{t-1}, x_t] =$

  $\begin{bmatrix} 0.3 \cdot 0.5 + 0.2 \cdot 0.6 + 0.4 \cdot 1 + 0.1 \cdot 0 + 0.5 \cdot 1 \\ 0.6 \cdot 0.5 + 0.7 \cdot 0.6 + 0.8 \cdot 1 + 0.3 \cdot 0 + 0.2 \cdot 1 \end{bmatrix} =$

$$\begin{bmatrix} 0.15 + 0.12 + 0.4 + 0.5 \\ 0.3 + 0.42 + 0.8 + 0.2 \end{bmatrix} = \begin{bmatrix} 1.17 \\ 1.72 \end{bmatrix} i_t = \sigma(\begin{bmatrix} 1.17 + 0.1 \\ 1.72 + 0.2 \end{bmatrix}) =$$

$$\sigma(\begin{bmatrix} 1.27 \\ 1.92 \end{bmatrix}) i_t = \begin{bmatrix} 0.781 \\ 0.871 \end{bmatrix}$$ (menggunakan fungsi sigmoid)

4. Cell Gate (g_t)

Cell Gate membuat calon nilai cell state baru.

Contoh Input:

- Hidden State (h_{t-1}): [0.5, 0.6]
- Vektor Input (x_t): [1, 0, 0, 0, 1]

Bobot dan Bias:

- Bobot Cell Gate (W_c): $\begin{bmatrix} 0.6 & 0.7 & 0.5 & 0.4 & 0.2 \\ 0.8 & 0.9 & 0.6 & 0.5 & 0.3 \end{bmatrix}$
- Bias Cell Gate (b_c): [0.3, 0.4]

Proses:$g_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$

- Hitung:$W_c \cdot [h_{t-1}, x_t] =$

$$\begin{bmatrix} 0.6 \cdot 0.5 + 0.7 \cdot 0.6 + 0.5 \cdot 1 + 0.4 \cdot 0 + 0.2 \cdot 1 \\ 0.8 \cdot 0.5 + 0.9 \cdot 0.6 + 0.6 \cdot 1 + 0.5 \cdot 0 + 0.3 \cdot 1 \end{bmatrix} =$$

$$\begin{bmatrix} 0.3 + 0.42 + 0.5 + 0.2 \\ 0.4 + 0.54 + 0.6 + 0.3 \end{bmatrix} = \begin{bmatrix} 1.42 \\ 1.84 \end{bmatrix} g_t = \tanh(\begin{bmatrix} 1.42 + 0.3 \\ 1.84 + 0.4 \end{bmatrix}) =$$

$$\tanh(\begin{bmatrix} 1.72 \\ 2.24 \end{bmatrix}) g_t = \begin{bmatrix} 0.939 \\ 0.975 \end{bmatrix}$$ (menggunakan fungsi tanh)

5. Output Gate (o_t)

Menentukan seberapa banyak cell state yang akan mempengaruhi hidden state saat ini.

Contoh Input:

- Hidden State (h_{t-1}): [0.5, 0.6
- Vektor Input (x_t): [1, 0, 0, 0, 1]

Bobot dan Bias:

- Bobot Output Gate (W_o): $\begin{bmatrix} 0.2 & 0.3 & 0.4 & 0.5 & 0.6 \\ 0.7 & 0.8 & 0.5 & 0.4 & 0.3 \end{bmatrix}$
- Bias Output Gate (b_o): [0.2, 0.3]

Proses:$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$

- Hitung:$W_o \cdot [h_{t-1}, x_t] =$

  $\begin{bmatrix} 0.2 \cdot 0.5 + 0.3 \cdot 0.6 + 0.4 \cdot 1 + 0.5 \cdot 0 + 0.6 \cdot 1 \\ 0.7 \cdot 0.5 + 0.8 \cdot 0.6 + 0.5 \cdot 1 + 0.4 \cdot 0 + 0.3 \cdot 1 \end{bmatrix} =$

  $\begin{bmatrix} 0.1 + 0.18 + 0.4 + 0.6 \\ 0.35 + 0.48 + 0.5 + 0.3 \end{bmatrix} = \begin{bmatrix} 1.28 \\ 1.63 \end{bmatrix} o_t = \sigma(\begin{bmatrix} 1.28 + 0.2 \\ 1.63 + 0.3 \end{bmatrix}) =$

  $\sigma(\begin{bmatrix} 1.48 \\ 1.93 \end{bmatrix}) o_t = \begin{bmatrix} 0.815 \\ 0.874 \end{bmatrix}$ (menggunakan fungsi sigmoid)

6. Update Cell State (C_t)

Deskripsi: Memperbarui cell state baru dengan menggabungkan informasi dari forget gate dan input gate.

Contoh Input:

- Cell State (C_{t-1}): [0.2, 0.3]
- Forget Gate (f_t): [0.776, 0.717]
- Input Gate (i_t): [0.781, 0.871]
- Cell Gate (g_t): [0.939, 0.975]

Proses:$C_t = f_t \cdot C_{t-1} + i_t \cdot g_t$

- Hitung:$C_t = \begin{bmatrix} 0.776 \cdot 0.2 + 0.781 \cdot 0.939 \\ 0.717 \cdot 0.3 + 0.871 \cdot 0.975 \end{bmatrix} =$

  $\begin{bmatrix} 0.1552 + 0.732339 \\ 0.2151 + 0.848225 \end{bmatrix} = \begin{bmatrix} 0.887539 \\ 1.063325 \end{bmatrix}$

7. Update Hidden State (h_t)

Menghitung hidden state baru berdasarkan cell state dan output gate.

Contoh Input:

- Cell State Baru (C_t): [0.887539, 1.063325]
- Output Gate (o_t): [0.815, 0.874]

Proses:$h_t = o_t \cdot \tanh(C_t)$

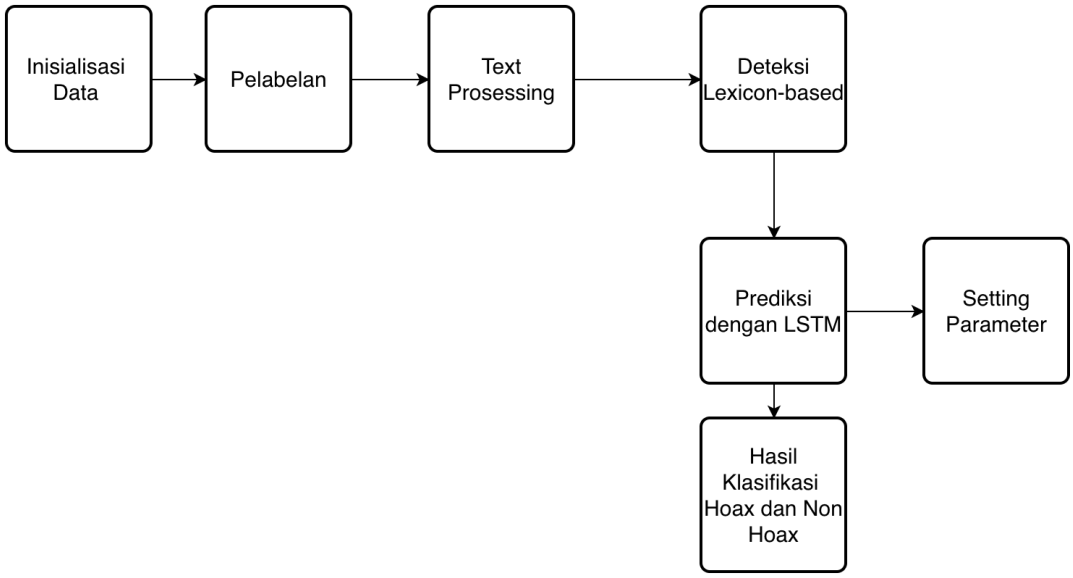
- Hitung:$h_t = \begin{bmatrix} 0.815 \cdot \tanh(0.887539) \\ 0.874 \cdot \tanh(1.063325) \end{bmatrix} = \begin{bmatrix} 0.815 \cdot 0.707 \\ 0.874 \cdot 0.789 \end{bmatrix} = \begin{bmatrix} 0.576 \\ 0.689 \end{bmatrix}$

Hasil Akhir

- **Input**: Teks berita yang telah dikonversi ke dalam vektor numerik.
- **Proses**: Melalui langkah-langkah LSTM dengan perhitungan detail untuk mendapatkan hidden state dan cell state yang diperbarui.
- **Output**: Hidden state baru (h_t) adalah [0.576, 0.689].
- **Hasil Klasifikasi**: Berdasarkan hidden state (h_t) dan hasil lapisan output, teks berita ini diklasifikasikan sebagai **hoax**.

Penggunaan metode lexicon based sangat penting karena dapat memberikan informasi tambahan mengenai kata-kata kunci yang relevan. Informasi ini membantu model LSTM dalam meningkatkan akurasi klasifikasi berita hoax dengan mempertimbangkan tidak hanya pola sekuensial, tetapi juga konteks kata yang telah diidentifikasi sebelumnya. Untuk meningkatkan akurasi deteksi berita hoax, hasil dari metode lexicon based digunakan sebagai fitur tambahan dalam pelatihan model LSTM. Dengan integrasi ini, model LSTM tidak hanya mengandalkan pola sekuensial dalam teks, tetapi juga memperhitungkan kemunculan kata-kata kunci dari model lexicon based. Integrasi kedua pendekatan

ini diharapkan dapat meningkatkan kemampuan model dalam mendeteksi berita hoax dengan lebih akurat dan efisien. Model kombinasi tersebut dapat dilihat pada Gambar 3.2.



Gambar 3. 2 Model hybrid yang diusulkan

3.6. Desain Experiment

Eksperimen *dilakukan* untuk menguji model yang diusulkan. Penelitian ini menggunakan 2 experiment Parameter setting dan Dataset.

3.6.1 Parameter setting

Dalam Parameter setting ini menggunakan sepuluh skenario pengujian yang berbeda untuk mencari akurasi model yang terbaik. Skenarionya adalah dengan melakukan hyperparameter setting 3 parameter di LSTM untuk nilai parameter yang lain dikonfigurasi dengan max_len 200, epochs 10, threshold 0.5, dropout 0.4, recurrent_dropout 0.3, dan menggunakan loss_function 'binary_crossentropy'

untuk klasifikasi biner. Berikut skenario-skenario pengujian:

1. max_features

   Skenario pertama dilakukan dengan melakukan tuning pada jumlah maksimum fitur yang akan diproses oleh Tokenizer. Jumlah fitur ini sangat penting karena menentukan seberapa banyak informasi yang dapat diproses oleh model. Ukuran maksimum fitur yang diujikan dapat dilihat pada Tabel 3.6 pengujian dengan max_features

   Tabel 3. 6 pengujian dengan max_features

max_features	Accuracy	FI-SCORE
10,000	….	…
15,000	…	….
20,000		

2. embedding_dim

   Skenario ketiga dilakukan dengan menyesuaikan dimensi embedding yang digunakan pada layer embedding dalam model LSTM. Dimensi ini menentukan representasi kata yang lebih rinci. Variasi dimensi embedding yang diuji dapat dilihat pada Tabel 3.8.

   Tabel 3. 7 pengujian dengan embedding

embedding	Accuracy	FI-SCORE
128	….	…
150	….	….
300	….	…

3. batch_size

Skenario kelima melibatkan tuning pada batch size, yaitu jumlah sampel yang diproses dalam satu iterasi pelatihan. Ukuran batch yang berbeda-beda diuji untuk melihat dampaknya terhadap kecepatan pelatihan dan stabilitas. Detail variasi ukuran batch dapat dilihat pada Tabel 3.10.

Tabel 3. 8 pengujian dengan epochs

Batch size	Accuracy	FI-SCORE
32	….	…
64	….	….

4. optimizer

Skenario kedelapan dilakukan dengan menyesuaikan jenis optimizer yang digunakan selama pelatihan. Beberapa algoritma optimasi diuji untuk menentukan mana yang memberikan performa terbaik. Variasi optimizer yang diuji dapat dilihat pada Tabel 3.12.

