FAKE REVIEW DETECTION TERHADAP ULASAN APLIKASI *MARKETPLACE* MENGGUNAKAN *BERT EMBEDDING* DAN *LONG SHORT TERM MEMORY*

TESIS

Oleh :
MUHAMMAD UDIN
NIM. 240605220001

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

FAKE REVIEW DETECTION TERHADAP ULASAN APLIKASI *MARKETPLACE* MENGGUNAKAN *BERT EMBEDDING* DAN *LONG SHORT TERM MEMORY*

TESIS

Diajukan Kepada:
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Magister Komputer (M.Kom)

Oleh :
MUHAMMAD UDIN
NIM. 240605220001

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

FAKE REVIEW DETECTION* TERHADAP ULASAN APLIKASI *MARKETPLACE* MENGGUNAKAN *BERT EMBEDDING* DAN *LONG SHORT TERM MEMORY

THESIS

Oleh :
MUHAMMAD UDIN
NIM. 240605220001

Telah diperiksa dan disetujui untuk diuji:
Tanggal: 2 Juni 2026

Pembimbing I	Pembimbing II
Dr. Cahyo Crysdian, M.Cs	Dr. H. M. Imamudin, LC. M.A.
NIP. 19740424 200901 1 008	NIP. 19740602 200901 1 010

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

Prof. Dr. Ir. Muhammad Faisal, M.T
NIP. 19740510 200501 1 007

FAKE REVIEW DETECTION* TERHADAP ULASAN APLIKASI *MARKETPLACE* MENGGUNAKAN BERT *EMBEDDING* DAN *LONG SHORT TERM MEMORY

THESIS

Oleh :
MUHAMMAD UDIN
NIM. 240605220001

Telah dipertahankan di Depan Dwan Penguji Thesis
Dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Magister Komputer (M.Kom)
Tanggal : 2 Juni 2026

Susunan Dewan		Tanda Tangan
Penguji I	: Dr. Irwan Budi Santoso, M.Kom NIP. 19770102 201101 1 004	( )
Penguji II	: Dr. Usman Pagalay, M.Si NIP. 19650414 200312 1 001	( )
Pembimbing I	: Dr. Cahyo Crysdian, S.T., M.Cs NIP. 19740424 200901 1 008	( )
Pembimbing II	: Dr. H. M. Imamudin, LC. M.A. NIP. 19740602 200901 1 010	( )

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

Prof. Dr. Muhammad Faisal, M.T
NIP. 19740510 200501 1 007

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini:

Nama	:	Muhammad Udin
NIM	:	240605220001
Program Studi	:	Magister Informatika
Fakultas	:	Sains dan Teknologi

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

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

Malang, 30 April 2026
Yang membuat pernyataan

Muhammad Udin
NIM. 240605220001

MOTTO

"Setiap proses yang dijalani dengan kerja keras
akan berakhir dengan hasil yang membanggakan"

PERSEMBAHAN

Thesis ini saya persembahkan untuk:

- Allah SWT yang memberikan rahmat, hidayah, kesehatan, rejeki, serta semua yang saya butuhkan sampai terselesaikannya Thesis ini.
- Istriku yang tercinta Arbiati Faizah yang sangat membantu memberikan semangat dan do'a.
- Ibu mertua Siti Romlah, terimakasih atas do'a dan restunya
- Bapak Dr. Cahyo Crysdian, S.T., M.Cs dan Bapak Dr. H. Mochamad Imamudin, LC., M.A, selaku dosen pembimbing, terima kasih atas semua masukan, arahan dan bimbingannya.
- Bapak Prof Suhartono, M. Kom, Bapak Dr. Zainal Arifin, M. Kom, Bapak Dr. Agung Teguh W Almais, M.T, Ibu Dr. Ririen Kusumawati, M. Kom, serta bapak ibu dosen yang tidak kami sebutkan, kami ucapkan terima kasih atas ilmu dan bimbingannya.
- Bapak Fery Sriafandi, S. Kom selaku Kepala SMK Plus Khoiriyah Hasyim Tebuireng, kami ucapkan terima kasih telah mendukung serta memberikan ijin studi.
- Teman-teman seperjuangan Pak Pietra, Pak Nanang, Pak Muchis, Pak Bagus, terimakasih atas masukan, ide dan sharingnnya. Selamat berjuang juga untuk teman-teman semua.
- Bapak Banu, Bapak Roihan serta bapak / ibu guru SMK Plus Khoiriyah Hasyim Tebuireng yang memberi dukungan.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Syukur *alhamdulillah* penulis haturkan kehadirat Allah SWT yang telah melimpahkan Rahmat dan Hidayat-Nya, sehingga penulis dapat menyelesaikan studi di Program Studi Magister Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang sekaligus menyelesaikan Thesis ini dengan baik.

Selanjutnya penulis haturkan ucapan terima kasih seiring do'a dan harapan jazakumullah ahsanal jaza' kepada semua pihak yang telah membantu terselesaikannya Thesis ini. Upacan terima kasih ini penulis sampaikan kepada:

1. Bapak Dr. Cahyo Crysdian, S.T., M.Cs dan Bapak Dr. H. Mochamad Imamudin, LC., M.A selaku dosen pembimbing Thesis, yang telah banyak memberikan pengarahan dan pengalaman yang berharga.
2. Segenap sivitas akademika Program Studi Magister Informatika, terutama seluruh Bapak / Ibu dosen, terima kasih atas segenap ilmu dan bimbingannya.
3. Istri tercinta yang senantiasa memberikan do'a dan semangat kepada penulis untuk menyelesaikan Thesis ini.
4. Semua rekan-rekan seperjuangan yang ikut membantu terselesaikannya Thesis ini.

Penulis menyadari bahwa dalam penyusunan Thesis ini masih terdapat kekurangan dan penulis berharap semoga Thesis ini bisa memberikan manfaat kepada para pembaca khususnya bagi penulis secara pribadi. *Amiinn Yaa Rabbal Alamin.*

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Malang, 30 April 2026

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Udin, Muhammad, 2026. ***Fake review Detection*** **Terhadap Ulasan Aplikasi** ***Marketplace*** **Menggunakan** ***BERT Embedding*** **Dan** ***Long Short Term Memory***. Thesis. Program Studi Magister Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana malik Ibrahim Malang. Pembimbing: (1) Dr. Cahyo Crysdian (2) Dr. H. M. Imamudin, LC., M.A.

Kata Kunci: *Fake review, IndoBERT, Word2Vec*, LSTM, SMOTE, ADASYN.

Ulasan palsu (*fake review*) pada platform *marketplace* dapat menyesatkan konsumen dan menurunkan tingkat kepercayaan terhadap sistem penilaian produk. Keberadaan ulasan palsu juga berdampak pada pengambilan keputusan pembelian yang tidak objektif serta merugikan pihak marketplace dan konsumen. Penelitian ini bertujuan untuk mengembangkan model deteksi *fake review* pada ulasan aplikasi *marketplace* menggunakan kombinasi representasi teks IndoBERT dan Word2Vec dengan *algoritma Long Short-Term Memory* (LSTM), serta menganalisis pengaruh teknik *oversampling* SMOTE dan ADASYN dalam mengatasi ketidakseimbangan data. Dataset yang digunakan berupa ulasan pengguna aplikasi Tokopedia berbahasa Indonesia yang telah melalui tahapan *preprocessing* meliputi *case folding, punctuation removal, word normalization*, dan *stemming*. Selanjutnya dilakukan proses *feature extraction* menggunakan IndoBERT dan Word2Vec, kemudian diklasifikasikan menggunakan LSTM. Evaluasi model dilakukan menggunakan metode *K-Fold Cross Validation* dengan metrik *accuracy, precision, recall*, dan F1-*Score*. Hasil penelitian menunjukkan bahwa teknik *oversampling* memberikan pengaruh yang berbeda pada setiap representasi fitur. Pada pendekatan IndoBERT, kombinasi IndoBERT + SMOTE + LSTM menghasilkan performa terbaik dengan *accuracy* sebesar 88,57%, *precision* 92,42%, *recall* 84,72%, dan F1-*Score* 88,40%. Sementara itu, pada pendekatan Word2Vec, kombinasi Word2Vec + ADASYN + LSTM memperoleh performa tertinggi dengan *accuracy* 92,42%, *precision* 86,61%, *recall* 99,19%, dan F1-*Score* 92,47%. Hasil tersebut menunjukkan bahwa Word2Vec lebih optimal dibandingkan IndoBERT pada dataset penelitian ini, sedangkan penerapan teknik *oversampling* terbukti mampu meningkatkan kinerja model dalam mendeteksi ulasan palsu. Penelitian ini diharapkan dapat berkontribusi dalam pengembangan sistem deteksi *fake review* berbahasa Indonesia yang lebih akurat untuk mendukung terciptanya ekosistem marketplace yang lebih terpercaya.

ABSTRACT

Udin, Muhammad, 2026. ***Fake review Detection* of Marketplace Application *Review*s Using BERT *Embedding* and Long Short Term Memory**. Thesis. Master of Informatics Study Program, Faculty of Science and Technology, State Islamic University of Maulana Malik Ibrahim Malang. Supervisors: (1) Dr. Cahyo Crysdian (2) Dr. H. M Imamudin, LC., MA

Keywords: *Fake review*, IndoBERT, Word2Vec, LSTM, SMOTE, ADASYN.

Fake reviews on marketplace platforms can mislead consumers and reduce trust in product rating systems. The presence of fake reviews also affects purchasing decisions, making them less objective and causing losses for both marketplaces and consumers. This study aims to develop a fake review detection model for marketplace application reviews by combining IndoBERT and Word2Vec text representations with the Long Short-Term Memory (LSTM) algorithm, as well as to analyze the impact of SMOTE and ADASYN oversampling techniques in addressing data imbalance issues. The dataset used consists of Indonesian-language user reviews from the Tokopedia application, which underwent several preprocessing stages, including case folding, punctuation removal, word normalization, and stemming. Feature extraction was then performed using IndoBERT and Word2Vec, followed by classification using LSTM. Model evaluation was conducted using the K-Fold Cross Validation method with accuracy, precision, recall, and F1-score as performance metrics. The results indicate that oversampling techniques have different effects on each feature representation approach. For the IndoBERT-based approach, the combination of IndoBERT + SMOTE + LSTM achieved the best performance, with an accuracy of 88.57%, precision of 92.42%, recall of 84.72%, and F1-score of 88.40%. Meanwhile, for the Word2Vec-based approach, the combination of Word2Vec + ADASYN + LSTM produced the highest performance, achieving an accuracy of 92.42%, precision of 86.61%, recall of 99.19%, and F1-score of 92.47%. These findings demonstrate that Word2Vec performed better than IndoBERT on the dataset used in this study, while the application of oversampling techniques effectively improved the model's ability to detect fake reviews. This research is expected to contribute to the development of more accurate Indonesian-language fake review detection systems and support the creation of a more trustworthy marketplace ecosystem.

الملخص

أودين، محمد، ٢٠٢٦. كشف المراجعات الزائفة (فايك ريفيو ديتكشن) في تقييمات تطبيقات السوق الإلكتروني باستخدام ،تمثيل بيرت وخوارزمية الذاكرة طويلة وقصيرة المدى. رسالة ماجستير، برنامج دراسات الماجستير في المعلوماتية .كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانغ
المشرفان: (١)د. جاهيو كريسديان. (٢) د.الحج محمد امان الدين ماجستير.

.الكلمات المفتاحية: المراجعات الزائفة، إندوبيرت، وورد تو فيك، الذاكرة طويلة وقصيرة المدى، إس موت، أداسين

تُعَدُّ المراجعات المزيَّفة في منصات الأسواق الإلكترونية من المشكلات التي قد تؤدي إلى تضليل المستهلكين وتقليل مستوى الثقة في أنظمة تقييم المنتجات. كما تؤثر هذه المراجعات في قرارات الشراء، مما يجعلها أقل موضوعية، ويتسبب في أضرار لكل من منصات التجارة الإلكترونية والمستهلكين. تهدف هذه الدراسة إلى تطوير نموذج للكشف عن المراجعات المزيَّفة في مراجعات تطبيقات الأسواق الإلكترونية من خلال دمج تمثيل النصوص باستخدام نموذج إندوبرت ونموذج وورد تو فيك مع خوارزمية الذاكرة الطويلة قصيرة المدى، بالإضافة إلى تحليل تأثير تقنيتي سموت وأداسين لمعالجة مشكلة عدم توازن البيانات .تتكون مجموعة البيانات المستخدمة من مراجعات مستخدمي تطبيق توكوبيديا المكتوبة باللغة الإندونيسية، والتي خضعت لعدة مراحل من المعالجة المسبقة للنصوص، تشمل توحيد الأحرف، وإزالة علامات الترقيم، وتطبيع الكلمات، واستخراج الجذور الصرفية. بعد ذلك، تم تنفيذ عملية استخراج السمات باستخدام نموذج إندوبرت ونموذج وورد تو فيك، ثم تصنيف البيانات باستخدام نموذج الذاكرة الطويلة قصيرة المدى. وتم تقييم أداء النموذج باستخدام أسلوب التحقق المتقاطع متعدد الطيات بالاعتماد على مقاييس الدقة، والدقة الإيجابية، والاسترجاع، والدرجة التوافقية .أظهرت النتائج أن تقنيات زيادة العينات الاصطناعية كان لها تأثيرات مختلفة تبعًا لطريقة تمثيل السمات المستخدمة. ففي نهج إندوبرت، حقق نموذج إندوبرت + سموت + الذاكرة الطويلة قصيرة المدى أفضل أداء، حيث بلغت الدقة ٨٨,٥٧٪، والدقة الإيجابية ٩٢,٤٢٪، والاسترجاع ٨٤,٧٢٪، والدرجة التوافقية ٨٨,٤٠٪. أما في نهج وورد تو فيك، فقد حقق نموذج وورد تو فيك + أداسين + الذاكرة الطويلة قصيرة المدى أعلى أداء، حيث بلغت الدقة ٩٢,٤٢٪، والدقة الإيجابية ٨٦,٦١٪، والاسترجاع ٩٩,١٩٪، والدرجة التوافقية ٩٢,٤٧٪ . تشير هذه النتائج إلى أن نموذج وورد تو فيك كان أكثر كفاءة من نموذج إندوبرت على مجموعة البيانات المستخدمة في هذه الدراسة، كما أثبتت تقنيات زيادة العينات الاصطناعية فعاليتها في تحسين أداء النموذج في الكشف عن المراجعات المزيَّفة. ومن المتوقع أن تسهم هذه الدراسة في تطوير أنظمة أكثر دقة للكشف عن المراجعات المزيَّفة باللغة الإندونيسية، بما يدعم بناء بيئة تجارة إلكترونية أكثر موثوقية.

BAB I

PENDAHULUAN

1.1 Latar Belakang

E-commerce singkatan dari *eletronic commerce*, merujuk pada kegiatan jual beli barang atau jasa yang dilakukan secara elektronik melalui internet yang melibatkan transfer uang dan pertukaran data elektronik untuk menjalankan transaksi bisnis. Saat ini, semakin banyak layanan e-*commerce* yang bermunculan dan dengan cepat mendapatkan popularitas karena pangsa pasar yang besar, banyak penyedia layanan e-*commerce* berlomba-lomba untuk menjadi yang terdepan. Saat ini, sarana *e-commerce* adalah bukan hanya lewat telepon dan televisi saja, tetapi kini lebih seiring menggunakan internet. Sebagian orang salah mengartikan antara *marketplace* dengan *e-commerce* dan menganggap keduanya sama. Padahal, pengertian *e-commerce* berbeda dengan *marketplace*. *Marketplace* merupakan salah satu model dari *e-commerce* yang bertindak sebagai perantara antara pembeli dan penjual. Contohnya seperti shopee, lazada, tokopedia, dan lain-lain. Jadi, *marketplace* bukan merupakan aktivitas jual belinya, melainkan perantara yang mempertemukan penjual dengan pembeli secara online. Sementara itu, bentuk lainnya *e-commerce* adalah berupa *website* atau aplikasi toko *online* yang dimiliki oleh suatu *brand*, perusahaan, atau bisnis rumahan.

Salah satu penyedia *e-commerce* yang umum digunakan masyarakat Indonesia saat ini, khususnya generasi milenial merupakan aplikasi tokopedia.

Tokopedia adalah salah satu perusahaan jual beli berbasis digital terbesar di Indonesia. Sejak resmi diluncurkan pada tanggal 17 Agustus 2009 yang didirikan oleh William Tanuwijaya dan Leontinus Alpha Edison. Pada tahun 2016 sampai 2023 tokopedia meluncurkan beberapa produk ataupun layanan yang dapat memudahkan konsumen dalam penggunaan aplikasi tokopedia. Adapun salah satu contoh produk yang telah diluncurkan oleh tokopedia yaitu teknologi finansial dan deals. Sedangkan pada layanan, tokopedia telah meluncurkan layanan yaitu dilayani tokopedia dan layanan pemenuhan pesanan (*fulfillment*). Di tahun 2023, Tokopedia telah memberdayakan lebih dari 14 juta penjual terdaftar, menawarkan lebih dari 40 produk digital yang dapat mempermudah kehidupan, dan memiliki lebih dari 1,8 miliar produk yang terdaftar.

Seiring dengan meningkatnya jumlah pengguna aplikasi tokopedia, timbul tantangan baru terkait bagaimana masyarakat merespon dan menilai aplikasi tersebut. Respon pengguna terhadap aplikasi seringkali tercermin melalui ulasan dan rating yang diberikan di platform distribusi aplikasi seperti google playstore. Ulasan ini memberikan informasi berharga bagi pengembang untuk mengevaluasi performa aplikasi serta memahami kebutuhan pengguna. Lebih dari itu, ulasan tersebut juga dapat menjadi cermin bagi calon pengguna yang mempertimbangkan untuk mengunduh dan menggunakan aplikasi tersebut (Gilbert et al. 2023).

Perkembangan teknologi digital mendorong pertumbuhan pesat *e-commerce* di berbagai negara, termasuk Indonesia. *Marketplace* menjadi

salah satu bentuk platform *e-commerce* yang paling diminati masyarakat karena menawarkan kemudahan dalam mencari produk, melakukan transaksi, hingga proses pengiriman. Salah satu fitur penting yang tersedia dalam *marketplace* adalah ulasan atau *review* konsumen, yang berfungsi sebagai sumber informasi dan pertimbangan utama dalam pengambilan keputusan pembelian. Ulasan dianggap lebih kredibel karena berasal dari pengalaman pengguna langsung. Menurut laporan Invesp (2023), lebih dari 90% konsumen global membaca ulasan daring sebelum memutuskan untuk membeli suatu produk, sehingga kehadiran ulasan menjadi faktor penting dalam membangun kepercayaan konsumen.

Sistem ulasan juga menghadapi tantangan serius berupa keberadaan ulasan palsu atau *fake review*. Ulasan palsu biasanya dibuat dengan tujuan tertentu, misalnya meningkatkan reputasi produk dengan ulasan positif yang tidak autentik, atau menjatuhkan kompetitor melalui ulasan negatif yang direkayasa. Praktik ini berdampak signifikan terhadap ekosistem *marketplace*. Laporan Invesp (2023) mencatat bahwa kerugian global akibat ulasan palsu mencapai USD 152 miliar per tahun. Di Indonesia, survei Cube Asia (2025) menunjukkan bahwa sekitar 5–10% transaksi di *marketplace* merupakan transaksi fiktif, yang sering kali berkaitan dengan manipulasi ulasan untuk memperkuat citra produk atau toko. Secara global, Tripadvisor juga melaporkan telah menghapus lebih dari 2,7 juta ulasan palsu pada tahun 2024, setara dengan satu dari 12 ulasan yang dikirimkan pengguna (Tripadvisor, 2025). Fenomena ini menunjukkan bahwa ulasan palsu bukan sekadar isu

lokal, melainkan persoalan global yang dapat menurunkan tingkat kepercayaan konsumen dan merugikan pihak terkait.

