IMPLEMENTASI MODEL LONG SHORT TERM MEMORY UNTUK KLASIFIKASI KATEGORI BERITA BERBAHASA INDONESIA

SKRIPSI

Oleh :
NUR MUHAMMAD NAJIBURROHMAN
NIM. 220605110165

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2026

IMPLEMENTASI MODEL LONG SHORT TERM MEMORY UNTUK KLASIFIKASI KATEGORI BERITA BERBAHASA INDONESIA

SKRIPSI

Diajukan kepada:
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh :
NUR MUHAMMAD NAJIBURROHMAN
NIM. 220605110165

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2026

HALAMAN PERSETUJUAN

IMPLEMENTASI MODEL LONG SHORT TERM MEMORY UNTUK KLASIFIKASI KATEGORI BERITA BERBAHASA INDONESIA

SKRIPSI

Oleh :
NUR MUHAMMAD NAJIBURROHMAN
NIM. 220605110165

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 18 Juni 2026

Pembimbing I,	Pembimbing II,
Dr. Zainal Abidin, M.Kom	Shoffin Nahwa Utama, M.T
NIP. 19760613 200501 1 004	NIP. 19860703 202012 1 003

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

Supriyono, M. Kom
NIP. 19841010 201903 1 012

HALAMAN PENGESAHAN

IMPLEMENTASI MODEL LONG SHORT TERM MEMORY UNTUK KLASIFIKASI KATEGORI BERITA BERBAHASA INDONESIA

SKRIPSI

Oleh :
NUR MUHAMMAD NAJIBURROHMAN
NIM. 220605110165

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 24 Juni 2026

Susunan Dewan Penguji

Ketua Penguji	: Dr. Ririen Kusumawati, S.Si., M.Kom. NIP. 19720309 200501 2 002	( )
Anggota Penguji I	: Fatchurrohman, M.Kom. NIP. 19700731 200501 1 002	( )
Anggota Penguji II	:Dr. Zainal Abidin, M.Kom. NIP. 19760613 200501 1 004	( )
Anggota Penguji III	: Shoffin Nahwa Utama, M.T NIP. 19860703 202012 1 003	( )

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

Supriyono, M. Kom
NIP. 19841010 201903 1 012

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama	: Nur Muhammad Najiburrohman
NIM	: 220605110165
Fakultas / Prodi	: Sains dan Teknologi / Teknik Informatika
Judul Skripsi	: Implementasi Model Long Short Term Memory Untuk Klasifikasi Kategori Berita Berbahasa Indonesia

Menyatakan dengan sebenarnya bahwa Skripsi 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 skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 23 Juni 2026
Yang membuat pernyataan,

Nur Muhammad Najiburrohman
NIM.2206051101665

MOTTO

"Sometimes you win and sometimes you learn. But have fun. Most people never win because they're more afraid of losing."

- Robert T. Kiyosaki

HALAMAN PERSEMBAHAN

Bismillahirrahmanirrahim

Dengan mengucapkan syukur kehadirat Allah SWT atas segala rahmat, kasih sayang, dan pertolongan-Nya, sehingga karya sederhana ini dapat terselesaikan.

Dengan segala kerendahan hati, skripsi ini saya persembahkan kepada:

Mama dan Papa Tercinta

Terima kasih atas cinta yang tak pernah berkurang, doa yang tak pernah putus, serta setiap pengorbanan yang tak pernah meminta balasan. Terima kasih telah menjadi tempat pulang, tempat menguatkan, dan alasan terbesar untuk terus berjuang ketika langkah terasa berat. Tidak ada kata yang mampu menggambarkan betapa besar kasih sayang dan perjuangan yang telah Mama dan Pama berikan.

Keluarga Tercinta

Terima kasih kepada seluruh keluarga yang selalu memberikan doa, perhatian, dukungan, dan semangat dalam setiap proses yang saya jalani

Untuk Diri Penulis Sendiri

Terima kasih karena telah memilih untuk tetap bertahan ketika keadaan terasa sulit, tetap bangkit setelah mengalami kegagalan, dan tetap melangkah meskipun perjalanan tidak selalu mudah. Terima kasih telah percaya bahwa setiap proses memiliki maknanya sendiri.

Semoga pencapaian ini menjadi awal dari perjalanan yang lebih besar, bukan akhir dari sebuah perjuangan.

KATA PENGANTAR

Puji syukur kehadirat Allah SWT atas segala limpahan rahmat, taufik, hidayah, serta karunia-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul "Implementasi Model Long Short-Term Memory untuk Klasifikasi Kategori Berita Berbahasa Indonesia" dengan baik. Shalawat serta salam semoga senantiasa tercurah kepada junjungan kita Nabi Muhammad SAW, beserta keluarga, sahabat, dan seluruh umatnya hingga akhir zaman. Penyusunan skripsi ini merupakan salah satu syarat untuk memperoleh gelar Sarjana Komputer pada Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Penulis menyadari bahwa penyusunan skripsi ini tidak akan terselesaikan tanpa adanya doa, dukungan, bimbingan, motivasi, serta bantuan dari berbagai pihak. Oleh karena itu, dengan segala kerendahan hati penulis menyampaikan rasa hormat dan terima kasih yang sebesar-besarnya kepada:

1. Prof. Dr. Hj. Ilfi Nur Diana, M.Si., CAHRM., CRMP., selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang
2. Dr. Agus Mulyono, M.Kes., selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang
3. Supriyono, M.Kom., selaku Ketua Program Studi Teknik Informatika yang telah memberikan arahan dan dukungan selama penulis menempuh pendidikan.
4. Dr. Zainal Abidin, M.Kom., selaku Dosen Wali sekaligus Dosen Pembimbing I yang telah meluangkan waktu, tenaga, dan pikiran untuk

memberikan bimbingan, arahan, motivasi, serta masukan yang sangat berharga dengan penuh kesabaran selama proses penyusunan skripsi ini.

5. Shoffin Nahwa Utama, M.T., selaku Dosen Pembimbing II yang telah memberikan banyak saran, ilmu, arahan, serta motivasi sehingga penelitian ini dapat terselesaikan dengan baik.
6. Dr. Ririen Kusumawati, S.Si., M.Kom., selaku Ketua Penguji yang telah memberikan kritik, saran, dan masukan yang sangat membangun demi penyempurnaan skripsi ini.
7. Fatchurrohman, M.Kom., selaku Anggota Penguji I yang telah memberikan evaluasi, kritik, dan saran yang sangat bermanfaat sehingga penelitian ini menjadi lebih baik.
8. Seluruh Dosen, Admin, dan Staf Program Studi Teknik Informatika yang telah memberikan ilmu pengetahuan, pengalaman, dan inspirasi selama masa perkuliahan. Semoga ilmu yang telah diberikan menjadi amal jariyah yang terus mengalir manfaatnya.
9. Kedua orang tua tercinta, Mama Erna Wiwik Hidayati dan Papa Wahyu Maretno Wibowo yang menjadi alasan terbesar penulis untuk terus berjuang. Terima kasih atas kasih sayang yang tak pernah berkurang, doa yang selalu dipanjatkan dalam setiap sujud, dukungan yang tiada henti, serta segala pengorbanan yang telah diberikan. Semoga pencapaian sederhana ini dapat menjadi salah satu bentuk kebahagiaan dan kebanggaan untuk Mama dan Papa.

10. Sahabat-sahabat terbaik penulis, yang selalu memberikan semangat, menjadi tempat berbagi cerita, bertukar pikiran, serta menemani dalam berbagai suka dan duka selama proses penyusunan skripsi. Terima kasih atas kebersamaan yang telah menjadi bagian berharga dalam perjalanan ini.
11. Teman-teman Teknik Informatika Angkatan 2022, yang telah menjadi rekan belajar, berdiskusi, dan berjuang bersama selama masa perkuliahan. Terima kasih atas kebersamaan, kerja sama, serta kenangan yang telah tercipta.
12. Annisa Adenanty Palupi, yang selalu hadir memberikan dukungan, semangat, perhatian, serta menjadi tempat berbagi cerita di setiap proses yang penulis lalui. Terima kasih telah menjadi seseorang yang selalu ada dalam berbagai keadaan, baik saat senang maupun saat menghadapi kesulitan, sehingga penulis mampu menyelesaikan perjalanan ini dengan lebih kuat.
13. Terakhir, kepada diri penulis sendiri. Terima kasih karena telah memilih untuk tetap bertahan, bangkit setiap kali menghadapi kegagalan, dan terus berjuang hingga mampu menyelesaikan perjalanan panjang ini. Skripsi ini menjadi bukti bahwa setiap doa, usaha, kesabaran, dan ketekunan akan menemukan hasilnya pada waktu yang tepat. Semoga pencapaian ini menjadi awal dari langkah-langkah yang lebih besar, penuh keberkahan, dan bermanfaat bagi banyak orang.

Malang, 23 Juni 2026

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Najiburrohman, Nur Muhammad. 2026. **Implementasi Model Long Short Term Memory Untuk Klasifikasi Kategori Berita Berbagasa Indonesia.** Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. Zainal Abidin, M.Kom (II) Shoffin Nahwa Utama, M.T.

Kata Kunci : Klasifikasi Berita, Text Classification Long Short-Term Memory, Natural Languange Processing, Deep Learning

Klasifikasi berita merupakan salah satu tugas dalam *Natural Language Processing* (NLP) yang bertujuan untuk mengelompokkan dokumen berita ke dalam kategori tertentu secara otomatis. Penelitian ini bertujuan untuk menganalisis performa metode *Long Short-Term Memory* (LSTM) dengan representasi *Word2Vec* dalam klasifikasi berita multi-kelas berbahasa Indonesia. Dataset yang digunakan adalah *Indonesian News Corpus* yang terdiri atas delapan kategori berita, yaitu Bisnis, Bola, *Lifestyle*, Nasional, Olahraga, Otomotif, Teknologi, dan *Travel*. Tahapan penelitian meliputi *preprocessing* data berupa *cleaning*, *case folding*, tokenisasi, *stopword removal*, *padding*, dan *label encoding*, kemudian dilanjutkan dengan pembentukan representasi kata menggunakan *Word2Vec*, pelatihan model LSTM, serta evaluasi menggunakan *K-Fold Cross Validation* dengan variasi *batch size* 32 dan 64 pada skenario 5-*fold* dan 10-*fold*. Kinerja model dievaluasi menggunakan metrik *accuracy*, *precision*, *recall*, dan *F1-score*. Hasil pengujian menunjukkan bahwa seluruh skenario menghasilkan nilai *accuracy* di atas 88%. Performa terbaik diperoleh pada penggunaan *batch size* 32 dengan 10-*fold cross validation*, yang menghasilkan *accuracy* sebesar 0,9203, *precision* sebesar 0,9207, *recall* sebesar 0,9203, dan *F1-score* sebesar 0,9203. Hasil tersebut menunjukkan bahwa kombinasi *Word2Vec* dan LSTM mampu memberikan performa klasifikasi yang baik serta memiliki kemampuan generalisasi yang stabil terhadap data berita berbahasa Indonesia.

ABSTRACT

Najiburrohman, Nur Muhammad. 2026. ***Implementation of Long Short-Term Memory Model for Indonesian News Category Classification***. Undergraduate Thesis. Informatics Engineering Study Program, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Advisors: (I) Dr. Zainal Abidin, M.Kom (II) Shoffin Nahwa Utama, M.T.

Keywords: News Classification, Text Classification, Long Short-Term Memory, Natural Language Processing, Deep Learning

News classification is one of the fundamental tasks in *Natural Language Processing* (NLP), aiming to automatically assign news articles to predefined categories based on their content. This study investigates the performance of the *Long Short-Term Memory* (LSTM) model combined with *Word2Vec* word embeddings for multi-class classification of Indonesian news articles. The dataset used in this research is the *Indonesian News Corpus*, which consists of eight categories: Business, Sports (Football), Lifestyle, National, Sports, Automotive, Technology, and Travel. The research process includes data preprocessing, consisting of cleaning, case folding, tokenization, stopword removal, padding, and label encoding, followed by word representation using *Word2Vec*, model training with LSTM, and performance evaluation using *K-Fold Cross Validation*. Two *batch size* values (32 and 64) and two validation scenarios (5-fold and 10-fold) were evaluated. The model performance was assessed using *accuracy*, *precision*, *recall*, and *F1-score*. Experimental results indicate that all evaluation scenarios achieved an accuracy above 88%. The best performance was obtained using a *batch size* of 32 with 10-fold cross-validation, achieving an *accuracy* of 0.9203, a *precision* of 0.9207, a *recall* of 0.9203, and an *F1-score* of 0.9203. These findings demonstrate that the combination of *Word2Vec* and LSTM provides effective and stable performance for multi-class classification of Indonesian news articles.

مستخلص البحث

.نجيب الرحمن، نور محمد. 2026 .*تطبيق نموذج الذاكرة طويلة وقصيرة المدى لتصنيف فئات الأخبار باللغة الإندونيسية*. بحث جامعي
قسم تقنية المعلومات، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرفان: (1)
.الدكتور زين العابدين، ماجستير في علوم الحاسوب، (2) شوفين نهوا أوتاما، ماجستير في التقنية

الكلمات المفتاحية: تصنيف الأخبار، تصنيف النصوص، الذاكرة طويلة وقصيرة المدى، معالجة اللغة الطبيعية، التعلم العميق

يُعدُّ تصنيف الأخبار إحدى مهام معالجة اللغات الطبيعية (*Natural Language Processing - NLP*)، ويهدف إلى تصنيف الوثائق الإخبارية تلقائيًا ضمن فئات محددة. تهدف هذه الدراسة إلى تحليل أداء نموذج الذاكرة طويلة وقصيرة المدى LSTM باستخدام تمثيل الكلمات Word2Vec في تصنيف الأخبار متعددة الفئات باللغة الإندونيسية. واعتمدت الدراسة على مجموعة بيانات *Indonesian News Corpus* التي تضم ثماني فئات إخبارية، وهي: الأعمال، وكرة القدم، ونمط الحياة، والأخبار الوطنية، والرياضة، والسيارات، والتكنولوجيا، والسفر. وشملت مراحل البحث المعالجة المسبقة للبيانات، والتي تضمنت تنظيف البيانات، وتحويل الأحرف إلى أحرف صغيرة *Case Folding*، وتقسيم النص إلى كلمات *Tokenization*، وإزالة الكلمات الشائعة *Stopword Removal*، وإضافة الحشو *Padding*، وترميز الفئات *Label Encoding* ، ثم إنشاء تمثيل الكلمات باستخدام Word2Vec، وتدريب نموذج LSTM، وتقييمه باستخدام أسلوب *K-Fold Cross Validation* مع استخدام حجمَي دفعة *Batch Size* مقدارهما 32 و64 ضمن سيناريوهين هما 5-*Fold* و10-*Fold*. وتم تقييم أداء النموذج باستخدام مقاييس الدقة *Accuracy* ، والدقة الإيجابية *Precision*، والاسترجاع *Recall* ، ومقياس *F1-Score*. وأظهرت النتائج أن جميع السيناريوهات حققت قيمة *Accuracy* تجاوزت 88%، في حين تحقق أفضل أداء عند استخدام Batch Size يساوي 32 مع 10-*Fold Cross Validation*، حيث بلغت قيمة *Accuracy* 0.9203، و*Precision* 0.9207، و*Recall* 0.9203، و*F1-Score* 0.9203، مما يدل على أن الجمع بين Word2Vec وLSTM يحقق أداءً جيدًا في تصنيف الأخبار، ويتمتع بقدرة مستقرة على التعميم عند التعامل مع بيانات الأخبار باللغة الإندونيسية.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Meningkatnya jumlah berita yang dipublikasikan setiap hari melalui berbagai portal berita menimbulkan tantangan dalam proses pengelolaan dan pengelompokan berita secara otomatis. Artikel berita yang diterbitkan mencakup berbagai kategori, seperti politik, ekonomi, teknologi, dan olahraga, sehingga proses kategorisasi menjadi semakin kompleks. Selama ini, sebagian besar media masih melakukan kategorisasi berita secara manual oleh editor, yang berpotensi menimbulkan bias dan inkonsistensi dalam penilaian antar individu. Oleh karena itu, diperlukan sistem klasifikasi berita otomatis yang mampu mengelompokkan berita secara akurat, efisien, dan konsisten untuk mendukung pengelolaan informasi digital (Sebastiani, 2001).

Dalam perspektif Islam, pengelolaan dan penyebaran informasi memiliki nilai moral yang sangat penting. Islam menekankan prinsip kejujuran dan kehati-hatian dalam menerima serta menyampaikan berita. Allah SWT berfirman:

يٰٓاَيُّهَا الَّذِيْنَ اٰمَنُوْٓا اِنْ جَاۤءَكُمْ فَاسِقٌۢ بِنَبَاٍ فَتَبَيَّنُوْٓا اَنْ تُصِيْبُوْا قَوْمًاۢ بِجَهَالَةٍ فَتُصْبِحُوْا عَلٰى مَا فَعَلْتُمْ نٰدِمِيْنَ

"Wahai orang-orang yang beriman, jika datang kepadamu orang fasik membawa suatu berita, maka periksalah dengan teliti agar kamu tidak menimpakan suatu musibah kepada suatu kaum tanpa mengetahui keadaannya yang menyebabkan kamu menyesal atas perbuatanmu itu." (QS. Al-Hujurat:6).

Menurut tafsir Ibnu Katsir, ayat ini menegaskan kewajiban setiap Muslim untuk tidak langsung menerima atau menyebarkan berita dari sumber yang tidak

terpercaya. Berita yang salah dapat menimbulkan fitnah, kerugian, dan penyesalan di kemudian hari. Verifikasi informasi menjadi sebuah amanah moral yang harus dijalankan dengan penuh kehati-hatian.

Dalam konteks penelitian ini, nilai-nilai tersebut dapat dijadikan dasar filosofis dalam pengembangan sistem klasifikasi berita yang tidak hanya berorientasi pada keakuratan teknis, tetapi juga selaras dengan prinsip kejujuran dan kehati-hatian dalam Islam. Pemanfaatan teknologi dalam proses klasifikasi berita dapat menjadi sarana untuk membantu manusia dalam memilah informasi yang benar dan relevan, sehingga mencegah penyebaran berita yang menyesatkan (*hoax*).

Berbagai metode telah dikembangkan untuk meningkatkan kinerja klasifikasi berita, terutama melalui pendekatan *machine learning* dan *deep learning*. Beberapa algoritma yang sering digunakan antara lain *Naïve Bayes*, *Support Vector Machine* (SVM), dan *Long Short-Term Memory* (LSTM), yang telah menunjukkan hasil yang baik dalam mengenali pola serta hubungan semantik pada data teks (Prasetio Widhi & Hatta Fudholi, 2023). Di antara berbagai metode tersebut, LSTM menjadi salah satu model yang banyak digunakan dalam tugas *Natural Language Processing* (NLP) karena mampu memproses data yang bersifat sekuensial sekaligus mempertahankan informasi dari langkah-langkah sebelumnya. Selain itu, arsitektur LSTM dirancang untuk mengatasi permasalahan *vanishing gradient* yang umum terjadi pada *Recurrent Neural Network* (RNN) konvensional, sehingga model dapat mempelajari dependensi jangka panjang secara lebih efektif.