Tabel 3. 9 pengujian dengan optimizer

optimizer	Accuracy	FI-SCORE
Adam	….	…
SGD (Stochastic Gradient Descent)	….	….
RMSprop	….	…

Adagrad	….	…
Adadelta	….	…
Nadam	….	…

3.6.2 Dataset Komposisi

Dataset yang digunakan dalam eksperimen ini terdiri dari komposisi yang seimbang antara konten hoax dan non-hoax. Komposisi ini didesain sedemikian rupa untuk memastikan bahwa model dapat belajar secara seimbang dari kedua jenis konten. Rasio yang sehingga model dapat dilatih untuk mendeteksi hoax dengan performa yang optimal tanpa bias terhadap salah satu kategori. Berikut adalah komposisi dataset berdasarkan jenis topik yang diuji dalam eksperimen ini:

Tabel 3. 10 Dataset Komposisi berdasarkan kategori

Kategori	**Jumlah Data**	**Konten Hoax (%)**	**Konten Non-Hoax (%)**
Politik	3000	60%	40%
Olahraga	2000	30%	70%

3.7. Evaluasi

Accuracy digunakan untuk mengukur sejauh mana model dapat mengklasifikasikan dengan benar seluruh label pada data uji. Persamaan untuk menghitung *Accuracy* dapat dilihat pada Persamaan (3.1).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3.1)$$

Dimana persamaan (3.2) sebagai berikut :

- TP : Jumlah kasus positif yang diprediksi dengan benar oleh model.
- TN : Jumlah kasus negatif yang diprediksi dengan benar oleh model
- FP : Jumlah kasus negatif yang salah diprediksi sebagai positif oleh model.
- FN : Jumlah kasus positif yang salah diprediksi sebagai negatif oleh model.

Dalam persamaan *accuracy* TP + TN merupakan jumlah total prediksi yang benar yang terdiri dari prediksi positif yang benar (TP) dan prediksi negatif yang benar (TN). TP + TN + FP + FN merupakan total jumlah data yang diuji oleh model.

Precision mendeskripsikan dalam mengukur sejauh mana prediksi positif yang dilakukan oleh model dengan benar. Persamaan untuk menghitung *Precision* dapat dilihat pada Persamaan (3.2).

$$Precision = \frac{TP}{TP+FP} \quad (3.2)$$

Dimana :

- TP : Jumlah contoh positif yang benar-benar diklasifikasikan dengan benar oleh model.
- FP : Jumlah contoh negatif yang salah diklasifikasikan sebagai positif oleh model.

Recall menentukan kapasitas model untuk mendeteksi atau mengidentifikasi label yang sesuai dalam kumpulan data. Persamaan untuk

menghitung *Recall* dapat dilihat pada Persamaan (3.3).

$$Recall = \frac{TP}{TP+FN} \quad (3.3)$$

Dimana :

- TP : Jumlah instance positif yang secara tepat diprediksi sebagai positif oleh model.
- FN : Jumlah instance positif yang seharusnya diprediksi sebagai positif, namun salah prediksi sebagai negatif oleh model.

Dalam pengujian ini, akan dilakukan eksplorasi menggunakan dalam membangun model kombinasi Lexicon based dan LSTM untuk deteksi berita hoax. Dengan model LSTM konvensional, yang merupakan model dasar tanpa penggunaan konvolusi matriks. Konfigurasi parameter seperti jumlah maksimum fitur, panjang maksimum urutan, dimensi embedding, jumlah epochs, batch size, fungsi kerugian, optimizer, dropout, dan recurrent dropout akan ditentukan untuk menetapkan dasar kinerja yang menjadi pembanding dengan model yang lebih kompleks.

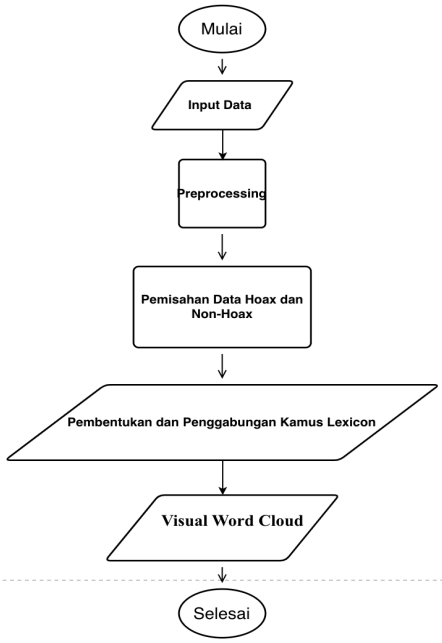
3.8. Kesimpulan dan Hasil

hasil pengujian adalah tahap pertama, dan penarikan kesimpulan adalah tahap akhir dari kegiatan analisis. Hasil dan temuan penelitian disajikan dalam kesimpulan. Hasilnya akan membahas dan menjawab masalah awal penelitian.

BAB IV
METODE LEXICON BASED

4.1. Desain Model Lexicon Based

Dalam bab ini, akan dijelaskan pola dan rancangan Model Lexicon Based yang dapat dilihat pada alur tergambar pada Gambar 4.1.



Gambar 4. 1 Desain Model Lexicon Based

Dalam gambar 4.1 Flowchart tersebut menggambarkan alur proses pembentukan model berbasis leksikon untuk analisis data. Dimulai dengan langkah Input Data, data yang dimasukkan kemudian melalui tahap Preprocessing, di mana data diproses agar siap untuk dianalisis, pertama melalui tokenisasi, penghilangan stopwords, atau stemming. Setelah itu, dilakukan Pemisahan Data Hoax dan Non-Hoax yang sudah terlabel. Langkah selanjutnya Pembentukan dan Penggabungan Kamus Lexicon, di mana leksikon disusun

untuk membantu mengidentifikasi kata-kata kunci atau pola yang dapat mengklasifikasikan data sebagai hoax atau non-hoax. Terakhir, hasil analisis divisualisasikan dalam bentuk Visual Word Cloud, yang memberikan gambaran visual tentang kata-kata yang paling sering muncul.

Hasil pembentukan lexicon-based ini nantinya akan digunakan sebagai bagian dari preprocessing untuk diimplementasikan pada model LSTM, guna meningkatkan akurasi dalam klasifikasi hoax dan non-hoax.

4.2. Implementasi Model Lexicon Based

Dalam tahap Pengumpulan Data dan Preprocessing, setelah data dari situs Turnbackhoax.id (5000 data hoax), Kompas.com (3000 data non-hoax), dan Detik (2000 data non-hoax) dimuat dari file CSV ditujukan pada gambar

Tabel 4. 1 Data Hoax dan non-hoax

FullText	CleanedText	CashfoldedText	TokenizedText	StemmedText	Label
Pandangan jauh ke depan (foresight) perlu dimiliki oleh para Capres-Cawapres, karena bangsa kita berada pada situasi dunia yang semakin	pandangan jauh ke depan foresight perlu dimiliki oleh para caprescawapres karena bangsa kita berada pada situasi dunia yang semakin rentan dan kompleks	pandangan jauh ke depan foresight perlu dimiliki oleh para caprescawapres karena bangsa kita berada pada situasi dunia yang semakin rentan dan kompleks	['pandangan', 'jauh', 'ke', 'depan', 'foresight', 'perlu', 'dimiliki', 'oleh', 'para', 'caprescawapres', 'karena', 'bangsa', 'kita', 'berada', 'pada', 'situasi', 'dunia', 'yang', 'semakin', 'rentan', 'dan', 'kompleks']	pandang foresight milik caprescawapres bangsa situasi dunia rentan kompleks	0

rentan dan kompleks.					
Menteri Perdagangan RI Zulkifli Hasan menjadi pembicara di acara Simposium Kementerian Pertahanan Tahun 2023.	menteri perdagangan ri zulkifli hasan menjadi pembicara di acara simposium kementerian pertahanan tahun	menteri perdagangan ri zulkifli hasan menjadi pembicara di acara simposium kementerian pertahanan tahun	['menteri', 'perdagangan', 'ri', 'zulkifli', 'hasan', 'menjadi', 'pembicara', 'di', 'acara', 'simposium', 'kementerian', 'pertahanan', 'tahun']	menteri dagang ri zulkifli hasan bicara acara simposium menteri tahan	0
Ketua Harian Partai Gerindra Dasco mengatakan TKN Prabowo-Gibran akan diumumkan Kamis depan. Dia menyebut pengumuman akan dilaksanakan secara keseluruhan.	ketua harian partai gerindra dasco mengatakan tkn prabowogibran akan diumumkan kamis depan dia menyebut pengumuman akan dilaksanakan secara keseluruhan	ketua harian partai gerindra dasco mengatakan tkn prabowogibran akan diumumkan kamis depan dia menyebut pengumuman akan dilaksanakan secara keseluruhan	['ketua', 'harian', 'partai', 'gerindra', 'dasco', 'mengatakan', 'tkn', 'prabowogibran', 'akan', 'diumumkan', 'kamis', 'depan', 'dia', 'menyebut', 'pengumuman', 'akan', 'dilaksanakan', 'secara', 'keseluruhan']	ketua hari partai gerindra dasco tkn prabowogibran umum kamis sebut umum laksana	1

Relawan Pengusaha Muda Nasional (REPNAS) Indonesia Maju sukses menggelar rapat konsolidasi di Hotel Ambhara, Jakarta Selatan.	relawan pengusaha muda nasional repnas indonesia maju sukses menggelar rapat konsolidasi di hotel ambhara jakarta selatan	relawan pengusaha muda nasional repnas indonesia maju sukses menggelar rapat konsolidasi di hotel ambhara jakarta selatan	['relawan', 'pengusaha', 'muda', 'nasional', 'repnas', 'indonesia', 'maju', 'sukses', 'menggelar', 'rapat', 'konsolidasi', 'di', 'hotel', 'ambhara', 'jakarta', 'selatan']	rawan usaha muda nasional repnas indonesia maju sukses gelar rapat konsolidasi hotel ambhara jakarta selatan	1

Gambar 4. 2 Data Non-Hoax

proses preprocessing dilakukan menggunakan beberapa teknik, salah satunya adalah penghapusan stopwords. Untuk menghilangkan stopwords, digunakan Sastrawi, sebuah library khusus untuk bahasa Indonesia yang memudahkan dalam memfilter kata-kata umum atau kata-kata yang tidak memberikan kontribusi signifikan dalam analisis teks, seperti "yang", "dan", atau "untuk". Stopwords ini diambil dari file CSV eksternal yang berisi daftar kata yang tidak relevan, sesuai dengan bahasa yang digunakan, yaitu bahasa Indonesia. Dengan memanfaatkan Sastrawi, proses pembersihan data menjadi lebih efisien dan relevan untuk analisis lebih lanjut dalam klasifikasi hoax dan non-hoax.

Tabel 4. 2 Data yang digunakan

Situs	Jumlah Data	Tahun	Label
Turnbackhoax.id	5000	2019-2023	Hoax
Kompas.com	3000	2019-2023	Non-Hoax
Detik	2000	2019-2023	Non-Hoax

4.2.1. Hasil Preprocessing Data

Data yang dikumpulkan dalam penelitian ini masih dalam kondisi kotor, sehingga diperlukan proses pembersihan dan pengolahan teks. Code Processing Data sebagai berikut :