Deteksi terhadap *fake review* menjadi tantangan dalam bidang *Natural Language Processing* (NLP). Hal ini dikarenakan ulasan di *marketplace* umumnya singkat, tidak terstruktur, menggunakan bahasa informal atau campuran, serta penuh dengan subjektivitas. Lebih jauh, ulasan palsu sering kali dirancang sedemikian rupa agar tampak meyakinkan, sehingga sulit dibedakan dari ulasan asli. Metode representasi teks tradisional seperti Bag-of-Words atau TF-IDF memiliki keterbatasan karena hanya menghitung frekuensi kata tanpa memahami konteks semantik (Masyam & Mohsin, 2024). Oleh karena itu, dibutuhkan pendekatan analisis teks yang mampu memahami konteks bahasa secara lebih mendalam.

Salah satu pendekatan terkini yang banyak digunakan adalah model representasi berbasis *Bidirectional Encoder Representations from Transformers* (BERT). BERT mampu menghasilkan representasi teks yang kontekstual, sehingga lebih efektif dalam memahami makna kata sesuai dengan kalimat dan relasinya (Pengfei Sun et al., 2024). Di sisi lain, model *Long Short-Term Memory* (LSTM) sebagai varian dari *Recurrent Neural Network* (RNN) terbukti unggul dalam mengolah data sekuensial serta mengingat informasi jangka panjang (Hochreiter & Schmidhuber, 1997). Kombinasi antara BERT *embedding* dan LSTM telah terbukti meningkatkan performa dalam berbagai tugas klasifikasi teks. Penelitian terbaru yang memanfaatkan IndoBERT *embedding* dengan LSTM menunjukkan hasil akurasi sekitar 80% dalam

mendeteksi *fake review* pada *e-commerce* (Rami Mohawesh et al., 2024). Studi lain yang menggunakan metode tradisional seperti SVM dan Random Forest dalam deteksi ulasan palsu di Shopee hanya mencapai akurasi 88,84% tanpa mengoptimalkan kekuatan *embedding* berbasis transformer (Khoirotulmuadiba Purifyregalia, 2025).

Dalam perspektif etika bisnis Islam, kejujuran merupakan prinsip fundamental yang harus dijunjung tinggi dalam setiap aktivitas muamalah, termasuk jual beli secara daring. Kejujuran tidak hanya berkaitan dengan penyampaian informasi produk yang benar, tetapi juga mencakup transparansi dalam kualitas barang, harga, serta pengalaman penggunaan yang tercermin dalam ulasan konsumen. Praktik ulasan palsu bertentangan secara langsung dengan nilai kejujuran karena mengandung unsur penipuan (tadlīs) dan manipulasi informasi yang dapat menyesatkan konsumen. Al-Qur'an secara tegas menekankan pentingnya kejujuran dalam transaksi, di dalam Surah Al-Muthaffifin ayat 1–3 Allah SWT berfirman :

وَيْلٌ لِّلْمُطَفِّفِينَ ۝١ الَّذِينَ إِذَا اكْتَالُوا عَلَى النَّاسِ يَسْتَوْفُونَ ۝٢ وَإِذَا كَالُوهُمْ أَو وَّزَنُوهُمْ يُخْسِرُونَ ۝٣

Artinya :

"Kecelakaan besarlah bagi orang-orang yang curang, (yaitu) orang-orang yang apabila menerima takaran dari orang lain mereka minta dipenuhi, dan apabila mereka menakar atau menimbang untuk orang lain, mereka mengurangi." (QS. Al-Muthaffifin/1-3)

Dalam tafsir Ibnu Katsir dijelaskan bahwa ayat ini tidak hanya terbatas pada kecurangan timbangan secara fisik, tetapi mencakup seluruh bentuk pengurangan hak orang lain dalam transaksi, termasuk penyampaian informasi

yang tidak benar (Ibnu Katsir, Tafsir al-Qur'an al-'Azhim, tafsir QS. Al-Muthaffifin [83]: 1–3). Prinsip ini diperkuat oleh hadis Rasulullah : "Barang siapa menipu kami, maka ia bukan termasuk golongan kami" (HR. Muslim No. 102). Hadis tersebut menunjukkan bahwa segala bentuk penipuan dan penyembunyian fakta dalam aktivitas muamalah merupakan perbuatan yang tercela. Dalam konteks perdagangan elektronik (*e-commerce*), ulasan palsu (*fake review*) dapat menjadi sarana untuk menipu konsumen dengan memberikan gambaran yang tidak sesuai dengan kondisi produk atau layanan yang sebenarnya. Oleh karena itu, praktik tersebut bertentangan dengan prinsip kejujuran (shidq) dan keadilan yang diajarkan dalam Islam. (Ibnu Katsir, tafsir QS. Al-Muthaffifin [83]: 1–3; HR. Muslim No. 102).

Selain kejujuran, Islam juga mengajarkan prinsip tabayun (klarifikasi dan verifikasi informasi) sebagai landasan dalam menyikapi informasi yang diterima, terutama informasi yang berpotensi menimbulkan kerugian. Dalam konteks *e-commerce*, tabayun menjadi sangat relevan karena konsumen sering kali menjadikan ulasan daring sebagai dasar pengambilan keputusan pembelian. Allah SWT berfirman dalam Surah Al-Hujurat ayat 6:

يَا أَيُّهَا الَّذِينَ آمَنُوا إِن جَاءَكُمْ فَاسِقٌ بِنَبَإٍ فَتَبَيَّنُوا أَن تُصِيبُوا قَوْمًا بِجَهَالَةٍ فَتُصْبِحُوا عَلَىٰ مَا فَعَلْتُمْ نَادِمِينَ

Artinya :

"Wahai orang-orang yang beriman, jika datang kepadamu orang fasik membawa suatu berita, maka telitilah kebenarannya (tabayun), agar kamu tidak menimpakan suatu musibah kepada suatu kaum tanpa mengetahui keadaannya."(QS. Al Hujurat/6).

Menurut tafsir Al-Qurthubi, ayat ini menegaskan kewajiban melakukan verifikasi atas setiap informasi sebelum dijadikan dasar tindakan, terutama ketika informasi tersebut berpotensi menimbulkan mudarat. Dalam konteks digital, ulasan palsu dapat dipandang sebagai "berita" yang tidak terverifikasi, sehingga penggunaan sistem otomatis untuk mendeteksi *fake review* sejalan dengan prinsip tabayun guna mencegah kerugian konsumen. Prinsip kehati-hatian dalam menerima informasi ini juga diperkuat oleh sabda Rasulullah: "Tinggalkanlah apa yang meragukanmu kepada apa yang tidak meragukanmu" (HR. At-Tirmidzi No. 2518 dan An-Nasa'i No. 5711).

Hadis tersebut mengajarkan bahwa setiap informasi yang masih diragukan kebenarannya hendaknya tidak langsung diterima atau dijadikan dasar pengambilan keputusan sebelum dilakukan pemeriksaan dan verifikasi yang memadai. Oleh karena itu, pengembangan sistem deteksi ulasan palsu (*fake review*) merupakan salah satu bentuk implementasi nilai tabayun dalam era digital untuk memastikan informasi yang beredar bersifat akurat, terpercaya, serta tidak merugikan konsumen maupun pelaku usaha.

Lebih lanjut, jual beli online dalam pandangan Islam pada dasarnya diperbolehkan (mubah) selama memenuhi rukun dan syarat jual beli yang sah. Beberapa kriteria utama jual beli yang sesuai syariah antara lain adanya pihak yang berakad (penjual dan pembeli), adanya objek transaksi yang jelas dan halal, adanya kesepakatan (ijab dan qabul), serta tidak mengandung unsur gharar (ketidakjelasan), penipuan, dan kedzaliman. Al-Qur’an menegaskan kebolehan jual beli yang dilakukan secara adil dalam Surah Al-Baqarah ayat

275: “Allah telah menghalalkan jual beli dan mengharamkan riba.” Dalam tafsir Ath-Thabari dijelaskan bahwa kebolehan jual beli dalam ayat ini mensyaratkan adanya kejelasan dan keadilan bagi semua pihak yang terlibat. Oleh karena itu, informasi produk dan ulasan konsumen yang tidak jujur dapat menimbulkan gharar karena menciptakan ketidakpastian dan ketidakseimbangan informasi antara penjual dan pembeli.

Dengan demikian, praktik penyebaran ulasan palsu tidak hanya menimbulkan dampak negatif dari sisi ekonomi dan kepercayaan konsumen, tetapi juga bertentangan dengan nilai-nilai fundamental Islam yang menjunjung tinggi kejujuran, keadilan, dan kehati-hatian dalam menerima informasi. Pengembangan sistem deteksi *fake review* berbasis kecerdasan buatan dapat dipandang tidak hanya sebagai solusi teknis, tetapi juga sebagai implementasi nilai-nilai syariah dalam konteks digital modern. Integrasi pendekatan teknologi dengan prinsip muamalah Islam ini diharapkan mampu menciptakan ekosistem *e-commerce* yang lebih aman, adil, dan berkelanjutan, serta mendukung tercapainya kemaslahatan bagi seluruh pihak yang terlibat.

Berdasarkan pemaparan tersebut, penelitian ini difokuskan pada pengembangan model deteksi ulasan palsu (*fake review*) pada aplikasi marketplace dengan memanfaatkan kombinasi BERT *embedding* sebagai metode representasi teks kontekstual dan *Long Short-Term Memory* (LSTM) sebagai model klasifikasi sekuensial. Penelitian ini diharapkan memberikan kontribusi ilmiah dalam memperkaya kajian deteksi *fake review* berbahasa Indonesia yang masih terbatas, khususnya dengan mengintegrasikan kekuatan

representasi berbasis transformer dan kemampuan pemodelan dependensi jangka panjang. Selain kontribusi teknis, penelitian ini juga memiliki implikasi praktis dalam meningkatkan kualitas sistem ulasan, mendukung pengambilan keputusan konsumen yang lebih akurat, serta membantu penyedia *marketplace* dalam menjaga integritas platform. Lebih jauh, hasil penelitian ini diharapkan dapat berkontribusi pada terciptanya ekosistem e-commerce yang lebih jujur, transparan, dan terpercaya, sejalan dengan prinsip etika bisnis dan nilai keadilan dalam transaksi digital yang berkelanjutan.

1.2 Pernyataan Masalah

a. Fitur representasi teks apa yang paling optimal untuk deteksi ulasan palsu (*fake review*) menggunakan BERT *embedding*?
b. Bagaimana kinerja metode *Long Short-Term Memory* (LSTM) yang dikombinasikan dengan BERT *embedding* dalam mendeteksi ulasan palsu *(fake review*)?

1.3 Tujuan Penelitian

a. Melakukan Analisis Fitur representasi teks apa yang paling optimal untuk deteksi ulasan palsu (*fake review*) menggunakan BERT *Embedding*.
b. Mengukur kinerja metode *Long Short-Term Memory* (LSTM) yang dikombinasikan dengan BERT *embedding* dalam mendeteksi ulasan palsu (*fake review*).

1.4 Manfaat Penelitian

a. Membantu stakeholder *marketplace* dalam mengidentifikasi dan meminimalkan keberadaan ulasan palsu, sehingga meningkatkan kepercayaan konsumen terhadap sistem penilaian produk.
b. Memberikan alat bantu bagi konsumen agar dapat membuat keputusan pembelian yang lebih tepat berdasarkan ulasan yang valid dan terpercaya.
c. Menjadi referensi bagi pengembang sistem *e-commerce* dalam membangun model deteksi *fake review* yang lebih akurat dan aplikatif pada platform berbahasa Indonesia.

1.5 Ruang Lingkup Penelitian

a. Data yang digunakan berupa ulasan pengguna pada aplikasi *marketplace* Tokopedia berbahasa Indonesia**.**
b. Dataset dibatasi pada ulasan teks bebas yang sering bersifat informal, tidak baku, dan tidak terstruktur (tanpa mempertimbangkan gambar, video, atau rating bintang).

BAB II

STUDI PUSTAKA

2.1 Deteksi *Fake review*

Penelitian yang dilakukan Joni Salminet et al. (2023) menyatakan maraknya *fake review* pada platform *e-commerce* dari dataset Yelp dapat menyesatkan konsumen dalam pengambilan keputusan. Dengan pendekatan berbasis *marchine learning* pada *review based features* dan *user based features* menemukan random forest memberikan hasil terbaik dengan fitur berbasis user terbukti sangat penting dalam membedakan *review* asli dan palsu, bahkan lebih signifikan daripada hanya fitur berbasis teks sehingga deteksi *fake review* lebih efektif jika menggunakan analisis konten teks dengan perilaku pengguna, bukan hanya melihat isi *review*.

Selanjutnya penelitian yang dilakukan Nour Qandos et al. (2024) menyatakan *fake review* di platform *e-commerce* sangat merugikan pelanggan dan bisnis, menurunkan kepercayaan. Dan bahasa arab memiliki keragaman dialek sehingga mempersulit proses deteksi *fake review* secara otomatis sehingga diperlukan metode baru untuk mendeteksi *fake review* dalam bahasa arab menggunakan arabic *fake review*s detection (AFRD) yang mencakup tiga domain diantaranya hotel, restoran dan produk. Dengan model deep learning yang digunakan pendekatan cascading multiscale domain-based (MCDB). Hasil dari pendekatan MCDB meningkatkan kinerja model dari 2% hingga 8%

dengan hotel akurasi meningkat menjadi 100%, restourant akurasi tertingg mencapai 100% dan product akurasi hingga 87,23%.

Penelitian Richa Gupta et al. (2024) penelitian ini mengusulkan sebuah model Siamese Bi-LSTM (SbiLM), yang menggabungkan Siamese *Neural Network* dengan *Bidirectional* LSTM dan Manhattan Distance untuk mengukur tingkat kemiripan antara pasangan ulasan. Model ini dirancang untuk mengatasi masalah ketidakseimbangan kelas dalam deteksi ulasan palsu tanpa memerlukan teknik sampling seperti under-samping yang sering kali mengakibatkan kehilangan informasi penting. Hasil menunjukkan bahwa SbiLM memiliki performa yang lebih unggul khususnya untuk kelas ulasan palsu, dengan hasil mengurangi tingkat *False Positive* dan *False Negative* secara signifikan. Selain itu model ini menunjukkan akurasi yang lebih tinggi dan loss yang lebih rendah dibandingkan dengan model deep learning tradisional seperti CNN, DNN, dan SMOTE-DNN pada dataset Yelp yang sangat tidak seimbang.

Penelitian Jinhao Liu et al (2024) didasari masalah deteksi ulasan palsu di platform *e-commerce* dengan ketidakseimbangan data dimana jumlah ulasa asli jauh dominan dibandingkan dengan ulasan palsu, dan kesulitan metode yang bergantung pada konteks teks yang sering terpengaruh oleh perubahan teks, perbedaan bahasa, dan konteks untuk menyembunyikan perilaku manipulatif. Penelitian menggunakan model RoBERTa yang dipadukan dengan fitur perilaku seperti aktivitas pengguna, jumlah rating positif atau negatif, dan urutan ulasan dengan augmentasi data menggunakan SMOTE untuk

menangani ketidakseimbang data. Hasil dari penelitian menunjukkan bahwa model yang menggabungkan Roberta dengan LSTM dan CNN serta fitur perilaku (Roberta+LSTM-CNN+Bfs) akurasi mencapai 87.65% dengan *recal*l tertinggi pada kelas ulasan palsu mencapai 89.87%.

Selanjutnya penelitian Maged Nasser et al (2025) didasari deteksi berita palsu di media sosial menjadi isu penting karena meningkatnta volume informasi yang tersebar dalam berbagai bentuk data seperti teks, gambar, video dan audio, sehingga membuat proses identifikasi berita palsu semakin kompleks terutama dengan adanya manipulasi canggih seperti deepfakes. Penelitian melakukan pendekatan berbasis deep learning dengan memanfaatkan model transformer, recurrent neural network (RNNs), dan Convolutional Neural Networks (CNN) mampu memberikan hasil yang lebih akurat dibandingkan metode tradisional, khususnya ketika menggabungkan data multimodal yang mencakup konten teks, visual, maupun audio. Hasil tinjauan menegaskan bahwa fusion multimodal lebih efektif daripada pendekatan single-model, dengan dataset popular seperti Twitter dan Weibo digunakan sebagai benchmark dan masih terdapat tantangan besar terkait keterbatasan dataset, keberagaman bahasa, serta perbedaan konteks budaya antarplatform.

Penelitian Rami Mohawesh, et al. (2024) melakukan penelitian deteksi ulasan palsu dengan menggunakan 2 dataset yaitu deception dataset yang berisi 3.032 ulasan dari sector hotel, dokter, restoran dan opspam dataset yang terdiri dari 1.600 ulasa hotel di Chicago. Dengan rendahnya kemampuan manusia

maupun model klasik dalam mendeteksi ulasan palsu yang hanya terfokus pada aspek linguistic tanpa menangkap makna semantic teks. Penelitian ini menawarkan model Roberta sebagai representasi semantic kontekstual dengan LSTM untuk menangkap pola sekuensialnya dengan tahapan meliputi pra prosesing data, tokenisasi, ekstraksi *embedding* dengan Roberta sedangkan pemprosesan sekuensial oleh LSTM. Hasil eksperimen menunjukkan bahwa pada opspam dataset model mencapai akurasi 96,03% sedangkan pada deception dataset akurasinya 93,15%, dengan demikian membuktikan bahwa integrasi Roberta dan LSTM lebih unggul dibandingkan metode terdahulu serta memberikan kontribusi penting bagi peningkatan keandalan deteksi ulasan palsu pada system *e-commerce* dan platform daring.

Selanjutnya penelitian yang dilakukan Ammar Oad et al. (2024) penelitian ini berfokus pada metode deteksi berita palsu berbasis Enhanced BERT dengan menggunakan dataset PolitiTweet yaitu dataset dengan mengumpulkan cuitan politik berbahasa inggris dari Twitter serta BuzzFeed yaitu dataset berisi berita hiburan yang sudah diberi label benar atau palsu. Dengan mengunakan Enhanced BERT yakni pengembangan dari BERTbase dengan tambahan linear layers, reLU activation functions, dropout layer yang dilatih dengan proses fine-tuning setelah tahap pra pemrosesan teks. Hasil penelitian menunjukkan bahwa Enhancd BERT secara konsisten memberikan performa lebih tinggi dibanding model pembanding dengan akurasi mencapai 98,23% pada PolitiTweet Dataset dan 85% pada BuzzFeed dataset. Dengan demikian penelitian membuktikan bahwa enhanced BERT tidak hanya unggul dalam

mengatasi masalah ketidakseimbangan data, tetapi juga mampu memperkuat generalisasi model dalam mendeteksi berita palsu pada konteks lintas budaya dan bahasa local.

Selanjutnya pada penelitian Gulsum Kayabasi Koru et. Al. (2024) meneliti sistem deteksi berita palsu dalam bahasa turki menggunakan pendekatan *transforme-based model* dengan dataset yag digunakan terdiri dari 10.000 cuitan twitter terkait isu politik, social, dan berita popular yang diberi label secara manual oleh ahli bahasa dan jurnalisme sebagai berita palsu atau benar. Penelitian ini menerapkan model BERTurk dengan tahapan penelitian meliputi pra pemrosesan data, tokenisasi dengan tokenizer BERTurk, pelatihan denan fine-tuning untuk klasifikasi biner. Hasil yang didapatkan BERTurk-large memberikan memberikan performa terbaik dengan akurasi 95% sedangkan BERTurk-Base memperoleh akurasi 92%. Dengan demikian penelitian ini memberikan kontribusi penting bagi pengembangan sistem deteksi berita palsu pada bahasa non-inggris dan membuka peluang untuk penerapan lintas bahasa dalam konteks yang lebih luas.