Sejumlah penelitian sebelumnya telah membuktikan efektivitas LSTM dalam klasifikasi teks. Misalnya, Rahmawati *et al.* (2023) berhasil meningkatkan akurasi klasifikasi berita Indonesia menggunakan LSTM dibandingkan metode tradisional seperti SVM. Penelitian oleh Tri Hermanto *et al*. (2021) menunjukkan bahwa kombinasi metode LSTM-CNN berbasis Word2Vec mampu mengklasifikasikan judul berita berbahasa Indonesia. Temuan-temuan tersebut menunjukkan bahwa model LSTM memiliki kemampuan yang kuat dalam mempelajari hubungan semantik antar kata pada teks berita.

1.2 Rumusan Masalah

Masalah yang diangkat adalah bagaimana metode *Long Short-Term Memory* (LSTM) dapat digunakan untuk mengklasifikasikan kategori berita online berbahasa Indonesia secara otomatis dan akurat.

1.3 Batasan Masalah

Penelitian ini memiliki beberapa batasan agar pembahasan lebih terarah dan fokus pada tujuan yang ingin dicapai. Adapun batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Data yang digunakan berasal dari dataset berita online berbahasa Indonesia, yaitu *Indonesian News Corpus*, yang berisi teks berita dari berbagai kategori seperti Teknologi, Otomotif, Bisnis Ekonomi, Olahraga, Nasional, Bola, *Lifestyle*, dan Travel.

2. Penelitian ini hanya berfokus pada teks isi berita (konten utama) tanpa mempertimbangkan elemen non-teks seperti gambar, video, atau metadata lainnya.
3. Metode yang digunakan dalam penelitian ini adalah *Long Short-Term Memory* (LSTM) sebagai model utama untuk melakukan klasifikasi kategori berita.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk membangun sebuah model klasifikasi berita online berbahasa Indonesia menggunakan *metode Long Short-Term Memory* (LSTM). Selanjutnya, penelitian ini bertujuan untuk mengevaluasi kinerja model LSTM dengan menggunakan metrik akurasi, presisi, *recall*, dan *F1-Score*, sehingga dapat diketahui tingkat keberhasilan model dalam melakukan klasifikasi berita secara tepat dan efektif.

1.5 Manfaat Penelitian

1. Manfaat Akademis

   Penelitian ini memperkaya kajian di bidang *machine learning* dan *natural language processing* (NLP), khususnya penerapan metode LSTM dalam klasifikasi berita berbahasa Indonesia.

2. Manfaat Praktis

   Hasil penelitian ini membantu proses klasifikasi berita secara otomatis agar lebih cepat, efisien, dan konsisten dibandingkan cara manual.

3. Manfaat Teknis

   Penelitian ini memberikan pemahaman tentang penerapan model LSTM serta dapat menjadi dasar pengembangan sistem klasifikasi teks berbasis *deep learning* di masa mendatang.

BAB II

STUDI PUSTAKA

2.1 Penelitian Terkait

Penelitian oleh Liu (2024) mengembangkan model *Long Short-Term Memory (LSTM)* untuk klasifikasi berita berbahasa Mandarin. Model ini memanfaatkan representasi vektor kata berbasis *Word2Vec* untuk menangkap makna semantik dalam teks. Dengan pelatihan menggunakan dataset berita besar, model tersebut berhasil mencapai akurasi di atas 94% dan F1-score 0,91. Studi ini menegaskan kemampuan LSTM dalam mengenali pola bahasa alami dengan struktur kalimat kompleks, khususnya pada bahasa yang memiliki karakter unik seperti Mandarin.

Khuntia & Gupta (2022) melakukan penelitian berjudul *Indian News Headlines Classification using Word Embeddings and LSTM* dengan fokus pada pengelompokan berita India berdasarkan judulnya. Proses pra-pemrosesan meliputi tokenisasi dan pembentukan embedding untuk mengubah kata menjadi representasi numerik yang bermakna. Hasil pengujian menunjukkan akurasi 93,6% dan F1-score 0,92, yang menandakan efektivitas LSTM dalam menangkap konteks teks pendek. Penelitian ini menunjukkan bahwa model LSTM dapat digunakan secara efisien untuk klasifikasi berbasis headline dengan informasi terbatas.

Penelitian oleh Nayoga *et al*. (2021) mengkaji penerapan metode *Long Short-Term Memory* (LSTM) dan *Bidirectional Long Short-Term Memory* (Bi-LSTM) untuk mendeteksi berita hoaks berbahasa Indonesia. Pada tahap praproses, data teks dipersiapkan melalui proses *case folding*, tokenisasi, dan *stopword removal* untuk

mempertahankan informasi yang relevan dalam setiap dokumen. Representasi kata menggunakan *FastText* diterapkan sebagai *word embedding* sebelum proses pelatihan model. Hasil pengujian menunjukkan bahwa model Bi-LSTM memberikan performa terbaik dengan memperoleh *accuracy* sebesar 91,2% dan *F1-score* sebesar 0,90. Temuan tersebut menunjukkan bahwa arsitektur LSTM, khususnya Bi-LSTM, memiliki kemampuan yang baik dalam memahami hubungan semantik yang kompleks pada teks berbahasa Indonesia.

Hasib *et al*. (2023) dalam penelitiannya mengusulkan kombinasi CNN dan LSTM untuk mengatasi ketidakseimbangan data berita. CNN digunakan untuk mengekstraksi fitur lokal, sementara LSTM berfungsi memahami konteks urutan kata. Model gabungan ini mencapai akurasi 93,4% dan F1-score 0,93. Temuan penelitian ini menunjukkan bahwa penggunaan arsitektur hibrida dapat menjadi alternatif yang efektif untuk meningkatkan performa klasifikasi berita multi-kategori pada dataset dengan distribusi kelas yang tidak seimbang.

Penelitian Abualigah *et al*. (2024) melakukan perbandingan beberapa model berbasis RNN, termasuk Bi-LSTM, CNN, dan DNN untuk deteksi berita palsu. Dengan memanfaatkan tahapan *text normalization* serta representasi kata menggunakan *word embedding*, model Bi-LSTM menunjukkan kinerja yang optimal dengan memperoleh *accuracy* sebesar 95,4% dan *F1-score* sebesar 0,95. Penelitian ini menunjukkan keunggulan Bi-LSTM dalam menangkap konteks dua arah yang diperlukan dalam analisis semantik, serta relevansinya untuk mendeteksi berita palsu dengan keakuratan tinggi.

Sementara itu, Ali *et al*. (2023) mengusulkan integrasi fitur linguistik dengan Bi-LSTM untuk mendeteksi berita palsu. Model ini memanfaatkan kombinasi antara fitur morfologis, sintaktis, dan semantik untuk memperkuat representasi teks. Dengan strategi ini, model mencapai akurasi 96% dan F1-score 0,96. Temuan ini memperlihatkan bahwa penggabungan teknik linguistik dengan LSTM mampu meningkatkan pemahaman konteks dan mengurangi kesalahan klasifikasi pada berita palsu.

Penelitian oleh Widhiyasana *et al.* (2021) menggunakan *Convolutional Long Short-Term Memory* (ConvLSTM) untuk klasifikasi berita berbahasa Indonesia. Data teks melalui tahapan *preprocessing* yang meliputi tokenisasi, normalisasi teks, dan *stopword removal* guna meningkatkan kualitas data masukan. Pada arsitektur yang digunakan, lapisan konvolusional berperan dalam mengekstraksi fitur-fitur penting dari teks, sedangkan lapisan LSTM dimanfaatkan untuk mempelajari hubungan atau dependensi antar kata dalam urutan teks. Model tersebut dievaluasi menggunakan dataset berita nasional dan memperoleh *accuracy* sebesar 94,8%, *precision* sebesar 93,7%, *recall* sebesar 94,1%, serta *F1-score* sebesar 93,9%. Hasil pengujian tersebut menunjukkan bahwa ConvLSTM mampu menangkap konteks semantik secara efektif sehingga memberikan performa yang baik dalam tugas klasifikasi teks berbahasa Indonesia.

Terakhir, Chen *et al*. (2022) mengembangkan model LSTM untuk klasifikasi multi-kategori berita dengan dataset besar yang mencakup berbagai topik seperti politik, ekonomi, dan hiburan. Tahapan praproses meliputi tokenisasi, padding, dan pembentukan embedding sebelum masuk ke lapisan LSTM. Model ini mencapai

akurasi 95,2% dan F1-score 0,94. Penelitian ini memberikan bukti kuat bahwa arsitektur LSTM sangat efektif dalam memahami konteks global teks berita serta relevan untuk tugas klasifikasi skala besar.

Tabel 2.1 Ringkasan Penelitian Terdahulu

No.	Peneliti (Tahun)	*Preprocessing*						Metode		Hasil
		Data Cleaning	Case Folding	Tokenizing	Normalization	Stopward Removal	Stemming	Pre	Main	
1.	Liu (2024)	-	-	-	-	✓	✓	Word2Vec	LSTM	Akurasi 94%, *F1-score* 0,91
2.	Khuntia & Gupta (2022)	-	-	✓	-	-	-	Word Embedding	LSTM	Akurasi 93,6%, F1-score 0,92
3.	Nayoga *et al.* (2021)	-	✓	✓	-	✓	-	FastText	LSTM & Bi-LSTM	Akurasi 91,2%, F1-score 0,90
4.	Hasib *et al.* (2023)	-	-	✓	✓	-	-	CNN	LSTM	Akurasi 93,4%, F1-score 0,93
5.	Abualigah *et al.* (2024)	-	-	-	✓	-	-	Word Embedding	Bi-LSTM, CNN, DNN	Akurasi 95,4%, F1-score 0,95
6.	Ali *et al.* (2023)	-	-	-	-	-	-	Linguistic Feature Extraction	Bi-LSTM	Akurasi 96%, F1-score 0,96
7.	Widhiyasana *et al.* (2021)	-	-	✓	✓	✓	-	CNN (Convolutional Layer)	ConvLSTM	Akurasi 94,8%, Precision 93,7%, Recall 94,1%, F1-score 93,9%
8.	Chen *et al*. (2022)	-	-	✓	-	-		Word Embedding	LSTM	Akurasi 95,2%, F1-score 0,94

Berbagai penelitian terdahulu menunjukkan bahwa metode *deep learning*, khususnya *Long Short-Term Memory* (LSTM), telah banyak diterapkan pada berbagai tugas klasifikasi teks dan mampu memberikan performa yang baik, termasuk dalam klasifikasi berita. Kemampuan LSTM dalam mempelajari hubungan antar kata pada data sekuensial memungkinkan model memahami konteks kalimat serta makna semantik secara lebih efektif dibandingkan beberapa metode klasifikasi konvensional. Selain itu, penggunaan teknik *word embedding*, seperti *Word2Vec*, *FastText*, dan *GloVe*, terbukti mampu menghasilkan representasi kata yang lebih baik sehingga dapat meningkatkan performa model LSTM dalam proses klasifikasi.

Berdasarkan hasil kajian tersebut, penelitian ini memanfaatkan metode LSTM untuk melakukan klasifikasi kategori berita berbahasa Indonesia. Fokus utama penelitian adalah menerapkan LSTM dengan tahapan *preprocessing* dan representasi teks yang sesuai dengan karakteristik bahasa Indonesia, guna menghasilkan model klasifikasi yang akurat dan efisien. Pendekatan ini diharapkan mampu memberikan kontribusi dalam pengembangan sistem klasifikasi berita otomatis yang lebih adaptif terhadap variasi topik dan gaya bahasa berita nasional.

2.2 Klasifikasi Teks

Klasifikasi teks merupakan salah satu bidang dalam *Natural Language Processing* (NLP) yang berfokus pada proses pengelompokan dokumen ke dalam kategori tertentu sesuai dengan isi atau karakteristik teks. Proses klasifikasi diawali dengan mengubah teks menjadi representasi numerik yang dapat diproses oleh algoritma *machine learning*. Selanjutnya, model mempelajari pola-pola linguistik

dari data latih untuk membedakan setiap kategori secara otomatis (Sebastiani, 2001). Pada penelitian ini, teknik klasifikasi teks diterapkan untuk mengelompokkan berita daring berbahasa Indonesia ke dalam beberapa kategori, seperti politik, ekonomi, olahraga, teknologi, dan kategori lainnya.

Menurut Aggarwal & Zhai (2012), klasifikasi teks terdiri dari dua komponen utama, yaitu *feature extraction* untuk mengubah teks menjadi vektor numerik, dan *classifier model* yang mempelajari hubungan antara fitur dan label kategori. Beberapa metode klasik yang sering digunakan antara lain *Naïve Bayes*, *Support Vector Machine* (SVM), dan *Decision Tree*. Namun, seiring perkembangan teknologi, metode *deep learning* seperti *Recurrent Neural Network* (RNN) dan *Long Short-Term Memory* (LSTM) menjadi lebih populer karena kemampuannya dalam menangkap konteks semantik dan hubungan urutan antar kata dalam teks (Kim, 2016).

Penelitian terdahulu menunjukkan bahwa model LSTM sangat efektif dalam tugas klasifikasi teks, termasuk berita, karena arsitekturnya memungkinkan jaringan untuk mempertahankan informasi penting dari urutan sebelumnya dan meminimalkan masalah *vanishing gradient* yang sering terjadi pada RNN konvensional (Hochreiter & Urgen Schmidhuber, 1997). Selain itu, Tri Hermanto *et al.* (2021) menunjukkan bahwa model LSTM-CNN dengan representasi Word2Vec mampu mengenali konteks kalimat secara lebih baik dalam klasifikasi berita online berbahasa Indonesia. Hasil serupa juga diperoleh oleh *Widhiyasana et al.* (2021), yang menerapkan *Convolutional Long Short-Term Memory* (Conv-LSTM) untuk klasifikasi berita dan melaporkan peningkatan akurasi dibandingkan

dengan metode tradisional seperti SVM dan *Naïve Bayes*. Dengan kemampuan tersebut, LSTM menjadi pilihan yang tepat untuk diterapkan dalam penelitian ini guna mengklasifikasikan berita online secara otomatis dan akurat.

2.3 Preprocessing

Preprocessing merupakan tahap awal yang sangat penting dalam pengolahan data teks sebelum dilakukan analisis atau pemodelan. Tahap ini bertujuan untuk membersihkan, menstandarkan, dan menyiapkan teks agar dapat diubah menjadi bentuk representasi numerik yang dapat diproses oleh algoritma pembelajaran mesin (Indurkhya & Damerau, 2020). Pada penelitian ini, preprocessing dilakukan terhadap data berita berbahasa Indonesia untuk memastikan bahwa setiap teks memiliki format yang seragam dan siap digunakan dalam proses *clustering* maupun *classification.* Tahapan *text preprocessing* dalam penelitian ini meliputi beberapa langkah berikut:

a. *Case Folding*

Case folding dilakukan dengan mengubah seluruh huruf dalam teks menjadi huruf kecil (*lowercase*). Langkah ini penting untuk menghindari redundansi data akibat perbedaan kapitalisasi kata, misalnya “Pemerintah” dan “pemerintah” yang semestinya diperlakukan sebagai kata yang sama (Manning, Raghavan, & Schütze, 2009).

b. *Tokenizing*

Tokenizing adalah proses memecah teks menjadi potongan-potongan kata atau token. Setiap token merepresentasikan unit makna yang nantinya akan digunakan dalam analisis atau pembelajaran model. Misalnya, kalimat “Pemerintah

mengeluarkan kebijakan baru" akan diubah menjadi daftar token ["pemerintah", "mengeluarkan", "kebijakan", "baru"].

c. *Stopword Removal*

Tahap ini menghapus kata-kata umum yang tidak memiliki kontribusi besar terhadap makna konteks seperti "yang", "dan", "di", atau "ke". Dengan menghapus *stopword*, ukuran fitur dapat diperkecil tanpa kehilangan makna semantik penting (Yan *et al.*, 2022).

d. *Padding*

Padding merupakan proses penyesuaian panjang urutan teks agar seluruh data memiliki dimensi input yang sama sebelum dimasukkan ke dalam model LSTM. Proses ini dilakukan dengan menambahkan nilai nol (*zero-padding*) pada urutan kata yang lebih pendek dari panjang maksimum yang telah ditentukan. Tujuannya agar model dapat memproses data dalam batch dengan ukuran seragam tanpa kehilangan konteks urutan kata. Menurut (Chollet, 2021), padding diperlukan karena jaringan saraf berurutan memerlukan input dengan ukuran tetap untuk menjaga stabilitas pelatihan dan efisiensi komputasi.

e. *Encoding Label*

Encoding label adalah proses mengubah kategori teks ke dalam bentuk numerik agar dapat diproses oleh model pembelajaran mesin. Teknik yang umum digunakan adalah *One-Hot Encoding*, di mana setiap kelas diubah menjadi vektor biner berdimensi sesuai jumlah kategori. Representasi ini memastikan bahwa tiap kelas memiliki bobot yang sama dan tidak menimbulkan interpretasi urutan antar label. (Farnham *et al.*, 2019) menjelaskan bahwa *One-Hot Encoding* efektif untuk

tugas klasifikasi karena mencegah model menganggap adanya hubungan ordinal antar kategori.

2.4 Representasi Teks

Representasi teks merupakan tahap penting dalam pemrosesan bahasa alami (*Natural Language Processing*/NLP) karena model pembelajaran mesin tidak dapat memahami data dalam bentuk teks mentah. Salah satu metode populer untuk mengubah teks menjadi representasi numerik adalah *Word2Vec*, yang dikembangkan oleh Mikolov *et al.* (2013) di *Google*. *Word2Vec* bekerja dengan cara memetakan setiap kata ke dalam ruang vektor berdimensi tinggi, di mana kedekatan antar vektor menunjukkan hubungan semantik antar kata. Dengan demikian, kata-kata yang memiliki makna serupa akan memiliki vektor yang berdekatan dalam ruang representasi tersebut.

Word2Vec memiliki dua arsitektur utama, yaitu *Continuous Bag of Words* (CBOW) dan *Skip-Gram*. CBOW memprediksi kata target berdasarkan konteks sekitarnya, sedangkan *Skip-Gram* melakukan hal sebaliknya, yaitu memprediksi konteks berdasarkan kata target. Pendekatan ini memungkinkan model untuk belajar representasi kata secara efisien melalui pelatihan pada korpus besar. Menurut Goldberg & Levy (2014), *Word2Vec* mampu menangkap hubungan sintaksis dan semantik antar kata dengan sangat baik, sehingga sering digunakan sebagai embedding layer dalam model deep learning seperti LSTM untuk tugas klasifikasi teks.