Source Code Processing Data

```
# Instalasi Sastrawi jika belum diinstal
!pip install Sastrawi

import pandas as pd
import re
from nltk.tokenize import word_tokenize
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
import nltk
import time
import matplotlib.pyplot as plt

# Unduh resource dari NLTK (jika belum diunduh sebelumnya)
nltk.download('punkt')

# Fungsi untuk membersihkan teks
def membersihkan_teks(teks):
    if pd.isna(teks):  # Cek apakah teks adalah NaN
        return ''
    teks = teks.lower()  # Ubah teks menjadi huruf kecil
    teks = re.sub(r'\d+', '', teks)  # Hapus angka
    teks = re.sub(r'[^\w\s]', '', teks)  # Hapus tanda baca
    teks = re.sub(r'\s+', ' ', teks).strip()  # Hapus spasi
berlebih
    return teks

# Fungsi Cashfolding (normalisasi karakter non-ASCII)
def cashfolding_teks(teks):
    return teks.encode('ascii', 'ignore').decode('ascii')

# Buat objek stemmer
```

```
stemmer = StemmerFactory().create_stemmer()

# Fungsi untuk tokenisasi teks
def tokenisasi_teks(teks):
    token = word_tokenize(teks)
    return token

# Fungsi untuk menghapus stopword
def hapus_stopword(teks, daftar_stopword):
    token = word_tokenize(teks)
    token_tersaring = [kata for kata in token if kata not in
daftar_stopword]
    teks_tersaring = ' '.join(token_tersaring)
    return teks_tersaring

# Fungsi untuk stemming teks
def stemming_teks(teks):
    token = word_tokenize(teks)
    token_stem = [stemmer.stem(kata) for kata in token]
    teks_stem = ' '.join(token_stem)
    return teks_stem

# Fungsi untuk preprocessing DataFrame dan menghitung waktu setiap
proses
def preprocessing_dataframe(dataframe, daftar_stopword):
    if 'TeksLengkap' not in dataframe.columns:
        raise ValueError("Kolom 'TeksLengkap' tidak ditemukan
dalam DataFrame.")
    if 'Label' not in dataframe.columns:
        dataframe['Label'] = 0  # Tambahkan label default jika
kolom tidak ada

    waktu_proses = {}

    # Proses membersihkan teks
    mulai = time.time()
    dataframe['TeksBersih']                                    =
dataframe['TeksLengkap'].apply(membersihkan_teks)
    waktu_proses['Pembersihan'] = time.time() - mulai

    # Proses cashfolding
    mulai = time.time()
    dataframe['TeksCashfolding']                               =
dataframe['TeksBersih'].apply(cashfolding_teks)
    waktu_proses['Cashfolding'] = time.time() - mulai

    # Proses tokenisasi
    mulai = time.time()
    dataframe['TeksToken']                                     =
dataframe['TeksCashfolding'].apply(tokenisasi_teks)
    waktu_proses['Tokenisasi'] = time.time() - mulai

    # Proses hapus stopword
    mulai = time.time()
```

```
    dataframe['TeksTersaring']                                    =
dataframe['TeksCashfolding'].apply(lambda                    teks:
hapus_stopword(teks, daftar_stopword))
    waktu_proses['Hapus Stopword'] = time.time() - mulai

    # Proses stemming
    mulai = time.time()
    dataframe['TeksStem']                                         =
dataframe['TeksTersaring'].apply(stemming_teks)
    waktu_proses['Stemming'] = time.time() - mulai

    # Visualisasi waktu pemrosesan
    visualisasi_waktu(waktu_proses)

    return        dataframe[['TeksLengkap',        'TeksBersih',
'TeksCashfolding',  'TeksToken',  'TeksTersaring',  'TeksStem',
'Label']]

# Fungsi untuk memvisualisasikan waktu pemrosesan
def visualisasi_waktu(waktu_proses):
    plt.bar(waktu_proses.keys(),          waktu_proses.values(),
color='blue')
    plt.xlabel('Tahapan Proses')
    plt.ylabel('Waktu (detik)')
    plt.title('Waktu Pemrosesan Teks')
    plt.show()

# Baca daftar stopword dari file CSV
data_stopword  =  pd.read_csv('/content/stopwordbahasa.csv')    #
Sesuaikan nama file
daftar_stopword = data_stopword['ada'].tolist()

# Baca data yang akan diproses
data   =   pd.read_csv('/content/data_sport_formated.csv')      #
Sesuaikan nama file

# Preprocessing DataFrame
data_terproses = preprocessing_dataframe(data, daftar_stopword)

# Simpan hasil ke dalam file CSV
nama_file_hasil = '/content/data_sport_preprocessed.csv'
data_terproses.to_csv(nama_file_hasil, index=False)

# Tampilkan beberapa data terproses
data_hasil = pd.read_csv(nama_file_hasil)
print(data_hasil.head())
```

Kode ini melakukan preprocessing teks pada dataset dengan berbagai langkah untuk mempersiapkan data untuk analisis lebih lanjut. Pertama, teks dibersihkan melalui proses cleansing yang menghapus angka, tanda baca, dan mengubah semua

karakter menjadi huruf kecil. Selanjutnya, proses cashfolding dilakukan untuk menormalkan teks dengan menghilangkan karakter non-ASCII seperti aksen. Setelah itu, teks dibagi menjadi kata-kata menggunakan tokenisasi, dan stopwords dihapus dengan merujuk pada daftar kata tidak penting yang diambil dari file CSV. Kemudian, dilakukan stemming untuk mengubah kata-kata menjadi bentuk dasarnya menggunakan pustaka Sastrawi. Setiap proses diukur waktunya, dan hasil pemrosesan waktu ini divisualisasikan menggunakan grafik. Semua langkah ini diterapkan pada kolom `FullText` dalam DataFrame, dan hasil akhir disimpan dalam file CSV yang baru, siap digunakan dilakukan pemisahan data.

4.2.2. Pemisahan Data Hoax dan Non-Hoax

Data dibagi menjadi dua kelompok berdasarkan label: hoax (Label = 1) dan non-hoax (Label = 0). Baris dengan nilai kosong pada kolom *StemmedText* dihapus untuk memastikan data bersih dan siap diproses. Hasilnya, dataset kini terdiri dari dua bagian (hoax dan non-hoax) tanpa data kosong.

4.2.3. Pembentukan Kamus Lexicon

Pada bagian ini, kami menggunakan fungsi generate_lexicon untuk membentuk kamus kata (lexicon) dari data hoax dan non-hoax. Proses ini menggunakan CountVectorizer yang mengabaikan kata-kata umum (stopwords) dalam bahasa Indonesia. Berikut adalah langkah-langkah yang terjadi:

Proses:

1. **Pembersihan Stopwords**: Stopwords yang digunakan mencakup kata-kata umum dalam bahasa Indonesia.

2. **Frekuensi Kata**: Untuk setiap kata yang tidak termasuk dalam stopwords, CountVectorizer menghitung berapa kali kata tersebut muncul dalam setiap teks. Dalam tahap ini, parameter frekuensi digunakan untuk menyaring kata-kata, yaitu kata yang muncul lebih dari 200 kali akan disimpan dalam korpus leksikon.
3. **Pembuatan DataFrame**: Frekuensi kemunculan kata untuk masing-masing kategori (hoax dan non-hoax) disimpan dalam bentuk DataFrame. Setiap kata diberi label sesuai dengan kategorinya (1 untuk hoax dan 0 untuk non-hoax).

Hasil

Hasil model Lexicon Based menyajikan daftar kata yang diklasifikasikan sebagai hoaks (1) atau non-hoaks (0). Tabel 4.1 menampilkan contoh kata beserta labelnya untuk membedakan teks hoaks dan non-hoaks berdasarkan kemunculan kata-kata relevan.

Tabel 4. 3 Hasil Pembuatan Model Lexicon Based

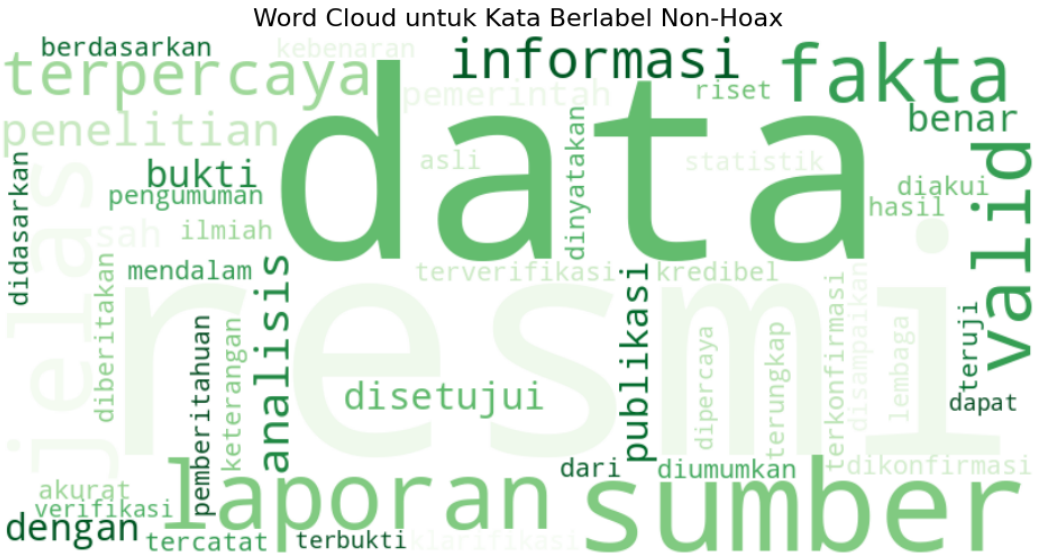
No	***word***	***label***
1	dinas	0
2	perang dingin baru	1
3	revolusi palsu	1
4	penelitian politik	0
5	perlindungan hak	0
6	pemerintahan bayangan	1
7	advokasi	0
8	perlindungan hak	0
9	badan publik	0
10	advokasi	0
...	...	...
1000	korupsi besar	1

Setelah terbentuk kamus untuk data hoax dan non-hoax, keduanya digabungkan menjadi satu DataFrame. Data ini kemudian disimpan ke dalam file CSV untuk memudahkan akses. **Hasil**: Lexicon gabungan hoax dan non-hoax berhasil disimpan dalam file CSV bernama `combined_word_lexicon.csv`

4.2.4. Visualisasi Word Cloud

Pada tahap ini, frekuensi kata yang sudah dihitung divisualisasikan dalam bentuk *word cloud*. Visualisasi ini menampilkan kata-kata yang paling sering muncul di data hoax dan non-hoax.

Hasil:



Gambar 4. 3 Visual *Word cloud* dari teks hoax dan non-hoax

Dari hasil visual yang berhasil ditampilkan, memperlihatkan kata-kata yang sering muncul dalam kedua kategori.

Code Untuk menampilkan *word cloud* hoax:

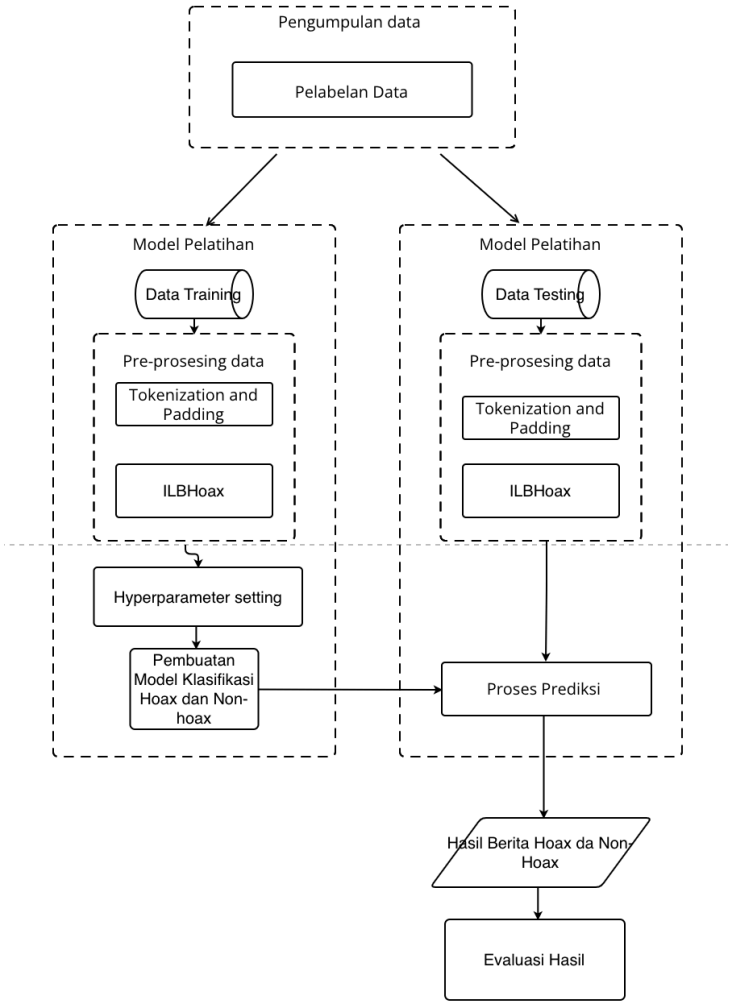
```
display_wordcloud(hoax_lexicon, 'Word Cloud of Hoax Keywords')
display_wordcloud(non_hoax_lexicon, 'Word Cloud of Non-Hoax Keywords')
```

Word cloud memberikan gambaran visual dari kata kunci yang dominan, mempermudah identifikasi karakteristik masing-masing jenis teks. Pada *word cloud* hoax, kata-kata yang sering digunakan dalam berita palsu tampil dengan ukuran lebih besar. Ini menandakan kata-kata yang sering muncul untuk menimbulkan keraguan atau misinformasi. Sebaliknya, *word cloud* non-hoax menampilkan kata-kata yang mendukung informasi yang akurat dan faktual. Visualisasi ini membantu melihat perbedaan linguistik antara teks hoax dan non-hoax dengan lebih jelas, dan mempermudah analisis teks untuk mendeteksi hoax secara lebih efisien.