Penelitian Juan Juan Peng (2024) penelitian berfokus pada pengembangan metode deteksi komentar palsu dalam platform *e-commerce* dengan menggunakan dataset JD.com yang berisi komentar asli maupun palsu, yang umumnya berupa promosi berlebihan atau manipulative. Penelitian ini menggunakan model gabungan berbasis *Triplet Convolutional Twin Network* (TCTN) dan CatBoost. TCTN digunakan untuk menghasilkan representasi semantic teks dengan memanfaatkan triplet loss sehingga *embedding* komentar

asli dan palsu dapat terpisah lebih jelas, sedangkan arsitektur twin network dengan CNN membantu mengekstraksi fitur semantic mendalam dari pasangan komentar. Representasi yang dihasilan kemudian diproses menggunakan CatBoost untuk klasifikasi akhir. Hasil dari penelitian ini menunjukkan bahwa kombinasi TCTN dan Catboost maupun CNN dapat capai akurasi diatas 95%. Dari hasil tersebut menegaskan bahwa *deep learning* untuk representasi teks dengan algoritma boosting untuk klasifikasi mampu meningkatkan kinerja deteksi komentar palsu, sehingga memberikan kontribusi praktis dalam menjaga kredibilitas platform *e-commerce*.

Selanjutnya penelitian Nishant Rai, et al. (2024) penelitian pada maraknya penyebaran ulasan palsu di platform daring yang dapat menyesatkan konsumen dan merusak kepercayaan terhadap layanan *online* meliputi keterbatasan dataset berlabel, perbedaan gaya bahasa serta model tradisional dalam kompleksitas semantic teks. Metode yang dilakukan mencakup pendekatan *machine learning* LSTM hingga transformer based seperti BERT. Hasil penelitian menunjukkan performa lebih baik dengan akurasi di atas 90% pada dataset benchmark, tapi performanya sering menurun drastic ketika diuji pada domain berbeda sehingga masih ada kesenjangan antara kinerja penelitian dan aplikasi nyata dapat disimpulkkan bahwa pendekatan multimodal serta cross-domain menjadi penting untuk meningatkan keandalan deteksi ulasan palsu.

Anna Glazkova (2020) meneliti perbandingan efektivitas berbagai teknik *oversampling* sintetis dalam menangani ketidakseimbangan kelas pada tugas klasifikasi teks multikategori. Dataset yang digunakan berupa korpus biografi

berbahasa Rusia yang diambil dari Wikipedia, terdiri atas 179 teks biografi yang dipecah menjadi 1.577 kalimat data latih dan 395 kalimat data uji, dengan sepuluh kategori topik seperti kelahiran, pendidikan, peristiwa pribadi, dan afiliasi. Distribusi kategori yang sangat tidak merata menjadi permasalahan utama karena menyebabkan model klasifikasi cenderung bias terhadap kelas mayoritas dan mengabaikan kelas minoritas. Untuk mengatasi hal ini, penelitian membandingkan beberapa metode *oversampling* sintetis, yaitu SMOTE, Borderline SMOTE, dan *Random Oversampling*, serta menguji kinerjanya pada algoritma KNN, SVM, Feedforward Neural Network, LSTM, dan BiLSTM. Teks direpresentasikan menggunakan tiga pendekatan: Bag-of-Words, Bag-of-Words + TF-IDF, dan Word2Vec, sedangkan evaluasi dilakukan menggunakan akurasi, *precision*, *recall*, dan macro F1-*Score*. Hasil penelitian menunjukkan bahwa teknik *oversampling* secara umum meningkatkan performa klasifikasi, dengan SMOTE dan *Random Oversampling* menghasilkan skor terbaik, terutama pada algoritma KNN dan SVM. Sebagai contoh, pada representasi Word2Vec, F1-*Score* KNN meningkat dari 49,42% menjadi lebih dari 70% setelah dilakukan *oversampling*. Sebaliknya, pengaruh teknik *oversampling* terhadap model LSTM dan BiLSTM tidak terlalu signifikan karena arsitektur ini lebih adaptif terhadap ketidakseimbangan data. Penelitian ini menyimpulkan bahwa *oversampling* sintetis efektif meningkatkan performa klasifikasi teks pada data tidak seimbang, khususnya untuk model berbasis pembelajaran tradisional, dan

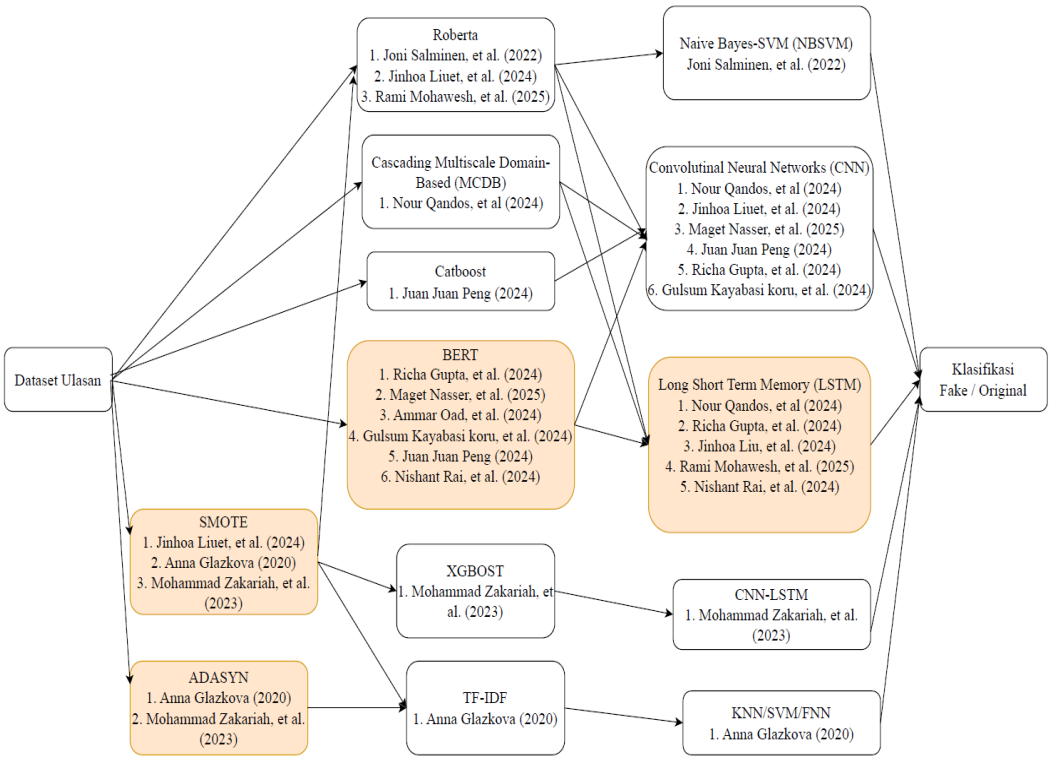
memberikan kontribusi penting dalam strategi penanganan class imbalance pada bahasa dengan sumber daya terbatas seperti Rusia.

Selanjutnya Penelitian yang dilakukan oleh Mohammad Zakariah et, al. (2023) melakukan pengembangan metode deteksi ulasan palsu dengan menggabungkan teknik klasifikasi *decision tree* dan AdaBoost. Dataset yang digunakan berasal dari Yelp dan TripAdvisor, mencakup ribuan ulasan pelanggan yang telah dilabeli sebagai ulasan asli atau palsu, termasuk fitur teks ulasan, rating bintang, metadata pengguna, dan informasi terkait lainnya. Masalah utama yang dikaji adalah meningkatnya jumlah ulasan palsu pada platform daring yang dapat menyesatkan konsumen serta menurunkan kepercayaan terhadap sistem rekomendasi dan reputasi layanan. Metode deteksi konvensional berbasis machine learning sering kali kurang efektif karena ketergantungan pada fitur manual dan kesulitan menangkap pola kompleks pada teks ulasan. Untuk mengatasi hal tersebut, penelitian ini menerapkan pendekatan decision tree sebagai weak learner dan memperkuatnya dengan AdaBoost sebagai teknik ensemble learning adaptif. Prosesnya meliputi pra-pemrosesan data untuk menghapus tanda baca, simbol, dan stopwords, dilanjutkan dengan ekstraksi fitur menggunakan TF-IDF dan seleksi fitur penting sebelum proses klasifikasi. Model ini dibandingkan dengan beberapa algoritma lain seperti Naïve Bayes, Logistic Regression, dan SVM. Hasil penelitian menunjukkan bahwa kombinasi decision tree dan AdaBoost memberikan performa terbaik, dengan akurasi 94,33%, precision 93,75%, *recal*l 94,82%, dan F1-*Score* 94,20%, melampaui model pembanding

lainnya. Temuan ini menegaskan bahwa teknik ensemble berbasis decision tree dan AdaBoost lebih stabil dan mampu menangkap variasi pola teks ulasan secara efektif, sehingga dapat menjadi solusi yang andal dalam mendeteksi ulasan palsu pada platform *e-commerce* dan layanan online.

2.2 Kerangka Teori

Pada beberapa penelitian yang dilakukan oleh peneliti-peneliti dahulu terdapat perbedaan yang digunakan sebagai input dan juga metode yang digunakan dalam tahap proses data meskipun output atau hasil yang diperoleh sama yaitu berupa klasifikasi fake dan original.



Gambar 2. 1 Kerangka Teori

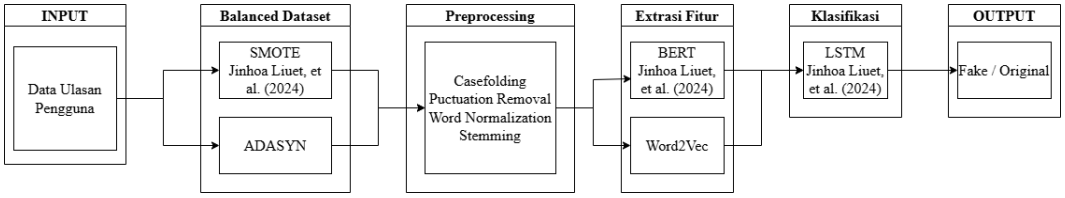
Dari kerangka teori diatas tersebut penelitian menggunakan input dataset berupa *review* dari media social dan *marketplace*. Dari 12 peneliti-peneliti dahulu menggunakan 10 pendekatan untuk menghasilkan output berupa klasifikasi Fake dan Original. yaitu metode RoBERTa, MCDB, Catboost, BERT, XGBOST, TF-IDF, NBSVM, CNN, LSTM, CNN-LSTM. Untuk metode NBSVM ada 1 literatur yang menggunakan metode tersebut. Sedangkan ada 6 literatur yang menggunakan metode CNN. Untuk metode LSTM ada 5 literatur dan juga metode metode CNN-LSTM ada 1 literatur.

BAB III

METODOLOGI

3.1 Kerangka Konsep

Kerangka konsep dapat ditunjukkan pada gambar 3.1 tentang kerangka konsep berikut ini :

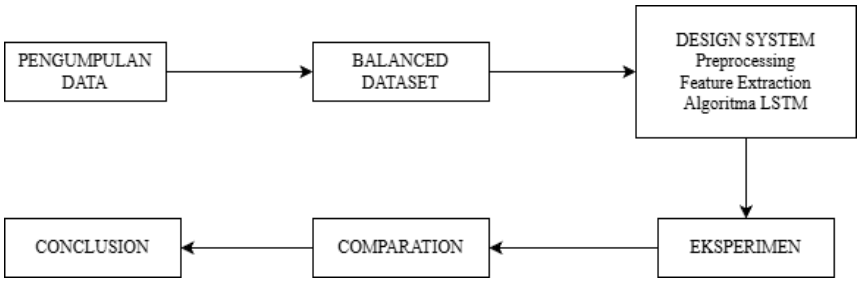


Gambar 3.1 Kerangka Konsep

Pada Gambar 3.1 tentang kerangka konsep menjelaskan tahap pertama yang dilakukan pada penelitian ini adalah pengumpulan data sebagai bahan penelitian. Penelitian ini menggunakan data ulasan pada *marketplace* tokopedia. Setelah pengumpulan data dilakukan proses Balance Dataset yaitu dengan Smote dan Adasyn. Data *Preprocessing* digunakan untuk mengambil semua informasi yang tersedia dengan cara membersihkan, memfilter, dan menggabungkan data-data yang akan digunakan. *Preprocessing* yang dilakukan pada tahap ini yaitu *case folding, punctuation removal, word normalization dan stemming*. Selanjutnya dilakukan proses word *embedding*. Dalam penelitian ini menggunakan IndoBert dan Word2Vec. Selanjutnya diproses dengan LSTM untuk mendeteksi *fake review*. Hal ini bertujuan untuk mengetahui klasifikasi original *review* atau fake feriew.

3.2 Desain Penelitian

Desain penelitian dapat ditunjukkan pada gambar 3.2 tentang desain penelitian berikut ini :



Gambar 3.2 Desain Penelitian

Desain penelitian pada gambar 3.2 menjelaskan rancangan penelitian yang dimulai dari pengumpulan data, mengembangkan algoritma untuk balanced dataset, mengembangkan algoritma *Preprocessing*, mengembangkan algoritma *feature extraction*, mengembangkan algoritma LSTM, eksperimen, comparation, conclusion.

3.3 Pengumpulan Data

Tahap pengumpulan data merupakan tahap yang sangat penting dalam sebuah penelitian, karena data yang terkumpul menjadi bahan utama dan inti dari objek penelitian. Pada penelitian ini, dataset yang digunakan merupakan dataset dari penelitian Habib Alamsyah (2023) yang tersedia secara terbuka pada repositori GitHub. Dataset tersebut berisi ulasan (*review*) berbahasa Indonesia yang berasal dari platform Tokopedia.

Dalam penelitian Habib Alamsyah (2023), proses pelabelan data dilakukan dengan menggunakan metode *Latent Dirichlet Allocation* (LDA) untuk

membantu mengelompokkan karakteristik ulasan secara tematik. LDA memandang setiap dokumen sebagai campuran beberapa topik, di mana setiap topik direpresentasikan oleh distribusi kata-kata yang saling berkaitan. Pada setiap ulasan diberikan dua label, yaitu *original* (OR) dan *computer generated* (CG). Label *original* (OR) ditandai dengan penggunaan kata-kata yang bersifat variatif, kontekstual, dan natural, yang mengindikasikan pengalaman pribadi pengguna seperti "barang sesuai", "harga murah", dan "pelayanan baik". Sebaliknya, label *computer generated* (CG) didominasi oleh kata-kata yang bersifat generik, repetitif, dan umum seperti "mantap", "cepat", dan "fast".

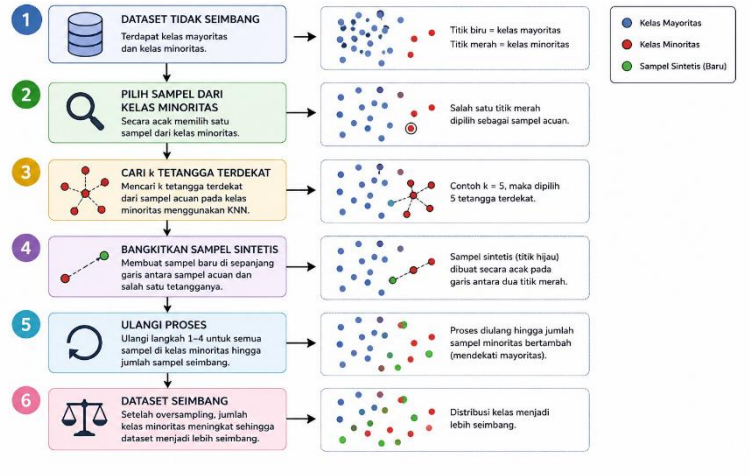
Untuk meningkatkan keakuratan dan validitas pelabelan, penentuan label pada dataset ini dilakukan secara semi-otomatis, yaitu dengan mengombinasikan hasil pemodelan topik menggunakan LDA dan proses validasi oleh pakar atau ahli, sehingga label yang dihasilkan tidak hanya berbasis pola statistik, tetapi juga mempertimbangkan aspek semantik dan konteks bahasa secara mendalam.

3.4 Balanced Dataset

3.4.1 SMOTE (*Synthetic Minority Oversampling Technique*)

SMOTE merupakan salah satu metode *oversampling* sintetis yang banyak digunakan untuk mengatasi permasalahan ketidakseimbangan kelas pada proses pembelajaran mesin. Pendekatan ini tidak hanya menduplikasi data kelas minoritas, tetapi juga membangkitkan data baru secara sintetis melalui proses interpolasi. Secara teknis, SMOTE bekerja dengan memilih satu sampel

dari kelas minoritas, kemudian menentukan sejumlah tetangga terdekat menggunakan algoritma K-*Nearest Neighbors* (K-NN). Setelah itu, satu tetangga dipilih secara acak, dan sampel baru dibentuk pada garis penghubung antara titik sampel dan tetangganya tersebut. Proses ini diulang hingga jumlah sampel sintetis yang diinginkan tercapai.



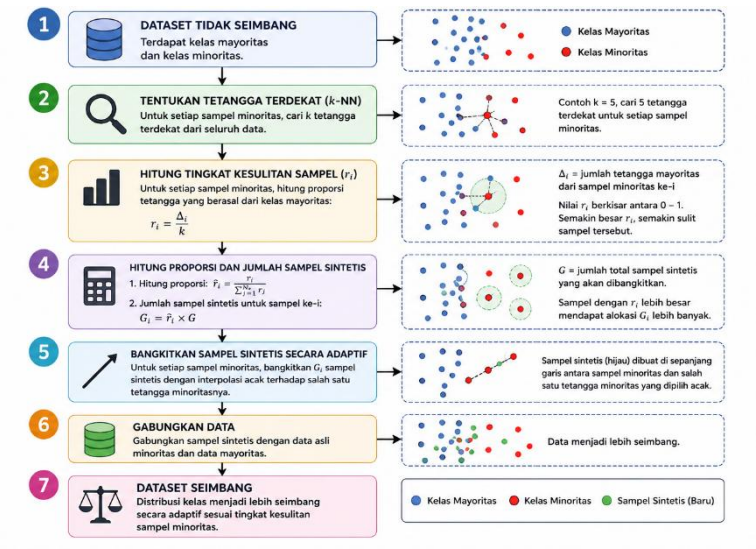
Gambar 3. 3 Alur SMOTE

Dari gambar 3.3 alur SMOTE, distribusi data kelas minoritas menjadi lebih merata dan representatif, sehingga mengurangi bias model terhadap kelas mayoritas. Berdasarkan penelitian Glazkova (2020), penerapan SMOTE pada klasifikasi teks multikelas terbukti mampu meningkatkan performa model klasifikasi, terutama pada algoritma KNN dan SVM, dibandingkan data asli tanpa *oversampling*.

3.4.2 ADASYN (*Adaptive Synthetic Sampling*)

ADASYN merupakan teknik *oversampling* adaptif yang fokus pada pembangkitan data sintetis di sekitar sampel minoritas yang sulit dipelajari oleh

model. ADASYN menghitung tingkat kepadatan lokal dari setiap sampel minoritas dan memberikan bobot lebih besar pada titik-titik dengan kepadatan rendah atau yang dikelilingi oleh banyak sampel mayoritas. Semakin sulit suatu sampel untuk diklasifikasi, semakin banyak data sintetis yang akan dibangkitkan di sekitarnya. Seperti terlihat pada gambar 3.4 Alur Proses ADASYN.



Gambar 3. 4 Alur ADASYN

Pada gambar 3.4. Alur ADASYN, metode ini lebih adaptif dalam memperkuat representasi kelas minoritas yang kompleks, sehingga dapat membantu model dalam belajar secara lebih seimbang. Penelitian yang dilakukan oleh Zakariah et al. (2023) menunjukkan bahwa penerapan ADASYN dalam sistem deteksi intrusi jaringan IoT mampu meningkatkan nilai akurasi, presisi, *recall*, dan F1-*Score* secara signifikan dibandingkan pendekatan tradisional.