Secara matematis, fungsi objektif dari model *Skip-Gram* ditunjukkan dengan rumus berikut (Mikolov *et al.*, 2013):

$$J = \frac{1}{T}\sum_{t=1}^{T} \sum_{-c \le j \le c, j \neq 0} \log P(w_{t+j} \mid w_t) \quad (2.1)$$

Keterangan:

1) J : fungsi objektif yang akan dimaksimalkan
2) T: jumlah total kata dalam korpus.
3) t: indeks posisi kata target dalam corpus, dengan nilai mulai dari 1 hingga T.
4) c : ukuran *context window*, yaitu jumlah kata di sekitar kata target yang dipertimbangkan.
5) j : indeks posisi relatif kata konteks terhadap kata target dalam jendela konteks.
6) w_t : kata target yang sedang diproses.
7) w_{t+j}: kata konteks di sekitar kata target.
8) $P(w_{t+j} \mid w_t)$: probabilitas kata konteks muncul berdasarkan kata target

Pada persamaan 2.1, simbol Jmenyatakan fungsi objektif (*objective function*) yang digunakan untuk mengukur performa model dan akan dimaksimalkan selama proses pelatihan. Simbol Tmenunjukkan jumlah total kata dalam korpus, sedangkan $t$merupakan indeks posisi kata target dalam korpus dengan rentang nilai dari 1 hingga T. Parameter $c$menyatakan ukuran *context window*, yaitu jumlah kata di sekitar kata target yang digunakan sebagai konteks. Simbol $j$merupakan indeks posisi relatif kata konteks terhadap kata target pada jendela konteks. Selanjutnya, w_t menyatakan kata target (*target word*) yang sedang diproses, sedangkan w_{t+j} merupakan kata konteks (*context word*) yang berada di sekitar kata target. Notasi $P(w_{t+j} \mid w_t)$ menunjukkan probabilitas kemunculan kata konteks berdasarkan kata target. Simbol logmerupakan fungsi logaritma yang digunakan untuk mempermudah proses optimasi model, sedangkan simbol $\sum$ menunjukkan proses penjumlahan terhadap seluruh kata target dan kata konteks dalam korpus. Adapun bentuk $\frac{1}{T}$ digunakan untuk menghitung rata-rata nilai fungsi objektif terhadap seluruh kata dalam korpus.. Probabilitas kemunculan kata konteks dihitung menggunakan fungsi *softmax* berikut:

$$P(w_O \mid w_I) = \frac{\exp(v_{w_O}^{\top} v_{w_I})}{\sum_{w=1}^{V} \exp(v_w^{\top} v_{w_I})} \quad (2.2)$$

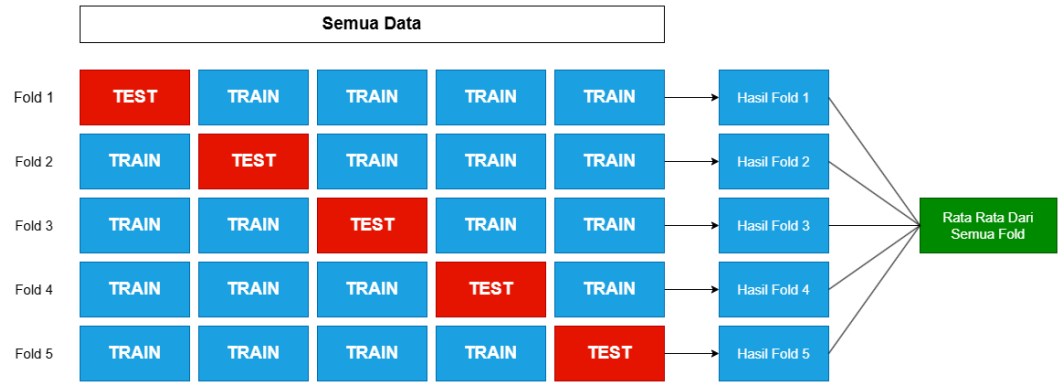
Keterangan:

1) w_I : kata input (target)
2) w_O : kata output (konteks)
3) v_{w_I}: vektor input dari kata target.
4) v_{w_O}: vektor output dari kata konteks.
5) V : ukuran kosakata (*vocabulary size*).

Pada persamaan 2.2, fungsi *softmax* pada metode *Skip-Gram*, simbol $P(w_O \mid w_I)$ menyatakan probabilitas kemunculan kata konteks berdasarkan kata target. Simbol $w_I$menunjukkan kata *input* (*input word*) atau kata target yang sedang diproses oleh model, sedangkan $w_O$merupakan kata *output* (*output word*) atau kata konteks yang berada di sekitar kata target. Notasi v_{w_I}menyatakan representasi vektor input dari kata target, sedangkan v_{w_O}merupakan representasi vektor output dari kata konteks. Simbol $v_{w_O}^{\top}$ menunjukkan operasi transpose pada vektor, sehingga $v_{w_O}^{\top} v_{w_I}$ merepresentasikan hasil perkalian dot product antara dua vektor kata. Sementara itu, simbol V menyatakan ukuran kosakata (*vocabulary size*), yaitu jumlah seluruh kata unik yang terdapat dalam korpus. Fungsi expmerupakan fungsi eksponensial yang digunakan dalam perhitungan *softmax* untuk mengubah nilai hasil perkalian vektor menjadi nilai probabilitas. Persamaan tersebut digunakan untuk menghitung seberapa besar kemungkinan suatu kata konteks muncul di sekitar kata target berdasarkan kedekatan representasi vektornya. Semakin tinggi nilai hasil perkalian antar vektor, maka semakin besar kemungkinan kedua kata memiliki hubungan semantik yang kuat.

2.5 K-Fold Cross Validation

K-Fold Cross Validation merupakan salah satu metode evaluasi yang banyak digunakan dalam machine learning untuk mengukur kemampuan generalisasi model terhadap data yang belum pernah dilihat sebelumnya. Metode ini bekerja dengan membagi dataset menjadi K bagian (fold) yang berukuran relatif sama. Pada setiap iterasi, satu fold digunakan sebagai data pengujian (*testing data*), sedangkan fold lainnya digunakan sebagai data pelatihan (*training data*). Proses tersebut dilakukan secara berulang sebanyak K kali hingga seluruh fold pernah digunakan sebagai data pengujian [(Rianansyah *et al.*, 2025).



Gambar 2.1 Ilustrasi K-Fold Cross Validation

Gambar 2.1 menunjukkan pada setiap iterasi satu bagian data digunakan sebagai data pengujian, sedangkan bagian lainnya digunakan sebagai data pelatihan. Hasil evaluasi dari setiap iterasi kemudian dirata-ratakan untuk memperoleh nilai performa akhir model. Dengan cara ini, setiap data memiliki kesempatan yang sama untuk menjadi data pelatihan maupun data pengujian

sehingga hasil evaluasi menjadi lebih objektif dan tidak bergantung pada satu skenario pembagian data tertentu(Oyedele, 2023).

Dibandingkan dengan metode *hold-out validation* yang hanya melakukan satu kali pembagian data*, K-Fold Cross Validation* mampu menghasilkan estimasi performa yang lebih stabil dan representatif. Metode ini juga dapat mengurangi risiko bias yang disebabkan oleh distribusi data yang tidak merata pada proses pembagian data. Oleh karena itu, *K-Fold Cross Validation* sering digunakan untuk mengevaluasi model klasifikasi dan prediksi, terutama pada penelitian yang bertujuan memperoleh hasil evaluasi yang lebih akurat.

2.6 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) merupakan pengembangan dari arsitektur *Recurrent Neural Network* (RNN) yang dirancang untuk mengatasi permasalahan *vanishing gradient* pada pelatihan jaringan saraf berurutan. Model ini diperkenalkan oleh Hochreiter & Urgen Schmidhuber (1997), dan sejak itu menjadi salah satu model paling efektif dalam menangani data sekuensial seperti teks, audio, maupun deret waktu (*time series*). Berbeda dengan RNN konvensional yang cenderung kehilangan informasi ketika urutan data terlalu panjang, LSTM memiliki mekanisme memori jangka panjang yang memungkinkan model untuk mempertahankan informasi penting selama proses pembelajaran berlangsung.

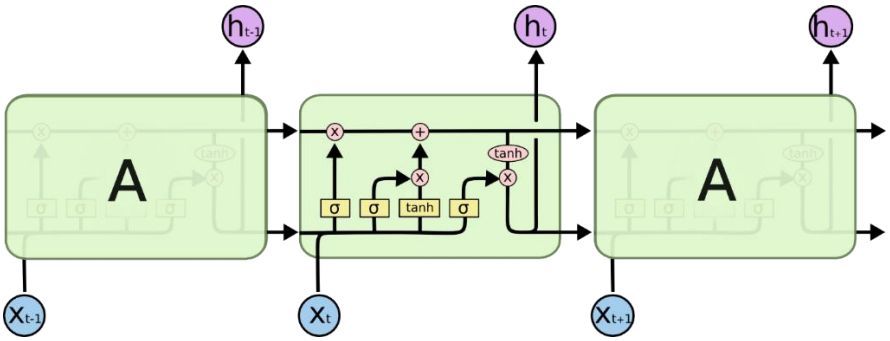
LSTM memiliki struktur internal yang lebih kompleks dibandingkan RNN biasa karena terdiri atas tiga komponen utama yang disebut gate, yaitu *input gate, forget gate,* dan *output gate*. Ketiga gerbang ini berperan penting dalam mengatur aliran informasi yang masuk, disimpan, dan dikeluarkan dari unit memori (*cell*

state). *Input gate* bertugas menentukan informasi baru yang akan ditambahkan ke memori, *forget gate* mengatur bagian informasi mana yang harus dilupakan, dan *output gate* menentukan informasi apa yang akan diteruskan ke lapisan berikutnya. Kombinasi ketiga gerbang ini membuat LSTM mampu menjaga keseimbangan antara informasi lama dan baru, sehingga konteks dari data urutan sebelumnya tetap relevan terhadap prediksi saat ini (Olah, 2015).

Kemampuan utama LSTM terletak pada cell state yang berfungsi seperti "jalur memori" utama. Aliran data melewati *cell state* dengan sedikit modifikasi, menjaga agar informasi penting tidak hilang selama proses propagasi. Selain itu, setiap gerbang dalam LSTM menggunakan fungsi sigmoid untuk menentukan seberapa besar informasi dapat diteruskan, serta fungsi aktivasi tanh untuk menjaga nilai agar tetap stabil dalam rentang tertentu. Dengan demikian, LSTM dapat mempelajari hubungan jangka panjang antar elemen dalam urutan teks (Kim, 2016).

Dalam konteks pemrosesan bahasa alami (*Natural Language Processing*), LSTM terbukti efektif untuk memahami konteks kalimat dan hubungan antar kata, terutama dalam tugas-tugas seperti klasifikasi teks, analisis sentimen, dan pemrosesan berita. Beberapa penelitian menunjukkan bahwa LSTM mampu mengenali pola semantik antar kata dan mempertahankan konteks kalimat secara akurat (Greff *et al.*, 2017). Dalam penelitian ini LSTM digunakan sebagai model

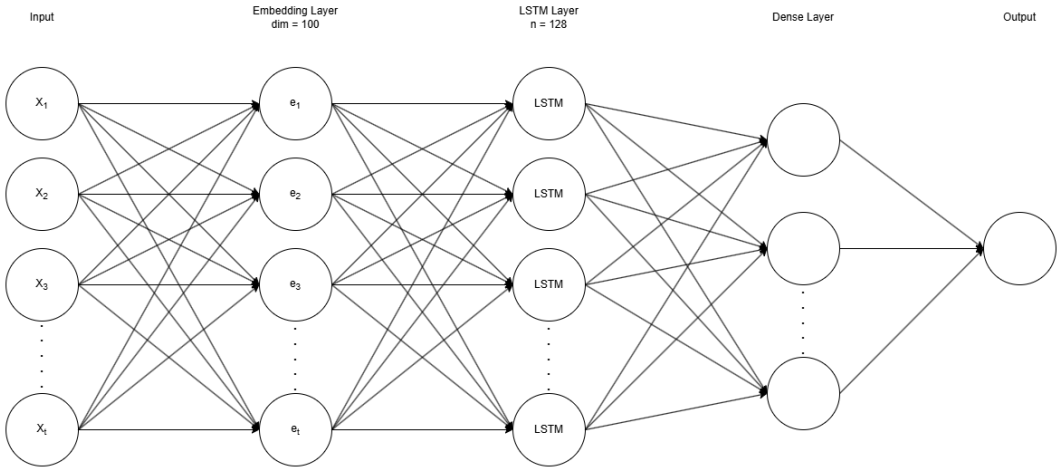
utama untuk klasifikasi kategori berita berdasarkan representasi vektor kata yang dihasilkan oleh *Word2Vec*.



Gambar 2.2 Arsitektur Long Short-Term Memory (LSTM)

Gambar 2.2 memperlihatkan aliran data pada arsitektur *Long Short-Term Memory* (LSTM), di mana komponen *input gate, forget gate,* dan *output gate* berperan penting dalam mengatur proses pembaruan dan penyimpanan informasi pada *cell state* serta *hidden state*. Setiap panah pada diagram menunjukkan arah propagasi informasi di setiap time step, menggambarkan bagaimana data masukan diolah dan diteruskan dari satu unit ke unit berikutnya dalam urutan waktu. Gambar ini diadaptasi dari Olah, (2015) yang memvisualisasikan struktur internal LSTM secara sistematis untuk menjelaskan mekanisme pengendalian memori jangka panjang pada jaringan saraf berulang.

Untuk memperoleh model klasifikasi yang optimal, dilakukan perancangan arsitektur jaringan seperti yang ditunjukkan pada Gambar 2.2.



Gambar 2.3 Alur Pemrosesan Data pada Model LSTM

Pada Gambar 2.3 dijelaskan alur pemrosesan data pada model LSTM. Model menerima masukan berupa urutan kata hasil dari tahap *preprocessing* dan *text representation* dengan panjang urutan maksimal sebanyak 200 token. Setiap token direpresentasikan ke dalam bentuk vektor berdimensi 100, yang dihasilkan melalui proses *Word2Vec embedding*. Representasi vektor ini memungkinkan model untuk memahami hubungan semantik antar kata dalam korpus berita yang digunakan.

1) Input Layer

Pada tahap awal pemrosesan, teks berita yang telah melalui proses pembersihan dan normalisasi dikonversi menjadi serangkaian token melalui *tokenizer* yang mengubah setiap kata menjadi indeks numerik berdasarkan posisi kata tersebut dalam *vocabulary*, sehingga teks yang bersifat simbolik dapat diproses oleh model deep learning tanpa kehilangan struktur urutannya. Setiap dokumen kemudian direpresentasikan sebagai urutan dengan panjang tetap, yaitu 200 token. apabila jumlah kata lebih sedikit dari 200, maka sistem menambahkan *padding* di

bagian akhir, sedangkan jika lebih panjang, token melebihi batas akan dipotong sehingga hanya 200 token pertama yang digunakan. Konsistensi panjang ini penting karena model LSTM membutuhkan tensor dengan dimensi seragam untuk dapat diproses secara paralel selama pelatihan. Urutan token tersebut akhirnya membentuk vektor input berdimensi tetap, $X = [x_1, x_2, \ldots, x_{200}]$, di mana setiap elemen x_t merupakan representasi numerik dari kata ke-t dalam teks, yang kemudian diteruskan menuju tahap embedding untuk memperoleh representasi semantik yang lebih bermakna.

2) Embedding Layer

Pada tahap *embedding*, setiap token hasil *preprocessing* dipetakan menjadi vektor representasi menggunakan model *Word2Vec* dengan arsitektur *Skip-Gram*. Berbeda dari *embedding* bawaan Keras, *Word2Vec Skip-Gram* mempelajari representasi kata dengan memprediksi konteks berdasarkan kata target sehingga hubungan semantik antar kata dapat ditangkap secara lebih mendalam. Model menghasilkan *embedding* berdimensi 100, sehingga setiap kata direpresentasikan sebagai vektor berukuran 100 elemen. Ketika urutan token sepanjang 200 posisi dimasukkan, setiap indeks kata digunakan untuk mengambil satu baris vektor dari *embedding matrix* hasil pelatihan *Word2Vec*, menghasilkan output berupa matriks berukuran 200×100 yang menyimpan representasi semantik setiap kata sesuai urutannya dalam dokumen. Dengan representasi ini, kata-kata yang memiliki makna atau konteks serupa akan memiliki vektor yang berdekatan, sehingga tahap LSTM selanjutnya dapat lebih efektif memahami pola bahasa yang ada dalam teks.

3) LSTM *Layer*

Layer LSTM bertanggung jawab menangkap hubungan antar kata dalam urutan kalimat, termasuk konteks jangka pendek maupun jangka panjang. Berbeda dari RNN biasa, LSTM memiliki struktur gerbang yang lebih kompleks, sehingga mampu mengatasi masalah *vanishing gradient*. Dengan demikian, LSTM dapat mempelajari pola teks yang panjang, seperti hubungan antar kalimat atau struktur wacana tertentu.

Pada model ini, LSTM memiliki 128 unit. Setiap unit merupakan komponen yang menyimpan informasi dalam bentuk *hidden state* dan *cell state*. LSTM memproses input embedding secara berurutan mulai dari kata ke-1 hingga kata ke-200. Pada setiap langkah waktu t, LSTM memperbarui nilai *hidden state* $h_t$dan *cell state* $c_t$menggunakan empat gerbang: *forget gate, input gate, candidate memory,* dan *output gate*.

Hidden state terakhir h_{200} dianggap sebagai representasi dokumen secara keseluruhan. Representasi ini tidak hanya berisi informasi kata individual, tetapi juga hubungan konteks yang terbentuk dari seluruh urutan kata. *Hidden state* tersebut sangat penting dan menjadi masukan utama untuk tahap klasifikasi selanjutnya.

Tahap pertama adalah menghitung nilai *forget gate*, yang menentukan bagian informasi dari *cell state* sebelumnya yang akan dipertahankan.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{2.3}$$

Keterangan:

1) f_t: nilai *forget gate*
2) σ: fungsi *sigmoid*
3) W_f: bobot *forget gate* yang didapat selama pelatihan
4) h_{t-1}: *hidden state* dari waktu sebelumnya

5) x_t: input pada waktu ke-t
6) b_f: bias pada *forget gate*

Langkah berikutnya adalah menghitung *input gate*, yang berfungsi untuk mengatur seberapa besar informasi baru yang akan dimasukkan ke dalam *cell state*. Persamaannya sebagai berikut:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{2.4}$$

Keterangan:
1) i_t: nilai *input gate*
2) σ: fungsi *sigmoid*
3) W_i: bobot *input gate* yang diperoleh selama pelatihan
4) h_{t-1}: *hidden state* dari waktu sebelumnya
5) x_t: input pada waktu ke-t
6) b_i: bias pada input gate

Selanjutnya, dilakukan perhitungan *candidate cell state*, yang menghasilkan informasi baru yang diusulkan untuk dimasukkan ke dalam *cell state*. Persamaannya sebagai berikut:

$$g_t = \tanh(W_g[h_{t-1}, x_t] + b_g) \tag{2.5}$$

Keterangan:
1) g_t: nilai *candidate cell state*
2) tanh: fungsi aktivasi *hyperbolic tangent*
3) W_g: bobot *candidate cell state*
4) h_{t-1}: *hidden state* dari waktu sebelumnya
5) x_t: input pada waktu ke-t
6) b_g: bias pada *candidate cell state*

Tahap berikutnya adalah menghitung *output gate*, yang menentukan bagian dari *cell state* yang akan diteruskan sebagai *hidden state* pada waktu ke-t. Persamaannya dituliskan sebagai berikut:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{2.6}$$

Keterangan:
1) o_t: nilai *output gate*
2) σ: fungsi *sigmoid*
3) W_o: bobot *output gate* yang diperoleh selama pelatihan
4) h_{t-1}: *hidden state* dari waktu sebelumnya
5) x_t: input pada waktu ke-t
6) b_o: bias pada *output gate*

Setelah nilai dari setiap gerbang dihitung, langkah berikutnya adalah memperbarui *cell state* (C_t) dan *hidden state* (h_t). *Cell state* diperbarui dengan menggabungkan hasil dari *forget gate* dan *input gate* untuk mempertahankan sebagian informasi lama serta menambahkan informasi baru yang relevan. Sedangkan *hidden state* diperoleh dari hasil *output gate* yang mengontrol informasi dari *cell state* yang telah diperbarui. Persamaan pembaruan *cell state* dan *hidden state* dituliskan sebagai berikut:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot g_t \tag{2.7}$$

Keterangan:
1) C_t: nilai *cell state* pada waktu ke-t
2) f_t: nilai *forget gate*
3) C_{t-1}: *cell state* dari waktu sebelumnya
4) i_t: nilai *input gate*
5) g_t: nilai *candidate cell state*

$$h_t = o_t \cdot \tanh(C_t) \tag{2.8}$$

Keterangan:
1) h_t: nilai *hidden state* pada waktu ke-t
2) o_t: nilai *output gate*
3) tanh: fungsi aktivasi *hyperbolic tangent*
4) C_t: nilai *cell state* pada waktu ke-t

Nilai *hidden state* terakhir (h_t) yang dihasilkan pada akhir urutan akan digunakan sebagai representasi dari keseluruhan teks yang diproses. Representasi ini kemudian diteruskan ke lapisan *dense* dengan fungsi aktivasi *softmax* untuk menghasilkan output klasifikasi kategori berita. Proses *forward* berlangsung secara berurutan untuk setiap token input (sebanyak 200 token), dengan memperbarui *cell state* dan *hidden state* pada setiap langkah sehingga model dapat menangkap konteks dan hubungan antar kata dalam teks. Pendekatan ini memungkinkan model untuk mempertahankan informasi penting dan mengabaikan bagian yang tidak relevan, sehingga meningkatkan akurasi dalam proses klasifikasi berita.