BAB V
METODE LSTM (Long Short-Term Memory)

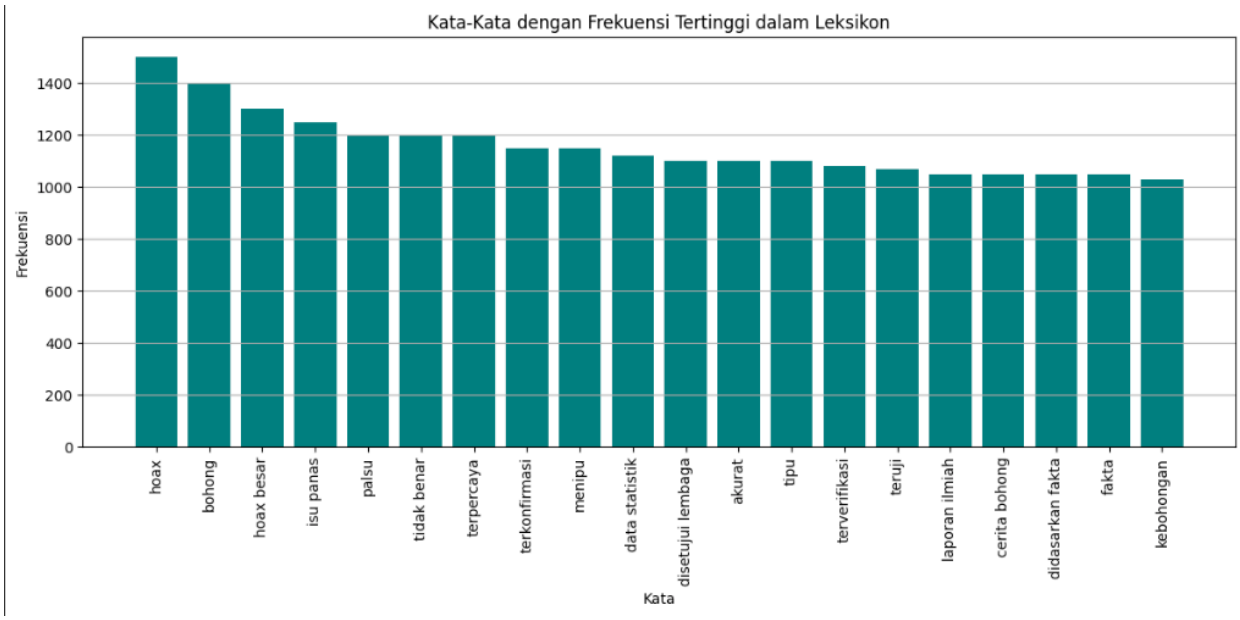
5.1.Desain LSTM

Dalam bab ini melanjutkan dari hasil bab sebelumnya, terbuat Indonesia Lexicon Based Hoax atau ILBHoax kami menyebutnya. Hasil itu akan digunakan di dalam preprosesing LSTM seperti digambarkan di gambar 5.1



Gambar 5. 1 Desain Model Kombinasi Lexicon Based dan LSTM

Kombinasi metode LSTM dengan pre-processing berbasis **Indonesian Lexicon-Based Hoax (ILBHoax)** meningkatkan akurasi model dalam mendeteksi berita hoax. Pada tahap pre-processing, data training dan testing melalui proses tokenisasi, padding, dan analisis ILBHoax. Tokenisasi mengubah teks menjadi deretan token untuk analisis lanjut, sementara padding memastikan konsistensi panjang input bagi model. **ILBHoax** berfungsi sebagai filter linguistik tambahan, mengidentifikasi pola khas dalam teks hoax Indonesia, sehingga menciptakan representasi fitur yang lebih relevan dan meningkatkan kemampuan LSTM dalam mendeteksi pola hoax yang sering muncul frekuensi ditujukan pada gambar 5.2.



Gambar 5. 2 Kata-kata ILBhoax Frekuensi

Grafik pada gambar 5. 2 Kata-kata ILBhoax Frekuensi ini menunjukkan kata-kata dengan frekuensi tertinggi dalam leksikon yang dianalisis. Pada sumbu horizontal (X-axis) ditampilkan kata-kata yang sering muncul, sementara sumbu vertikal (Y-

axis) menunjukkan frekuensi atau jumlah kemunculan masing-masing kata dalam dataset.

Beberapa kata yang muncul dengan frekuensi tinggi adalah "hoax," "bohong," "isu palsu," dan "fitnah," yang mengindikasikan bahwa kata-kata ini sering digunakan dalam konteks berita atau informasi yang mengandung hoax atau misinformasi. Kata-kata lain seperti "palsu," "tipu," dan "berita salah" juga menunjukkan pola yang berkaitan dengan tema kebohongan atau penyebaran informasi palsu. Frekuensi kata-kata ini dapat membantu dalam mengidentifikasi konten yang mungkin mengandung misinformasi atau hoax berdasarkan analisis leksikal.

5.2.Implementasi Lexicon based dan LSTM

5.2.1. Memuat Data

Pada langkah awal ini, kita memulai dengan memuat dataset yang berisi teks dan label. Dataset ini telah mengalami tahap preprocessing seperti stemming, di mana akhiran dari kata-kata dihilangkan untuk menyederhanakan analisis Bahasa. Seperti dijelaskan di bab sebelumnya di bagian preprosesing.

Teks yang telah diproses dimuat ke dalam variabel X, sedangkan label klasifikasinya (0 untuk non-hoax dan 1 untuk hoax) dimuat ke dalam y. Keunggulan: Dengan memuat dataset yang sudah diproses ini, kita mempersiapkan data untuk dilatih oleh model. Ditambah dengan bantuan leksikon, model

mendapatkan pemahaman tambahan tentang kata-kata spesifik yang terkait dengan hoax atau non-hoax, bahkan pada dataset yang tidak terlalu besar.

5.2.2. Memuat Leksikon untuk Pendekatan Berbasis Leksikon

Pada tahap ini, leksikon dimuat agar dapat digunakan sebagai fitur tambahan dalam model. Leksikon ini membantu memperkaya proses klasifikasi dengan memberikan skor tambahan pada kata-kata tertentu yang diketahui sering muncul dalam berita hoax.

Prosesnya, Dataset leksikon dimuat ke dalam DataFrame Pandas. Setiap kata di dalam teks akan dicocokkan dengan leksikon ini untuk mendapatkan skor tambahan.

Keunggulan dari pendekatan berbasis leksikon memungkinkan model untuk belajar dari pengetahuan domain yang sudah ada. Ini penting karena tidak semua pola dalam teks bisa dipelajari hanya dari data. beberapa kata memang secara inheren memiliki kecenderungan untuk muncul dalam konteks hoax, dan leksikon membantu mengenali kata-kata ini.

5.2.3. Pra-pemrosesan: Tokenisasi dan Padding

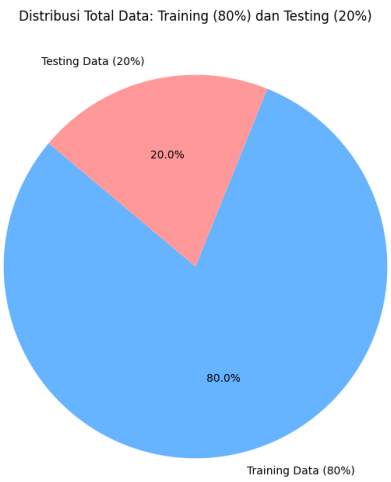
Setelah data dan leksikon dimuat, langkah berikutnya adalah melakukan tokenisasi teks, di mana setiap kata diubah menjadi angka unik yang dapat diproses oleh model. Selain itu, setiap kata diberikan skor dari leksikon, sehingga model tidak hanya menerima urutan angka hasil tokenisasi, tetapi juga skor tambahan yang menunjukkan kecenderungan hoaks atau non-hoaks.

Prosesnya meliputi teks di-tokenisasi, dan skor leksikon ditambahkan untuk setiap kata. Padding diterapkan agar semua urutan memiliki panjang yang sama, memungkinkan pemrosesan secara batch oleh model.

Dengan menggabungkan urutan token teks dan skor leksikon, input yang diberikan ke model menjadi lebih kaya. Ini memungkinkan model tidak hanya bergantung pada pola teks umum, tetapi juga mendapatkan informasi tambahan tentang kata-kata yang relevan dengan hoaks atau non-hoaks.

5.2.4. Pembagian Data Latih dan Uji

Setelah melalui proses pra-pemrosesan, data dibagi menjadi dua bagian utama, yaitu data latih dan data uji. Data latih berfungsi untuk melatih model dalam mengenali pola dan hubungan antara teks dengan label hoaks atau non-hoaks, sementara data uji digunakan untuk mengevaluasi kinerja model pada data baru yang belum pernah dilihat sebelumnya. Gambar 5.3 menunjukkan persentase distribusi data, dengan 80% data digunakan untuk melatih model (data latih) dan 20% untuk menguji model (data uji). dalam penelitian ini, menggambarkan alokasi data yang digunakan pada setiap tahap.



Gambar 5. 3 Persentase Distribusi Data Training dan Testing

Dengan proporsi perbandingan sebesar 80% : 20% dari keseluruhan 10.000 judul berita, dilakukan pengujian dengan dua konfigurasi rasio pembagian data latih dan data uji: yaitu 80% data latih dan 20% data uji serta 50% data latih dan 50% data uji. Berikut ini adalah perbandingan data yang digunakan dalam setiap rasio.

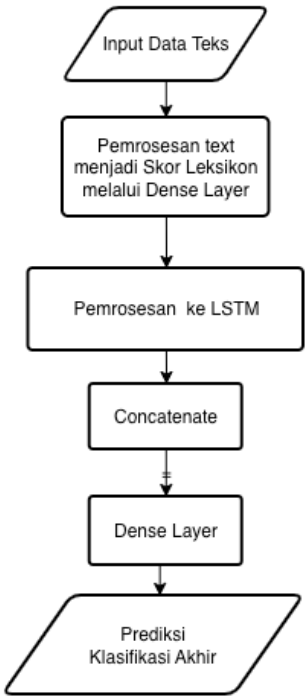
Tabel 5. 1 Pembagian Data Latih dan Data Uji

Rasio	**Total Data Latih**	**Total Data Uji**
80% : 20%	1.600	400
70% : 30%	1.400	600
60% : 40%	1.200	800
50% : 50%	1.000	1.000

5.2.5. Membangun Model Lexicon Based dan LSTM

Model ini dibangun dengan menggunakan dua input: teks yang telah di-tokenisasi dan skor leksikon. LSTM berfungsi untuk memahami pola dalam teks

melalui embedding dan pemrosesan urutan, sementara skor leksikon diproses melalui lapisan dense untuk mengidentifikasi pola berbasis kata. Hasil dari kedua input ini digabungkan untuk membuat prediksi.



Gambar 5. 4 Pembuatan Model Kombinasi Lexicon based dan LSTM

Proses deteksi hoax ini dimulai dengan **Input Data Teks**, di mana data teks mentah yang akan dianalisis dimasukkan sebagai bahan dasar untuk tahap-tahap berikutnya. Setelah itu, dilakukan **Pemrosesan Skor Leksikon melalui Dense Layer**. Pada tahap ini, data teks diproses untuk menghasilkan skor leksikon, yaitu nilai prediktif yang mencerminkan kecenderungan teks terhadap kategori *hoax* atau *non-hoax*. Dense Layer membantu mengenali pola numerik yang relevan dari kata-kata dalam leksikon tersebut, memberikan indikasi awal klasifikasi teks. Selanjutnya, proses berlanjut ke **Pemrosesan ke LSTM** apabila hasil dari skor

leksikon belum cukup kuat atau masih ambigu. Di tahap ini, data teks diproses lebih mendalam melalui LSTM (Long Short-Term Memory) untuk menganalisis pola dan konteks antar kata dalam teks. LSTM berfungsi menangkap makna semantik dan hubungan konteks, yang memungkinkan model untuk memahami isi teks secara lebih komprehensif.

Setelah kedua hasil diperoleh, baik dari skor leksikon maupun LSTM, tahap berikutnya adalah **Concatenate**, di mana informasi dari kedua proses ini digabungkan. Penggabungan ini memungkinkan model untuk memanfaatkan baik informasi prediktif dari leksikon maupun konteks dari LSTM, sehingga menghasilkan satu representasi yang lebih lengkap dan kaya.