Untuk menghindari terjadinya data leakage, proses *oversampling* menggunakan SMOTE dan ADASYN dilakukan setelah tahap pembagian data training dan testing. Teknik balancing hanya diterapkan pada data training, sedangkan data testing dipertahankan dalam kondisi asli tanpa penambahan sampel sintetis. Pendekatan ini bertujuan agar proses evaluasi model tetap objektif dan mampu menggambarkan kemampuan generalisasi model terhadap data yang belum pernah digunakan selama proses pelatihan.

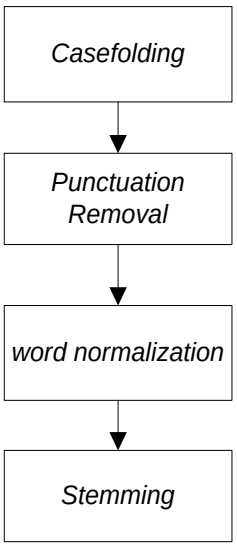
Dalam penelitian ini, Term Frequency-Inverse Document Frequency (TF-IDF) diterapkan sebagai instrumen transformasi fitur numerik untuk memfasilitasi algoritma Synthetic Minority Over-sampling Technique (SMOTE) dalam menangani ketidakseimbangan data teks. Hal ini dilakukan karena SMOTE bekerja berdasarkan kalkulasi jarak Euclidean antar tetangga terdekat (k-Nearest Neighbors), sebuah operasi matematis yang tidak dapat dilakukan langsung pada data teks mentah yang bersifat non-numerik. Melalui representasi vektor TF-IDF, distribusi spasial kelas minoritas dapat dipetakan secara matematis untuk menentukan target jumlah sampel sintetis yang akurat guna mencapai keseimbangan kelas. Berbeda dengan pendekatan SMOTE standar yang menghasilkan vektor numerik abstrak dan tidak dapat dikonversi kembali menjadi teks natural, prosedur dalam penelitian ini hanya menggunakan kuota perhitungan SMOTE sebagai dasar untuk menduplikasi teks asli secara acak. Strategi ini dipilih karena lebih terstruktur dibandingkan Random Over-Sampling (ROS) konvensional yang tidak mempertimbangkan bobot fitur.

3.5 Design System

3.5.1 Preprocessing

Dari pengumpulan data, data yang didapatkan masih belum terstruktur dan tidak konsisten. Tahap *Preprocessing* akan membuang komponen-komponen pada data yang dianggap tidak memiliki makna penting dan membuat data menjadi semakin konsisten. Tujuannya agar data sesuai dengan format untuk pengolahan di tahapan berikutnya.

Preprocessing memiliki banyak macam jenis. Hal ini kemudian yang disesuaikan dengan kebutuhan dari dataset yang digunakan. Untuk setiap penelitian bisa memiliki proses *Preprocessing* yang berbeda. Pada penelitian ini *Preprocessing* yang dilakukan adalah *case folding*, *punctuation removal*, *word normalization* dan *stemming*. Untuk *punctuation removal* meliputi menghapus tag HTML, menghapus tanda baca, menghapus *NON-ASCII* dan menghapus spasi berlebih. Alur *Preprocessing* yang dilakukan pada penelitian ini digambarkan pada Gambar 3.5 Alur *Preprocessing.*



Gambar 3. 5 Alur *Preprocessing*

a. *Case folding*

Casefolding merupakan tahap penyamakan seluruh huruf pada data menjadi huruf besar (*uppercase*) atau kecil (*lowercase*). Pada penelitian ini, seluruh huruf akan diubah ke dalam huruf kecil (*lowercase*). Pada proses ini, yang diperbolehkan hanyalah “a” hingga “z”. Contoh penerapan *casefolding* dapat dilihat pada Tabel 3.1

Tabel 3. 1 Contoh penerapan *casefolding*

Sebelum	Sesudah
Krg cocok.. tidak ada hasil.. bukan jelek krn yg beli byk.. bahan ok alami. produk lain ditoko ini aku beli bagus cck cm yg ini krg. Tyt tiap org beda2 ya ada yg cck ada yg tdk!. ☺ ☺	krg cocok.. tidak ada hasil.. bukan jelek krn yg beli byk.. bahan ok alami. produk lain ditoko ini aku beli bagus cck cm yg ini krg. tyt tiap org beda2 ya ada yg cck ada yg tdk!. ☺ ☺

b. *Punctuation Removal*

Punctuation removal dalam penelitian ini meliputi menghapus tag HTML, menghapus tanda baca, menghapus *non-ascii* dan menghapus spasi berlebih.

1. Menghapus Tag HTML

Pada saat dataset dikumpulkan dengan teknik scraping, terdapat tag HTML yang terdeteksi pada beberapa data. Contoh tag HTML yang ada pada yaitu
 yang melambangkan adanya jeda baris antar kalimat. Tag ini perlu dihapus karena tidak memiliki makna yang penting pada data. Contoh penerapan penghapusan tag HTML dapat dilihat pada Tabel 3.2

Tabel 3. 2 Contoh menghapus tag HTML

Sebelum	**Sesudah**
krg cocok.. tidak ada hasil.. bukan jelek krn yg beli byk.. bahan ok alami. produk lain ditoko ini aku beli bagus cck cm yg ini krg. tyt tiap org beda2 ya ada yg cck ada yg tdk!. ☺ ☺	krg cocok.. tidak ada hasil.. bukan jelek krn yg beli byk.. bahan ok alami. produk lain ditoko ini aku beli bagus cck cm yg ini krg. tyt tiap org beda2 ya ada yg cck ada yg tdk!. ☺ ☺

2. Menghapus Tanda Baca

Dalam tahap ini, segala tanda baca seperti !@#$% akan dihilangkan. Tanda baca tersebut tidak memiliki makna apapun sehingga perlu dihilangkan agar model lebih dapat mudah mempelajari data. Contoh penerapan tanda baca dapat dilhat pada Tabel 3.3

Tabel 3. 3 Contoh menghapus tanda baca

Sebelum	**Sesudah**
krg cocok.. tidak ada hasil.. bukan jelek krn yg beli byk.. bahan ok alami. produk lain ditoko ini aku beli bagus cck cm yg ini krg. tyt tiap org beda2 ya ada yg cck ada yg tdk!. ☺ ☺	krg cocok.. tidak ada hasil.. bukan jelek krn yg beli byk.. bahan ok alami. produk lain ditoko ini aku beli bagus cck cm yg ini krg. tyt tiap org beda2 ya ada yg cck ada yg tdk. ☺ ☺

3. Menghapus NON-*ASCII*

Tahap ini akan menghapus seluruh karakter yang bukan merupan karakter *ascii*. Karater seperti emoji yang ikut terambil pada saat pengumpulan data akan terhapus. Untuk contoh penerapan penghapusan karakter *NON-ASCII* dapat dilihat pada Tabel 3.4

Tabel 3. 4 Contoh menghapus *NON-ASCII*

Sebelum	**Sesudah**
krg cocok.. tidak ada hasil.. bukan jelek krn yg beli byk.. bahan ok alami. produk lain ditoko ini aku beli bagus cck cm yg ini krg. tyt tiap org beda2 ya ada yg cck ada yg tdk. ☺ ☺	krg cocok.. tidak ada hasil.. bukan jelek krn yg beli byk.. bahan ok alami. produk lain ditoko ini aku beli bagus cck cm yg ini krg. tyt tiap org beda2 ya ada yg cck ada yg tdk.

4. Menghapus Spasi Berlebih

Tahap ini melakukan penghapusan spasi berlebih yang ada dalam data. Tabel 3.5 merupakann contoh penerapan menghapus spasi berlebih.

Tabel 3. 5 Contoh menghapus spasi berlebih

Sebelum	Sesudah
krg cocok.. tidak ada hasil.. bukan jelek krn yg beli byk.. bahan ok alami. produk lain ditoko ini aku beli bagus cck cm yg ini krg. tyt tiap org beda2 ya ada yg cck ada yg tdk.	krg cocok.. tidak ada hasil.. bukan jelek krn yg beli byk.. bahan ok alami. produk lain ditoko ini aku beli bagus cck cm yg ini krg. tyt tiap org beda2 ya ada yg cck ada yg tdk.

c. *Word Normalization*

Word normalization dilakukan untuk menormalisasikan kata yang tidak baku menjadi kata baku berdasarkan *vocabulary* yang ada. Untuk contoh penerapan *word normalization* dapat dilihat pada Tabel 3.6.

Tabel 3. 6 Contoh *word normalization*

Sebelum	Sesudah
krg cocok.. tidak ada hasil.. bukan jelek krn yg beli byk.. bahan ok alami. produk lain ditoko ini aku beli bagus cck cm yg ini krg. tyt tiap org beda2 ya ada yg cck ada yg tdk.	kurang cocok.. tidak ada hasil.. bukan jelek karena yang beli banyak.. bahan ok alami. produk lain ditoko ini aku beli bagus cocok cuma yang ini kurang. ternyata tiap orang beda-beda ya ada yang cocok ada yang tidak.

Pada tabel 3.7 merupakan contoh beberapa kata informal beserta kata formalnya yang digunakan untuk mengubah data teks agar menjadi lebih mudah diolah pada tahap selanjutnya.

Tabel 3. 7 Contoh *vocabulary word normalization*

Tidak Baku	Baku
Krng	kurang
Krn	karena
Byk	banyak
Cck	cocok
Cm	cuma
Tyt	ternyata
Tdk	tidak

d. *Stemming*

Stemming dilakukan untuk menghilangkan imbuhan, awalan, dan akhiran dari sebuah kata. Selanjutnya setiap kata hanya mengandung kata dasar. Pada penelitian ini, untuk *stemming* menggunakan library Sastrawi. Untuk contoh penerapan *stemming* dapat dilihat pada tabel 3.8

Tabel 3. 8 Contoh *stemming*

Sebelum	**Sesudah**
kurang cocok.. tidak ada hasil.. bukan jelek karena yang beli banyak.. bahan ok alami. produk lain ditoko ini aku beli bagus cocok cuma yang ini kurang. ternyata tiap orang beda-beda ya ada yang cocok ada yang tidak.	kurang cocok.. tidak ada hasil.. bukan jelek karena beli banyak.. bahan ok alami. produk lain toko ini aku beli bagus cocok ini kurang. nyata tiap orang beda ya ada cocok ada tidak.

3.5.2 *Feature extraction* dengan *Word embedding*

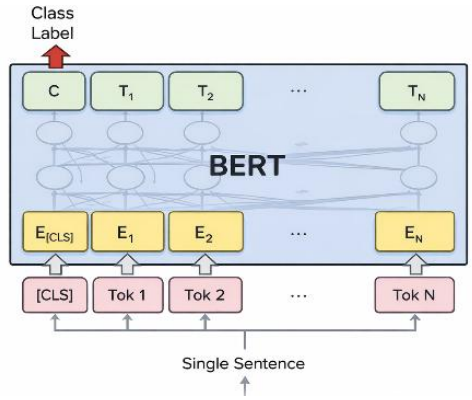
Pada tahap ini dilakukan *word embedding*. *Word embedding* merupakan metode yang digunakan untuk membuat sebuah vektor dengan cara merubah representasi kata yang memungkinkan kata–kata dengan arti yang sama untuk memiliki representasi yang serupa. Pada penelitian ini untuk *word embedding* menggunakan IndoBert dan Word2Vec.

a. IndoBert

Model *pre-trained* BERT pada penelitian ini yang pertama menggunakan model IndoBERT. IndoBert ini dilatih dari dataset indo4B. Dataset indo4B merupakan dataset bahasa Indonesia yang didapat dari sumber publik seperti sosial media, blog, berita dan *website*.

BERT dilatih dengan dataset yang tidak memiliki label. Pada tahap ini, BERT dilatih untuk menggunakan dua tugas yang diawasi sendiri, yaitu

Masked LM (*Masked Language Model*) dan *Next Sentence Prediction*. BERT akan belajar untuk memahami suatu bahasa dan konteksnya. Untuk IndoBERT, pada tahap pra-pelatihan, ia dilatih menggunakan dataset Indo4B. pada dasarnya model BERT hanya akan menggunakan token keluaran [C] untuk memprediksi label data. Ilustrasi untuk tugas klasifikasi kalimat tunggal dapat dilihat pada Gambar 3.6 Arsitektur BERT.



Gambar 3. 6 Arsitektur BERT

Pada Gambar 3.6 tersebut menjekaskan arsitektur BERT (*Bidirectional Encoder Representations from Transformers*) yang diterapkan pada tugas klasifikasi satu kalimat. Proses dimulai dengan pemberian token khusus [CLS] pada awal urutan teks, yang diikuti oleh token-token kata hasil tokenisasi. Setiap token kemudian direpresentasikan dalam bentuk vektor *embedding*, yang merupakan kombinasi dari token *embedding*, segment *embedding*, dan positional *embedding*, sehingga mampu merepresentasikan informasi leksikal, segmentasi, dan posisi token dalam kalimat.

Pada model IndoBERT, setiap token hasil tokenisasi direpresentasikan dalam bentuk vektor *embedding* berdimensi 768. Vektor tersebut merupakan

representasi numerik yang diperoleh dari proses pembelajaran Transformer dan mengandung informasi semantik, sintaksis, serta konteks penggunaan kata dalam suatu kalimat. Setiap dimensi pada vektor tidak memiliki arti tunggal yang dapat diinterpretasikan secara langsung, melainkan secara bersama-sama membentuk representasi makna suatu token. Dengan mekanisme self-attention, setiap token dapat memahami hubungan dengan token lain dalam kalimat sehingga menghasilkan representasi kontekstual yang lebih kaya dibandingkan metode *embedding* tradisional. Sebagai contoh, kata yang sama dapat memiliki representasi vektor yang berbeda apabila digunakan pada konteks kalimat yang berbeda.

Output IndoBERT untuk satu ulasan berbentuk matriks *embedding* berukuran n × 768, dengan n menyatakan jumlah token hasil tokenisasi. Setiap baris matriks merepresentasikan satu token, sedangkan 768 kolom merepresentasikan fitur-fitur laten yang menggambarkan karakteristik linguistik dan semantik token tersebut. Matriks *embedding* ini kemudian digunakan sebagai masukan bagi model *Long Short-Term Memory* (LSTM). LSTM memproses setiap vektor token secara berurutan untuk mempelajari hubungan antar kata, dependensi jangka panjang, serta pola sekuensial yang terdapat dalam ulasan. Dengan demikian, kombinasi IndoBERT dan LSTM memungkinkan model memanfaatkan representasi kontekstual yang kuat dari IndoBERT sekaligus kemampuan LSTM dalam menangkap pola urutan kata untuk meningkatkan akurasi deteksi *fake review*.

Seluruh *embedding* token tersebut diproses melalui sejumlah lapisan *encoder* Transformer pada BERT yang bekerja secara bidirectional. Mekanisme self-attention memungkinkan setiap token mempelajari hubungan kontekstual dengan token lain dalam kalimat secara menyeluruh, sehingga menghasilkan representasi kontekstual yang kaya dan informatif. Output dari proses ini berupa vektor representasi untuk setiap token, termasuk representasi khusus token [CLS] yang dilambangkan sebagai vektor C.

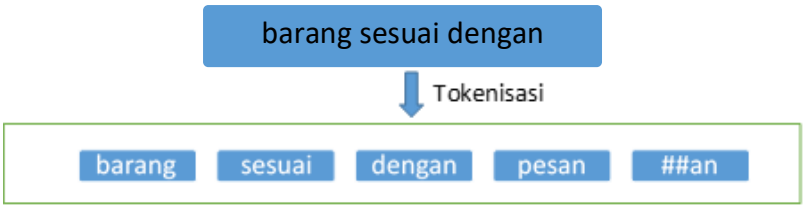
Vektor C berfungsi sebagai representasi global dari keseluruhan kalimat dan digunakan sebagai masukan ke lapisan klasifikasi (*classification layer*) untuk menghasilkan prediksi label kelas. Dengan memanfaatkan pemodelan konteks dua arah dan representasi semantik tingkat tinggi, arsitektur BERT terbukti efektif dalam meningkatkan kinerja berbagai tugas pemrosesan bahasa alami, khususnya pada permasalahan klasifikasi teks.

Proses tokenisasi pada kalimat dengan menggunakan *tokenizer*, dengan langkah-langkah sebagai berikut:

1. Setiap kalimat ditokenisasi menjadi per kata atau sub kata menggunakan *WordPiece*. Untuk melakukan tokenisasi pada sebuah kata, *tokenizer* akan memeriksa apakah tiap kata pada kalimat terdapat pada *vocabulary*. Jika tidak ada, *tokenizer* akan memecah kata menjadi sub-sub kata yang kemungkinan kemunculunnya pada *vocabulary* paling besar. Jika *tokenizer* juga tidak menemukan sub kata pada *vocabulary*, kata tersebut akan dipecah menjadi per karakter. Akan tetapi, jika semua kata diubah menjadi sub kata atau karakter individual, akan terjadi *overload*. Kata–kata

yang tidak ada pada vocabulary akan diganti dengan token [UNK] atau *unknown*. Namun jika semua kata diubah menjadi token tersebut, banyak informasi yang akan hilang. Oleh karena itu, kata–kata dapat dipecah menjadi sub kata dengan simbol ##. BERT melakukan ini karena dua hal yaitu pertama untuk mempercepat *processing* dan mengurangi jumlah *parameters* yang harus dilatih, dan kedua untuk mengatasi masalah *out-of-vocabulary*.

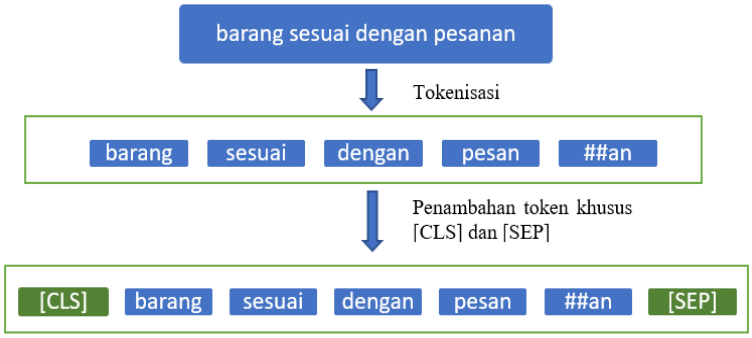
Dalam Gambar 3.7 menunjukkan tahap pertama sebuah kalimat diubah menjadi sebuah token–token kata dan sub kata. Misalkan kalimat *input*-nya adalah "barang sesuai dengan pesanan". Saat dilakukan pengecekan, ternyata kata "pesanan" tidak ada pada *vocabulary*. Sehingga, kata "pesanan" dipecah menjadi sub kata yaitu "pesan" dan "##an" dimana token pertama lebih sering muncul (prefiks) pada *vocabulary* sedangkan token kedua diawali dengan ## untuk menunjukkan bahwa token tersebut adalah sufiks yang mengikuti sub kata lainnya.