4) Dense Layer

Setelah *hidden state* terakhir dari LSTM diperoleh, nilai tersebut diteruskan ke dalam *dense*. *Dense layer* ini berfungsi sebagai pemetaan non-linear dari representasi dokumen ke ruang fitur yang lebih ringkas. Aktivasi yang digunakan adalah ReLU (*Rectified Linear Unit*), yang mampu mempercepat konvergensi dan mengatasi masalah *vanishing gradient* pada tahap *feed-forward*. Operasi *dense layer d*apat dituliskan dalam bentuk rumus:

$$z = ReLU(W_d h_{200} + b_d) \tag{2.9}$$

Keterangan:
1) z: vektor keluaran dari *dense layer*
2) W_d: matriks bobot pada *dense layer*
3) h: nilai *hidden state* terakhir dari LSTM
4) b_d: bias pada *dense layer*
5) ReLU(·): fungsi aktivasi *Rectified Linear Unit*

Lapisan ini membantu model dalam memisahkan pola-pola latent yang dipelajari oleh LSTM sehingga dapat dipetakan lebih jelas menuju kategori berita yang tersedia. Dengan adanya *dense layer*, model mampu menangkap struktur data pada level yang lebih tinggi.

5) Output Layer

Tahap akhir dari proses adalah layer *output* yang memiliki 8 *neuron* sesuai jumlah kategori berita. Layer ini menggunakan fungsi aktivasi *softmax* untuk mengubah nilai linear menjadi probabilitas. Fungsi *softmax* mengubah keluaran linear dari *layer output* menjadi distribusi probabilitas. Secara matematis, *softmax* didefinisikan sebagai:

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad (2.10)$$

Keterangan:

1) z_i: nilai *logit* ke-i dari *output layer*
2) K: jumlah kelas pada *output*
3) e^{z_i}: eksponensial dari logit
4) $\sum_{j=1}^{K} e^{z_j}$: normalisasi agar jumlah probabilitas = 1

Pada arsitektur jaringan saraf, nilai *logit* z diperoleh dari hasil operasi linear terhadap bobot dan bias *output layer*, sehingga persamaan lengkapnya dinyatakan sebagai:

$$\hat{y} = \text{softmax}(W_o z + b_o) \quad (2.11)$$

Keterangan:

1) $\hat{y}$: vektor probabilitas untuk setiap kelas
2) W_o: bobot pada *output layer*
3) z: output dari *dense layer* sebelumnya
4) b_o: bias pada output layer
5) softmax(·): fungsi aktivasi yang mengubah *logit* menjadi probabilitas

Fungsi softmax memastikan bahwa total probabilitas yang dihasilkan berjumlah 1, sehingga model dapat menentukan kelas dengan probabilitas tertinggi sebagai hasil prediksi akhir.

BAB III

METODOLOGI PENELITIAN

3.1 Rancangan Penelitian

Rancangan penelitian ini disusun untuk memberikan panduan yang sistematis dalam pelaksanaan penelitian agar berjalan sesuai dengan tujuan yang ingin dicapai. Penelitian ini berfokus pada penerapan metode *Long Short-Term Memory* (LSTM) untuk melakukan klasifikasi kategori berita berbahasa Indonesia. Model ini dipilih karena kemampuannya dalam memahami konteks dan urutan kata pada teks berita. Dataset yang digunakan dalam penelitian ini adalah *Indonesian News Corpus*, yang berisi kumpulan berita dari berbagai kategori seperti politik, bisnis, olahraga, hiburan, dan teknologi.

Gambar 3.1 Rancangan Penelitian

Gambar 3.1 menunjukkan alur umum proses penelitian yang dilakukan, di mana setiap tahap saling berkaitan untuk mencapai hasil klasifikasi yang optimal. Penelitian diawali dengan identifikasi masalah dan studi literatur untuk memahami teori serta pendekatan yang relevan terhadap klasifikasi teks berita. Selanjutnya, dilakukan pengumpulan dan persiapan data dengan menerapkan tahapan preprocessing seperti *case folding*, *tokenization*, *stopword removal*, *padding*, dan *encoding label* agar data siap digunakan dalam proses pembelajaran model. Setelah itu, teks direpresentasikan menggunakan *Word2Vec* untuk mengubah kata menjadi vektor numerik yang dapat diproses oleh model LSTM.

Tahap selanjutnya adalah perancangan dan implementasi model LSTM yang digunakan untuk mempelajari pola dalam teks dan mengklasifikasikan berita ke dalam kategori yang sesuai. Terakhir, dilakukan pengujian dan evaluasi model menggunakan metrik evaluasi seperti akurasi, presisi, *recall*, dan *F1-score* untuk menilai kinerja sistem klasifikasi yang dikembangkan.

3.2 Persiapan Data

Tahap persiapan data merupakan langkah awal dalam proses penelitian yang bertujuan untuk memastikan bahwa data yang digunakan memiliki kualitas dan struktur yang sesuai untuk diolah menggunakan model *Long Short-Term Memory* (LSTM). Proses ini menjadi krusial karena kualitas data akan sangat memengaruhi kinerja dan akurasi model klasifikasi yang dibangun.

Dataset yang digunakan dalam penelitian ini adalah *Indonesian News Corpus*, yaitu kumpulan teks berita berbahasa Indonesia yang dikembangkan untuk keperluan penelitian di bidang *Natural Language Processing (NLP)*, khususnya

pada tugas klasifikasi teks, analisis sentimen, dan pemrosesan bahasa alami lainnya. Dataset ini bersifat terbuka (*open source*) dan dapat diakses melalui repositori dengan tautan berikut : https://data.mendeley.com/datasets/2zpbjs22k3/1

Dataset ini berisi 150,466 berita berbahasa Indonesia yang dikumpulkan dari berbagai portal berita daring nasional selama enam bulan mulai dari Juli 2015 hingga Desember 2015, mencakup beragam kategori Teknologi, Otomotif, Bisnis Ekonomi, Nasional, Olahraga, Bola, Lifestyle, dan Travel.. Setiap berita dalam dataset terdiri atas beberapa komponen utama yang digunakan untuk proses analisis dan klasifikasi.

Dataset *Indonesian News Corpus* memiliki beberapa atribut utama yang berperan penting dalam proses klasifikasi teks, antara lain sebagai berikut:

Tabel 3.1 Atribut Dataset

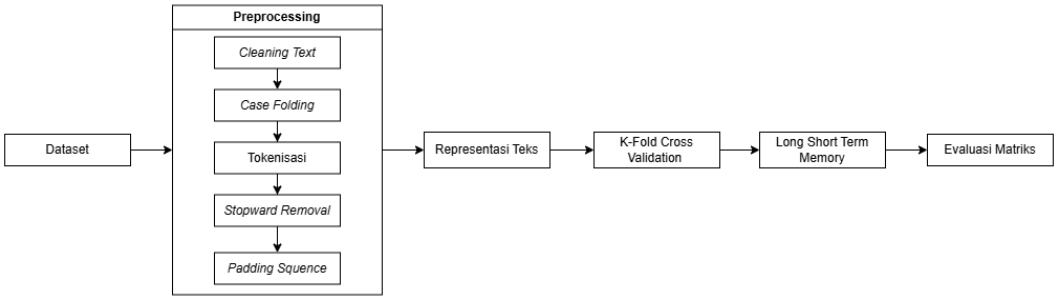
Nama Atribut	Tipe Data	Deskripsi
sumber	String	Menunjukkan asal atau portal berita tempat artikel dipublikasikan
tanggal	String	Menunjukkan tanggal publikasi berita
kategori	String	Label atau kelas berita yang menunjukkan topik utama artikel
judul	String	Judul berita yang berfungsi sebagai ringkasan singkat dari isi berita.
isi	String	Konten utama atau teks penuh berita yang akan digunakan sebagai data utama dalam proses klasifikasi.
link	String	Tautan URL ke halaman berita asli di portal sumber.

Dalam penelitian ini, atribut yang digunakan secara langsung adalah judul, isi, dan kategori. Atribut judul dan isi digabungkan untuk membentuk teks lengkap yang menjadi input bagi model LSTM, sementara kategori digunakan sebagai label target yang akan diprediksi oleh model.

Pemilihan dataset ini didasarkan pada kelengkapan dan keragamannya dalam mencakup berbagai topik berita nasional, sehingga diharapkan dapat meningkatkan kemampuan model dalam mengenali pola bahasa serta konteks semantik dalam teks berita berbahasa Indonesia.

3.3 Desain Sistem

Dalam penelitian ini diperlukan perancangan desain sistem untuk menggambarkan alur kerja metode yang digunakan secara keseluruhan. Desain sistem tersebut berfungsi sebagai panduan dalam menjelaskan bagaimana data diproses, dimodelkan, hingga menghasilkan keluaran berupa klasifikasi kategori berita. Pada gambar 3.2 ditunjukkan rancangan sistem berbasis LSTM yang meliputi proses praproses teks, pembentukan *word embedding*, serta pemodelan menggunakan LSTM hingga tahap prediksi akhir.

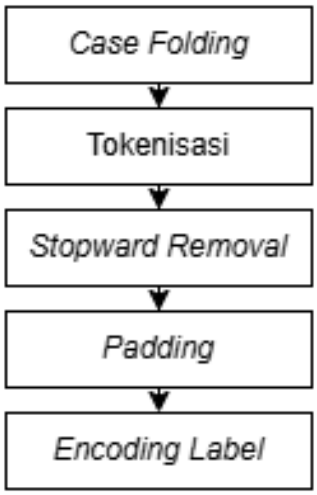


Gambar 3.2 Desain Sistem Penelitian

3.3.1 *Preprocessing*

Tahap *preprocessing* dilakukan untuk menyiapkan teks berita dari *Indonesian News Corpus* agar dapat diolah secara optimal oleh model *Long Short-Term Memory (LSTM)*. Proses ini bertujuan untuk membersihkan, menormalkan, dan menyeragamkan data teks sehingga menghasilkan representasi yang konsisten dan

mudah dipahami oleh algoritma pembelajaran mesin. Setiap tahap preprocessing saling berhubungan untuk memastikan kualitas data yang digunakan pada proses pelatihan dan pengujian model.



Gambar 3.3 Alur *Preprocessing*

a. Case Folding

Langkah pertama adalah mengubah seluruh huruf dalam teks menjadi huruf kecil (*lowercase*). Tujuan dari tahap ini adalah untuk menyeragamkan penulisan kata sehingga kata yang sama dengan perbedaan huruf kapital tidak dianggap berbeda.

Tabel 3.2 Contoh Proses *Case Folding*

Sebelum *Case Folding*	JAKARTA, KOMPAS.com Ponsel Android Huawei Honor 4C resmi meluncur di pasar Indonesia, Rabu (1/7/2015). Perusahaan asal Tiongkok itu merancangnya sebagai ponsel kelas menengah dengan banderol harga Rp 2,3 juta.
Setelah *Case Folding*	jakarta kompas com ponsel android huawei honor 4c resmi meluncur di pasar indonesia rabu 1 7 2015 perusahaan asal tiongkok itu merancangnya sebagai ponsel kelas menengah dengan banderol harga rp 2 3 juta

b. Tokenisasi

Setelah huruf diseragamkan, kalimat kemudian dipecah menjadi token atau potongan kata. Tokenisasi bertujuan agar setiap kata dapat diproses secara individual dalam tahap representasi teks.

Tabel 3.3 Contoh Proses Tokenisasi

Sebelum Tokenisasi	jakarta kompas com ponsel android huawei honor 4c resmi meluncur di pasar indonesia rabu 1 7 2015 perusahaan asal tiongkok itu merancangnya sebagai ponsel kelas menengah dengan banderol harga rp 2 3 juta
Setelah Tokenisasi	['jakarta', 'kompas', 'com', 'ponsel', 'android', 'huawei', 'honor', '4c', 'resmi', 'meluncur', 'di', 'pasar', 'indonesia', 'rabu', '1', '7', '2015', 'perusahaan', 'asal', 'tiongkok', 'itu', 'merancangnya', 'sebagai', 'ponsel', 'kelas', 'menengah', 'dengan', 'banderol', 'harga', 'rp', '2', '3', 'juta']

c. Stopword Removal

Pada tahap ini dilakukan penghapusan kata-kata umum yang tidak memberikan makna penting dalam konteks kalimat, seperti “yang”, “di”, “dan”, “akan”, dan sebagainya. Tujuannya adalah agar model lebih fokus pada kata bermakna utama.

Tabel 3.4 Contoh Proses *Stopword Removal*

Sebelum *Stopword Removal*	['jakarta', 'kompas', 'com', 'ponsel', 'android', 'huawei', 'honor', '4c', 'resmi', 'meluncur', 'di', 'pasar', 'indonesia', 'rabu', '1', '7', '2015', 'perusahaan', 'asal', 'tiongkok', 'itu', 'merancangnya', 'sebagai', 'ponsel', 'kelas', 'menengah', 'dengan', 'banderol', 'harga', 'rp', '2', '3', 'juta']
Setelah *Stopword Removal*	['jakarta', 'kompas', 'com', 'ponsel', 'android', 'huawei', 'honor', '4c', 'resmi', 'meluncur', 'pasar', 'indonesia', 'rabu', '2015', 'perusahaan', 'tiongkok', 'merancangnya', 'ponsel', 'kelas', 'menengah', 'banderol', 'harga', 'rp', 'juta']

d. *Padding*

Karena setiap teks memiliki panjang token yang berbeda, maka dilakukan proses *padding* untuk menyeragamkan panjang urutan kata agar sesuai dengan

parameter *input length* model LSTM. Kata yang kurang dari panjang maksimum akan diisi dengan nilai nol (0) di bagian akhir.

Tabel 3.5 Contoh Proses *Padding*

Sebelum *Padding*	['jakarta', 'kompas', 'com', 'ponsel', 'android', 'huawei', 'honor', '4c', 'resmi', 'meluncur', 'pasar', 'indonesia', 'rabu', '2015', 'perusahaan', 'tiongkok', 'merancangnya', 'ponsel', 'kelas', 'menengah', 'banderol', 'harga', 'rp', 'juta']
Setelah *Padding*	['jakarta', 'kompas', 'com', 'ponsel', 'android', 'huawei', 'honor', '4c', 'resmi', 'meluncur']

e. *Encoding Label*

Langkah terakhir adalah mengubah label kategori berita menjadi bentuk numerik dengan menggunakan metode *One-Hot Encoding*. Proses ini dilakukan agar label dapat dikenali oleh model LSTM dalam tahap pelatihan.

Tabel 3.6 Label *One-Hot Encoding*

Kategori	Bisnis Ekonomi	Bola	Lifestyle	Nasional	Olahraga	Otomotif	Teknologi	Travel
Bisnis Ekonomi	1	0	0	0	0	0	0	0
Bola	0	1	0	0	0	0	0	0
Lifestyle	0	0	1	0	0	0	0	0
Nasional	0	0	0	1	0	0	0	0
Olahraga	0	0	0	0	1	0	0	0
Otomotif	0	0	0	0	0	1	0	0
Teknologi	0	0	0	0	0	0	1	0
Travel	0	0	0	0	0	0	0	1

3.3.2 Representasi Teks

Setiap kata hasil preprocessing diubah menjadi vektor numerik menggunakan Word2Vec dengan arsitektur Skip-Gram dan teknik Negative Sampling, dengan dimensi vektor $d = 100$dan jendela konteks $m = 5$. Untuk mengilustrasikan proses perhitungannya, digunakan tiga kata dari korpus berita: 'jakarta', 'kompas', dan 'android', yang diasumsikan muncul dalam kalimat:

['jakarta', 'kompas', 'com', 'ponsel', 'android', 'huawei', 'honor', '4c', 'resmi', 'meluncur', 'pasar', 'indonesia', 'rabu', '2015', 'perusahaan', 'tiongkok', 'merancangnya', 'ponsel', 'kelas', 'menengah', 'banderol', 'harga', 'rp', 'juta']

Dengan $m = 5$, kata "kompas" berperan sebagai kata pusat (w), dengan kata konteks "jakarta" (c_1) dan "android" (c_2). Karena vektor 100 dimensi tidak praktis untuk ditampilkan, ilustrasi menggunakan $d = 4$.

Tabel 3.7 Inisialisasi vektor kata

Kata	Vektor(4 Dimensi)
$u_{kompas}(kata\ pusat)$	$[0.30, 0.20, -0.10, 0.50]$
$u_{jakarta}(konteks)$	$[0.20, -0.10, 0.40, 0.30]$
$u_{android}(konteks)$	$[0.10, 0.40, 0.20, -0.20]$

Menghitung probabilitas konteks menggunakan fungsi softmax pada persamaan 2.2. Dimana yang pertama adalah menghitung *dot product* untuk masing-masing pasangan:

$$v_{jakarta}^{\top} u_{kompas} = (0.20)(0.30) + (-0.10)(0.20) + (0.40)(-0.10) + (0.30)(0.50) = 0.15$$

$$v_{android}^{\top} u_{kompas} = (0.10)(0.30) + (0.40)(0.20) + (0.20)(-0.10) + (-0.20)(0.50) = -0.01$$

Asumsikan jumlah kata dalam kosakata $| V | = 3$ yang disederhanakan, sehingga:

$$P(\text{jakarta} \mid \text{kompas}) = \frac{e^{0.15}}{e^{0.15} + e^{-0.01} + e^{0}} = \frac{1.162}{1.162 + 0.990 + 1.000} \approx 0.369$$

Nilai-nilai ini kemudian digunakan pada fungsi objektif Skip-Gram sesuai Persamaan 2.1 untuk memaksimalkan log-probabilitas kata konteks yang benar:

$$J = \frac{1}{1}[\log P\,(jakarta|kompas) + \log P(androdi|kompas)]$$

$$= \log(0.369) + \log(0.314) = -0.997 + (-1.158) = -2.155$$

Nilai loss ini diminimalkan melalui backpropagation, sehingga vektor "jakarta" dan "android" secara bertahap bergerak mendekati vektor "kompas" dalam

ruang semantik. Setelah pelatihan selesai, setiap kata direpresentasikan sebagai vektor berdimensi 100 yang kemudian disusun menjadi matriks $n \times 100$ sebagai input model LSTM.