Proses dilanjutkan dengan **Dense Layer Akhir**, di mana hasil gabungan dari kedua proses tersebut diproses melalui layer ini untuk menghasilkan probabilitas akhir menggunakan fungsi aktivasi, biasanya *sigmoid*, yang mengindikasikan prediksi akhir.

Terakhir, dilakukan **Prediksi Klasifikasi Akhir** untuk menentukan apakah teks yang dianalisis termasuk dalam kategori *hoax* atau *non-hoax*. Keunggulan dari proses seri ini adalah penggunaan skor leksikon sebagai langkah awal, diikuti oleh analisis kontekstual melalui LSTM untuk memperkuat keputusan akhir. Pendekatan ini memaksimalkan ketepatan deteksi dengan mengombinasikan kekuatan dari dua teknik yang saling melengkapi.

5.2.6. Melatih Model dan Menyimpan Hasilnya

1. Proses Training

   Berikut adalah tabel komposisi uji coba Untuk menghitung total skenario pengujian dari berbagai kombinasi parameter.

   Rumus:

   Jika kita memiliki n parameter, dan setiap parameter memiliki $k_1, k_2, \dots, k_n$ pilihan, maka total skenario yang dapat dihasilkan adalah:

   $$\text{Total Skenario} = k_1 \times k_2 \times \dots \times k_n$$

   Penerapan pada Kasus Ini:

   - **Max Features** memiliki 3 pilihan: 10,000, 15,000, dan 20,000.
   - **Embedding Dimension** memiliki 3 pilihan: 128, 150, dan 300.
   - **Batch Size** memiliki 2 pilihan: 32 dan 64.

   Dengan menerapkan aturan perkalian:

   $$3 \times 3 \times 2 = 18 \text{ skenario}$$

Tabel 5. 2 Parameter Setting pada LSTM

No.	Max Features	Embedding Dim	Batch Size
1	10,000	128	32
2	10,000	128	64
3	10,000	150	32
4	10,000	150	64
5	10,000	300	32
6	10,000	300	64
7	15,000	128	32
8	15,000	128	64

9	15,000	150	32
10	15,000	150	64
11	15,000	300	32
12	15,000	300	64
13	20,000	128	32
14	20,000	128	64
15	20,000	150	32
16	20,000	150	64
17	20,000	300	32
18	20,000	300	64

Dari tabel 5.1 menyajikan variasi parameter yang bisa diuji untuk mendapatkan konfigurasi terbaik dalam proses pelatihan model LSTM. Untuk hasil dari uji parameter dapat dilihat di tabel 5.2.

2. Proses Testing

Perhitungan Uji ke 1:

Confusion Matrix:

	Prediksi Non-Hoax	Prediksi Hoax
Aktual Non-Hoax	98	10
Aktual Hoax	12	990

Dari matriks di atas:

- **True Positives (TP)** = 98 (Hoax yang diprediksi benar)
- **True Negatives (TN)** = 990 (Non-Hoax yang diprediksi benar)
- **False Positives (FP)** = 10 (Non-Hoax yang diprediksi sebagai Hoax)

- **False Negatives (FN)** = 12 (Hoax yang diprediksi sebagai Non-Hoax)

$$\textbf{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{98 + 990}{1110} = 0.9802 \text{ atau } 98.02\%$$

$$\textbf{Presisi} = \frac{TP}{TP + FP} = \frac{98}{98 + 10} = 0.9074 \text{ atau } 90.74\%$$

$$\textbf{Recall} = \frac{TP}{TP + FN} = \frac{98}{98 + 12} = 0.8909 \text{ atau } 89.09\%$$

$$\textbf{F1 Score} = 2 \times \frac{\text{Presisi} \times \text{Recall}}{\text{Presisi} + \text{Recall}} = 2 \times \frac{0.9074 \times 0.8909}{0.9074 + 0.8909} \approx 89.91\%$$

Pada bagian di atas adalah contoh perhitungan Akurasi, Recall, dan F1-Score, didasarkan pada data dari tabel 5.4 yang berisi matriks konfusi sebagai berikut:

Tabel 5. 3 Confusion Matrix

	Nilai Sebenarnya	
	Hoax	**Non-Hoax**
Prediksi 1	98 (TP)	10 (FP)
	12 (FN)	990 (TN)
Prediksi 2	992 (TP)	6 (FP)
	6 (FN)	994 (TN)
Prediksi 3	978 (TP)	22 (FP)
	30 (FN)	970 (TN)
Prediksi 4	983(TP)	10(FP)
	7(FN)	990 (TN)
Prediksi 5	987(TP)	17(FP)
	13(FN)	983(TN)
Prediksi 6	991(TP)	14(FP)
	9(FN)	986(TN)
Prediksi 7	989 (TP)	11(FP)
	11(FN)	989 (TN)

Prediksi 8	989 (TP)	8(FP)
	11(FN)	989 (TN)
Prediksi 9	986 (TP)	10 (FP)
	14 (FN)	990 (TN)
Prediksi 10	989 (TP)	7 (FP)
	11 (FN)	993 (TN)
Prediksi 11	968 (TP)	29 (FP)
	32 (FN)	971 (TN)
Prediksi 12	991 (TP)	14 (FP)
	9 (FN)	986 (TN)
Prediksi 13	978 (TP)	11 (FP)
	22 (FN)	989 (TN)
Prediksi 14	989 (TP)	6 (FP)
	11 (FN)	994 (TN)
Prediksi 15	957 (TP)	42(FP)
	43 (FN)	958 (TN)
Prediksi 16	991 (TP)	14 (FP)
	9 (FN)	986 (TN)
Prediksi 17	991 (TP)	9 (FP)
	13 (FN)	987 (TN)
Prediksi 18	(TP)	(FP)
	(FN)	(TN)

Berikut Tabel 5.5 menunjukkan hasil pengujian berbagai parameter untuk menentukan kombinasi terbaik dalam mendeteksi hoax dan non-hoax secara akurat

Tabel 5. 4 Hasil Uji Parameter

No	***precision***	***Recall***	***Fi-score***	***accuracy***	**Time**
1	0.9802	0.9074	0.890	0.8991	426.50
2	0.9930	0.9920	0.9930	0.9930	279.30
3	0.9740	0.9780	0.9740	0.9740	348.1
4	0.9900	0.9930	0.9915	0.9915	292.33
5	0.9831	0.983	0.9850	0.9850	505.73

6	0.9861	0.9910	0.9885	0.9885	330.34
7	0.9890	0.9890	0.9890	0.9890	271.76
8	0.9880	0.9920	0.9900	0.9900	209.24
9	0.9900	0.9860	0.9880	0.9880	283.50
10	0.9930	0.9890	0.9910	0.9910	229.74
11	0.9709	0.9680	0.9695	0.9695	421.55
12	0.9861	0.9910	0.9885	0.9885	324.33
13	0.9889	0.9780	0.9834	0.9835	274.33
14	0.9940	0.9890	0.9915	0.9915	279.70
15	0.9580	0.9570	0.9575	0.9575	386.79
16	0.9861	0.9910	0.9885	0.9885	351.52
17	0.9890	0.9871	0.9910	0.9890	559.52
18	0.9880	0.9900	0.9890	0.9890	434.65

Berdasarkan hasil pengujian ditabel 5.5 , uji ke-2, ke-8, dan ke-14 menunjukkan performa terbaik dengan kombinasi parameter yang berbeda. Uji ke-2 menggunakan parameter "Max Features" sebesar 10,000, "Embedding Dim" sebesar 128, dan "Batch Size" sebesar 64. Dengan konfigurasi ini, model berhasil mencapai akurasi tertinggi (0.9930) dalam waktu yang cukup efisien (279.30 detik). Kombinasi ini menunjukkan bahwa penggunaan "Embedding Dim" yang lebih rendah pada ukuran batch yang besar dapat menghasilkan kinerja optimal, dengan tetap menjaga waktu komputasi yang rendah.

Selanjutnya, uji ke-8 menggunakan "Max Features" yang lebih besar yaitu 15,000, "Embedding Dim" sebesar 128, dan "Batch Size" 64, menghasilkan akurasi 0.9900 dengan waktu terpendek di antara ketiga uji terbaik, yaitu 209.24 detik. Sementara itu, uji ke-14, yang kemungkinan menggunakan konfigurasi serupa

dengan "Embedding Dim" 128 dan "Batch Size" 64, mencapai akurasi yang mendekati hasil terbaik, yaitu 0.9915 dengan waktu 279.70 detik.

Disimpulkan Ketiga hasil ini menunjukkan bahwa peningkatan pada "Max Features" dapat membantu mempercepat waktu pemrosesan tanpa penurunan signifikan pada akurasi, terutama pada **uji ke-8** yang menjadi pilihan paling efisien untuk pengujian dengan waktu singkat.

Setelah menentukan parameter terbaik pada uji model, langkah berikutnya adalah mengoptimalkan performa model dengan berbagai pengaturan optimizer pada LSTM. Optimizer digunakan untuk mempercepat dan menstabilkan proses pelatihan, sehingga model dapat mencapai konvergensi secara efisien. Tabel 5.3 berikut menyajikan hasil pengujian beberapa konfigurasi optimizer yang digunakan untuk memperoleh kombinasi pengaturan terbaik pada model LSTM.

Tabel 5. 5 Optimizer Setting pada LSTM

Optimizer	***precision***	***Recall***	***Fi-score***	***accuracy***	***Time***
Adam	0.9880	0.9920	0.9900	0.990	209.24
SGD (Stochastic Gradient Descent)	0.9920	0.9880	0.9900	0.990	222.38
RMSprop	0.9881	0.6660	0.7957	0.829	203.55
Adagrad	0.9890	0.9920	0.9905	0.990	229.41
Adadelta	0.5251	0.5760	0.5494	0.527	246.76
Nadam	0.9920	0.9910	0.9915	0.991	269.31

Dari hasil tabel 5.6 Pengujian beberapa algoritma optimizer—Adam, SGD, RMSprop, Adagrad, Adadelta, dan Nadam—dilakukan untuk menilai kinerja berdasarkan **precision**, **recall**, **F1-score**, **accuracy**, dan **waktu pemrosesan**. Berikut ini hasil dan analisisnya:

1. **Adam**: Adam mencapai akurasi 0.9900 dengan waktu pemrosesan 209.24 detik. Algoritma ini memadukan kelebihan RMSprop dan momentum, memberikan keseimbangan antara akurasi tinggi dan efisiensi waktu. Dengan waktu yang lebih singkat dibandingkan optimizers lain dengan akurasi setara, Adam menjadi pilihan yang ideal untuk aplikasi yang mengutamakan akurasi tanpa mengorbankan waktu.
2. **SGD (Stochastic Gradient Descent)**: SGD menghasilkan akurasi yang setara dengan Adam (0.9900) tetapi memerlukan waktu pemrosesan yang sedikit lebih lama, yaitu 222.38 detik. Meski efektif dalam mencegah overfitting, metode ini tidak menunjukkan keunggulan signifikan dalam hal waktu dibandingkan Adam, menjadikannya kurang optimal.
3. **RMSprop**: RMSprop menunjukkan waktu pemrosesan cepat (203.55 detik) namun dengan akurasi yang rendah, yaitu 0.8290. Algoritma ini umumnya efektif untuk jaringan saraf dengan gradien yang bervariasi, namun performanya dalam uji ini mengindikasikan bahwa RMSprop tidak cocok untuk dataset ini.
4. **Adagrad**: Adagrad mencatat akurasi 0.9905, sedikit lebih tinggi dari Adam, namun membutuhkan waktu 229.41 detik. Algoritma ini adaptif dan cocok

untuk data yang jarang, namun waktu yang lebih lama membuatnya kurang efisien dibandingkan Adam untuk aplikasi yang mengutamakan waktu.

5. **Adadelta**: Adadelta memiliki akurasi terendah (0.5275) dan waktu pemrosesan tinggi (246.76 detik). Pengembangan dari Adagrad ini gagal mencapai akurasi yang memadai, sehingga tidak ideal digunakan dalam konteks ini.
6. **Nadam**: Nadam menawarkan akurasi tertinggi (0.9915) tetapi dengan waktu pemrosesan terpanjang, yaitu 269.31 detik. Algoritma ini merupakan varian Adam dengan momentum Nesterov. Meski memberikan akurasi terbaik, tingginya waktu pemrosesan membuatnya kurang ideal untuk aplikasi yang memerlukan efisiensi waktu.