Gambar 3. 7 Tahap Tokenisasi

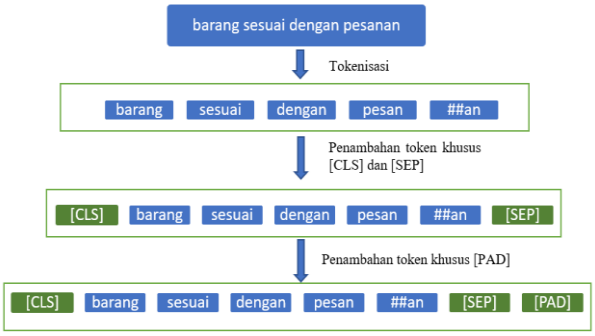
2. Setiap kalimat diberi token – token khusus yaitu [CLS] di awal kalimat dan [SEP] pada akhir kalimat. Token [CLS] menjadi indikator awal sebuah kalimat. Token [SEP] adalah token yang digunakan untuk memisahkan kalimat satu dengan kalimat selanjutnya. Kalimat yang sudah diberi token

khusus ini menjadi *token embeddings*. Gambar 3.8 menunjukkan tahap *token embeddings.*



Gambar 3. 8 Tahap *Token Embeddings*

3. Setelah itu kalimat – kalimat disesuaikan dengan panjang maksimum yang telah ditentukan dengan mengurangi atau memberi *padding* dengan token khusus [PAD]. Gambar 3.9 menujukkan tahap pemberian *padding* dimana panjang maksimum yang ditetapkan adalah 8. Sehingga perlu ditambahkan token [PAD] pada akhir kalimat setelah token [SEP].



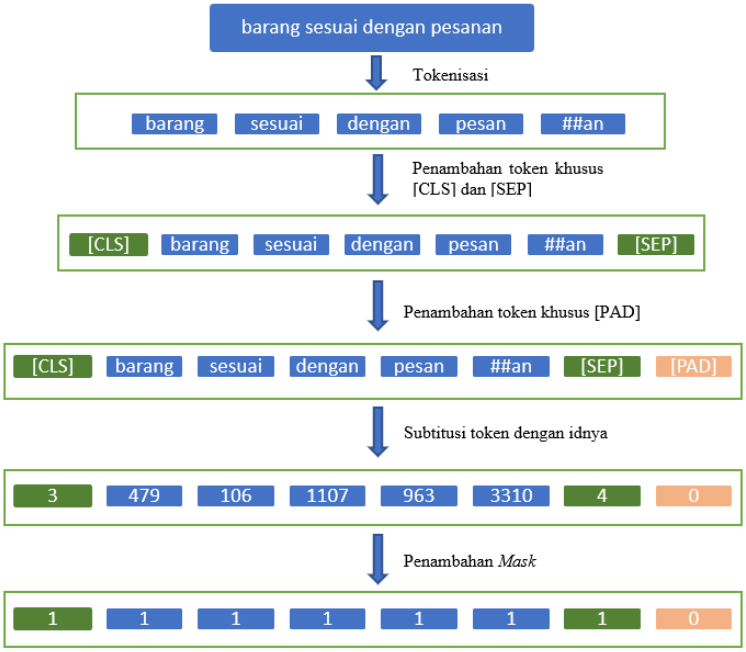
Gambar 3. 9 Tahap Pemberian Padding

4. Kemudian setiap kalimat dicocokkan dengan bilangan unik atau ID sesuai dengan vocabulary dan bilangan unik tersebut disimpan sebagai token id. Bilangan unik atau ID tersebut diperoleh berdasarkan indeks kata pada vocabulary, karena pada vocabulary disusun berdasarkan tingkat kemunculannya. Seperti pada model indobert-base-p1, token [CLS] berada

di indeks 3 dan di ikuti oleh token [SEP] yang berada di indeks 4. Subtitusi token ke id sudah menjadi ketetapan. Sehingga jika terdapat suatu kata seperti kata “barang”, indeks atau id dari kata tersebut adalah 479. Gambar 3.10 menujukkan tahap subtusi id dari setiap token pada kalimat.

Gambar 3. 10 Tahap Subtitusi Token Dengan ID

5. *Attention mask*, yaitu suatu variabel yang digunakan untuk mencegah model memproses *padding* sebagai bagian dari *input*. Variabel *mask* terdiri dari nilai 0 dan 1. Dimana nilai 0 berarti token tidak perlu masuk dalam perhitungan model. Sedangkan nilai 1 berarti token yang perlu diperhitungakan oleh model. Gambar 3.11 menujukkan tahap *attention mask* dari setiap token pada kalimat.



Gambar 3. 11 Tahap *Attention Mask*

b. Word2Vec

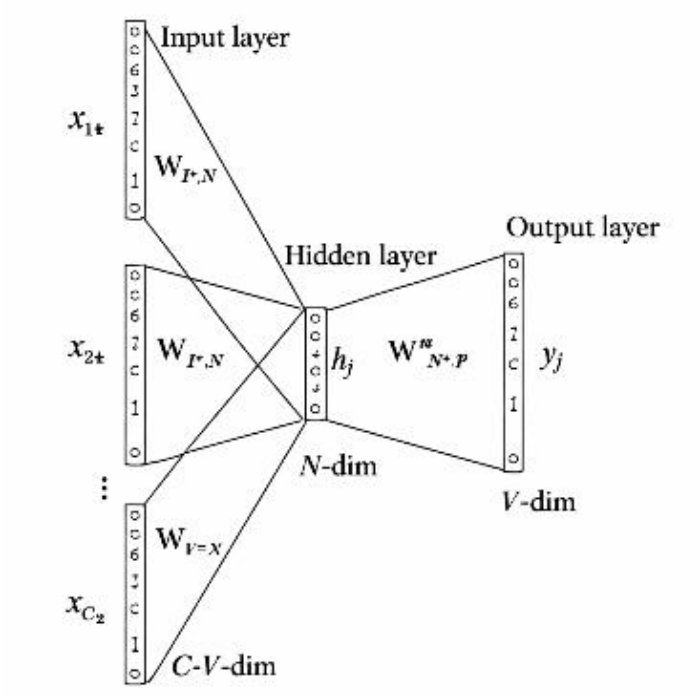
Model Word2vec merupakan model word *embedding* yang mempertimbangkan korpus sebagai input dan output berupa vektor. Word2vec memiliki kemampuan membaca dan mengolah teks dalam ukuran yang besar dan mengubah setiap kata menjadi vektor. Model Word2vec mampu memahami sintaks dan makna semantik kata dari bahasa alami kemudian merepresentasikan setiap kata dengan sebuah vektor. Word2vec mencapai kinerja terbaik dalam NLP dengan mengelompokan kata serupa yang memiliki vektor yang sama.

Neural Network digunakan pada model Word2vec untuk menghasilkan output berupa ruang vektor dari input yang berupa korpus teks. Model Word2vec memiliki dua jenis arsitektur bernama *Continous Bag of Words* (CBOW) dan Skipgram. Arsitektur CBOW memprediksi kata saat ini

berdasarkan konteks, sedangkan arsitektur Skip-Gram memprediksi dalam jangkauan sebelum atau setelah kata sekarang dimana kata sekarang merupakan input, sehingga arsitektur Skip-Gram dianggap sebagai arsitektur yang efisien dalam mempelajari vektor kata yang tidak terstruktur dalam jumlah besar . Terdapat 2 jenis arsitektur neural network dari Word2vec yaitu "*Continuous Bag of Word* (CBOW)" dan "Skip-Gram".

1. *Continous Bag of Word* (CBOW)

Arsitektur Word2vec CBOW adalah kebalikan dari Word2vec skip-gram. Tujuannya adalah untuk memprediksi kata (output) ketika diberikan konteks disekitar kata tersebut (input). Data input Word2vec Skip-gram berbentuk n-hot encoded vector. Ketika proses training berjalan, kata yang sedang menjadi inputan akan bernilai 1 sedangkan kata yang lainnya akan bernilai 0. Karena targetnya hanya 1 kata, maka target output akan berbentuk one-hot encoded vector. Prosesnya tidak jauh berbeda dengan arsitektur Skip-gram. Perbedaannya hanya terletak pada data inputnya yang berupa n-hot encoded vector dan target outputnya berupa one-hot encoded vector. Setelah dilakukan training hingga mencapai eror minimum, maka kita dapat mengambil vektor representasi kata dengan cara mengalikan one-hot encoded vector masing masing kata dengan bobot W.

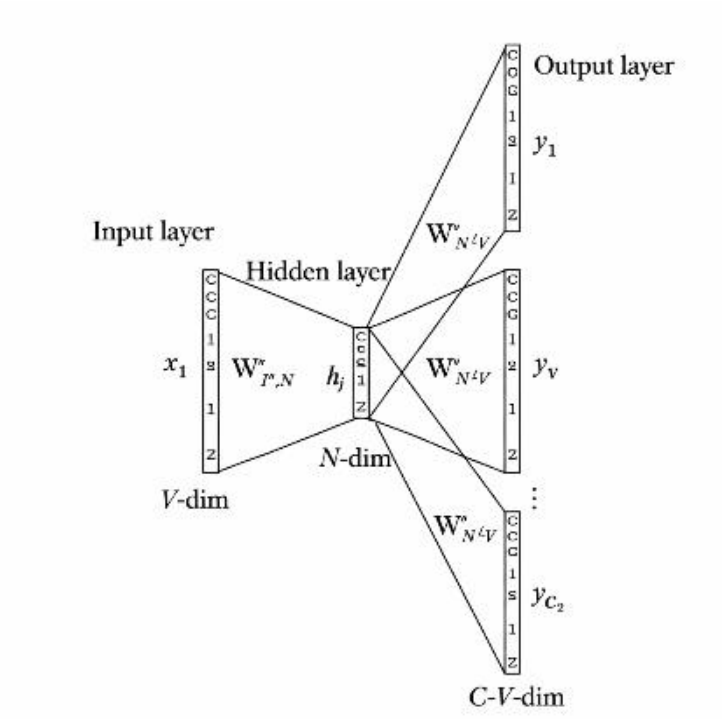


Gambar 3. 12 *Continuous Bag of Word* (CBOW)

2. Skip-Gram

Skip-Gram merupakan model yang menggunakan kata untuk memprediksi target konteks. Tujuan dari arsitektur skip-gram adalah untuk memprediksi konteks (output) di sekitar *current word* (input). Data *input* berbentuk *one-hot encoded vector* sehingga bentuk datanya angka 0 dan 1. Inisialisasi bobot pada W dan W' adalah random. Bobot W dan W' merupakan matrik dengan ukuran W = V x N dan W' = N x V. Pada proses *feedforward*, vektor *input* akan di dot *product* dengan bobot W dan menghasilkan nilai pada layer projection. Kemudian layer projection di dot product dengan bobot W' dan menghasilkan vektor output. Setelah mendapatkan nilai *output* pada tahap *feedforward*, maka akan dihitung nilai eror nya dengan menggunakan metode *cross entropy* yaitu Target - *Output*. Selanjutnya adalah tahap *backpropagation*

dengan memanfaatkan teknik *gradient descent* yaitu dengan melakukan update bobot W dan W'. Proses ini akan diulang kembali ke tahap *feedforward* hingga tercapai nilai eror minimum.



Gambar 3. 13 Arsitektur Skip-Gram

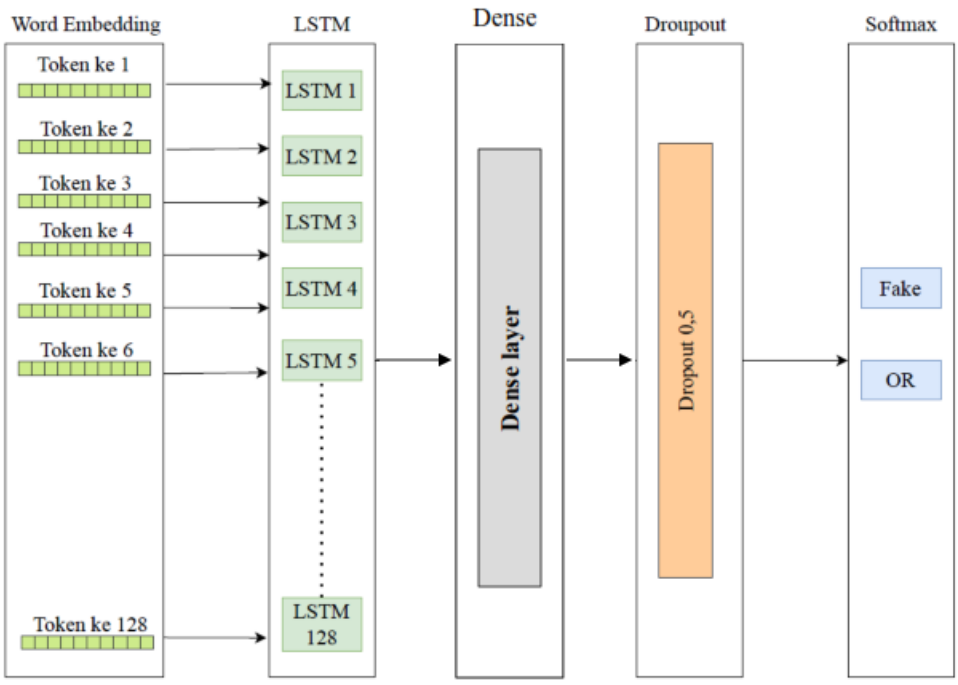
Setelah didapatkan nilai eror minimum pada *cross entropy*, maka vektor yang merepresentasikan kata tersebut diambil dari bobot W dengan cara mengalikan *dot* product antara *one-hot encoded vector* masing-masing kata dengan bobot W, sedangkan bobot pada W' akan diabaikan.

Tujuan dan manfaat Word2vec adalah untuk mengelompokkan vektor dari kata yang mirip di dalam ruang vektor. Word2vec membuat vektor yang didistribusikan representasi numerik dari fitur kata. Dengan menggunakan data yang cukup, Word2vec data memprediksi secara akurat arti kata berdasarkan

riwayat kemunculannya. Prediksi tersebut dapat digunakan untuk menentukan asosiasi sebuah kata dengan kata-kata lainnya yang mirip.

3.5.3 LSTM

Setelah data diproses dengan BERT *embedding*, selanjutnya akan dilakukan proses klasifikasi menggunakan LSTM. Untuk proses model LSTM, pertama hasil BERT *embedding* yang berupa token diproses ke LSTM pertama dengan jumlah layer adalah 128. Setelah itu diproses dense layer selanjutnya diproses dropout sebesar 0,5. Terakhir, dilakukan softmax untuk mengekstraksi fake dan original untuk detail arsitektur LSTM dapat dilihat pada Gambar 3.14.



Gambar 3. 14 Arsitektur LSTM

Pada gambar 3.14 Arsitektur LSTM menjelaskan lapisan *Long Short-Term Memory* (LSTM) berfungsi untuk memodelkan urutan token hasil *word embedding*. Pada proses pembaruan LSTM dikendalikan oleh *forget gate*, input *gate*, dan output *gate*. Berikut adalah contoh simulasi perhitungan LSTM secara manual. Nilai bobot (W, U) dan nilai bias (b) ditentukan secara acak oleh program. Tetapi pada perhitungan manual yang dilakukan ini untuk nilai bobot (W, U) dan nilai bias (b) dapat diasumsikan seperti pada Tabel 3.9 agar memudahkan penjelasan.

Tabel 3. 9 Contoh Nilai Bobot Dan Bias

w_{xf}	w_{hf}	b_f	w_{xi}	w_{hi}	b_i	w_{xc}	w_{hc}	b_c	w_{xo}	w_{ho}	b_o
0,007	0,015	0,002	0,015	0,008	0,003	0,045	0,015	0,002	0,006	0,025	0,001

Selain nilai bobot dan bias seperti Tabel 3.9, juga dibutuhkan nilai h_{t-1} dan c_{t-1}. Pada perhitungan ini, nilai $h_{t-1} = 0$ dan $c_{t-1} = 0$, karena belum ada proses sebelumnya. Input yang akan dilakukan perhitungan ini berupa kalimat "barang sesuai dengan pesanan". Dimana kalimat input telah dilakukan proses *embedding* dengan menggunakan BERT maka didapat nilai vektor sebagai berikut.

$$\begin{bmatrix} 3 \\ 479 \\ 106 \\ 1107 \\ 963 \\ 3310 \\ 4 \end{bmatrix}$$

Matrix tersebut merupakan input X1 hingga X7. Berikut contoh perhitungan dengan metode LSTM dengan nilai X1 = 3.

1. Forget *Gate*s

   Forget *gate* menentukan seberapa besar informasi pada cell state sebelumnya dipertahankan atau dilupakan. Forget *gate* dirumuskan sebagai persamaan 3.1

$$f_t = \sigma(w_{xf} * x_t + w_{hf} * h_{t-1} + b_f) \quad (3.1)$$

$$f_t = \sigma(0{,}007 * 3 + 0{,}015 * 0 + 0{,}002)$$

$$f_t = \sigma(0{,}023)$$

$$f_t = 0{,}50575$$

   Keterangan:

   f_t = *Forget gate*

   σ = Fungsi aktivasi *sigmoid*

   w_{xf} = Bobot *input* pada *forget gate*

   x_t = *Input* ke t

   w_{hf} = *Bobot hidden state* pada *forget gate*

   h_{t-1} = *Output hidden state* sebelumnya

   b_f = *Bias* pada *forget gate*

2. Input *Gate*s

   Input *gate* mengatur informasi baru yang akan disimpan dalam cell state. Input *gate* dihitung menggunakan persamaan 3.2

$$i_t = \sigma(w_{xi} * x_t + w_{hi} * h_{t-1} + b_i) \quad (3.2)$$

$$i_t = \sigma(0{,}015 * 3 + 0{,}008 * 0 + 0{,}003)$$

$$i_t = \sigma(0{,}048)$$

$$i_t = 0{,}51199$$

Kemudian menghitung fungsi aktivasi tanh dengan menggunakan persamaan 3.3. Dimana hasil dari perhitungan tersebut akan digunakan untuk menghitung cell *gate*s.

$$\hat{C}_t = tanh(w_{xc} * x_t + w_{hc} * h_{t-1} + b_c) \quad (3.3.)$$

$$\hat{C}_t = tanh(0{,}045 * 3 + 0{,}015 * 0 + 0{,}002)$$

$$\hat{C}_t = tanh(0{,}137)$$

$$\hat{C}_t = 0{,}13615$$

Keterangan :

i_t = *Input gate*

$\hat{C}_t$ = *Cell* aktivasi

σ = Fungsi aktivasi *sigmoid*

tanh = Fungsi aktivasi *tanh*

w_{xf} = Bobot *input* pada *forget gate*

x_t = *Input* ke t

h_{t-1} = *Output hidden state* sebelumnya

w_{xi} = *Bobot input* pada *input gate*

w_{xc} = *Bobot input* pada *cell* aktivasi

w_{hi} = *Bobot hidden state* pada *input gate*

w_{hc} = *Bobot hidden state* pada *cell* aktivasi

b_i = *Bias* pada *input gate*

b_c = *Bias* pada *cell* aktivasi

3. Cell *Gate*s

   Cell *gate*s diperbarui dengan mengombinasikan informasi lama dan informasi baru. Mekanisme ini memungkinkan LSTM mempertahan informasi penting dalam jangka panjang sehingga permasalahan vanishing gradient dapat diminimalkan. Cell *gate*s dirumuskan pada persamaan 3.4

$$c_t = f_t * c_{t-1} + i_t * \hat{C}_t \quad (3.4)$$

$$c_t = 0{,}50575 * 0 + 0{,}51199 * 0{,}13615$$

$$c_t = 0{,}64814$$

   Keterangan :

   c_t = *cell gates*

   f_t = *forget gate*

   c_{t-1} = *output cell gate* sebelumnya

   i_t = *input gate*

   $\hat{C}_t$ = *cell* aktivasi

4. Output *Gate*s

   Output *gate* menentukan bagian dari cell *gate*s yang akan dikeluarkan sebagai hidden state. Output *gate* dirmuskan pada persamaan 3.5

$$o_t = \sigma(w_{xo} * x_t + w_{ho} * h_{t-1} + b_o) \quad (3.5)$$

$$o_t = \sigma(0{,}006 * 3 + 0{,}025 * 0 + 0{,}001)$$

$$o_t = \sigma(0{,}019)$$

$$o_t = 0{,}50475$$

Kemudian menghitung nilai pada memory cell menggunakan ativasi tanh dan dikalikan dengan hasil o_t sehingga menghasilkan nilai yang akan dikeluarkan. Untuk perhitungan dengan menggunakan permaaan 3.6.

$$h_t = o_t * tanh(c_t) \quad (3.6)$$

$$h_t = 0{,}50475 * tanh(0{,}64814)$$

$$h_t = 0{,}50475 * 0{,}57042$$

$$h_t = 0{,}28792$$

Keterangan :

o_t = *Output gate*

c_t = *Cell gate*

σ = Fungsi aktivasi *sigmoid*

tanh = Fungsi aktivasi *tanh*

w_{xo} = Bobot *input* pada *output gate*

x_t = *Input* ke t

h_{t-1} = *Output hidden state* sebelumnya

w_{ho} = *Bobot hidden state* pada *output gate*

b_o = *Bias* pada *output gate*

Setiap blok LSTM 1 hingga LSTM 128 merepresentasikan proses iteratif perhitungan persamaan di atas untuk setiap token dalam urutan teks. Hidden state terakhir atau agregasi hidden state digunakan sebagai masukan ke lapisan berikutnya (Dropout dan Softmax) untuk keperluan klasifikasi ulasan palsu.