3.3.3 *K-Fold Cross Validation*

Setelah proses representasi kata menggunakan Word2Vec, data yang telah ditransformasikan ke dalam bentuk vektor digunakan pada tahap evaluasi model menggunakan metode K-Fold Cross Validation. Metode ini digunakan untuk memperoleh hasil evaluasi yang lebih stabil dan representatif dengan cara membagi dataset menjadi beberapa bagian (*fold*) yang berukuran relatif sama.

Pada penelitian ini digunakan dua skenario pengujian, yaitu 5-Fold Cross Validation dan 10-Fold Cross Validation. Pada skenario 5-fold, dataset dibagi menjadi lima bagian, di mana empat fold digunakan sebagai data pelatihan dan satu fold digunakan sebagai data pengujian pada setiap iterasi. Sementara itu, pada skenario 10-fold, dataset dibagi menjadi sepuluh bagian dengan sembilan fold digunakan sebagai data pelatihan dan satu fold digunakan sebagai data pengujian. Proses ini dilakukan secara berulang hingga seluruh fold pernah digunakan sebagai data pengujian.

Hasil evaluasi pada setiap iterasi kemudian dirata-ratakan untuk memperoleh nilai performa akhir model. Penggunaan K-Fold Cross Validation dalam penelitian ini bertujuan untuk mengurangi bias akibat pembagian data tertentu serta menghasilkan evaluasi yang lebih objektif terhadap performa model klasifikasi berita. Data pada setiap fold selanjutnya digunakan sebagai masukan bagi model Long Short-Term Memory (LSTM) untuk proses pelatihan dan pengujian.

Performa model dievaluasi menggunakan metrik *accuracy*, *precision*, *recall*, dan *F1-score* pada masing-masing skenario pengujian.

3.3.4 *Long Short Term Memory* (LSTM)

Matriks input berukuran $n \times 100$ hasil representasi *Word2Vec* diproses secara sekuensial oleh model LSTM dengan $n = 128$ unit tersembunyi, sebagaimana ditunjukkan pada arsitektur di atas. Pada setiap langkah waktu t, model menerima vektor kata $x_t \in \mathbb{R}^{100}$, *hidden state* sebelumnya $h_{t-1} \in \mathbb{R}^{128}$, dan *cell state* $C_{t-1} \in \mathbb{R}^{128}$, kemudian memprosesnya melalui empat gerbang (Persamaan 2.xx - 2.xx). Implementasi perhitungan menggunakan *embedding* kata "kompas" sebagai x_t pada satu langkah waktu. Karena dimensi asli (100 dan 128) tidak praktis ditampilkan, ilustrasi menggunakan $d = 2$ dan $n = 2$.

Tabel 3.8 Nilai awal yang digunakan

Variable	Nilai
x_t (embedding "kompas")	[0.30,0.20]
h_{t-1}	[0.10,−0.05]
C_{t-1}	[0.20,0.15]

Dengan input gabungan: $[h_{t-1}, x_t\ ] = [0.10, -0.05, 0.30, 0.20]$

Forget gate : memutuskan informasi mana dari C_{t-1} yang dibuang:

$$W_f = \begin{bmatrix} 0.90 & 0.20 & 0.50 & 0.30 \\ 0.10 & -0.20 & 0.10 & 0.00 \end{bmatrix}, b_f = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Sehingga :

Baris 1:$(0.90)(0.10) + (0.20)(-0.05) + (0.50)(0.30) + (0.30)(0.20) =$

$0.09 - 0.01 + 0.15 + 0.06 = 0.29$

Baris 2:$(0.10)(0.10) + (-0.20)(-0.05) + (0.10)(0.30) + (0.00)(0.20) =$

$0.01 + 0.01 + 0.03 + 0.00 = 0.05$

$$W_f \cdot [h_{t-1}, x_t\ ] + b_f = \begin{bmatrix} 0.29 \\ 0.05 \end{bmatrix}$$

Kemudian fungsi *sigmoid* diterapkan :

$$f_t = \sigma\begin{pmatrix} 0.29 \\ 0.05 \end{pmatrix} = \begin{bmatrix} \frac{1}{1+e^{-0.29}} \\ \frac{1}{1+e^{-0.05}} \end{bmatrix} \approx \begin{bmatrix} 0.572 \\ 0.512 \end{bmatrix}$$

Input gate : memutuskan informasi baru mana yang disimpan:

$$W_i = \begin{bmatrix} 0.60 & 0.20 & -0.20 & 0.10 \\ 0.60 & -0.20 & 0.30 & 0.30 \end{bmatrix}, b_i = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Baris 1: $(0.60)(0.10) + (0.20)(-0.05) + (-0.20)(0.30) + (0.10)(0.20) =$

$0.06 - 0.01 - 0.06 + 0.02 = 0.01$

Baris 2: $(0.60)(0.10) + (-0.20)(-0.05) + (0.30)(0.30) + (0.30)(0.20) =$

$0.06 + 0.01 + 0.09 + 0.06 = 0.22$

$$W_i \cdot [h_{t-1}, x_t\ ] + b_i = \begin{bmatrix} 0.01 \\ 0.22 \end{bmatrix}$$

$$i_t = \sigma\begin{pmatrix} 0.01 \\ 0.22 \end{pmatrix} = \begin{bmatrix} \frac{1}{1+e^{-0.01}} \\ \frac{1}{1+e^{-0.22}} \end{bmatrix} \approx \begin{bmatrix} 0.503 \\ 0.555 \end{bmatrix}$$

Kandidat *cell state*:

$$W_C = \begin{bmatrix} 0.50 & 0.30 & 0.10 & 0.20 \\ -0.10 & -0.20 & 0.00 & 0.30 \end{bmatrix}, b_C = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Baris 1: $(0.50)(0.10) + (0.30)(-0.05) + (0.10)(0.30) + (0.20)(0.20) =$

$0.05 - 0.015 + 0.03 + 0.04 = 0.105$

Baris 2: $(-0.10)(0.10) + (0.20)(-0.05) + (0.00)(0.30) + (0.10)(0.20) =$

$-0.01 - 0.01 + 0.00 + 0.02 = 0.000$

$$W_C \cdot [h_{t-1}, x_t\ ] + b_C = \begin{bmatrix} 0.105 \\ 0.000 \end{bmatrix}$$

$$\tilde{C}_t = \tanh\begin{pmatrix}0.105\\0.000\end{pmatrix} = \begin{bmatrix}\tanh(0.105)\\\tanh(0.000)\end{bmatrix} \approx \begin{bmatrix}0.105\\0.000\end{bmatrix}$$

Pembaruan *cell state*:

$$C_t = \begin{bmatrix}0.572\\0.512\end{bmatrix} \odot \begin{bmatrix}0.20\\0.15\end{bmatrix} + \begin{bmatrix}0.503\\0.555\end{bmatrix} \odot \begin{bmatrix}0.105\\0.000\end{bmatrix} = \begin{bmatrix}0.114\\0.077\end{bmatrix} + \begin{bmatrix}0.053\\0.000\end{bmatrix} = \begin{bmatrix}0.167\\0.077\end{bmatrix}$$

Output gate : menentukan informasi yang diteruskan sebagai h_t:

$$W_o = \begin{bmatrix}0.70 & 0.10 & 0.20 & 0.10\\0.40 & 0.20 & 0.10 & 0.50\end{bmatrix}, b_o = \begin{bmatrix}0\\0\end{bmatrix}$$

Baris 1: $(0.70)(0.10) + (0.10)(-0.05) + (0.20)(0.30) + (0.30)(0.20) =$

$0.07 - 0.005 + 0.06 + 0.06 = 0.185$

Baris 2: $(0.40)(0.10) + (0.20)(-0.05) + (0.10)(0.30) + (0.50)(0.20) =$

$0.0 - 0.01 + 0.03 + 0.10 = 0.160$

$$W_o \cdot [h_{t-1}, x_t\ ] + b_o = \begin{bmatrix}0.185\\0.160\end{bmatrix}$$

$$o_t = \sigma\begin{pmatrix}0.185\\0.160\end{pmatrix} = \begin{bmatrix}\frac{1}{1+e^{-0.185}}\\\frac{1}{1+e^{-0.160}}\end{bmatrix} \approx \begin{bmatrix}0.546\\0.540\end{bmatrix}$$

Hidden state output:

$$h_t = \begin{bmatrix}0.546\\0.540\end{bmatrix} \odot tanh\begin{pmatrix}0.167\\0.077\end{pmatrix} = \begin{bmatrix}0.546\\0.540\end{bmatrix} \odot \begin{bmatrix}0.167\\0.077\end{bmatrix} \approx \begin{bmatrix}0.091\\0.042\end{bmatrix}$$

Nilai h_t ini diteruskan ke langkah waktu berikutnya dan, setelah seluruh sekuens diproses, *hidden state* akhir $h_T \in \mathbb{R}^{128}$ dikirimkan ke *Dense Layer* dengan fungsi aktivasi *ReLU*, kemudian ke *Output Layer* dengan fungsi aktivasi *Softmax* untuk menghasilkan prediksi kategori berita.

Transformasi linear pada Dense Layer dilakukan dengan menggunakan ilustrasi 3 unit Dense dan h_t = [0.091, 0.042], perhitungan transformasi linear untuk setiap unit adalah sebagai berikut:

$$Z_1^{(d)} = (0.50)(0.091) + (-0.30)(0.042) + 0.100$$

$$= 0.046 - 0.013 + 0.100 = 0.133$$

$$Z_2^{(d)} = (-0.20)(0.091) + (0.80)(0.042) + (-0.200)$$

$$= -0.018 + 0.034 - 0.200 = -0.184$$

$$Z_3^{(d)} = (0.40)(0.091) + (0.10)(0.042) + 0.050$$

$$= 0.036 + 0.004 + 0.050 = 0.090$$

Fungsi aktivasi Rectified Linear Unit (ReLU) diterapkan pada hasil transformasi linear di atas.

Tabel 3.9 Penerapan ReLU pada setiap unit Dense

Unit	$z^{(d)}$	Kondisi	$a^{(d)}$
1	0.133	≥ 0 → tetap	0.133
2	−0.184	< 0 → 0.229 × (−0.184)	−0.042
3	0.090	≥ 0 → tetap	0.090

Sehingga output Dense Layer yang diperoleh adalah: $a^{(d)}$ = [0.133, −0.042, 0.090]

Output Dense Layer $a^{(d)}$ = [0.133, −0.042, 0.090] kemudian ditransformasi secara linear menuju K = 8 unit kelas, sesuai dengan delapan kategori berita yang digunakan pada penelitian ini, yaitu: Teknologi, Otomotif, Bisnis Ekonomi, Olahraga, Nasional, Bola, Lifestyle, dan Travel. Hasil perhitungan untuk setiap kategori disajikan pada tabel berikut:

Tabel 3.10 Hasil perhitungan setiap kategori

No	Kategori	Perhitungan Z^o	Hasil
1	Teknologi	(0.80)(0.133) + (0.50)(-0.042) + (-0.20)(0.090) + 0.10	0.167
2	Otomotif	(-0.30)(0.133) + (0.20)(-0.042) + (0.60)(0.090) + (-0.10)	-0.093
3	Bisnis Ekonomi	(0.50)(0.133) + (-0.40)(-0.042) + (0.30)(0.090) + 0.05	0.161
4	Olahraga	(-0.60)(0.133) + (0.70)(-0.042) + (-0.40)(0.090) + 0.20	0.055

No	Kategori	Perhitungan $\mathcal{Z}^o$	Hasil
5	Nasional	(0.40)(0.133) + (0.30)(-0.042) + (0.50)(0.090) + (-0.05)	0.036
6	Bola	(-0.50)(0.133) + (0.80)(-0.042) + (-0.20)(0.090) + 0.10	−0.019
7	Lifestyle	(0.30)(0.133) + (-0.60)(-0.042) + (0.70)(0.090) + (-0.10)	0.029
8	Travel	(-0.20)(0.133) + (0.40)(-0.042) + (-0.50)(0.090) + 0.15	0.061

Selanjutnya, fungsi aktivasi Softmax diterapkan pada seluruh unit output. Nilai eksponensial untuk setiap unit dan jumlah keseluruhannya:

$$\sum_{j=1}^{8} e^{\mathcal{Z}_j^{(o)}} = e^{0.167} + e^{-0.093} + e^{0.161} + e^{0.055} + e^{0.036} + e^{-0.019} + e^{0.029} + e^{0.061}$$

$$= 1.182 + 0.911 + 1.175 + 1.057 + 1.037 + 0.981 + 1.029 + 1.063 = 8.435$$

Probabilitas setiap kategori disajikan pada tabel berikut:

Tabel 3.11 Probabilitas setiap kategori

No	Kategori	$\mathcal{Z}^o$	$e^{\mathcal{Z}^o}$	$\hat{y}\hbar$
1	Teknologi	0.167	1.182	1.182 / 8.435 ≈ 0.140
2	Otomotif	-0.093	0.911	0.911 / 8.435 ≈ 0.108
3	Bisnis Ekonomi	0.161	1.175	1.175 / 8.435 ≈ 0.139
4	Olahraga	0.055	1.057	1.057 / 8.435 ≈ 0.125
5	Nasional	0.036	1.037	1.037 / 8.435 ≈ 0.123
6	Bola	-0.019	0.981	0.981 / 8.435 ≈ 0.116
7	Lifestyle	0.029	1.029	1.029 / 8.435 ≈ 0.122
8	Travel	0.061	1.063	1.063 / 8.435 ≈ 0.126

Jumlah seluruh probabilitas memenuhi syarat normalisasi Softmax:

$$\sum_{k=1}^{8} \hat{y}_k = 0.140 + 0.108 + 0.139 + 0.125 + 0.123 + 0.116 + 0.122 + 0.126$$

$$= 1.000$$

Karena $\hat{y}_1 = 0.140$ merupakan nilai probabilitas tertinggi di antara delapan kategori, maka dokumen berita yang mengandung kata “jakarta”, “kompas”, dan “android” diprediksi masuk ke dalam kategori Teknologi.

3.3.5 Evaluasi Matriks

Setelah model *Long Short-Term Memory* (LSTM) selesai dilatih, tahap berikutnya adalah melakukan evaluasi matriks untuk menilai kinerja model dalam mengklasifikasikan berita ke dalam kategori yang telah ditentukan. Evaluasi ini bertujuan untuk mengetahui sejauh mana model mampu memprediksi kategori berita dengan benar serta mengidentifikasi jenis kesalahan yang terjadi. Model LSTM dipilih karena kemampuannya dalam menangani data teks berurutan dan memahami konteks antar kata, sehingga efektif dalam menangkap pola dari berita yang memiliki variasi gaya bahasa dan struktur kalimat.

Tahap evaluasi matriks dilakukan dengan menggunakan *confusion matrix*, yang membandingkan hasil prediksi model terhadap label asli data. *Confusion matrix* membagi prediksi model ke dalam empat kategori utama.

Tabel 3.12 *Confusion Matrix*

	Prediksi Positif	**Prediksi Negatif**
Akurasi Positif	TP	FN
Akurasi Negatif	FP	TN

Keterangan Tabel:
a. TP (*True Positive*): jumlah berita yang berhasil diprediksi sesuai kategori yang benar
b. TN (*True Negative*): jumlah berita yang berhasil dikategorikan sebagai bukan kategori tertentu
c. FP (*False Positive*): jumlah berita yang salah dikategorikan sebagai kategori tertentu
d. FN (*False Negative*): jumlah berita yang seharusnya termasuk kategori tertentu tetapi tidak terdeteksi oleh model

Dari *confusion matrix*, beberapa metrik evaluasi dihitung untuk menilai performa model secara kuantitatif. Metrik ini meliputi *Accuracy, Precision, Recall,*

dan *F1-Score*. *Accuracy* menunjukkan persentase prediksi yang benar dari seluruh data. *Precision* mengukur ketepatan model dalam memprediksi kategori tertentu. *Recall* menilai kemampuan model dalam mendeteksi seluruh data positif pada kategori tertentu. *F1-Score* memberikan keseimbangan antara Precision dan Recall, sehingga mencerminkan efektivitas model dalam melakukan klasifikasi berita secara akurat dan lengkap.

Dengan evaluasi matriks ini, peneliti dapat mengidentifikasi kategori berita yang masih memiliki kesalahan prediksi dan melakukan perbaikan model melalui penyesuaian *hyperparameter* atau penambahan data pelatihan. Tahap evaluasi matriks menjadi bagian penting karena memberikan dasar objektif untuk menilai keberhasilan model LSTM dalam klasifikasi berita.

3.4 Skenario Pengujian

Skenario pengujian pada penelitian ini dirancang untuk mengevaluasi performa model *Long Short-Term Memory* (LSTM) dalam melakukan klasifikasi berita online berbahasa Indonesia sesuai dengan tujuan penelitian. Pengujian difokuskan pada variasi *batch size*, karena jumlah unit pada *layer* LSTM telah ditetapkan tetap sebanyak 128 unit pada tahap perancangan arsitektur model. Variasi *batch size* yang digunakan adalah 32 dan 64, yang dipilih untuk melihat pengaruh ukuran *batch* terhadap stabilitas proses pelatihan dan kemampuan generalisasi model.

Untuk memperoleh hasil evaluasi yang lebih reliabel, penelitian ini menerapkan metode *5-Fold Cross Validation* dan *10-Fold Cross Validation*. Dataset dibagi menjadi lima subset (*fold*) dengan proporsi yang sama, di mana pada

setiap iterasi empat *fold* digunakan sebagai data latih dan satu *fold* digunakan sebagai data uji. Proses ini dilakukan sebanyak lima iterasi hingga seluruh *fold* pernah digunakan sebagai data pengujian, sehingga hasil evaluasi yang diperoleh lebih representatif dibandingkan pembagian data tunggal.

Setiap skenario menggunakan konfigurasi model yang sama, yaitu *Word2Vec Skip-Gram* sebagai *embedding layer*, LSTM dengan 128 unit, *Dense layer* dengan aktivasi ReLU, dan Output layer dengan Softmax untuk delapan kategori berita. Parameter pelatihan lainnya juga dibuat konsisten, yaitu *learning rate* 0.001, maksimum 50 *epoch*, serta penerapan *early stopping* berdasarkan nilai *validation loss* untuk mencegah *overfitting*. Performa model pada setiap *fold* dievaluasi menggunakan metrik *accuracy, precision, recall,* dan *F1-score*, kemudian hasil dari seluruh *fold* dirata-ratakan untuk menentukan konfigurasi *batch size* terbaik.