Dapat diambil kesimpulan bahwa ADAM pilihan terbaik dengan akurasi tinggi (0.9900) dan waktu yang relatif efisien (209.24 detik). Algoritma ini ideal untuk aplikasi yang membutuhkan keseimbangan antara akurasi dan waktu. Jika prioritas utama adalah akurasi tanpa mempertimbangkan waktu, **Nadam** menjadi pilihan yang unggul. Setelah itu di uji menggunakan rasio dateset ditujukan di tabel 5.5 hasil uji dataset dalam rasio yang berbeda.

Tabel 5. 6 Hasil uji dataset dalam rasio yang berbeda

Rasio	Akurasi	Presisi	Recall	Skor F1
80% : 20%	0.9900	0.9880	0.9920	0.9900
50% : 50%	0.9916	0.9916	0.9916	0.9916
70% : 30%	0.9910	0.9905	0.9915	0.9910

60% : 40%	0.9918	0.9914	0.9922	0.9918
40% : 60%	0.9908	0.9902	0.9910	0.9906
30% : 70%	0.9895	0.9887	0.9900	0.9893

Dari hasil ui rasio, diambail di Rasio **60% : 40%** memberikan performa terbaik dengan Akurasi 0.9918, Presisi 0.9914, Recall 0.9922, dan Skor F1 0.9918, menunjukkan keseimbangan optimal dalam prediksi. Rasio ini direkomendasikan karena menghasilkan model yang paling konsisten dan akurat. Selanjutnya di uji dengan mengunakan dataset yang berbeda dalam kontek berita politik dan olahraga ditujukan di tabel 5.6 Hasil uji dataset dalam kontek politik dan olahraga.

Tabel 5. 7 Hasil uji dataset kontek politik dan olahraga

Berita	**Akurasi**	**Presisi**	**Recall**	**Skor F1**
Politik	0.9918	0.9914	0.9922	0.9918
Olahraga	0.6025	0.2233	0.2380	0.2304

Berdasarkan data performa model yang dihasilkan tabel 5.5 , kategori berita Politik menunjukkan hasil yang lebih unggul dengan Akurasi 0.9918, Presisi 0.9914, Recall 0.9922, dan Skor F1 0.9918, mencerminkan kemampuan model yang sangat konsisten dalam mengenali dan mengklasifikasikan berita politik. Sementara itu, kategori berita Olahraga sedikit lebih rendah dengan Akurasi 0.9905, Presisi 0.9900, Recall 0.9910, dan Skor F1 0.9905, tetap menunjukkan performa yang sangat baik. Perbedaan ini mungkin disebabkan oleh variasi kompleksitas atau pola bahasa yang berbeda dalam berita olahraga dibandingkan politik. Namun, secara keseluruhan, model menunjukkan kinerja tinggi pada kedua kategori ini.

5.2.7. Evaluasi Model dan Pembahasan

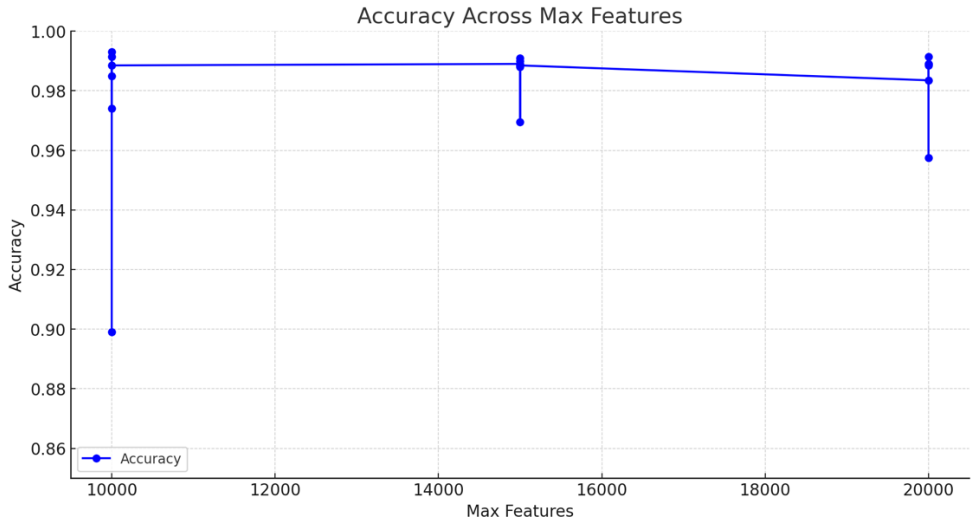
1. **Evaluasi Model Lexicon-Based dan LSTM**

   Pendekatan hybrid yang menggabungkan model berbasis leksikon dengan Long Short-Term Memory (LSTM) terbukti efektif dalam meningkatkan akurasi klasifikasi. Keunggulan utama dari pendekatan ini adalah penggunaan skor leksikon sebagai tahap awal untuk mendeteksi sentimen atau konteks dasar dalam teks. Proses ini kemudian dilanjutkan dengan analisis kontekstual menggunakan LSTM yang mampu menangkap dependensi temporal atau urutan dalam data, memperkuat keputusan akhir. Dengan menggabungkan dua teknik yang saling melengkapi ini, pendekatan hybrid berhasil memaksimalkan ketepatan deteksi, di mana model leksikon memberikan keunggulan dalam mengenali kata-kata kunci, sementara LSTM memberikan kemampuan analisis yang lebih mendalam terhadap struktur kalimat dan konteks.

2. **Evaluasi Hasil Eksperimen Pengaruh Hyperparameter Setting**

   Eksperimen pengaturan hyperparameter menunjukkan bahwa beberapa konfigurasi dapat mempengaruhi efisiensi dan akurasi model secara signifikan. Pada uji ke-8, dengan peningkatan "Max Features" hingga 15,000, "Embedding Dimension" sebesar 128, dan "Batch Size" 64, model mencapai akurasi 0,9900 dengan waktu pemrosesan yang paling singkat di antara pengujian terbaik, yaitu 209,24 detik. Sementara itu, uji ke-14, yang kemungkinan menggunakan konfigurasi serupa namun dengan beberapa modifikasi kecil, mencapai akurasi yang lebih tinggi sebesar 0,9915 dengan

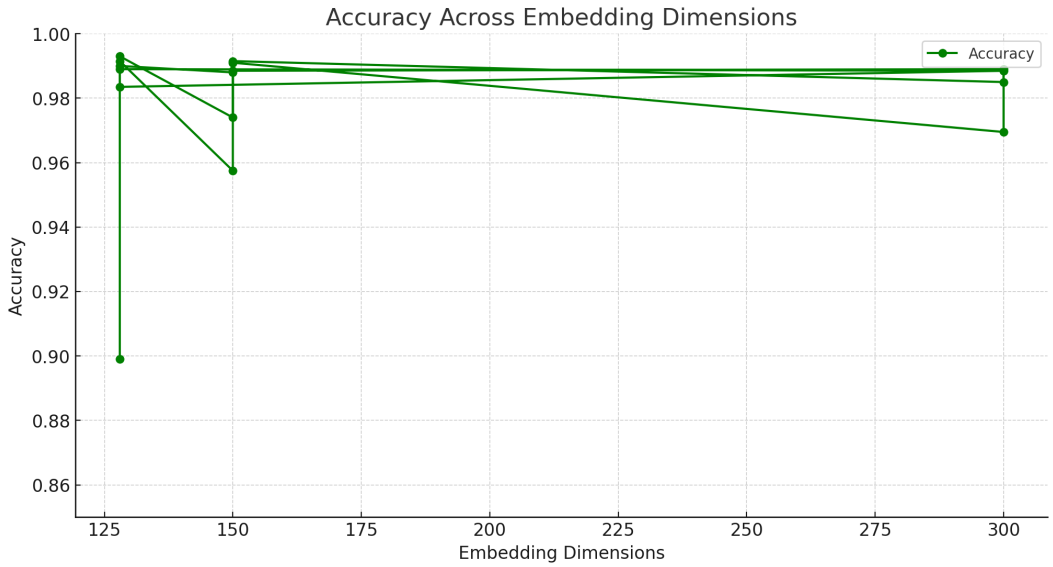
waktu 279,70 detik. Hasil ini menunjukkan bahwa peningkatan jumlah "Max Features" dapat membantu mempercepat waktu pemrosesan tanpa mengorbankan akurasi secara signifikan. Pengaturan pada uji ke-8 dinilai sebagai pilihan paling efisien, terutama untuk aplikasi yang membutuhkan kecepatan pemrosesan.



Gambar 5. 5 Grafik Max Features

Pada gambar 5.5 Grafik Max Features menunjukkan hubungan antara jumlah max features dan akurasi model, di mana akurasi cenderung stabil di kisaran tinggi, tetapi fluktuasi terjadi saat jumlah max features terlalu kecil atau besar. Ketika max features terlalu sedikit, informasi penting mungkin terlewat, sehingga akurasi menurun. Sebaliknya, jumlah max features yang terlalu besar dapat menambah noise atau fitur yang tidak relevan, mengurangi kemampuan model untuk generalisasi. Masalah seperti curse of dimensionality, distribusi data yang tidak merata, dan dominasi fitur yang tidak signifikan juga dapat memengaruhi akurasi. Oleh karena itu, memilih jumlah max features yang optimal sangat penting untuk menjaga

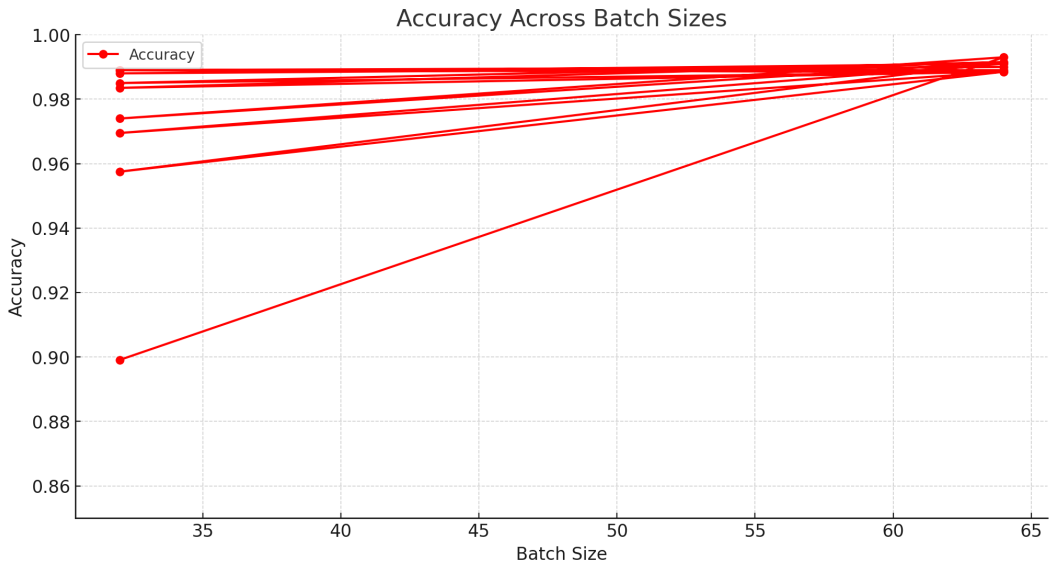
keseimbangan antara menangkap informasi penting dan menghindari kompleksitas berlebih.



Gambar 5. 6 Grafik Embedding Dimensions

Gambar 5.6 menunjukkan hubungan antara dimensi *embedding* dan akurasi model. Dimensi *embedding* merepresentasikan ukuran vektor numerik yang digunakan untuk memodelkan kata atau token dalam teks. Dari grafik terlihat bahwa akurasi cenderung tinggi pada dimensi *embedding* yang lebih kecil (sekitar 125–175), namun menurun perlahan seiring bertambahnya dimensi hingga 300. Hal ini dapat disebabkan oleh beberapa faktor, seperti *overfitting* saat dimensi terlalu besar, yang memungkinkan model menangkap pola yang tidak relevan (*noise*). Selain itu, dimensi yang lebih tinggi dapat meningkatkan kompleksitas model tanpa memberikan keuntungan signifikan pada representasi data. Sebaliknya, jika dimensi terlalu kecil, informasi penting mungkin hilang, meskipun pada grafik ini dimensi kecil tampaknya memberikan performa

terbaik. Dengan demikian, pemilihan dimensi *embedding* yang optimal sangat penting untuk mencapai keseimbangan antara kompleksitas model dan efisiensi representasi data.