Dengan demikian, LSTM mampu mempelajari pola sekuensial kompleks pada teks ulasan, seperti pengulangan kata, struktur kalimat manipulatif, dan

konteks sentimen, yang menjadi karakteristik utama dalam deteksi ulasan palsu.

3.6 Eksperimen

Perancangan skenario eksperimen atau uji coba dalam penelitian ini dilakukan untuk memperoleh evaluasi performa model yang komprehensif dan objektif, khususnya dalam konteks permasalahan klasifikasi dengan ketidakseimbangan kelas. Oleh karena itu, eksperimen disusun ke dalam enam skenario uji coba yang dirancang secara terkontrol dengan mempertahankan konsistensi pembagian data, serta memvariasikan teknik *oversampling* yang digunakan.

Skenario uji coba dalam penelitian ini dilakukan dengan mengombinasikan dua metode ekstraksi fitur, yaitu IndoBERT dan Word2Vec, dengan tiga teknik penyeimbangan data (*oversampling*), yakni baseline (tanpa penyeimbangan), SMOTE, dan ADASYN. Setiap kombinasi diuji dalam enam skenario berbeda, di mana masing-masing metode *feature extraction* dipasangkan dengan ketiga teknik *oversampling* tersebut. Seluruh skenario menggunakan proporsi pembagian data yang sama, yaitu 80% untuk data pelatihan. Dengan desain eksperimen ini, penelitian bertujuan untuk membandingkan pengaruh teknik *oversampling* terhadap kinerja model pada dua pendekatan representasi teks yang berbeda secara konsisten dan terkontrol. sebagaimana ditunjukkan pada Tabel 3.10. tersebut.

Tabel 3. 10 Skenario Uji Coba

Skenario	Feature Extraction	Teknik *Oversampling*	Pembagian Data
1	IndoBert	BASELINE	80%
2	IndoBert	SMOTE	80%
3	IndoBert	ADASYN	80%
4	Word2Vec	BASELINE	80%
5	Word2Vec	SMOTE	80%
6	Word2Vec	ADASYN	80%

Pembagian data sebesar 80% sebagai data latih dan 20% sebagai data uji dipilih karena rasio tersebut merupakan praktik umum dalam penelitian pembelajaran mesin dan terbukti mampu memberikan keseimbangan antara kecukupan data untuk proses pelatihan dan representativitas data untuk evaluasi. Dengan menggunakan rasio pembagian data yang sama pada seluruh skenario, pengaruh variabel pembagian data dapat dikendalikan, sehingga perbedaan kinerja model yang dihasilkan dapat dikaitkan secara langsung dengan perlakuan eksperimen yang diberikan.

Ketidakseimbangan distribusi kelas pada dataset ulasan berpotensi menyebabkan model cenderung memihak kelas mayoritas. Untuk mengatasi permasalahan tersebut, penelitian ini menerapkan tiga teknik *oversampling*, yaitu *Synthetic Minority Over-sampling Technique* (SMOTE) dan *Adaptive Synthetic Sampling* (ADASYN). SMOTE menghasilkan sampel sintetis dengan pendekatan interpolasi linier antar data minoritas, sedangkan ADASYN secara adaptif memfokuskan pembangkitan sampel sintetis pada data minoritas yang sulit diklasifikasikan. Penggunaan kedua metode ini bertujuan untuk mengevaluasi efektivitas masing-masing teknik dalam meningkatkan kinerja klasifikasi, terutama pada kelas minoritas.

Rencana dari uji coba ini akan dianalisis menggunakan metrik Akurasi, Presisi, *Recall*, dan F1-*Score* untuk setiap kelas. Perbandingan antara kedua skenario akan digunakan untuk menentukan metode *oversampling* yang paling efektif dalam meningkatkan kinerja klasifikasi, khususnya pada kelas minoritas. Interpretasi dari hasil perbandingan ini akan menjadi dasar dalam penentuan model akhir terbaik yang diusulkan dalam penelitian ini.

3.7 Comparation

3.7.1 K-*fold* Cors Validation

K-*Fold* Cross Validation digunakan sebagai metode evaluasi untuk menilai kinerja dan kemampuan generalisasi model klasifikasi berbasis deep learning, yaitu Bidirectional *Encoder* Representations from Transformers (BERT) dan *Long Short-Term Memory* (LSTM), dalam mendeteksi ulasan palsu. Penerapan metode ini bertujuan untuk memastikan bahwa hasil evaluasi model tidak dipengaruhi oleh satu skenario pembagian data latih dan data uji tertentu.

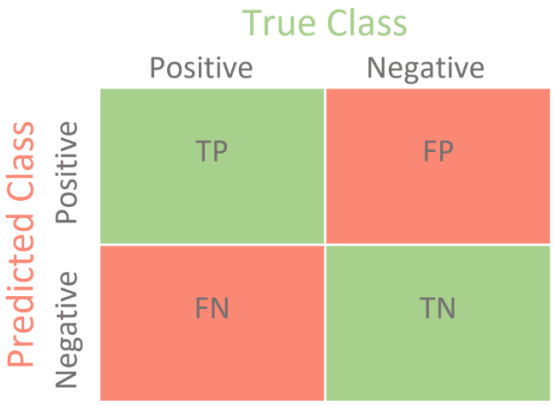
Pada tahap evaluasi, dataset ulasan yang telah melalui proses praproses dibagi ke dalam K-*fold* dengan ukuran yang relatif sama. Pada setiap iterasi, satu *fold* digunakan sebagai data uji (*testing set*), sedangkan K−1 *fold* lainnya digunakan sebagai data latih (*training set*). Model BERT dan LSTM dilatih ulang pada setiap iterasi sesuai dengan skenario eksperimen yang telah ditetapkan, kemudian dilakukan pengujian menggunakan data uji pada *fold* yang bersesuaian.

Kinerja model pada setiap *fold* dievaluasi menggunakan *confusion matrix* untuk memperoleh nilai akurasi, *precision, recall,* dan F1-*Score*. Selanjutnya, nilai rata-rata dari seluruh *fold* digunakan sebagai performa akhir masing-masing model. Dengan pendekatan ini, evaluasi model BERT dan LSTM menjadi lebih konsisten dan representatif, sehingga mampu menggambarkan kemampuan model secara objektif dalam mendeteksi ulasan palsu.

3.7.2 Confusion matrix

Evaluasi merupakan tahapan yang bertujuan untuk mengukur kinerja dari model yang telah dibuat. Pada penelitian ini untuk evaluasi menggunakan *confusion matrix* yang memberikan gambaran rinci mengenai kemampuan model dalam mengklasifikasikan ulasan secara benar maupun salah. Untuk *confusion matrix* dapat dilihat pada Gambar 3.15 Dalam konteks deteksi ulasan palsu, kelas positif merepresentasikan ulasan palsu, sedangkan kelas negatif merepresentasikan ulasan asli. Berdasarkan Gambar 3.15 *confusion matrix* terdiri atas empat komponen utama, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). *True Positive* (TP) adalah jumlah ulasan palsu yang berhasil dideteksi dan diklasifikasikan dengan benar sebagai ulasan palsu oleh model. *True Negative* (TN) merupakan jumlah ulasan asli yang berhasil diprediksi dengan benar sebagai ulasan asli. *False Positive* (FP) adalah ulasan asli yang secara keliru diprediksi sebagai ulasan palsu, sehingga berpotensi menyebabkan kesalahan dalam mengidentifikasi ulasan yang sebenarnya valid. Sementara itu, *False Negative* (FN) merupakan ulasan

palsu yang gagal terdeteksi oleh model dan justru diprediksi sebagai ulasan asli, yang berpotensi menurunkan tingkat keandalan sistem dalam mendeteksi aktivitas manipulatif.



Gambar 3. 15 *Confusion matrix*

Berdasarkan nilai *confusion matrix* dapat diperoleh nilai *Accuracy*, *Precision*, *Recall*, dan F1-*Score*. *Accuracy* menunjukkan persentase keseluruhan ulasan yang berhasil diklasifikasikan dengan benar oleh model terhadap total data uji. Nilai akurasi yang tinggi mengindikasikan bahwa model mampu melakukan klasifikasi secara umum dengan baik, namun metrik ini dapat menjadi kurang representatif pada kondisi data yang tidak seimbang antara ulasan palsu dan ulasan asli.

Precision mengukur tingkat ketepatan model dalam mendeteksi ulasan palsu, yaitu seberapa besar proporsi ulasan yang diprediksi sebagai palsu benar-benar merupakan ulasan palsu. Metrik ini penting untuk meminimalkan kesalahan *False Positive*, karena kesalahan tersebut dapat menyebabkan ulasan asli dianggap sebagai ulasan palsu dan berpotensi merugikan pihak penjual maupun pengguna.

Recall mengukur kemampuan model dalam menemukan seluruh ulasan palsu yang terdapat dalam data uji. Nilai *recal*l yang tinggi menunjukkan bahwa model memiliki kemampuan yang baik dalam mendeteksi sebagian besar ulasan palsu, sehingga dapat menekan jumlah *False Negative*, yaitu ulasan palsu yang lolos dari proses deteksi.

Sementara itu, F1-*Score* merupakan rata-rata harmonis antara Precision dan *Recal*l, yang digunakan untuk memberikan keseimbangan antara tingkat ketepatan dan tingkat cakupan deteksi ulasan palsu, terutama pada kondisi dataset yang bersifat tidak seimbang.

Berikut rumus untuk menghitung *accuracy, precesion, recall* dan F1-*Score* pada pembentukan model klasifikasi ditunjukan pada persamaan (3.7), persamaan (3.8), persamaan (3.9) dan persamaan (3.10).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3.7)$$

$$Precesion = \frac{TP}{TP+FP} \quad (3.8)$$

$$Recall = \frac{TP}{TP+FN} \quad (3.9)$$

$$F1-Score = 2 * \frac{Recall\ x\ Preceision}{Recall+Precision} \quad (3.10)$$

BAB IV

PERSIAPAN DATA DAN *PREPROCESSING*

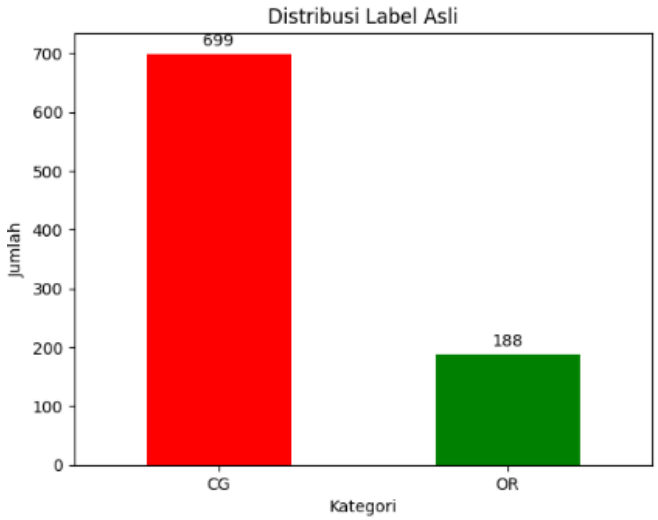
4.1 Persiapan Data

Pada tahap persiapan data, dataset dalam penelitian ini dibagi ke dalam tiga skenario utama, yaitu baseline, SMOTE, dan ADASYN, guna menganalisis pengaruh teknik penyeimbangan data terhadap kinerja model klasifikasi. Skenario baseline menggunakan data asli tanpa perlakuan khusus, sehingga distribusi kelas yang tidak seimbang tetap dipertahankan sebagai acuan awal. Selanjutnya, pada skenario SMOTE (*Synthetic Minority Over-sampling Technique*), dilakukan proses penambahan data sintetis pada kelas minoritas dengan cara menghasilkan sampel baru berdasarkan kedekatan antar data, sehingga distribusi kelas menjadi lebih seimbang. Sementara itu, skenario ADASYN (*Adaptive Synthetic Sampling*) merupakan pengembangan dari SMOTE yang secara adaptif menghasilkan data sintetis dengan mempertimbangkan tingkat kesulitan dalam mempelajari sampel tertentu, sehingga lebih fokus pada area data yang kompleks. Pembagian ini bertujuan untuk membandingkan efektivitas masing-masing pendekatan dalam meningkatkan performa model, khususnya dalam menangani permasalahan ketidakseimbangan data pada proses klasifikasi.

4.1.1 *BASELINE*

Pada penelitian ini, dataset yang digunakan merupakan dataset dari penelitian Habib Alamsyah (2023) yang tersedia secara terbuka pada repositori GitHub. Dataset tersebut berisi ulasan (*review*) berbahasa Indonesia yang berasal dari platform Tokopedia dengan jumlah total ulasan 887 ulasan.

Setelah dilakukan proses pelabelan terhadap seluruh data, diperoleh distribusi kelas yang tidak seimbang antara dua kategori yang digunakan, yaitu CG (Computer Generated/*fake review*) dan OR (Original *Review*/asli) dan diperjelas pada gambar 4.1 tentang distribusi label asli berikut ini :



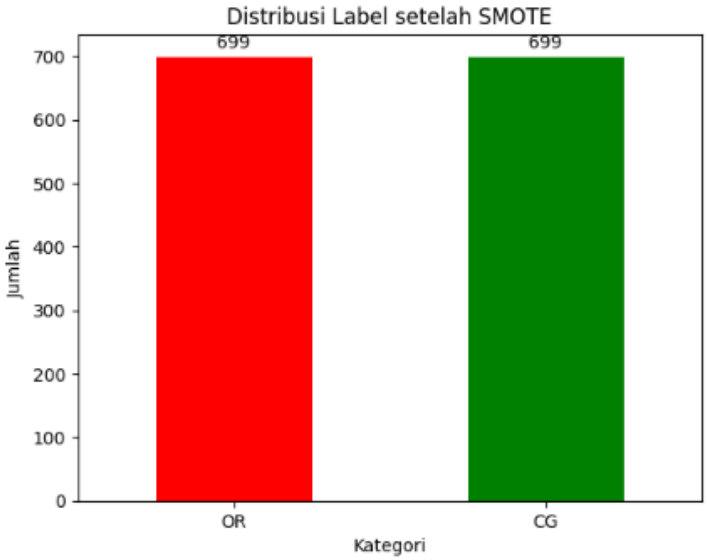
Gambar 4. 1 Distribusi Label Asli

Berdasarkan gambar 4.1 Distribusi label asli tersebut dari total 887 ulasan, sebanyak 699 ulasan diklasifikasikan ke dalam kategori CG, sedangkan 188 ulasan termasuk dalam kategori OR. Hal ini menunjukkan bahwa proporsi data sangat didominasi oleh kelas CG, dengan persentase sekitar 78,8%, dibandingkan kelas OR yang hanya sekitar 21,2%.

Ketidakseimbangan distribusi kelas (*class imbalance*) ini menjadi salah satu tantangan utama dalam proses pengembangan model klasifikasi, karena model cenderung bias terhadap kelas mayoritas. Jika tidak ditangani dengan baik, kondisi ini dapat menyebabkan performa model yang kurang optimal, terutama dalam mendeteksi kelas minoritas (OR). Oleh karena itu, diperlukan strategi penanganan data tidak seimbang, seperti teknik *oversampling* (misalnya SMOTE atau ADASYN), agar model dapat belajar secara lebih seimbang dan menghasilkan performa klasifikasi yang lebih akurat.

4.1.2 SMOTE

Setelah dilakukan penanganan ketidakseimbangan data menggunakan metode SMOTE (*Synthetic Minority Over-sampling Technique*), distribusi kelas pada dataset mengalami perubahan yang signifikan menjadi seimbang dan diperjelas pada gambar 4.2 tentag distirbusi label setelah dilakukan SMOTE berikut ini:



Gambar 4. 2 Distribusi label setelah SMOTE

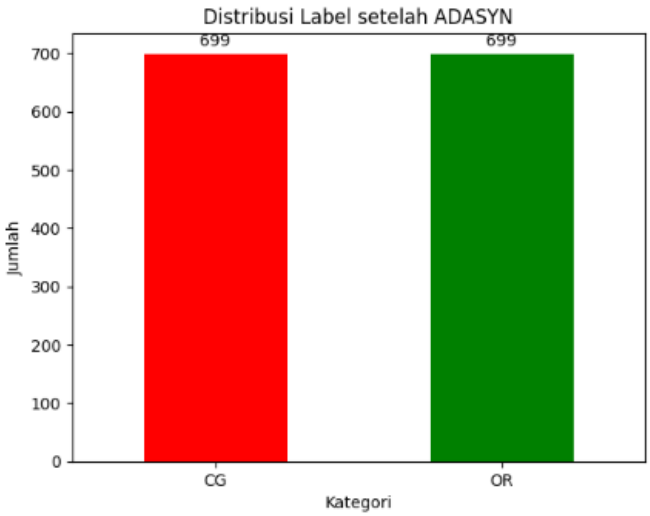
Berdasarkan gambar 4.2 distribusi label setelah SMOTE jumlah data pada masing-masing kelas, yaitu OR (Original *Review*) dan CG (Computer Generated/*fake review*), menjadi sama banyak, yaitu masing-masing 699 ulasan. Dengan demikian, total data setelah proses SMOTE meningkat dari 887 ulasan menjadi 1.398 ulasan.

Proses SMOTE bekerja dengan cara menghasilkan data sintetis pada kelas minoritas (dalam hal ini kelas OR) berdasarkan kemiripan fitur dengan tetangga terdekatnya (nearest neighbors), sehingga tidak hanya menduplikasi data, tetapi menciptakan variasi baru yang tetap representatif. Hal ini bertujuan untuk mengurangi bias model terhadap kelas mayoritas (CG) yang sebelumnya mendominasi dataset.

Dengan distribusi data yang telah seimbang, model klasifikasi diharapkan mampu belajar secara lebih adil terhadap kedua kelas, sehingga meningkatkan kemampuan dalam mendeteksi ulasan asli (OR) maupun ulasan palsu (CG). Selain itu, penerapan SMOTE juga berpotensi meningkatkan nilai evaluasi seperti *recal*l dan F1-*Score* pada kelas minoritas, yang sebelumnya cenderung rendah akibat keterbatasan jumlah data. Namun demikian, perlu diperhatikan bahwa penggunaan data sintetis juga dapat berisiko menimbulkan *overfitting* apabila tidak diimbangi dengan validasi model yang tepat. Oleh karena itu, evaluasi lebih lanjut tetap diperlukan untuk memastikan performa model yang optimal dan generalisasi yang baik.

4.1.3 ADASYN

Setelah dilakukan penanganan ketidakseimbangan data menggunakan metode ADASYN (*Adaptive Synthetic Sampling*), distribusi kelas pada dataset menjadi seimbang antara kedua kategori, yaitu CG (*Computer Generated*/*fake review*) dan OR (*Original Review*) dan diperjelas pada gambar 4.3 tentang distribusi label setelah di lakukan ADASYN berikut ini:



Gambar 4. 3 Distribusi label setelah ADASYN

Berdasarkan gambar 4.3 tentang distribnusi label setelah ADASYN masing-masing kelas memiliki jumlah data yang sama, yaitu 699 ulasan, sehingga total data setelah proses ADASYN menjadi 1.398 ulasan.