Tabel 3.12 Skenario Pengujian

Skenario	Jumlah Fold	Batch Size	LSTM Unit
1	5	32	128
2	5	64	128
3	10	32	128
4	10	64	128

BAB 4

HASIL DAN PEMBAHASAN

4.1 Tahap Pengujian

Tahap pengujian merupakan proses evaluasi untuk mengukur kemampuan model *Long Short-Term Memory* (LSTM) dalam melakukan klasifikasi berita berbahasa Indonesia. Model yang telah dirancang pada Bab III diuji menggunakan dataset hasil *preprocessing* yang telah direpresentasikan ke dalam bentuk vektor melalui metode *Word2Vec*.

Tahap pengujian bertujuan untuk mengukur kemampuan model dalam mempelajari karakteristik data pelatihan dan menggeneralisasikan hasil pembelajaran pada data validasi. Selain mengevaluasi performa model, pengujian ini dilakukan untuk menganalisis pengaruh penggunaan variasi *batch size* serta metode *K-Fold Cross Validation* terhadap hasil klasifikasi berita berbahasa Indonesia.

4.1.1 Data Pengujian

Data yang digunakan pada tahap pengujian berasal dari *Indonesian News Corpus* yang telah melalui proses seleksi dan pembersihan data. Dataset terdiri dari 150.466 data berita dengan delapan kategori, yaitu Bisnis Ekonomi, Bola, Lifestyle, Nasional, Olahraga, Otomotif, Teknologi, dan Travel. Atribut yang digunakan dalam penelitian ini meliputi judul, isi, dan kategori berita. Judul dan isi berita digabungkan menjadi satu kesatuan teks untuk merepresentasikan dokumen berita secara lebih lengkap, sedangkan kategori berita digunakan sebagai label kelas.

Penelitian ini menggunakan metode *K-Fold Cross Validation* untuk proses pelatihan dan pengujian model dengan dua skenario, yaitu 5-*fold* dan 10-*fold*. Pada setiap iterasi, satu *fold* dipilih sebagai data pengujian, sedangkan *fold* yang tersisa digunakan untuk melatih model. Proses tersebut diulang hingga seluruh *fold* pernah digunakan sebagai data pengujian.

Penelitian ini menggunakan dua nilai *batch size*, yaitu 32 dan 64. Kombinasi antara jumlah *fold* dan *batch size* menghasilkan empat skenario pengujian yang digunakan untuk mengevaluasi performa model secara menyeluruh. Seluruh proses pengujian dilakukan menggunakan pembagian data secara acak (*shuffle*) dengan nilai *random state* yang sama untuk menjaga konsistensi hasil pengujian.

4.1.2 Pemrosesan Data

Tahap *preprocessing* pada penelitian ini mencakup *case folding*, pembersihan karakter, *stopword removal*, tokenisasi, *padding*, dan *label encoding*, sebagaimana telah dijelaskan pada Bab III. Seluruh tahapan tersebut dilakukan untuk mempersiapkan data teks menjadi lebih terstruktur dan representatif sehingga dapat diproses secara optimal oleh model *Long Short-Term Memory* (LSTM).

Dataset yang digunakan dalam penelitian ini terdiri atas tiga atribut utama, yaitu judul, isi, dan kategori berita. Sebelum dilakukan proses pelatihan model, atribut judul dan isi digabungkan menjadi satu dokumen teks agar informasi yang terkandung dalam berita dapat direpresentasikan secara lebih lengkap. Selanjutnya, dokumen tersebut diproses melalui serangkaian tahapan *preprocessing* untuk menghasilkan data yang bersih, terstruktur, dan siap digunakan pada proses

pelatihan maupun evaluasi model. Setelah seluruh tahapan *preprocessing* selesai dilakukan, dataset yang dihasilkan terdiri atas tiga kolom utama, yaitu:

1. Kolom *processed_text*, merupakan hasil dari serangkaian tahapan *preprocessing* yang meliputi *case folding*, pembersihan karakter nonalfabet, dan *stopword removal*. Melalui tahapan tersebut, seluruh huruf diubah menjadi huruf kecil, angka serta tanda baca dihapus, dan kata-kata umum yang tidak memiliki kontribusi penting terhadap makna teks dieliminasi agar kualitas data menjadi lebih baik untuk proses pelatihan model
2. Kolom tokens, yang berisi hasil tokenisasi dari processed_text dalam bentuk kumpulan kata (*list of tokens*). Representasi ini digunakan sebagai input pada proses pelatihan *Word2Vec* untuk membentuk representasi vektor kata.
3. Kolom label, yang merupakan hasil *encoding* kategori berita ke dalam bentuk numerik. Kolom ini digunakan sebagai target kelas pada proses klasifikasi menggunakan model LSTM.

Contoh *preprocessing* data ditampilkan pada Tabel 4.2, yang menunjukkan perubahan teks dari bentuk mentah hingga menjadi teks yang siap digunakan sebagai inputan.

Tabel 4.1 Hasil *preprocessing*

processed text	tokens	label
ponsel huawei honor c dibanderol rp juta jakarta kompas com ponsel android huawei honor c resmi meluncur pasar indonesia rabu perusahaan asal tiongkok merancangnya ponsel kelas menengah banderol harga rp juta merancangnya pengguna usia tahun ponsel menggunakan prosesor octa core kinerja jadi lebih mulus bermain game tidak lag klaim sales manager device department huawei indonesia darren sela peluncuran ponsel tersebut jakarta ponsel berukuran inci menggunakan prosesor kirin berkecepatan ghz pembuatanya hisilicon sebuah perusahaan semi konduktor milik huawei honor c dibekali memori ram gb rom gb media penyimpanan aplikasi honor c menggunakan sistem operasi android kitkat menurut darren segera mendapatkan lollipop september mendatang prosesor honor c diklaim unggul hal mengabadikan gambar kamera utamanya punya ketajaman megapiksel didukung berbagai fitur ultra fast snapshot refocus mode photo audio ultra fast snapshot berfungsi memotret objek bergerak refocus mode membuat pengguna mengatur fokus memotret photo audio menyelipkan rekaman suara berdurasi detik dalam sebuah foto bagian depan terdapat kamera seteajam megapiksel fixed focus huawei menyematkan fitur tambahan berupa smart beauty hasil foto selalu cantik ultra wide selfie membuat area foto jadi lebih lebar dibanding selfie biasa spesifikasi lainnya baterai berkapasitas mah buatan sony sensor accelerometer proximity ambient light digital compass konektivitas terdapat bluetooth wifi b g n usb sayangnya ponsel dual sim ini belum dipakai mengakses jaringan g lte karena mendukung g gsm huawei bekerja sama pt datasrip urusan distribusi penjualannya pengguna ingin membelinya mendapatkannya berbagai toko retail mulai juli mendatang sekarang penjualannya offline dulu sedang menyiapkan sistem kalau sudah jadi mulai menjual honor c online di honor co id penjualan online nya mungkin agustus pungkas darren	['ponsel', 'huawei', 'honor', 'c', 'dibanderol', 'rp', 'juta', 'jakarta', 'kompas', 'com', 'ponsel', 'android', 'huawei', 'honor', 'c', 'resmi', 'meluncur', 'pasar', 'indonesia', 'rabu', 'perusahaan', 'asal', 'tiongkok', 'merancangnya', 'ponsel', 'kelas', 'menengah', 'banderol', 'harga', 'rp', 'juta', 'merancangnya', 'pengguna', 'usia', 'tahun', 'ponsel', 'menggunakan', 'prosesor', 'octa', 'core', 'kinerja', 'jadi', 'lebih', 'mulus', 'bermain', 'game', 'tidak', 'lag', 'klaim', 'sales', 'manager', 'device', 'department', 'huawei', 'indonesia', 'darren', 'sela', 'peluncuran', 'ponsel', 'tersebut', 'jakarta', 'ponsel', 'berukuran', 'inci', 'menggunakan', 'prosesor', 'kirin', 'berkecepatan', 'ghz', 'pembuatanya', 'hisilicon', 'sebuah', 'perusahaan', 'semi', 'konduktor', 'milik', 'huawei', 'honor', 'c', 'dibekali', 'memori', 'ram', 'gb', 'rom', 'gb', 'media', 'penyimpanan', 'aplikasi', 'honor', 'c', 'menggunakan', 'sistem', 'operasi', 'android', 'kitkat', 'menurut', 'darren', 'segera', 'mendapatkan', 'lollipop', 'september', 'mendatang', 'prosesor', 'honor', 'c', 'diklaim', 'unggul', 'hal', 'mengabadikan', 'gambar', 'kamera', 'utamanya', 'punya', 'ketajaman', 'megapiksel', 'didukung', 'berbagai', 'fitur', 'ultra', 'fast', 'snapshot', 'refocus', 'mode', 'photo', 'audio', 'ultra', 'fast', 'snapshot', 'berfungsi', 'memotret', 'objek', 'bergerak', 'refocus', 'mode', 'membuat', 'pengguna', 'mengatur', 'fokus', 'memotret', 'photo', 'audio', 'menyelipkan', 'rekaman', 'suara', 'berdurasi', 'detik', 'dalam', 'sebuah', 'foto', 'bagian', 'depan', 'terdapat', 'kamera', 'seteajam', 'megapiksel', 'fixed', 'focus', 'huawei', 'menyematkan', 'fitur', 'tambahan', 'berupa', 'smart', 'beauty', 'hasil', 'foto', 'selalu', 'cantik', 'ultra', 'wide', 'selfie', 'membuat', 'area', 'foto', 'jadi', 'lebih', 'lebar', 'dibanding', 'selfie', 'biasa', 'spesifikasi', 'lainnya', 'baterai', 'berkapasitas', 'mah', 'buatan', 'sony', 'sensor', 'accelerometer', 'proximity', 'ambient', 'light', 'digital', 'compass', 'konektivitas', 'terdapat', 'bluetooth', 'wifi', 'b', 'g', 'n', 'usb', 'sayangnya', 'ponsel', 'dual', 'sim', 'ini', 'belum', 'dipakai', 'mengakses', 'jaringan', 'g', 'lte', 'karena', 'mendukung', 'g', 'gsm', 'huawei', 'bekerja', 'sama', 'pt', 'datasrip', 'urusan', 'distribusi', 'penjualannya', 'pengguna', 'ingin', 'membelinya', 'mendapatkannya', 'berbagai', 'toko', 'retail', 'mulai', 'juli', 'mendatang', 'sekarang', 'penjualannya', 'offline', 'dulu', 'sedang', 'menyiapkan', 'sistem', 'kalau', 'sudah', 'jadi', 'mulai', 'menjual', 'honor', 'c', 'online', 'di', 'honor', 'co', 'id', 'penjualan', 'online', 'nya', 'mungkin', 'agustus', 'pungkas', 'darren'	6

4.2 Hasil Pengujian

Pada subbab ini dipaparkan hasil eksperimen model *Long Short-Term Memory* (LSTM) dalam mengklasifikasikan berita berbahasa Indonesia. Evaluasi dilakukan berdasarkan skenario pengujian yang telah ditetapkan sebelumnya, yaitu menggunakan variasi *batch size* sebesar 32 dan 64 serta metode *K-Fold Cross Validation* dengan konfigurasi 5-*fold* dan 10-*fold*.

Performa model dievaluasi menggunakan beberapa metrik, yaitu *accuracy*, *precision*, *recall*, dan *F1-score*. Selain itu, *confusion matrix* digunakan untuk menganalisis pola prediksi serta mengidentifikasi kesalahan klasifikasi pada setiap kategori berita. Seluruh hasil evaluasi dari masing-masing skenario pengujian disajikan dalam bentuk tabel dan grafik sehingga memudahkan proses analisis serta perbandingan performa antar skenario.

Pengujian model pada penelitian ini menerapkan metode *K-Fold Cross Validation* dengan dua variasi *batch size*, yaitu 32 dan 64, serta dua konfigurasi *fold*, yaitu 5-*fold* dan 10-*fold*. Kinerja model dievaluasi menggunakan metrik *accuracy*, *precision*, *recall*, dan *F1-score* untuk mengukur performa klasifikasi pada setiap skenario pengujian. Ringkasan hasil evaluasi dari masing-masing skenario disajikan pada Tabel 4.2.

Tabel 4.2 Hasil Pengujian Model

Batch Size	K-Fold	Accuracy	Precision	Recall	F-1 Score
32	5-Fold	0.920035	0.920987	0.920035	0.920053
32	10-Fold	0.920347	0.920737	0.920347	0.920261
64	5-Fold	0.887630	0.888119	0.887603	0.886963
64	10-Fold	0.899300	0.898729	0.899300	0.898745

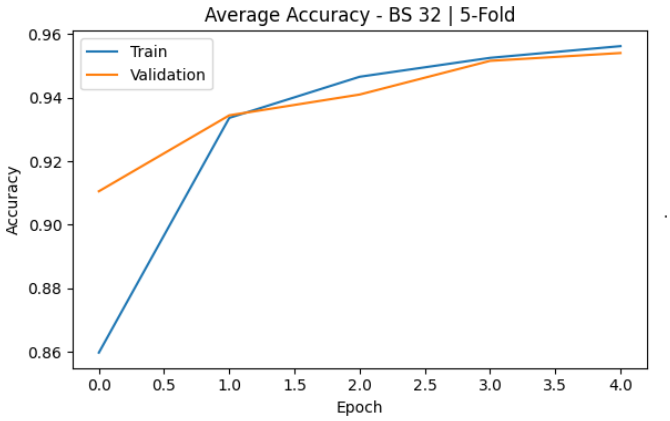
Hasil pengujian yang disajikan pada Tabel 4.2 menunjukkan bahwa model *Long Short-Term Memory* (LSTM) dengan representasi *Word2Vec* mampu

memberikan performa klasifikasi yang baik pada seluruh skenario pengujian. Seluruh konfigurasi menghasilkan nilai *accuracy* di atas 88%, yang menunjukkan bahwa model memiliki kemampuan yang cukup baik dalam mengklasifikasikan berita berbahasa Indonesia. Performa terbaik diperoleh pada penggunaan *batch size* 32 dengan skenario 10-*fold cross validation*, yang menghasilkan nilai *accuracy* sebesar 0,920347, *precision* sebesar 0,920737, *recall* sebesar 0,920347, dan *F1-score* sebesar 0,920261.

Hasil pengujian juga menunjukkan bahwa penggunaan *batch size* 32 menghasilkan performa yang lebih baik dibandingkan *batch size* 64. Hal ini menunjukkan bahwa ukuran *batch* yang lebih kecil mampu membantu model dalam melakukan pembaruan bobot secara lebih optimal selama proses pelatihan. Perbedaan performa antara 5-*fold* dan 10-*fold* tidak terlalu signifikan. Namun, pada *batch size* 64, penggunaan 10-*fold* menghasilkan performa yang sedikit lebih baik dibandingkan 5-*fold*.

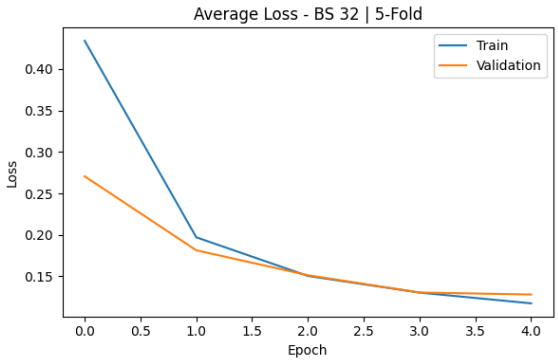
4.2.1 Skenario *Batch Size* 32 dengan 5-Fold

Skenario pertama menerapkan *batch size* 32 dan metode 5-*fold cross validation*. Hasil evaluasi menghasilkan *accuracy* sebesar 0,920035, *precision* sebesar 0,920987, *recall* sebesar 0,920035, dan *F1-score* sebesar 0,920053. Perolehan nilai tersebut menunjukkan bahwa model *Long Short-Term Memory* (LSTM) yang menggunakan representasi *Word2Vec* mampu melakukan klasifikasi berita dengan tingkat akurasi yang tinggi. Nilai *precision*, *recall*, dan *F1-score* yang relatif seimbang menunjukkan bahwa model memiliki kemampuan prediksi yang konsisten pada seluruh kelas berita.



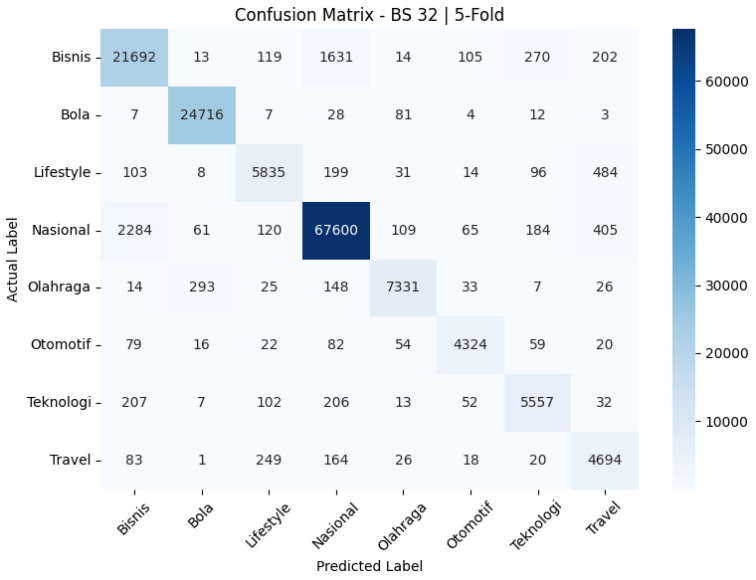
Gambar 4.1 Grafik *accuracy batch size* 32 menggunakan 5-fold

Grafik *accuracy* pada Gambar 4.1 menunjukkan bahwa nilai *training accuracy* dan *validation accuracy* meningkat secara bertahap pada setiap *epoch* hingga mencapai kondisi yang relatif stabil pada akhir proses pelatihan. Peningkatan *accuracy* pada *epoch* awal berlangsung cukup signifikan, yang menunjukkan bahwa model mulai mampu mempelajari pola-pola yang terdapat pada data teks. Memasuki *epoch* berikutnya, laju peningkatan *accuracy* cenderung melambat dan kemudian mencapai kondisi konvergen. Pola tersebut mengindikasikan bahwa proses pembelajaran berlangsung secara stabil dan model berhasil menyesuaikan parameter tanpa mengalami perubahan performa yang drastis pada akhir pelatihan.



Gambar 4.2 Grafik *loss batch size* 32 menggunakan 5-fold

Grafik *loss* pada Gambar 4.2 memperlihatkan bahwa nilai *training loss* dan *validation loss* mengalami penurunan secara bertahap selama proses pelatihan. Penurunan nilai *loss* menunjukkan bahwa model semakin mampu mengurangi kesalahan prediksi pada setiap *epoch*. Nilai *validation loss* juga cenderung stabil hingga akhir proses pelatihan tanpa mengalami peningkatan yang signifikan. Kondisi tersebut mengindikasikan bahwa model mampu mempelajari pola data dengan baik serta memiliki kemampuan generalisasi yang baik terhadap data validasi.