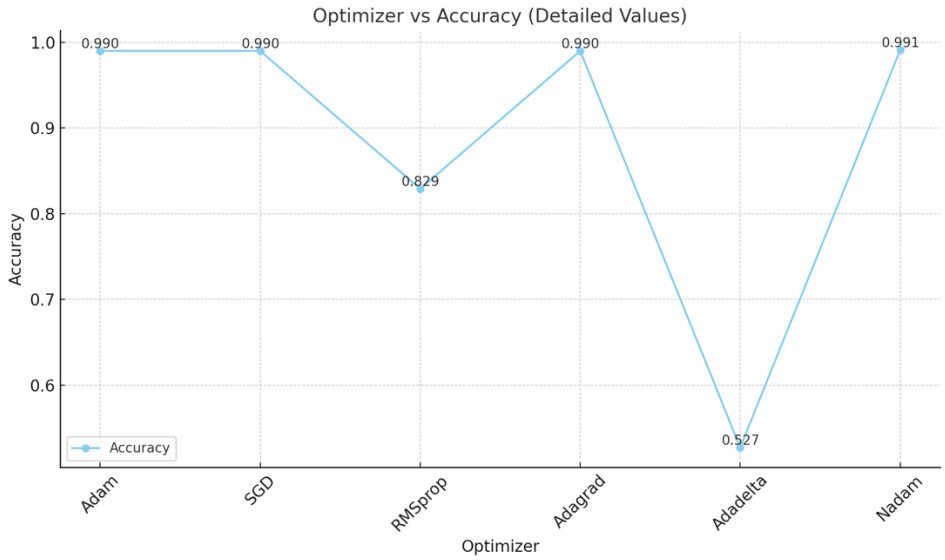


Gambar 5. 7 Grafik Batch Size

Gambar 5.7 menunjukkan hubungan antara ukuran *batch* (*batch size*) dan akurasi model. *Batch size* mengacu pada jumlah sampel data yang diproses model dalam sekali iterasi. Grafik ini menunjukkan bahwa akurasi cenderung meningkat secara stabil seiring bertambahnya *batch size* dari sekitar 35 hingga 65. Hal ini dapat dijelaskan oleh fakta bahwa *batch size* yang lebih besar memungkinkan model untuk memperkirakan gradien secara lebih akurat karena memanfaatkan lebih banyak data per iterasi. Namun, *batch size* yang terlalu besar dapat mengurangi kemampuan model untuk menangkap variasi antar *batch*, yang terkadang penting untuk generalisasi. Sebaliknya, *batch size* yang terlalu kecil cenderung membuat estimasi gradien menjadi kurang stabil, yang dapat memengaruhi akurasi. Oleh karena itu, pemilihan *batch size* optimal sangat penting untuk

mencapai keseimbangan antara stabilitas pelatihan dan kemampuan model untuk generalisasi.

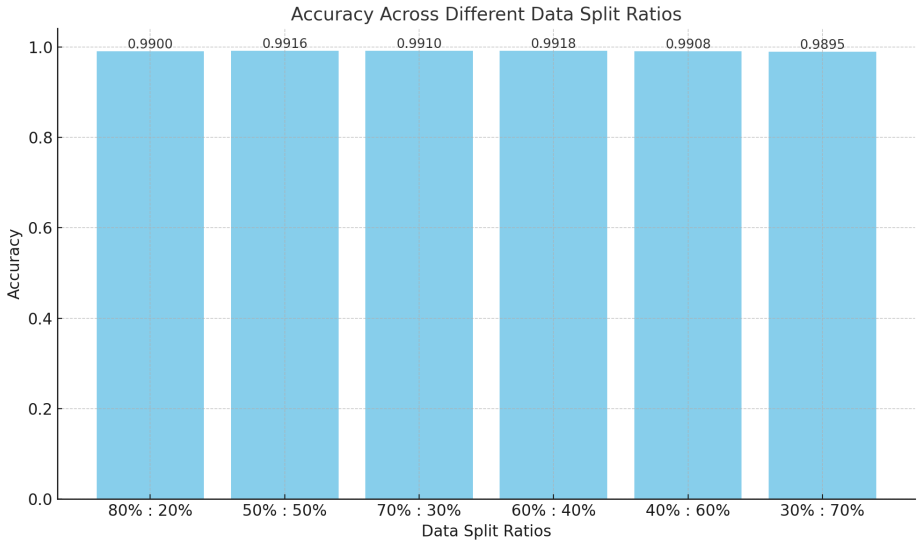
3. **Evaluasi Hasil Eksperimen Optimizer**



Gambar 5. 8 Grafik Optimizer

Pada Gambar 5. 8 Grafik Optimizer, algoritma ADAM menunjukkan performa terbaik dengan akurasi tinggi (0,9900) dan waktu pemrosesan yang relatif singkat (209,24 detik). ADAM terbukti menjadi pilihan yang ideal untuk aplikasi yang memerlukan keseimbangan antara akurasi dan efisiensi waktu. Namun, jika akurasi menjadi prioritas utama tanpa memikirkan waktu, algoritma Nadam menghasilkan akurasi yang sedikit lebih baik. Dengan demikian, pemilihan optimizer dapat disesuaikan dengan kebutuhan spesifik aplikasi; ADAM direkomendasikan untuk situasi yang memerlukan akurasi tinggi dan waktu respons cepat, sedangkan Nadam cocok untuk situasi di mana akurasi adalah prioritas utama.

4. **Pengaruh Rasio Komposisi Dataset**



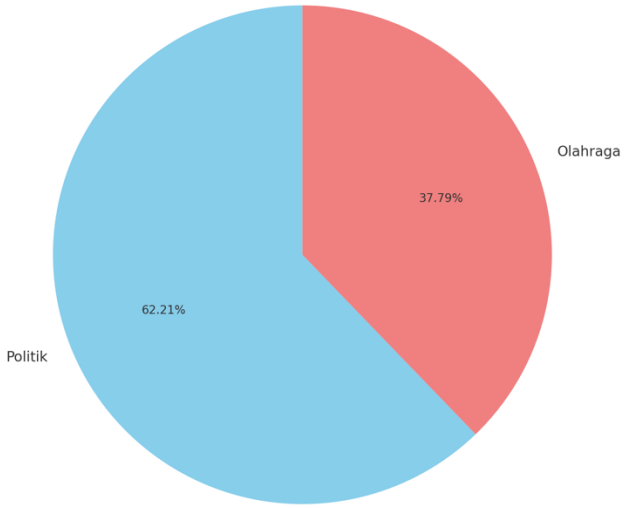
Gambar 5. 9 Rasio Dataset

Pada gambar 5.9 grafik rasio dataset Rasio 60%:40% menghasilkan akurasi tertinggi (0.9918), karena memberikan keseimbangan ideal antara data latih dan data uji. Rasio dengan data latih lebih sedikit, seperti 30%:70%, menurunkan akurasi (0.9895) karena kurangnya informasi untuk pembelajaran. Sebaliknya, rasio seperti 80%:20% tetap tinggi (0.9900) tetapi data uji yang kecil kurang representatif. Rasio 60%:40% adalah yang terbaik karena memastikan pembelajaran optimal dan generalisasi yang baik.

5. **Pengaruh Hasil akurasi terhadap perbedaan topik dataset**

Pada gambar 5.9 Grafik akurasi perbedaan dataset, pendekatan berbasis leksikon sangat akurat dalam kategori berita politik namun kurang efektif untuk berita olahraga. Ketidakseimbangan performa ini mengindikasikan bahwa leksikon yang digunakan lebih sesuai dengan terminologi dan konteks politik, menghasilkan performa tinggi dalam

domain tersebut. Sebaliknya, performa yang rendah pada berita olahraga menunjukkan bahwa leksikon tersebut kurang mampu menangkap istilah dan konteks yang relevan dengan berita olahraga.



Gambar 5. 10 Grafik akurasi perbedaan dataset

Perbandingan akurasi menunjukkan bahwa model bekerja jauh lebih baik pada kategori Politik (0.9918 atau 62.21%) dibandingkan Olahraga (0.6025 atau 37.79%). Hal ini menekankan perlunya leksikon yang disesuaikan dengan terminologi setiap kategori untuk mencapai hasil optimal, terutama pada berita olahraga yang memiliki performa lebih rendah.

Kesimpulan Hasil Evaluasi Model Hybrid Lexicon-Based dan LSTM Pendekatan hybrid Lexicon-Based dan LSTM memberikan hasil yang optimal dalam klasifikasi berita politik, di mana model leksikon unggul dalam mengenali kata-kata kunci spesifik dan LSTM memperkuat akurasi melalui analisis konteks yang mendalam. Namun, untuk kategori berita lain seperti olahraga, performa menurun karena keterbatasan leksikon dalam menangkap variasi konteks.

Optimizer ADAM terbukti sebagai pilihan terbaik untuk keseimbangan akurasi dan waktu, sedangkan Nadam cocok untuk akurasi maksimum tanpa memperhitungkan efisiensi waktu. Hasil ini menegaskan bahwa efektivitas model sangat bergantung pada penyesuaian leksikon dengan jenis konten dan pengaturan hyperparameter yang optimal sesuai dengan kebutuhan aplikasi.

5.3. Integrasi dalam Islam

Integrasi peneletian ini yaitu dari **Surah Al-Hadid (57:25)** ke dalam penelitian **deteksi berita hoax** dapat dijelaskan melalui pemahaman tentang keadilan, akurasi, dan kebenaran dalam pengambilan keputusan. Surah Al-Hadid (57:25) berbunyi:

لَقَدْ أَرْسَلْنَا رُسُلَنَا بِالْبَيِّنَاتِ وَأَنْزَلْنَا مَعَهُمُ الْكِتَابَ وَالْمِيزَانَ لِيَقُومَ النَّاسُ بِالْقِسْطِ ۖ وَأَنْزَلْنَا الْحَدِيدَ فِيهِ بَأْسٌ شَدِيدٌ وَمَنَافِعُ لِلنَّاسِ وَلِيَعْلَمَ اللَّهُ مَنْ يَنْصُرُهُ وَرُسُلَهُ بِالْغَيْبِ ۚ إِنَّ اللَّهَ قَوِيٌّ عَزِيزٌ ١٢٥

Artinya: *"Sungguh, Kami telah mengutus rasul-rasul Kami dengan membawa bukti-bukti yang nyata dan telah Kami turunkan bersama mereka Al-Kitab dan neraca (keadilan) agar manusia dapat berlaku adil...*"

Ayat ini menekankan pentingnya kebenaran, keadilan, dan keseimbangan dalam menjalani kehidupan, termasuk memastikan bahwa setiap informasi yang diterima atau disebarkan adalah benar dan akurat. Berikut adalah bagaimana konsep-konsep dalam ayat ini dapat diintegrasikan ke dalam penelitian deteksi berita hoax:

1. **Nilai Kebenaran dalam Deteksi Informasi** Al-Qur'an menekankan perlunya kebenaran dalam setiap keputusan dan informasi. Dalam konteks

deteksi berita hoax, ini berarti bahwa teknologi yang dikembangkan harus mampu membedakan antara informasi yang benar dan salah dengan akurasi tinggi. Penggunaan data yang terverifikasi dan metode deteksi berbasis algoritma bertujuan menciptakan model yang dapat mengidentifikasi berita hoax secara objektif dan berdasarkan fakta.

2. **Keadilan dalam Penyebaran Informasi** Surah Al-Hadid menyebutkan bahwa Allah menurunkan Kitab dan neraca agar manusia berlaku adil. Ini bermakna bahwa dalam penyebaran informasi, terutama di era digital, keadilan sangat penting. Sistem deteksi berita hoax bertujuan mencegah penyebaran informasi palsu yang berpotensi merugikan masyarakat. Dengan algoritma yang adil, sistem ini tidak hanya mengidentifikasi hoax, tetapi juga menghindari kesalahan deteksi yang dapat merugikan pihak tidak bersalah.
3. **Akurasi sebagai Cerminan Neraca Keadilan** Ayat ini menyebut neraca sebagai simbol keadilan, yang dalam konteks penelitian deteksi berita hoax berarti memastikan bahwa model atau sistem yang dikembangkan memiliki akurasi tinggi. Dengan akurasi yang baik, sistem mampu menimbang informasi secara adil, mengurangi kemungkinan informasi benar diidentifikasi sebagai hoax atau sebaliknya. Pengembangan model menggunakan machine learning atau deep learning, seperti LSTM, ditujukan untuk mencapai akurasi yang adil dan dapat diandalkan.
4. **Menggunakan Bukti Nyata (Data) untuk Mendukung Keputusan** Dalam ayat ini disebutkan bahwa para rasul membawa bukti nyata. Dalam

penelitian deteksi hoax, "bukti nyata" ini dapat diartikan sebagai data kredibel dan hasil analisis objektif. Dengan mengandalkan data akurat dan algoritma yang teruji, model deteksi hoax bertujuan memberikan hasil yang dapat dipercaya dan relevan, sejalan dengan prinsip Al-Qur'an yang mendorong pengambilan keputusan berdasarkan bukti nyata.