Berbeda dengan SMOTE, ADASYN bekerja secara adaptif dengan menghasilkan data sintetis yang lebih difokuskan pada sampel-sampel minoritas yang sulit dipelajari (*hard-to-learn examples*). Dengan kata lain, ADASYN tidak hanya menyeimbangkan jumlah data, tetapi juga memperhatikan tingkat kompleksitas distribusi data, sehingga area yang lebih

sulit dalam ruang fitur akan mendapatkan lebih banyak sampel sintetis. Dalam konteks penelitian ini, kelas OR yang sebelumnya merupakan kelas minoritas diperkuat dengan penambahan data sintetis hingga jumlahnya setara dengan kelas CG.

Hasil distribusi yang seimbang ini diharapkan dapat meningkatkan performa model klasifikasi, khususnya dalam mengenali pola pada kelas minoritas yang sebelumnya kurang terwakili. Selain itu, pendekatan adaptif dari ADASYN memungkinkan model untuk lebih sensitif terhadap variasi data yang kompleks, sehingga berpotensi meningkatkan metrik evaluasi seperti *recal*l dan F1-*Score*. Namun demikian, seperti halnya SMOTE, penggunaan data sintetis dari ADASYN juga perlu diimbangi dengan proses evaluasi yang ketat untuk menghindari risiko overfitting serta memastikan bahwa model tetap mampu melakukan generalisasi dengan baik pada data baru.

4.2 PREPROCESSING

Setelah dilakukan pembagian data ke dalam tiga *oversampling* yaitu baseline, SMOTE, dan ADASYN, tahap selanjutnya adalah melakukan *Preprocessing* pada masing-masing skenario tersebut. Proses ini bertujuan untuk memastikan bahwa seluruh data, baik data asli maupun data hasil *oversampling*, memiliki kualitas yang konsisten, bersih, dan siap digunakan dalam proses pelatihan model.

Pada skenario baseline, *Preprocessing* dilakukan terhadap data asli tanpa adanya penambahan data sintetis. Sementara itu, pada skenario SMOTE dan

ADASYN, *Preprocessing* diterapkan pada data yang telah melalui proses *oversampling*, di mana distribusi kelas telah diseimbangkan melalui penambahan data pada kelas minoritas. Hal ini penting karena data hasil *oversampling* berpotensi memiliki variasi baru yang tetap perlu dinormalisasi agar sesuai dengan standar pengolahan teks.

Tahapan *Preprocessing* yang dilakukan meliputi *casefolding*, yaitu mengubah seluruh teks menjadi huruf kecil untuk menjaga konsistensi; *punctuation removal*, yaitu menghapus tanda baca yang tidak relevan; *word normalization*, yaitu mengubah kata tidak baku menjadi bentuk baku; serta *stemming*, yaitu mengembalikan kata ke bentuk dasarnya dengan menghilangkan imbuhan. Keempat tahapan ini diterapkan secara konsisten pada ketiga skenario data.

Dengan menerapkan *preprocessing* pada data baseline, SMOTE, dan ADASYN, diharapkan seluruh dataset memiliki struktur yang seragam dan bebas dari noise. Hal ini memungkinkan model klasifikasi untuk belajar dari data yang lebih representatif dan seimbang, sehingga dapat meningkatkan performa serta menghasilkan evaluasi yang lebih akurat dalam membandingkan ketiga pendekatan yang digunakan.

4.2.1 BASELINE

a. Casefolding

Pada tahap ini data diproses untuk mengubah semua kata besar/kapital menjadi huruf kecil. *Case folding* membatu dalam menormalisasi teks.

Mengubah semua huruf menjadi huruf kecil membuat teks lebih konsisten. Hal ini membantu dalam perbandingan dan pengelompokan teks yang memiliki kasus berbeda. Pemodelan bahasa atau klasifikasi teks yang membuat semua huruf menjadi kecil dapat membantu mengurangi kosakata dan memfasilitasi pemahaman teks. Seperti terlihat pada tabel 4.1 tentang hasil proses *case folding* pada baseline berikut ini.

Tabel 4. 1 Hasil proses *case folding baseline*

	review	label
0	yang ini belum dicoba	OR
1	pesanan sesuai terima kasih	OR
2	terima kasih atas bonusnya	OR
3	terima kasih	OR
4	respon seller cepat dan ramah, packing rapi, p...	OR
...	... ...	...
882	packing rapi dan cepat sampai, semoga cocok bi...	CG
883	happy repeat order customer ðÿ˜„	CG
884	produknya bagus sekali! reguler deh beli di sini	CG
885	good stuff, recommended seller!!	CG
886	respon cepat dan pengiriman cepat, barang sesuai	CG

b. Punctuation Removal

Punctuation removal merupakan tahap *Preprocessing* yang bertujuan untuk menghapus seluruh tanda baca dari teks ulasan, seperti titik (.), koma (,), tanda seru (!), tanda tanya (?), serta simbol lainnya yang tidak memiliki kontribusi signifikan terhadap proses analisis. Penghapusan tanda baca dilakukan untuk mengurangi noise dalam data dan menyederhanakan struktur teks, sehingga model dapat lebih fokus pada kata-kata yang mengandung informasi penting. Dalam konteks analisis ulasan, tanda baca sering kali tidak memengaruhi makna secara langsung, terutama pada model berbasis pembelajaran mesin yang lebih mengandalkan frekuensi dan pola kata. Dengan

demikian, setelah tahap ini, teks menjadi lebih bersih dan terstruktur, serta mempermudah proses ekstraksi fitur pada tahap selanjutnya. Seperti terlihat pada tabel 4.2 tentang hasil proses *punctuation removal* pada basline berikut ini.

Tabel 4. 2 Hasil Proses *Punctuation Removal Baseline*

	review	label
0	yang ini belum dicoba	OR
1	pesanan sesuai terima kasih	OR
2	terima kasih atas bonusnya	OR
3	terima kasih	OR
4	respon seller cepat dan ramah, packing rapi p...	OR
...	...	...
882	packing rapi dan cepat sampai semoga cocok bi...	CG
883	happy repeat order customer ðÿ˜„	CG
884	produknya bagus sekali reguler deh beli di sini	CG
885	good stuff recommended seller	CG
886	respon cepat dan pengiriman cepat barang sesuai	CG

c. Word Normalization

Word normalization merupakan tahap *Preprocessing* yang bertujuan untuk mengubah kata-kata tidak baku, singkatan, atau variasi penulisan menjadi bentuk yang lebih standar sesuai kaidah bahasa. Proses ini sangat penting dalam pengolahan data ulasan, karena teks yang berasal dari pengguna umumnya mengandung bahasa informal, seperti penggunaan kata singkat, slang, atau kesalahan penulisan. Melalui normalisasi, kata-kata seperti “gk”, “nggak”, atau “ga” akan diseragamkan menjadi “tidak”, sehingga mengurangi variasi kata yang sebenarnya memiliki makna yang sama. Dengan demikian, *word normalization* membantu meningkatkan konsistensi data, memperkecil jumlah kosakata yang tidak perlu, serta mempermudah model dalam mengenali

pola dan hubungan antar kata pada tahap klasifikasi. Speerti terlihat pada tabel 4.3 tentang hasil proses *word normalization* pada baseline berikut ini.

Tabel 4. 3 Hasil Proses Word Normalization Baseline

	review	label
0	yang ini belum dicoba	OR
1	pesanan sesuai terima kasih	OR
2	terima kasih atas bonusnya	OR
3	terima kasih	OR
4	respon seller cepat dan ramah packing rapi pen...	OR
...	...	...
882	packing rapi dan cepat sampai semoga cocok bia...	CG
883	senang ulang order customer ðÿ˜ „	CG
884	produknya bagus sekali reguler deh beli di sini	CG
885	good stuff recommended seller	CG
886	respon cepat dan pengiriman cepat barang sesuai	CG

d. Stemming

Stemming merupakan tahap lanjutan dalam p*reprocessing* yang bertujuan untuk mengubah kata menjadi bentuk dasarnya dengan cara menghilangkan imbuhan, baik awalan, akhiran, maupun sisipan. Proses ini dilakukan untuk menyederhanakan variasi kata yang memiliki makna dasar yang sama, sehingga dapat mengurangi kompleksitas fitur dalam data teks. Dalam konteks ulasan, kata-kata seperti “membeli”, “dibeli”, dan “pembelian” akan direduksi menjadi bentuk dasar yang sama, yaitu “beli”. Dengan demikian, *stemming* membantu model dalam mengenali pola kata secara lebih efektif tanpa terpengaruh oleh variasi bentuk kata. Selain itu, tahap ini juga berkontribusi dalam meningkatkan efisiensi pemrosesan data serta akurasi model klasifikasi karena jumlah kosakata menjadi lebih terkontrol dan representatif. Seperti terlihat tabel 4.4 tentang hasil proses *stemming* pada baseline berikut ini.

Tabel 4. 4 Hasil Proses *Stemming Baseline*

	review	label
0	Coba	OR
1	pesan sesuai terima kasih	OR
2	terima kasih bonus	OR
3	terima kasih	OR
4	respon seller cepat ramah packing rapi kirim c...	OR
...	...	...
882	packing rapi cepat moga cocok biar langgan	CG
883	senang ulang order customer	CG
884	produk bagus reguler beli	CG
885	good stuff recommended seller	CG
886	respon cepat kirim cepat barang sesuai	CG

4.2.2 SMOTE

a. Casefolding

Pada proses *case folding* pada data yang telah melalui teknik *oversampling* SMOTE. Pada tahap ini, seluruh teks ulasan, baik data asli maupun data sintetis hasil SMOTE, diubah menjadi huruf kecil secara keseluruhan. Terlihat bahwa kalimat seperti “Pesanan sesuai terima kasih” dan “Respon seller cepat dan ramah” telah dikonversi menjadi bentuk lowercase tanpa adanya perbedaan penulisan huruf kapital. Proses ini bertujuan untuk menyeragamkan representasi teks sehingga kata yang sama tidak dianggap berbeda hanya karena perbedaan penggunaan huruf besar dan kecil. Penerapan *case folding* pada data hasil SMOTE sangat penting untuk menjaga konsistensi antara data asli dan data sintetis, sehingga model klasifikasi dapat mempelajari pola secara lebih optimal tanpa terganggu oleh variasi format penulisan. Seperti terlihat pada tabel 4.5 tentang hasil proses *case folding* pada SMOTE berikut ini.

Tabel 4. 5 Hasil Proses *casefolding* pada SMOTE

	Review	label
0	yang ini belum dicoba	OR
1	pesanan sesuai terima kasih	OR
2	terima kasih atas bonusnya	OR
3	terima kasih	OR
4	respon seller cepat dan ramah, packing rapi, p...	OR
...	...	...
1393	Puas	OR
1394	ok dicoba belum barang	OR
1395	di hantar hujan basah wraping plastik saat kon...	OR
1396	selanjutnya effect penilaian baru maksimal lah...	OR
1397	pesanan sesuai barang iklan sampai cepat	OR

b. Punctuation Removal

Pada tahap *punctuation removal* pada data yang telah melalui teknik *oversampling* SMOTE. Seluruh tanda baca seperti titik, koma, tanda elipsis, dan simbol lainnya dihilangkan dari teks ulasan, baik pada data asli maupun data sintetis hasil SMOTE. Bahwa kalimat yang sebelumnya mengandung tanda baca kini menjadi lebih bersih dan hanya terdiri dari rangkaian kata, seperti pada bagian "respon seller cepat dan ramah packing rapi …" yang telah disederhanakan tanpa simbol tambahan. Proses ini bertujuan untuk mengurangi noise dalam data serta memastikan bahwa model hanya memproses informasi yang relevan berupa kata-kata. Penerapan *punctuation removal* setelah SMOTE juga penting untuk menjaga konsistensi struktur teks antara data asli dan data hasil *oversampling*, sehingga kualitas data tetap terjaga dan dapat meningkatkan kinerja model klasifikasi. Seperti terlihat pada tabel 4.6 tentang hasil proses punctuation removal pada SMOTE berikut ini.

Tabel 4. 6 Hasil Proses *Casefolding* pada SMOTE

	review	label
0	yang ini belum dicoba	OR
1	pesanan sesuai terima kasih	OR
2	terima kasih atas bonusnya	OR
3	terima kasih	OR
4	respon seller cepat dan ramah, packing rapi, p...	OR
...	...	...
1393	Puas	OR
1394	ok dicoba belum barang	OR
1395	di hantar hujan basah wraping plastik saat kon...	OR
1396	selanjutnya effect penilaian baru maksimal lah...	OR
1397	pesanan sesuai barang iklan sampai cepat	OR

c. Word Normalization

Pada tahap ini, kata-kata dalam ulasan, baik dari data asli maupun data sintetis hasil SMOTE, diseragamkan ke dalam bentuk baku sesuai dengan kaidah bahasa. Terlihat bahwa kata-kata yang sebelumnya berpotensi memiliki variasi penulisan atau bentuk tidak baku telah dinormalisasi menjadi lebih konsisten, seperti penggunaan kata “ok” yang tetap dipertahankan dalam bentuk yang umum digunakan, serta struktur kalimat yang menjadi lebih seragam. Proses ini bertujuan untuk mengurangi keragaman kosakata yang memiliki makna sama namun ditulis dengan cara berbeda, sehingga memudahkan model dalam mengenali pola bahasa. Dengan diterapkannya word normalization pada data hasil SMOTE, kualitas dan konsistensi data menjadi lebih terjaga, yang pada akhirnya dapat meningkatkan efektivitas model dalam proses klasifikasi. Seperti terlihat pada tabel 4.7 tentang hasil proses *Word Normalization* berikut ini.

Tabel 4. 7 Hasil Proses *Word Normalization* pada SMOTE

	review	label
0	yang ini belum dicoba	OR
1	pesanan sesuai terima kasih	OR
2	terima kasih atas bonusnya	OR
3	terima kasih	OR
4	respon seller cepat dan ramah packing rapi pen...	OR
...	...	...
1393	Puas	OR
1394	ok dicoba belum barang	OR
1395	di hantar hujan basah wraping plastik saat kon...	OR
1396	selanjutnya effect penilaian baru maksimal lah...	OR
1397	pesanan sesuai barang iklan sampai cepat	OR

d. Stemming

Selanjutnya proses *stemming* pada data yang telah melalui teknik *oversampling* SMOTE. Pada setiap kata dalam ulasan, baik dari data asli maupun data sintetis hasil SMOTE, diubah ke bentuk dasarnya dengan menghilangkan imbuhan seperti awalan dan akhiran. Terlihat bahwa beberapa kata mengalami penyederhanaan, misalnya “pesanan” menjadi “pesan”, “dihantar” menjadi “hantar”, serta “penilaian” menjadi “nilai”. Proses ini bertujuan untuk mengurangi variasi bentuk kata yang memiliki makna dasar yang sama, sehingga jumlah kosakata menjadi lebih sederhana dan terstruktur. Dengan penerapan *stemming* pada data hasil SMOTE, model klasifikasi dapat lebih mudah mengenali pola kata tanpa terpengaruh oleh perbedaan bentuk morfologis, sehingga berpotensi meningkatkan akurasi dan efisiensi dalam proses pembelajaran. Seperti terlihat pada tabel 4.8 tentang hasil proses *stemming* pada SMOTE berikut ini.

Tabel 4. 8 Hasil Proses *Stemming* pada SMOTE

	review	label
0	Coba	OR
1	pesan sesuai terima kasih	OR
2	terima kasih bonus	OR
3	terima kasih	OR
4	respon seller cepat ramah packing rapi kirim c...	OR
...	...	...
1393	Puas	OR
1394	ok coba barang	OR
1395	hantar hujan basah wraping plastik kondisi	OR
1396	effect nilai maksimal blm	OR
1397	pesan sesuai barang iklan cepat	OR

4.2.3 ADASYN

a. Casefolding

Pada proses *case folding* data yang telah melalui teknik *oversampling* ADASYN. Pada seluruh teks ulasan, baik data asli maupun data sintetis yang dihasilkan oleh ADASYN, diubah menjadi huruf kecil secara menyeluruh. Terlihat bahwa kalimat seperti “Pesanan sesuai terima kasih” dan “Respon seller cepat dan ramah, packing rapi” telah dikonversi menjadi bentuk lowercase tanpa adanya variasi huruf kapital. Selain itu, label seperti CG (*Computer Generated*) dan OR (*Original Review*) tetap dipertahankan sebagai penanda kelas dalam dataset. Proses ini bertujuan untuk menyeragamkan representasi teks sehingga tidak terjadi perbedaan makna akibat penggunaan huruf besar dan kecil. Dengan diterapkannya *case folding* pada data hasil ADASYN, konsistensi antara data asli dan data sintetis dapat terjaga, sehingga model klasifikasi mampu mempelajari pola secara lebih optimal dan akurat. Seperti terlihat pada tabel 4.9 tentang hasil proses *case folding* pada ADASYN berikut ini.

Tabel 4. 9 Hasil Proses *Case folding* pada ADASYN

	review	label
7	yang ini belum dicoba	CG
8	pesanan sesuai terima kasih	CG
9	terima kasih atas bonusnya	CG
10	terima kasih	CG
11	respon seller cepat dan ramah, packing rapi, pen...	CG
...	...	...
1393	responsive kondisi orderan jg baik seller sesu...	OR
1394	di tapi hantar hujan basah wraping maaf plasti...	OR
1395	berkhasiat kurang hujan basah wraping hantar p...	OR
1396	kondisi hantar wraping hujan basah plastik di ...	OR
1397	ngaruh ngga sih ke saya habis botol sdh gak	OR

b. Punctuation removal

Selanjutnya proses *punctuation removal* pada data yang telah melalui teknik *oversampling* ADASYN. Pada seluruh tanda baca seperti koma, titik, serta simbol lainnya dihapus dari teks ulasan, baik pada data asli maupun data sintetis yang dihasilkan oleh ADASYN. Terlihat bahwa kalimat seperti "respon seller cepat dan ramah, packing rapi" telah diubah menjadi rangkaian kata tanpa tanda baca, sehingga teks menjadi lebih sederhana dan bersih. Proses ini bertujuan untuk menghilangkan elemen yang tidak terlalu berpengaruh terhadap makna utama teks, sehingga model dapat lebih fokus pada kata-kata yang memiliki nilai informasi. Dengan diterapkannya punctuation removal pada data hasil ADASYN, konsistensi struktur teks antara data asli dan data sintetis tetap terjaga, sehingga dapat meningkatkan kualitas data serta mendukung performa model dalam proses klasifikasi. Seperti terlihat pada tabel 4.10 tentang hasil proses *punctuation removal* pada ADASYN berikut ini.

Tabel 4. 10 Hasil Proses *Punctuation Removal* pada ADASYN

	review	label
7	yang ini belum dicoba	CG
8	pesanan sesuai terima kasih	CG
9	terima kasih atas bonusnya	CG
10	terima kasih	CG
11	respon seller cepat dan ramah packing rapi pen...	CG
...	...	...
1393	responsive kondisi orderan jg baik seller sesu...	OR
1394	di tapi hantar hujan basah wraping maaf plasti...	OR
1395	berkhasiat kurang hujan basah wraping hantar p...	OR
1396	kondisi hantar wraping hujan basah plastik di ...	OR
1397	ngaruh ngga sih ke saya habis botol sdh gak	OR

c. Word Normalization

Pada proses *word normalization* data yang telah melalui teknik *oversampling* ADASYN. Pada kata-kata dalam ulasan, baik yang berasal dari data asli maupun data sintetis hasil ADASYN, diseragamkan ke dalam bentuk yang lebih baku dan konsisten. Terlihat adanya perbaikan pada kata-kata tidak baku atau singkatan, seperti “jg” yang dinormalisasi menjadi “juga” serta “gak” menjadi “tidak”. Proses ini bertujuan untuk mengurangi variasi penulisan kata yang memiliki makna sama, sehingga kosakata dalam dataset menjadi lebih terstruktur dan mudah dipahami oleh model. Dengan diterapkannya word normalization pada data hasil ADASYN, kualitas data menjadi lebih baik karena informasi yang terkandung dalam teks lebih konsisten, yang pada akhirnya dapat meningkatkan kemampuan model dalam mengenali pola dan melakukan klasifikasi secara lebih akurat. Seperti terlihat pada tabel 4.11 tentang hasil proses *word normalization* pada ADASYN seperti berikut ini.