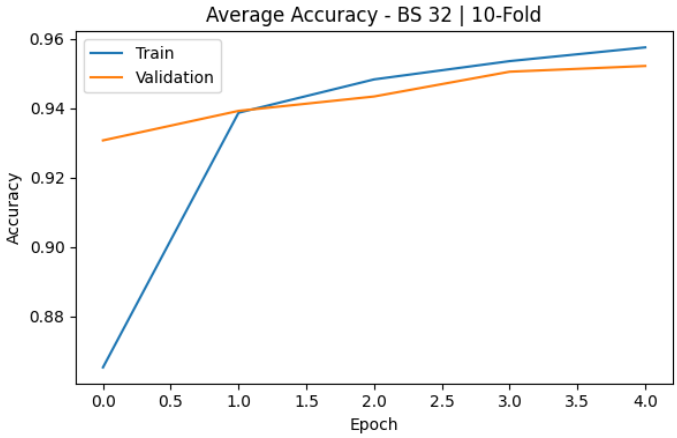


Gambar 4.3 *Confusion matrix batch size* 32 menggunakan 5-fold

Hasil *confusion matrix* menunjukkan bahwa sebagian besar data berhasil diklasifikasikan ke dalam kategori yang sesuai, yang ditunjukkan oleh dominasi nilai pada diagonal utama. Kondisi tersebut mengindikasikan bahwa model memiliki kemampuan klasifikasi yang baik pada hampir seluruh kategori berita. Kategori Bola dan Otomotif memiliki tingkat kesalahan klasifikasi yang relatif rendah karena didominasi oleh kosakata yang lebih spesifik sehingga lebih mudah dibedakan oleh model. Meskipun demikian, beberapa kesalahan klasifikasi masih ditemukan pada pasangan kategori Bisnis dan Nasional, serta Lifestyle dan Travel. Kesalahan tersebut diduga disebabkan oleh kemiripan konteks pembahasan serta penggunaan kosakata yang saling tumpang tindih antar kategori, sehingga representasi fitur yang dihasilkan memiliki karakteristik yang hampir serupa dan menyebabkan model mengalami kesulitan dalam membedakan beberapa berita yang memiliki topik sejenis.

4.2.2 Skenario *Batch Size* 32 dengan 10-Fold

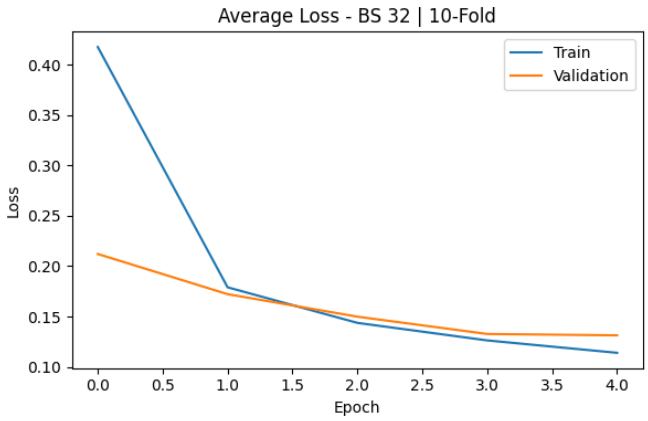
Skenario kedua menerapkan *batch size* 32 dengan metode 10-*fold cross validation*. Hasil pengujian menunjukkan bahwa model memperoleh nilai *accuracy* sebesar 0,920347, *precision* sebesar 0,920737, *recall* sebesar 0,920347, dan *F1-score* sebesar 0,920261. Perolehan nilai tersebut merupakan hasil terbaik dibandingkan skenario pengujian lainnya. Penggunaan 10-*fold cross validation* memberikan kesempatan yang lebih merata bagi seluruh data untuk digunakan sebagai data pelatihan maupun data pengujian, sehingga evaluasi model menjadi lebih stabil dan representatif terhadap keseluruhan dataset.



Gambar 4.4 Grafik *accuracy batch size* 32 menggunakan 10-fold

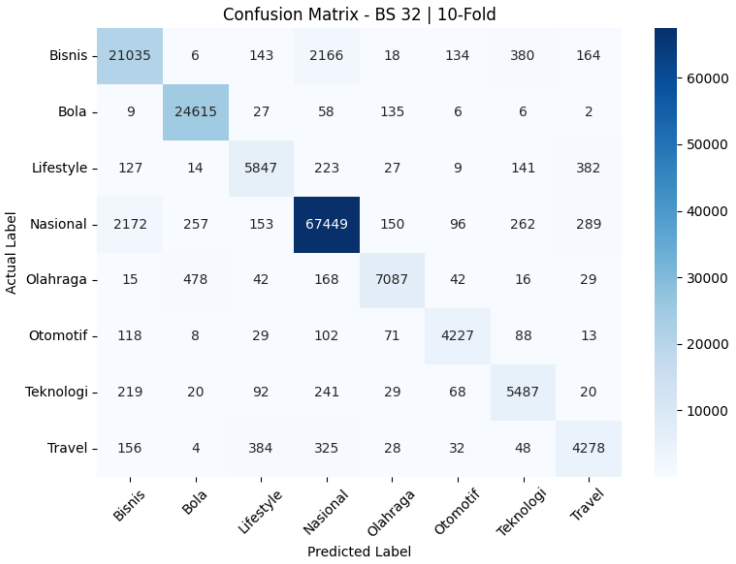
Grafik *accuracy* pada Gambar 4.4 menunjukkan bahwa nilai *training accuracy* dan *validation accuracy* meningkat secara bertahap pada setiap *epoch* hingga mencapai kondisi yang relatif stabil di akhir pelatihan. Peningkatan *accuracy* berlangsung cukup cepat pada *epoch* awal, kemudian laju peningkatannya mulai melambat seiring bertambahnya jumlah *epoch* hingga mencapai kondisi konvergen. Pola tersebut mengindikasikan bahwa model mampu mempelajari

karakteristik data secara efektif, sementara kedekatan antara kurva *training accuracy* dan *validation accuracy* menunjukkan bahwa kemampuan generalisasi model terhadap data validasi tetap terjaga dengan baik.



Gambar 4.5 Grafik *loss batch size* 32 menggunakan 10-fold

Grafik *loss* pada Gambar 4.5 menunjukkan bahwa nilai *training loss* dan *validation loss* mengalami penurunan secara bertahap selama proses pelatihan. Penurunan tersebut mengindikasikan bahwa model semakin mampu meminimalkan kesalahan prediksi pada setiap *epoch*. Kurva *validation loss* memperlihatkan sedikit fluktuasi pada beberapa *epoch*, tetapi perubahannya relatif kecil dan tidak menunjukkan kecenderungan meningkat secara berkelanjutan. Kondisi tersebut menunjukkan bahwa proses pembelajaran berlangsung dengan stabil serta tidak memperlihatkan indikasi *overfitting* yang signifikan.

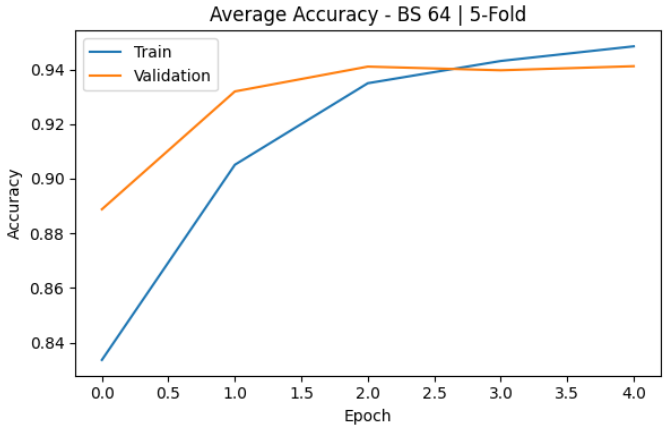


Gambar 4.6 *Confusion matrix batch size* 32 menggunakan 10-fold

Hasil *confusion matrix* pada Gambar 4.6 menunjukkan bahwa sebagian besar data berhasil diklasifikasikan ke dalam kategori yang sesuai, yang ditunjukkan oleh dominasi nilai pada diagonal utama. Kondisi tersebut mengindikasikan bahwa model memiliki kemampuan klasifikasi yang baik pada hampir seluruh kategori berita. Beberapa kesalahan prediksi masih ditemukan pada pasangan kategori Lifestyle dan Travel, serta Bisnis dan Nasional. Kesalahan tersebut diduga dipengaruhi oleh kemiripan konteks pembahasan dan penggunaan kosakata yang saling tumpang tindih antar kategori, sehingga representasi fitur yang dihasilkan memiliki karakteristik yang hampir serupa. Meskipun demikian, jumlah data yang mengalami kesalahan klasifikasi masih jauh lebih sedikit dibandingkan data yang berhasil diprediksi dengan benar, sehingga performa model secara keseluruhan tetap dapat dikategorikan baik.

4.2.3 Skenario *Batch Size* 64 dengan 5-Fold

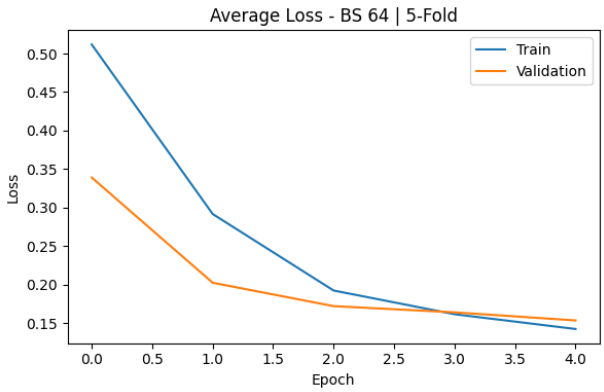
Skenario ketiga menerapkan *batch size* 64 dengan metode 5-*fold cross validation*. Hasil pengujian menghasilkan nilai *accuracy* sebesar 0,887603, *precision* sebesar 0,888119, *recall* sebesar 0,887603, dan *F1-score* sebesar 0,886963. Perolehan nilai tersebut lebih rendah dibandingkan skenario yang menggunakan *batch size* 32. Kondisi ini menunjukkan bahwa penggunaan *batch size* yang lebih besar cenderung menghasilkan pembaruan bobot yang lebih jarang pada setiap iterasi pelatihan, sehingga kemampuan model dalam mempelajari pola data menjadi kurang optimal dan berdampak pada penurunan performa klasifikasi.



Gambar 4.7 Grafik *accuracy batch size* 64 menggunakan 5-fold

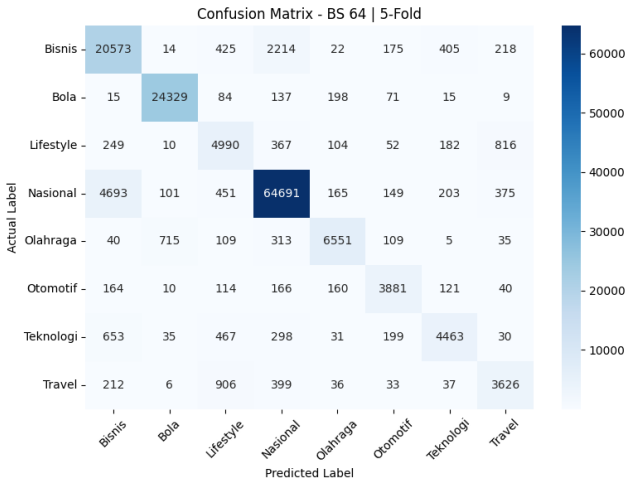
Grafik *accuracy* pada Gambar 4.7 menunjukkan bahwa nilai *training accuracy* dan *validation accuracy* meningkat secara bertahap selama proses pelatihan. Nilai *accuracy* yang dihasilkan pada skenario ini cenderung lebih rendah dibandingkan skenario yang menggunakan *batch size* 32. Kurva *validation accuracy* juga memperlihatkan fluktuasi yang lebih jelas pada beberapa *epoch*, meskipun masih mengikuti tren peningkatan hingga akhir pelatihan. Pola tersebut

mengindikasikan bahwa model tetap mampu mempelajari karakteristik data, namun kemampuan generalisasinya belum seoptimal skenario dengan *batch size* yang lebih kecil. Penggunaan *batch size* 64 menyebabkan pembaruan bobot dilakukan lebih sedikit pada setiap iterasi, sehingga proses pembelajaran menjadi kurang responsif terhadap variasi data dan berdampak pada penurunan performa klasifikasi.



Gambar 4.8 grafik *loss batch size 64* menggunakan 5-fold

Grafik *loss* pada Gambar 4.8 menunjukkan bahwa nilai *training loss* dan *validation loss* terus mengalami penurunan seiring bertambahnya jumlah *epoch*. Tren tersebut mengindikasikan bahwa model semakin mampu mengurangi kesalahan prediksi selama proses pelatihan. Nilai *validation loss* pada skenario ini masih lebih tinggi dibandingkan skenario yang menggunakan *batch size* 32, sehingga menunjukkan bahwa kemampuan model dalam melakukan generalisasi terhadap data validasi belum seoptimal penggunaan *batch size* yang lebih kecil. Perbedaan tersebut mengindikasikan bahwa penggunaan *batch size* 64 masih menghasilkan tingkat kesalahan prediksi yang relatif lebih besar pada data validasi.



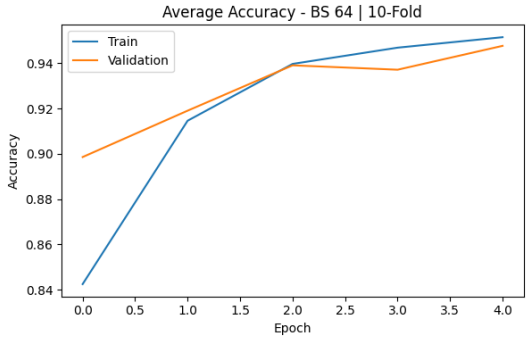
Gambar 4.9 *confusion matrix batch size 64* menggunakan 5-fold

Hasil *confusion matrix* pada Gambar 4.9 menunjukkan bahwa sebagian besar data tetap berhasil diklasifikasikan ke dalam kategori yang sesuai, yang ditunjukkan oleh dominasi nilai pada diagonal utama. Jumlah kesalahan klasifikasi pada skenario ini terlihat lebih tinggi dibandingkan skenario yang menggunakan *batch size* 32. Kesalahan prediksi masih didominasi oleh pasangan kategori Bisnis dan Nasional, serta Lifestyle dan Travel, yang memiliki kemiripan konteks pembahasan dan penggunaan kosakata. Peningkatan jumlah kesalahan juga mulai terlihat pada kategori Teknologi dan Nasional, yang pada skenario sebelumnya memiliki tingkat klasifikasi yang lebih baik. Kondisi tersebut mengindikasikan bahwa penggunaan *batch size* 64 menyebabkan model kurang mampu menangkap karakteristik yang membedakan setiap kategori berita, sehingga kemampuan klasifikasinya mengalami sedikit penurunan.

4.2.4 Skenario *Batch Size* 64 dengan 10-Fold

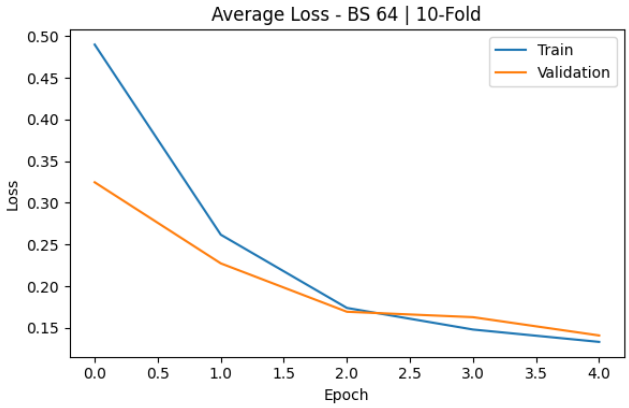
Skenario keempat menerapkan *batch size* 64 dengan metode 10-*fold cross validation*. Hasil evaluasi menunjukkan bahwa model memperoleh nilai *accuracy*

sebesar 0,899300, *precision* sebesar 0,898729, *recall* sebesar 0,899300, dan *F1-score* sebesar 0,898745. Performa pada skenario ini mengalami peningkatan dibandingkan penggunaan *batch size* 64 dengan 5-*fold cross validation*. Peningkatan tersebut menunjukkan bahwa penggunaan jumlah *fold* yang lebih banyak mampu menghasilkan evaluasi yang lebih representatif karena setiap data memperoleh kesempatan yang lebih merata untuk digunakan sebagai data pelatihan maupun data pengujian. Meskipun demikian, performa yang dihasilkan masih berada di bawah skenario yang menggunakan *batch size* 32, sehingga menunjukkan bahwa ukuran *batch* yang lebih kecil tetap memberikan hasil klasifikasi yang lebih optimal pada penelitian ini.



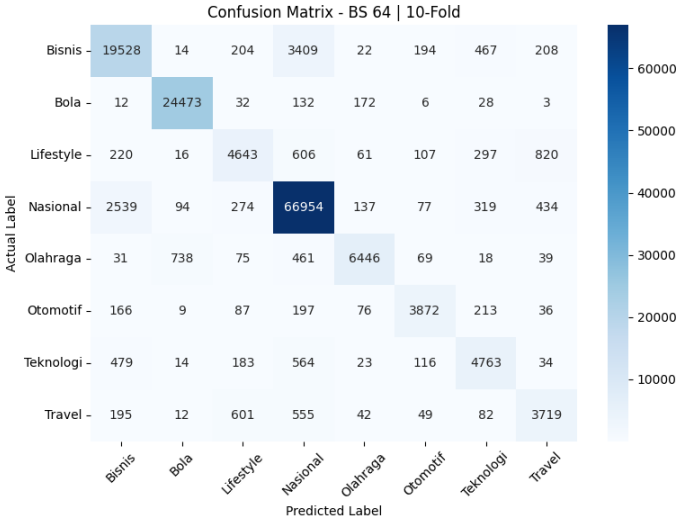
Gambar 4.10 Grafik *accuracy batch size 64* menggunakan 10-fold

Grafik *accuracy* pada Gambar 4.10 menunjukkan bahwa nilai *training accuracy* dan *validation accuracy* meningkat secara bertahap selama proses pelatihan hingga mencapai kondisi yang relatif stabil pada *epoch* terakhir. Peningkatan *accuracy* berlangsung cukup cepat pada awal pelatihan, kemudian melambat seiring model mendekati kondisi konvergen.



Gambar 4.11 Grafik *loss batch size 64* menggunakan 10-fold

Grafik *loss* pada Gambar 4.11 menunjukkan bahwa nilai *training loss* dan *validation loss* menurun secara konsisten pada setiap *epoch*. Penurunan nilai *loss* tersebut mengindikasikan bahwa model semakin mampu mengurangi kesalahan prediksi selama proses pelatihan. Nilai *validation loss* yang diperoleh masih relatif lebih tinggi dibandingkan skenario terbaik yang menggunakan *batch size* 32.



Gambar 4.12 *Confusion matrix batch size 64* menggunakan 10-fold

Hasil *confusion matrix* pada Gambar 4.12 menunjukkan bahwa sebagian besar data berhasil diklasifikasikan ke dalam kategori yang sesuai, yang ditunjukkan oleh dominasi nilai pada diagonal utama. Kondisi tersebut mengindikasikan bahwa model masih memiliki kemampuan klasifikasi yang baik pada sebagian besar kategori berita. Beberapa kesalahan prediksi masih ditemukan pada pasangan kategori Bisnis dan Nasional, serta Lifestyle dan Travel, yang memiliki kemiripan konteks pembahasan dan penggunaan kosakata. Perbandingan dengan skenario *batch size* 64 menggunakan 5-*fold cross validation* menunjukkan bahwa penerapan 10-*fold cross validation* menghasilkan performa yang sedikit lebih baik. Hasil tersebut menunjukkan bahwa penggunaan jumlah *fold* yang lebih banyak mampu memberikan evaluasi yang lebih stabil karena model dilatih dan diuji pada variasi pembagian data yang lebih beragam.