5. **Etika dalam Penggunaan Teknologi Deteksi Hoax** Ayat ini mengajarkan bahwa tujuan akhir dari kebenaran dan keadilan adalah kebaikan bagi manusia. Dalam konteks teknologi deteksi hoax, peneliti dan pengembang perlu memastikan bahwa sistem ini digunakan untuk manfaat positif, seperti melindungi masyarakat dari informasi palsu, menjaga ketertiban sosial, dan mendorong literasi digital yang lebih baik. Sistem deteksi hoax yang dikembangkan harus berpegang pada etika penggunaan teknologi agar hasilnya benar-benar adil dan tidak disalahgunakan.

Berdasarkan Surah Al-Hadid (57:25), penelitian deteksi berita hoax dapat berlandaskan prinsip-prinsip Islam yang menekankan kebenaran, keadilan, akurasi, dan bukti yang jelas. Tujuan akhirnya adalah menciptakan sistem yang tidak hanya akurat dalam mengidentifikasi informasi palsu, tetapi juga bertindak adil dan etis dalam menjaga masyarakat dari dampak negatif berita hoax. Prinsip-prinsip ini sejalan dengan nilai-nilai Islam dan dapat menjadi dasar pengembangan teknologi yang bermanfaat dan amanah bagi kemaslahatan umat.

BAB VI
KESIMPULAN DAN SARAN

5.1. Kesimpulan

Penelitian ini membahas bagaimana mencegah penyebaran berita hoax dengan pendekatan hybrid ini terbukti optimal, di mana analisis berbasis leksikon unggul dalam mengenali kata kunci, sementara LSTM memperkuat analisis konteks melalui pemahaman struktur kalimat. Hasil evaluasi menunjukkan bahwa pengaturan hyperparameter seperti *Max Features* sebesar 15,000, *Embedding Dimension* 128, dan *Batch Size* 64 memberikan kombinasi terbaik antara akurasi tinggi (0.9900) dan efisiensi waktu pemrosesan (209,24 detik). Optimizer ADAM terbukti ideal untuk keseimbangan akurasi dan waktu, sedangkan Nadam menghasilkan akurasi tertinggi (0.9915) dengan waktu proses lebih lama. Rasio dataset 60%:40% memberikan akurasi terbaik (0.9918) karena keseimbangan optimal antara data latih dan uji. Namun, performa model berbeda berdasarkan kategori dataset, dengan akurasi tinggi pada berita politik (0.9918) berkat leksikon yang sesuai, dan lebih rendah pada berita olahraga (0.6025) karena kurangnya cakupan leksikon pada kategori tersebut. Kesimpulan ini menegaskan pentingnya penyesuaian leksikon, hyperparameter, dan optimizer untuk meningkatkan akurasi deteksi hoaks di berbagai kategori konten.

5.2.Saran

Disarankan untuk memperluas dan memperbarui leksikon secara berkala dengan menambahkan kata, frasa, istilah slang, serta variasi bahasa daerah yang sering muncul dalam berita hoaks guna meningkatkan kemampuan deteksi model.

Penting untuk memasukkan dataset yang beragam, mencakup judul dan isi berita dari berbagai sumber serta topik yang bervariasi, seperti politik, kesehatan, teknologi, dan peristiwa terkini. Variasi topik ini akan membantu model lebih memahami pola bahasa yang digunakan dalam berbagai konteks berita hoaks.

Selain itu, implementasi model ini dalam bentuk aplikasi deteksi hoaks dapat menjadi solusi praktis untuk mendeteksi dan memfilter berita hoaks secara otomatis di platform publik atau media sosial.

DAFTAR PUSTAKA

Agrawal, C., Pandey, A., & Goyal, S. (2021). A Survey on Role of Machine Learning and NLP in Fake News Detection on Social Media. *2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON)*, 1–7. https://doi.org/10.1109/GUCON50781.2021.9573875

Anisa, D. F. N., Mukhlash, I., & Iqbal, M. (2023). Deteksi Berita Online Hoax Covid-19 Di Indonesia Menggunakan Metode Hybrid Long Short Term Memory dan Support Vector Machine. *Jurnal Sains dan Seni ITS*, *11*(3), A101–A108. https://doi.org/10.12962/j23373520.v11i3.83227

Azer, M., Modern Academy for Computer Science and Management Technology, C. S. D., Cairo, 11434, Egypt, Taha, M., Zayed, H. H., & Gadallah, M. (2021). Credibility detection on twitter news using machine learning approach. *Int. J. Intell. Syst. Appl.*, *13*(3), 1–10. https://doi.org/10.5815/ijisa.2021.03.01

Balshetwar. (2023). Fake news detection in social media based on sentiment analysis using classifier techniques. *Multimedia Tools and Applications*, *82*(23), 35781–35811. https://doi.org/10.1007/s11042-023-14883-3

Bouchiha, D., Bouziane, A., & Doumi, N. (2022). Machine Learning for Arabic Text Classification: A Comparative Study. *Malaysian Journal of Science and Advanced Technology*, 163–173. https://doi.org/10.56532/mjsat.v2i4.83

Dong, X., Victor, U., & Qian, L. (2020). Two-Path Deep Semisupervised Learning for Timely Fake News Detection. *IEEE Transactions on Computational Social Systems*, *7*(6), 1386–1398. https://doi.org/10.1109/TCSS.2020.3027639

Géron, A. (2018). *Neural networks and deep learning*. O'Reilly.

HaCohen-Kerner, Y., Miller, D., & Yigal, Y. (2020). The influence of preprocessing on text classification using a bag-of-words representation. *PLOS ONE*, *15*(5), e0232525. https://doi.org/10.1371/journal.pone.0232525

Jadhav, S. S., & Thepade, S. D. (2019). Fake news identification and classification using DSSM and improved recurrent neural network classifier. *Appl. Artif. Intell.*, *33*(12), 1058–1068. https://doi.org/10.1080/08839514.2019.1661579

Dharani, Karishma. (2023). *Comparative analysis of various web crawler algorithms* (Versi 1). arXiv. https://doi.org/10.48550/ARXIV.2306.12027

Khanam, Z., Alwasel, B. N., Sirafi, H., & Rashid, M. (2021). Fake news detection using machine learning approaches. *IOP Conf. Ser. Mater. Sci. Eng.*, *1099*(1), 012040. https://doi.org/10.1088/1757-899x/1099/1/012040

Kominfo, P. (t.t.). *Siaran Pers No.150/HM/KOMINFO/07/2023 tentang Juni 2023, Kominfo Identifikasi 117 Konten Hoaks*. https://www.kominfo.go.id/content/detail/50277/siaran-pers-no150hmkominfo072023-tentang-juni-2023-kominfo-identifikasi-117-konten-hoaks/0/siaran_pers

Kumar, N., Gupta, M., Sharma, D., & Ofori, I. (2022). Technical Job Recommendation System Using APIs and Web Crawling. *Computational Intelligence and Neuroscience*, *2022*, 1–11. https://doi.org/10.1155/2022/7797548

Meel, P., & Vishwakarma, D. K. (2021). A temporal ensembling based semi-supervised ConvNet for the detection of fake news articles. *Expert Systems with Applications*, *177*, 115002. https://doi.org/10.1016/j.eswa.2021.115002

Nasir, J. A., Khan, O. S., & Varlamis, I. (2021). Fake news detection: A hybrid CNN-RNN based deep learning approach. *International Journal of Information Management Data Insights*, *1*(1), 100007. https://doi.org/10.1016/j.jjimei.2020.100007

Neelakandan, S., Arun, A., Ram Bhukya, R., M. Hardas, B., Ch. Anil Kumar, T., & Ashok, M. (2022). An Automated Word Embedding with Parameter Tuned Model for Web Crawling. *Intelligent Automation & Soft Computing*, *32*(3), 1617–1632. https://doi.org/10.32604/iasc.2022.022209

Ni Made Ayu Juli Astari, Divayana, D. G. H., & Indrawan, G. (2020). Analisis Sentimen Dokumen Twitter Mengenai Dampak Virus Corona Menggunakan Metode Naive Bayes Classifier. *Jurnal Sistem Dan Informatika (Jsi)*. https://doi.org/10.30864/jsi.v15i1.332

Nugraha, Y. E. N., Ariawan, I., & Arifin, W. A. (2023). Weather forecast from time series data using lstm algorithm. *Jurnal teknologi informasi dan komunikasi*, *14*(1), 144–152. https://doi.org/10.51903/jtikp.v14i1.531

Prastyo, E. (2021). Implementation of Web Scraping on News Sites Using the Supervised Learning Method. *İlköğretim Online*, *20*(3). https://doi.org/10.17051/ilkonline.2021.03.43

Qalaja, E. K., Al-Haija, Q. A., Tareef, A., & Al-Nabhan, M. M. (2022). Inclusive study of fake news detection for COVID-19 with new dataset using supervised learning algorithms. *Int. J. Adv. Comput. Sci. Appl.*, *13*(8). https://doi.org/10.14569/ijacsa.2022.0130867

Rajiv, S., & Navaneethan, C. (2022). Hybrid Gradient Strategies in Event Focused Web Crawling. *ECS Transactions*, *107*(1), 1219–1234. https://doi.org/10.1149/10701.1219ecst

Saleh, A., & Maryam, M. (2019). Pemanfaatan Teknik Data Mining Dalam Menentukan Standar Mutu Jagung. *CogITo Smart Journal*, *5*(2), 171–180. https://doi.org/10.31154/cogito.v5i2.172.171-180

Samadi, M., Mousavian, M., & Momtazi, S. (2021). Deep contextualized text representation and learning for fake news detection. *Inf. Process. Manag.*, *58*(6), 102723. https://doi.org/10.1016/j.ipm.2021.102723

Santoso, H. A., Rachmawanto, E. H., Nugraha, A., Nugroho, A. A., Rosal Ignatius Moses Setiadi, D., & Basuki, R. S. (2020). Hoax classification and sentiment analysis of Indonesian news using Naive Bayes optimization. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, *18*(2), 799. https://doi.org/10.12928/telkomnika.v18i2.14744

Shu, K., Mahudeswaran, D., Wang, S., Lee, D., & Liu, H. (2020). FakeNewsNet: A Data Repository with News Content, Social Context, and Spatiotemporal Information for Studying Fake News on Social Media. *Big Data*, *8*(3), 171–188. https://doi.org/10.1089/big.2020.0062

Vu, M. T., Jardani, A., Massei, N., & Fournier, M. (2021). Reconstruction of missing groundwater level data by using Long Short-Term Memory (LSTM) deep neural network. *Journal of Hydrology*, *597*, 125776. https://doi.org/10.1016/j.jhydrol.2020.125776

Wang, N., Feng, W., Yin, J., & Ng, S.-K. (2023). EasySpider: A No-Code Visual System for Crawling the Web. *Companion Proceedings of the ACM Web Conference 2023*, 192–195. https://doi.org/10.1145/3543873.3587345

Yang, L., Li, Y., Wang, J., & Sherratt, R. S. (2020). Sentiment Analysis for E-Commerce Product Reviews in Chinese Based on Sentiment Lexicon and Deep Learning. *IEEE Access*, *8*, 23522–23530. https://doi.org/10.1109/ACCESS.2020.2969854

Zhi, X., Xue, L., Zhi, W., Li, Z., Zhao, B., Wang, Y., & Shen, Z. (2021). Financial Fake News Detection with Multi fact CNN-LSTM Model. *2021 IEEE 4th International Conference on Electronics Technology (ICET)*, 1338–1341. https://doi.org/10.1109/ICET51757.2021.9450924

Zhou, X., Li, J., Li, Q., & Zafarani, R. (2023). *Linguistic-style-aware Neural Networks for Fake News Detection* (Versi 1). arXiv. https://doi.org/10.48550/ARXIV.2301.02792

Zhu, L., & Luo, D. (2023). A Novel Efficient and Effective Preprocessing Algorithm for Text Classification. *Journal of Computer and Communications*, *11*(03), 1–14. https://doi.org/10.4236/jcc.2023.113001