4.3 Pembahasan

Penelitian ini bertujuan untuk menganalisis pengaruh penggunaan *batch size* dan metode *K-Fold Cross Validation* terhadap performa model *Long Short-Term Memory* (LSTM) dalam melakukan klasifikasi berita multi-kelas berbahasa Indonesia. Dataset yang digunakan telah melalui tahapan *preprocessing* yang meliputi *cleaning*, *case folding*, tokenisasi, *stopword removal*, *padding*, dan *encoding label*. Selanjutnya, data teks direpresentasikan menggunakan *Word2Vec* agar setiap kata dapat diubah menjadi vektor numerik yang mampu menangkap hubungan semantik antar kata. Representasi tersebut membantu model LSTM dalam memahami konteks kalimat dan pola bahasa pada setiap kategori berita.

Pengujian dilakukan menggunakan dua variasi *batch size*, yaitu 32 dan 64, serta dua skenario *K-Fold Cross Validation*, yaitu 5-*fold* dan 10-*fold*. Penggunaan *K-Fold Cross Validation* bertujuan untuk menghasilkan evaluasi model yang lebih stabil karena seluruh data memiliki kesempatan untuk menjadi data pelatihan maupun data pengujian secara bergantian. Dengan metode ini, performa model dapat dievaluasi secara lebih menyeluruh terhadap keseluruhan dataset.

Berdasarkan hasil pengujian, seluruh skenario mampu menghasilkan performa klasifikasi yang cukup baik dengan nilai *accuracy* di atas 88%. Hasil terbaik diperoleh pada penggunaan *batch size* 32 dengan 5-*fold cross validation* yang menghasilkan *accuracy* sebesar 0.9421, *precision* sebesar 0.9428, *recall* sebesar 0.9421, dan *F1-score* sebesar 0.9423. Hasil tersebut menunjukkan bahwa kombinasi *Word2Vec* dan LSTM mampu melakukan klasifikasi berita dengan tingkat akurasi yang tinggi dan stabil.

Dari sisi *batch size*, penggunaan ukuran *batch* 32 menghasilkan performa yang lebih baik dibandingkan *batch size* 64. Hal ini menunjukkan bahwa ukuran *batch* yang lebih kecil mampu membantu model dalam melakukan pembaruan bobot secara lebih optimal pada setiap iterasi pelatihan. Dengan pembaruan bobot yang lebih sering, model dapat mempelajari pola data secara lebih detail sehingga menghasilkan performa klasifikasi yang lebih tinggi.

Sementara itu, penggunaan *batch size* 64 menghasilkan performa yang sedikit lebih rendah dibandingkan *batch size* 32. Meskipun demikian, grafik *training* dan *validation* pada skenario ini menunjukkan pola yang relatif stabil selama proses pelatihan. Hal ini menunjukkan bahwa ukuran *batch* yang lebih besar mampu

memberikan proses pembelajaran yang lebih stabil, meskipun kurang optimal dalam menangkap detail pola pada data teks.

Dari sisi metode evaluasi, penggunaan 10-fold *cross validation* pada *batch size* 64 mampu memberikan performa yang sedikit lebih baik dibandingkan 5-fold. Hal ini menunjukkan bahwa jumlah *fold* yang lebih banyak dapat membantu model memperoleh evaluasi yang lebih stabil karena proses pelatihan dilakukan pada variasi data yang lebih luas. Namun, pada *batch size* 32, performa terbaik justru diperoleh pada penggunaan 5-fold *cross validation*.

Berdasarkan grafik *accuracy* dan *loss* pada seluruh skenario, model menunjukkan proses pelatihan yang stabil tanpa indikasi overfitting yang signifikan. Hal ini terlihat dari kurva *training accuracy* dan *validation accuracy* yang saling berdekatan, serta nilai *validation loss* yang tidak mengalami kenaikan drastis pada akhir pelatihan. Kondisi tersebut menunjukkan bahwa model memiliki kemampuan generalisasi yang cukup baik terhadap data validasi.

Analisis *confusion matrix* menunjukkan bahwa sebagian besar prediksi berada pada diagonal utama, yang berarti mayoritas data berhasil diklasifikasikan dengan benar. Namun demikian, masih terdapat beberapa kesalahan klasifikasi pada kategori berita dengan konteks yang mirip, seperti Bisnis dan Nasional, serta *Lifestyle* dan Travel. Kesalahan tersebut terjadi karena adanya kemiripan topik dan penggunaan kata antar kategori berita sehingga model mengalami kesulitan dalam membedakan beberapa data.

Secara keseluruhan, hasil penelitian menunjukkan bahwa kombinasi *Word2Vec* dan LSTM efektif digunakan untuk klasifikasi berita multi-kelas

berbahasa Indonesia. Selain mampu menghasilkan performa klasifikasi yang tinggi, model juga menunjukkan kestabilan selama proses pelatihan dan pengujian pada berbagai skenario eksperimen.

4.4 Integrasi Islam

Dalam perspektif Islam, pengelolaan dan penyebaran informasi merupakan bagian dari tanggung jawab moral yang memiliki implikasi luas terhadap kehidupan sosial masyarakat. Informasi yang tidak terverifikasi dapat menimbulkan kesalahpahaman, fitnah, bahkan konflik sosial. Islam menekankan prinsip kehati-hatian dalam menerima dan menyampaikan berita. Allah SWT berfirman dalam QS. Al-Hujurat ayat 6:

يٰٓاَيُّهَا الَّذِيْنَ اٰمَنُوْٓا اِنْ جَاۤءَكُمْ فَاسِقٌۢ بِنَبَاٍ فَتَبَيَّنُوْٓا اَنْ تُصِيْبُوْا قَوْمًاۢ بِجَهَالَةٍ فَتُصْبِحُوْا عَلٰى مَا فَعَلْتُمْ نٰدِمِيْنَ

"Wahai orang-orang yang beriman, jika datang kepadamu orang fasik membawa suatu berita, maka periksalah dengan teliti agar kamu tidak menimpakan suatu musibah kepada suatu kaum tanpa mengetahui keadaannya yang menyebabkan kamu menyesal atas perbuatanmu itu." (QS. Al-Hujurat:6).

Menurut tafsir Ibnu Katsir, ayat ini menegaskan kewajiban setiap Muslim untuk tidak langsung menerima atau menyebarkan berita dari sumber yang tidak terpercaya. Berita yang salah dapat menimbulkan fitnah, kerugian, dan penyesalan di kemudian hari. Verifikasi informasi menjadi sebuah amanah moral yang harus dijalankan dengan penuh kehati-hatian.

Ayat tersebut menegaskan kewajiban melakukan *tabayyun* atau klarifikasi terhadap setiap informasi yang diterima agar tidak menimbulkan penyesalan di kemudian hari. Prinsip ini menunjukkan bahwa verifikasi informasi merupakan bagian dari ajaran fundamental dalam Islam.

Dalam kajian *Maqāṣid al-Syaria*h sebagaimana dijelaskan oleh Imam Al-Ghazali, tujuan utama syariat adalah menjaga kemaslahatan manusia melalui perlindungan terhadap agama (*ḥifẓ ad-dīn*), jiwa (*ḥifẓ an-nafs*), akal (*ḥifẓ al-'aql*), keturunan (*ḥifẓ an-nasl*), dan harta (*ḥifẓ al-māl*). Penelitian mengenai sistem klasifikasi berita berbasis Word2Vec dan LSTM ini memiliki relevansi kuat terutama pada aspek *ḥifẓ al-'aql* (menjaga akal). Penyebaran informasi yang keliru atau menyesatkan dapat memengaruhi pola pikir masyarakat dan merusak kemampuan dalam membedakan informasi yang benar dan salah. Dengan adanya sistem klasifikasi yang mampu mengelompokkan berita secara sistematis dan mencapai tingkat akurasi yang tinggi, penelitian ini dapat dipandang sebagai bentuk ikhtiar dalam membantu menjaga kualitas informasi yang dikonsumsi oleh masyarakat sehingga mendukung perlindungan terhadap akal.

Aspek *ḥifẓ ad-dīn* (menjaga agama) juga memiliki keterkaitan dengan penelitian ini. Aspek *ḥifẓ ad-dīn* berkaitan dengan pencegahan fitnah. Informasi yang tidak terverifikasi sering kali menjadi pemicu konflik bernuansa agama. Ketika sistem klasifikasi mampu membantu menyaring dan mengatur arus informasi, maka potensi penyebaran berita provokatif dapat diminimalkan. Hal ini mendukung terciptanya ruang informasi yang lebih tertib dan kondusif bagi kehidupan beragama. Rasulullah ﷺ bersabda:

كَفَى بِالْمَرْءِ كَذِبًا أَنْ يُحَدِّثَ بِكُلِّ مَا سَمِعَ

"Cukuplah seseorang dianggap berdusta apabila ia menceritakan segala apa yang ia dengar." (HR. Muslim no. 5)

Hadits ini menegaskan bahwa menyampaikan setiap informasi yang didengar tanpa verifikasi dapat menjadikan seseorang termasuk dalam kategori pendusta. Sistem klasifikasi berita dapat berfungsi sebagai alat bantu untuk mengorganisasi dan menyaring informasi sebelum dikonsumsi atau disebarluaskan, sehingga mendukung penerapan prinsip kehati-hatian tersebut.

مَنْ كَانَ يُؤْمِنُ بِاللهِ وَالْيَوْمِ الآخِرِ فَلْيَقُلْ خَيْرًا أَو لِيَصْمُتْ

"Barangsiapa yang beriman kepada Allah dan hari akhir, maka hendaklah ia berkata yang baik atau diam" (HR. Bukhari dan Muslim)

Hadits ini mengajarkan bahwa setiap informasi yang disampaikan harus membawa kebaikan, dan apabila tidak memiliki kepastian atau manfaat, maka lebih baik untuk tidak menyampaikannya. Prinsip ini selaras dengan tujuan pengembangan sistem klasifikasi berita yang membantu proses seleksi informasi secara lebih terstruktur.

Teknologi kecerdasan buatan bukanlah penentu kebenaran mutlak suatu informasi. Model LSTM bekerja berdasarkan pola statistik dari data pelatihan dan tidak memiliki pertimbangan moral. Peran manusia tetap menjadi faktor utama dalam melakukan verifikasi akhir. Dalam Islam, teknologi dipandang sebagai sarana (wasilah) untuk mencapai kemaslahatan selama digunakan secara bertanggung jawab dan tidak menimbulkan mudarat.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

penelitian yang dilakukan menggunakan metode *Long Short-Term Memory* (LSTM) dengan representasi *Word2Vec* mampu digunakan dengan baik untuk klasifikasi berita multi-kelas berbahasa Indonesia. Model berhasil mempelajari hubungan kontekstual antar kata sehingga mampu menghasilkan performa klasifikasi yang tinggi pada seluruh skenario pengujian.

Pengujian dilakukan menggunakan dua variasi *batch size*, yaitu 32 dan 64, serta dua skenario *K-Fold Cross Validation*, yaitu 5-fold dan 10-fold. Hasil pengujian menunjukkan bahwa seluruh skenario menghasilkan performa yang cukup baik dengan nilai *accuracy* di atas 88%. Skenario terbaik diperoleh pada penggunaan *batch size* 32 dengan 5-fold *cross validation* yang menghasilkan *accuracy* sebesar 0.9421, *precision* sebesar 0.9428, *recall* sebesar 0.9421, dan *F1-score* sebesar 0.9423.

Berdasarkan hasil pengujian, penggunaan *batch size* 32 menghasilkan performa yang lebih baik dibandingkan *batch size* 64. Hal ini menunjukkan bahwa ukuran *batch* yang lebih kecil mampu membantu model dalam melakukan pembaruan bobot secara lebih optimal selama proses pelatihan. Sementara itu, penggunaan 10-fold *cross validation* memberikan evaluasi yang lebih stabil pada beberapa skenario karena model diuji pada variasi data yang lebih luas.

Hasil analisis grafik *accuracy* dan *loss* menunjukkan bahwa model mampu melakukan proses pelatihan dengan stabil tanpa mengalami *overfitting.*

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa saran yang dapat digunakan untuk pengembangan penelitian selanjutnya. Penelitian berikutnya dapat menggunakan metode representasi kata yang lebih modern, seperti *FastText, GloVe*, maupun model berbasis transformer seperti *BERT* untuk membandingkan performa terhadap *Word2Vec*.

Penelitian selanjutnya dapat mencoba variasi arsitektur model lain, seperti Bidirectional LSTM (BiLSTM), GRU, maupun kombinasi CNN-LSTM untuk meningkatkan kemampuan model dalam memahami konteks teks secara lebih mendalam. Pengujian terhadap parameter lain seperti jumlah unit LSTM, jumlah *layer, learning rate*, dan *dropout* juga dapat dilakukan untuk memperoleh konfigurasi model yang lebih optimal.

Penggunaan dataset dengan jumlah data yang lebih besar dan kategori berita yang lebih beragam juga dapat dilakukan agar model memiliki kemampuan generalisasi yang lebih baik. Selain itu, proses *preprocessing* dapat dikembangkan lebih lanjut dengan penanganan kata tidak baku, singkatan, maupun bahasa campuran yang sering ditemukan pada berita daring berbahasa Indonesia. Terakhir, model klasifikasi yang telah dikembangkan dapat diimplementasikan ke dalam sistem atau aplikasi berbasis *web* maupun *mobile* sehingga dapat digunakan secara langsung dalam proses klasifikasi berita secara otomatis

DAFTAR PUSTAKA

Abualigah, L., Al-Ajlouni, Y. Y., Daoud, M. S., Altalhi, M., & Migdady, H. (2024). Fake news detection using recurrent neural network based on bidirectional LSTM and GloVe. *Social Network Analysis and Mining*, *14*(1). https://doi.org/10.1007/s13278-024-01198-w

Aggarwal, C. C., & Zhai, C. X. (2012). A survey of text classification algorithms. In *Mining Text Data* (Vol. 9781461432234, pp. 163–222). Springer US. https://doi.org/10.1007/978-1-4614-3223-4_6

Ali, A. A., Latif, S., Ghauri, S. A., Song, O. Y., Abbasi, A. A., & Malik, A. J. (2023). Linguistic Features and Bi-LSTM for Identification of Fake News. *Electronics (Switzerland)*, *12*(13). https://doi.org/10.3390/electronics12132942

Al-Mubarakfuri, S., & Al-Atsari, A. I. (2011). Shahih Tafsir Ibnu Katsir.

Chen, J., Feng, Z., & Jiang, W. (2022). Analysis and Comparison of Chinese News Text Classification Methods Based on Deep Learning. In *Highlights in Science, Engineering and Technology AMMSAC* (Vol. 2022).

Chollet, F. (n.d.). *M A N N I N G*.

Farnham, B., Tokyo, S., Boston, B., Sebastopol, F., & Beijing, T. (n.d.). *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems SECOND EDITION*.

Goldberg, Y., & Levy, O. (2014). *word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method*. http://arxiv.org/abs/1402.3722

Greff, K., Srivastava, R. K., Koutnik, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(10), 2222–2232. https://doi.org/10.1109/TNNLS.2016.2582924

Hasib, K. M., Azam, S., Karim, A., Marouf, A. Al, Shamrat, F. M. J. M., Montaha, S., Yeo, K. C., Jonkman, M., Alhajj, R., & Rokne, J. G. (2023). MCNN-LSTM: Combining CNN and LSTM to Classify Multi-Class Text in Imbalanced News Data. *IEEE Access*, *11*, 93048–93063. https://doi.org/10.1109/ACCESS.2023.3309697

Hochreiter, S., & Urgen Schmidhuber, J. ¨. (n.d.). *Long Short-Term Memory*.

Khuntia, M., & Gupta, D. (2022). Indian News Headlines Classification using Word Embedding Techniques and LSTM Model. *Procedia Computer Science*, *218*, 899–907. https://doi.org/10.1016/j.procs.2023.01.070

Kim, K. G. (2016). Book Review: Deep Learning. *Healthcare Informatics Research*, *22*(4), 351. https://doi.org/10.4258/hir.2016.22.4.351

Liu, C. (2024). Long short-term memory (LSTM)-based news classification model. *PLoS ONE*, *19*(5 May). https://doi.org/10.1371/journal.pone.0301835

Manning, Raghavan, & Schütze,. (n.d.).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient Estimation of Word Representations in Vector Space*. http://arxiv.org/abs/1301.3781

Nayoga, B. P., Adipradana, R., Suryadi, R., & Suhartono, D. (2021). Hoax Analyzer for Indonesian News Using Deep Learning Models. *Procedia Computer Science*, *179*, 704–712. https://doi.org/10.1016/j.procs.2021.01.059

Prasetio Widhi, E., & Hatta Fudholi, D. (n.d.). *IMPLEMENTATION OF DEEP LEARNING FOR FAKE NEWS CLASSIFICATION IN BAHASA INDONESIA*. *03*(02), 370–381. https://doi.org/10.59141/jrssem.v3i2.546

Rahmawati, A., Sulandari, W., Subanti, S., & Yudhanto, Y. (2023). Penerapan Metode Reccurent Neural Network dengan Pendekatan Long Short-Term Memory (LSTM) untuk Meramalkan Harga Saham Hybe Corporation The Application of Recurrent Neural Network Method with the Long Short-Term Memory (LSTM) Approach to Forecast Hybe Corporation's Stock Price. *Jurnal Bumigora Information Technology (BITe)*, *5*(1), 65–76. https://doi.org/10.30812/bite/v5i1.2973

Rianansyah, A., Utami, E., & Ariatmanto, D. (2025). IMPLEMENTATION OF WORD EMBEDDING IN DETECTING POLITICAL FAKE NEWS IN INDONESIA USING LONG SHORT-TERM MEMORY ALGORITHM. *JIPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, *10*(4), 3124–3136. https://doi.org/10.29100/jipi.v10i4.6680

Sebastiani, F. (n.d.). *Machine Learning in Automated Text Categorization*. Retrieved www.ira.uka.de/bibliography/Ai/automated.text.

Tri Hermanto, D., Setyanto, A., & Luthfi, E. T. (n.d.). *Algoritma LSTM-CNN untuk Sentimen Klasifikasi dengan Word2vec pada Media Online LSTM-CNN Algorithm for Sentiment Clasification with Word2vec On Online Media*.

Widhiyasana, Y., Semiawan, T., Gibran, I., Mudzakir, A., & Noor, M. R. (2021). Penerapan Convolutional Long Short-Term Memory untuk Klasifikasi Teks Berita Bahasa Indonesia (Convolutional Long Short-Term Memory

Implementation for Indonesian News Classification). In *Jurnal Nasional Teknik Elektro dan Teknologi Informasi* | (Vol. 10, Number 4).

Yan, H., Gui, L., & He, Y. (2022). *Hierarchical Interpretation of Neural Text Classification under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0) license*. https://doi.org/10.1162/coli