Tabel 4. 11 Hasil Proses *Word Normalization* pada ADASYN

	review	label
7	yang ini belum dicoba	CG
8	pesanan sesuai terima kasih	CG
9	terima kasih atas bonusnya	CG
10	terima kasih	CG
11	respon seller cepat dan ramah packing rapi pen...	CG
...	...	...
1393	responsif kondisi orderan juga baik seller ses...	OR
1394	di tapi hantar hujan basah wraping maaf plasti...	OR
1395	berkhasiat kurang hujan basah wraping hantar p...	OR
1396	kondisi hantar wraping hujan basah plastik di ...	OR
1397	ngaruh tidak sih ke aku habis botol sdh tidak	OR

d. Stemming

Selanjutnya dilakukan proses *stemming* pada data yang telah melalui teknik *oversampling* ADASYN. Pada setiap kata dalam ulasan, baik dari data asli maupun data sintetis hasil ADASYN, diubah ke bentuk dasarnya dengan menghilangkan imbuhan seperti awalan dan akhiran. Berdasarkan gambar, terlihat bahwa beberapa kata mengalami penyederhanaan, misalnya “pesanan” menjadi “pesan”, “dihantar” menjadi “hantar”, sementara kata seperti “terima kasih” dan “kondisi” tetap dipertahankan sebagai bentuk dasar. Selain itu, kalimat seperti “respon seller cepat ramah packing rapi kirim cepat” menjadi lebih terstruktur tanpa mengubah makna utamanya.

Proses ini bertujuan untuk mengurangi variasi kata yang memiliki makna sama sehingga kosakata menjadi lebih sederhana dan konsisten, terutama pada data hasil ADASYN yang mengandung data sintetis. Dengan *stemming*, model dapat lebih mudah mengenali pola tanpa terpengaruh perbedaan bentuk morfologis, sekaligus mengurangi kompleksitas fitur. Hal ini membantu

meningkatkan efisiensi proses pelatihan serta berpotensi meningkatkan akurasi model dalam mendeteksi ulasan palsu. Seperti terlihat pada tabel 4.12 tentang hasil proses *stemming* pada ADASYN seperti berikut ini.

Tabel 4. 12 Hasil Proses *Stemming* pada ADASYN

	review	label
7	coba	CG
8	pesan sesuai terima kasih	CG
9	terima kasih bonus	CG
10	terima kasih	CG
11	respon seller cepat ramah packing rapi kirim c...	CG
...	...	...
1393	responsif kondisi order seller sesuai barang h...	OR
1394	hantar hujan basah wraping maaf plastik kondisi	OR
1395	khasiat hujan basah wraping hantar plastik kon...	OR
1396	kondisi hantar wraping hujan basah plastik	OR
1397	ngaruh habis botol sdh	OR

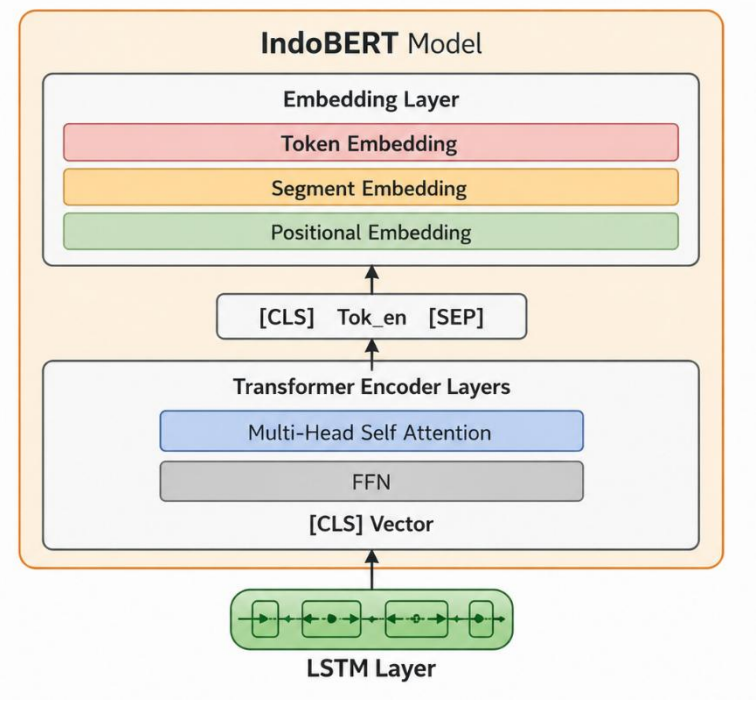
BAB V

METODE INDOBERT DAN LSTM

5.1 Desain

5.1.1 *Feature extraction* dengan IndoBERT

Pada tahap feature extraction, penelitian ini mengimplementasikan model *Bidirectional Encoder Representations from Transformers* (BERT) dengan varian IndoBERT sebagai pendekatan utama dalam merepresentasikan teks ulasan berbahasa Indonesia ke dalam bentuk vektor numerik yang kontekstual. Pemilihan IndoBERT didasarkan pada kemampuannya dalam memahami karakteristik linguistik bahasa Indonesia secara lebih komprehensif dibandingkan model BERT generik, mengingat model ini telah dilatih menggunakan korpus besar berbahasa Indonesia dari berbagai domain.



Gambar 5. 1 Alur *Feature Ectraction* IndoBert

Alur proses *feature extraction* dalam penelitian ini dapat dilihat pada Gambar 5.1 tentang alur feature Ectranction. Arsitektur ini terdiri dari tiga komponen utama, yaitu *Embedding* Layer, Transformer *Encoder* Layers, dan LSTM Layer. Pada tahap awal, teks ulasan yang telah melalui proses tokenisasi direpresentasikan dalam bentuk token dengan format khusus [CLS] token [SEP]. Token [CLS] digunakan sebagai representasi keseluruhan kalimat, sedangkan [SEP] berfungsi sebagai penanda akhir urutan teks. Representasi token tersebut kemudian diproses pada *Embedding* Layer yang terdiri atas Token *Embedding*, Segment *Embedding*, dan Positional *Embedding*.

Pada *Embedding* Layer, IndoBERT menggabungkan tiga jenis representasi, yaitu Token *Embedding*, Segment *Embedding*, dan Positional *Embedding*. Token *Embedding* berfungsi mengubah setiap token menjadi representasi vektor yang mengandung informasi semantik. Segment *Embedding* digunakan untuk membedakan segmen atau kalimat yang diproses oleh model, sedangkan Positional *Embedding* memberikan informasi posisi token dalam urutan teks. Ketiga representasi tersebut dijumlahkan untuk menghasilkan representasi masukan yang kaya akan informasi konteks, makna, dan urutan kata sebelum diproses lebih lanjut oleh *Transformer Encoder*. Pendekatan ini memungkinkan IndoBERT memahami hubungan antar kata secara kontekstual sehingga menghasilkan representasi fitur yang lebih baik untuk tugas klasifikasi teks.

Selanjutnya, representasi vektor dari *Embedding* Layer diteruskan ke Transformer *Encoder* Layers, yang merupakan inti dari arsitektur IndoBERT.

Pada bagian ini terdapat mekanisme Multi-Head Self-Attention yang memungkinkan model memahami hubungan antar kata dalam satu kalimat secara kontekstual. Mekanisme perhatian ini mampu menangkap makna kata berdasarkan kata-kata lain yang muncul di sekitarnya, sehingga dapat mengatasi ambiguitas makna yang sering ditemukan dalam bahasa alami. Setelah proses perhatian dilakukan, hasilnya diteruskan ke *Feed Forward Network* (FFN) untuk menghasilkan representasi fitur yang lebih kaya dan mendalam. Output dari *encoder* kemudian menghasilkan [CLS] Vector, yaitu vektor yang merepresentasikan informasi keseluruhan kalimat atau ulasan.

Pada penelitian deteksi ulasan palsu, [CLS] *Vector* yang dihasilkan oleh IndoBERT digunakan sebagai fitur masukan ke LSTM Layer. Penggunaan LSTM bertujuan untuk mempelajari pola sekuensial dan ketergantungan jangka panjang yang masih terdapat dalam representasi fitur hasil IndoBERT. Kombinasi IndoBERT dan LSTM memungkinkan model tidak hanya memahami konteks semantik dari setiap kata, tetapi juga mampu menangkap pola urutan kata yang menjadi karakteristik ulasan asli maupun ulasan palsu. Dengan demikian, arsitektur IndoBERT-LSTM dapat menghasilkan representasi fitur yang lebih informatif sehingga meningkatkan kemampuan model dalam membedakan kelas ulasan secara lebih akurat.

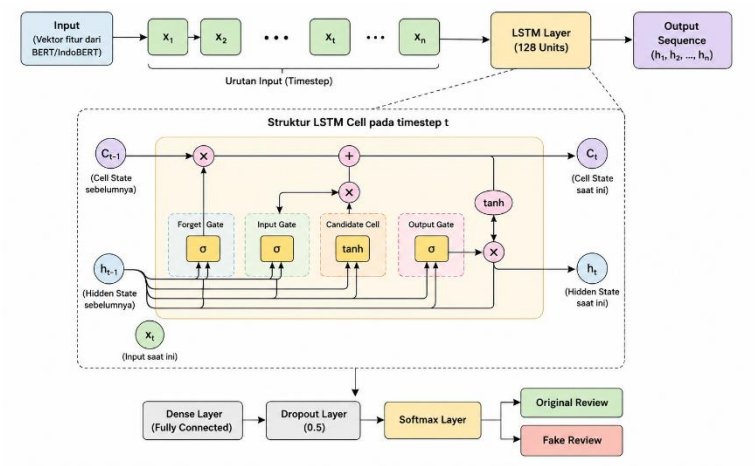
Secara keseluruhan, IndoBERT berfungsi sebagai *feature extractor* berbasis transformer yang menghasilkan representasi kontekstual dari teks bahasa Indonesia, sedangkan LSTM berperan sebagai *classifier sequence learning* yang mengolah representasi tersebut untuk melakukan prediksi kelas.

Pendekatan ini sangat efektif dalam tugas analisis teks karena mampu menggabungkan keunggulan pemahaman konteks global dari transformer dengan kemampuan pemodelan urutan data dari LSTM, sehingga menghasilkan performa klasifikasi yang lebih baik dibandingkan penggunaan metode konvensional.

5.1.2 Klasifikasi dengan Algoritma LSTM

Pada tahap klasifikasi, penelitian ini mengimplementasikan algoritma *Long Short-Term Memory* (LSTM) sebagai model utama untuk mendeteksi ulasan palsu (*fake review*) pada dataset ulasan marketplace. LSTM merupakan pengembangan dari *Recurrent Neural Network* (RNN) yang dirancang untuk mengatasi permasalahan vanishing gradient, sehingga mampu menangkap dependensi jangka panjang dalam data sekuensial seperti teks.

Arsitektur model yang digunakan dalam penelitian ini mengadopsi pendekatan hibrida dengan mengintegrasikan representasi teks berbasis transformer melalui IndoBERT dan kemampuan pemodelan sekuensial dari LSTM. Alur proses klasifikasi sebagaimana ditunjukkan pada Gambar 5.2 menggambarkan tahapan yang terstruktur mulai dari input data hingga proses pengambilan keputusan akhir.



Gambar 5. 2 Alur Algoritma LSTM

Berdasarkan Gambar 5.2 tentang alur kerja *Long Short-Term Memory* (LSTM) dalam melakukan klasifikasi teks. Proses dimulai dari input berupa vektor fitur yang dihasilkan dari tahap sebelumnya, misalnya representasi teks dari BERT atau IndoBERT. Vektor input tersebut kemudian disusun dalam bentuk urutan (sequence) atau timestep (x_1, x_2,...,x_n) dan diproses secara berurutan oleh lapisan LSTM. Pada setiap timestep, LSTM menerima tiga informasi utama, yaitu input saat ini (x_t), hidden state sebelumnya (h_{t-1}), dan cell state sebelumnya (C_{t-1}). Melalui mekanisme ini, LSTM mampu mempertahankan informasi penting dari urutan data yang panjang sehingga dapat mengatasi permasalahan *vanishing gradient* yang sering terjadi pada *Recurrent Neural Network* (RNN) konvensional.

Di dalam setiap sel LSTM terdapat tiga gerbang utama, yaitu *Forget Gate*, *Input Gate*, dan *Output Gate*. *Forget Gate* bertugas menentukan informasi mana dari *cell state* sebelumnya yang perlu dipertahankan atau dihapus. Selanjutnya, *Input Gate* menentukan informasi baru yang relevan untuk

ditambahkan ke memori, sedangkan *Candidate Cell* menghasilkan kandidat informasi baru yang akan disimpan. Hasil dari kedua komponen tersebut digunakan untuk memperbarui *cell state* menjadi (C_t). Setelah itu, *Output Gate* menentukan informasi yang akan dikeluarkan sebagai *hidden state* (h_t), yang menjadi representasi keluaran pada timestep saat ini sekaligus masukan bagi proses pada timestep berikutnya.

Setelah seluruh urutan data diproses, keluaran dari LSTM dengan 128 unit diteruskan ke *Dense Layer* (*Fully Connected Layer*) untuk melakukan transformasi fitur dan meningkatkan kemampuan model dalam mempelajari pola klasifikasi. Selanjutnya, *Dropout Layer* dengan nilai 0,5 diterapkan untuk mengurangi risiko overfitting dengan menonaktifkan sebagian neuron secara acak selama proses pelatihan. Hasil dari *Dropout Layer* kemudian diteruskan ke *Softmax Layer*, yang menghitung probabilitas setiap kelas berdasarkan fitur yang telah dipelajari. Pada tahap akhir, model menghasilkan prediksi berupa dua kategori, yaitu *Original Review* dan *Fake review*.

Secara keseluruhan, arsitektur ini memanfaatkan kemampuan LSTM dalam menangkap hubungan dan ketergantungan antar kata dalam urutan teks, sehingga model dapat mempelajari pola linguistik yang membedakan ulasan asli dan ulasan palsu. Dengan adanya *Dense Layer, Dropout Layer, dan Softmax Layer*, proses klasifikasi menjadi lebih optimal dan mampu menghasilkan prediksi yang lebih akurat terhadap data ulasan yang dianalisis.

5.2 Uji Coba

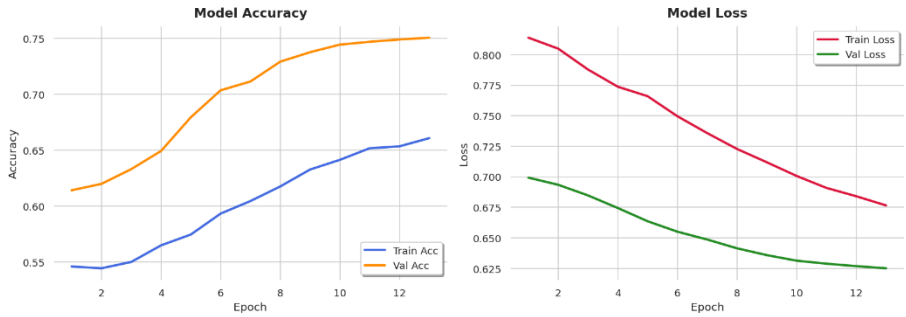
5.2.1 *BASELINE*

Pada tahap eksperimen, dataset yang digunakan terlebih dahulu dibagi menjadi dua bagian utama, yaitu data training dan data testing dengan perbandingan 80:20. Sebanyak 80% data digunakan untuk melatih model agar mampu mempelajari pola dari data ulasan, sedangkan 20% sisanya digunakan sebagai data pengujian untuk mengevaluasi kinerja model terhadap data yang belum pernah dilihat sebelumnya. Pembagian ini bertujuan untuk mengukur kemampuan generalisasi model sehingga hasil evaluasi yang diperoleh lebih objektif dan tidak bias terhadap data pelatihan.

Selanjutnya, untuk meningkatkan keandalan evaluasi model, penelitian ini menerapkan metode K-*Fold Cross Validation* dengan nilai K sebesar 10. Dalam metode ini, dataset dibagi menjadi K bagian yang sama besar, kemudian model dilatih dan diuji sebanyak K kali dengan kombinasi data training dan testing yang berbeda pada setiap iterasi. Pada setiap proses, satu *fold* digunakan sebagai data pengujian, sedangkan *fold* lainnya digunakan sebagai data pelatihan. Hasil evaluasi dari seluruh iterasi kemudian dirata-ratakan untuk memperoleh nilai performa akhir yang lebih stabil dan representatif. Penggunaan K-*Fold Cross Validation* ini diharapkan dapat mengurangi bias akibat pembagian data tunggal serta meningkatkan validitas hasil penelitian.

Selain itu, parameter pelatihan model juga disesuaikan dengan menggunakan jumlah *epoch* sebesar 50% dari *baseline* yang telah ditentukan sebelumnya. *Epoch* merupakan jumlah iterasi penuh dalam proses pelatihan

model terhadap seluruh data training. Pengurangan jumlah *epoch* ini bertujuan untuk mengoptimalkan efisiensi waktu komputasi sekaligus menghindari terjadinya *overfitting*, yaitu kondisi ketika model terlalu menyesuaikan diri dengan data pelatihan sehingga menurunkan kinerjanya pada data baru. Dengan demikian, diharapkan model tetap mampu mencapai performa yang optimal meskipun dengan jumlah *epoch* yang lebih efisien.



Gambar 5. 3 Grafik model *Accuracy* dan model Loss pada LSTM

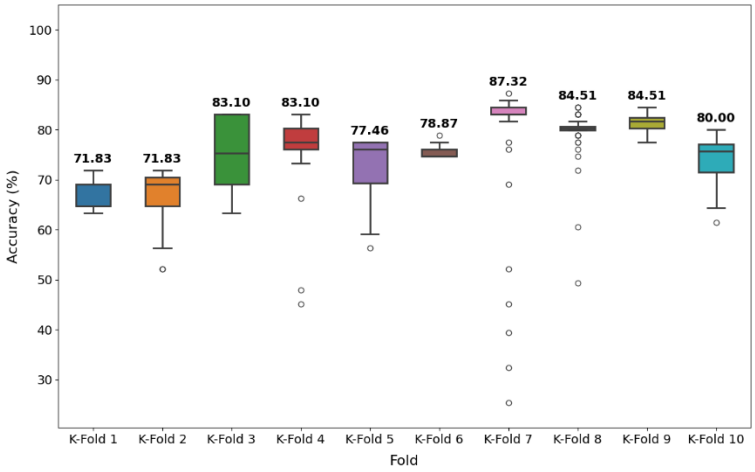
Berdasarkan Gambar 5.3, dapat dianalisis bahwa performa model LSTM menunjukkan peningkatan yang cukup baik selama proses pelatihan. Pada grafik *accuracy*, terlihat bahwa akurasi data training meningkat secara bertahap dari sekitar 0,61 pada *epoch* awal hingga mencapai sekitar 0,75 pada *epoch* terakhir. Peningkatan ini menunjukkan bahwa model semakin mampu mengenali pola dalam data pelatihan. Sementara itu, akurasi data validasi juga mengalami peningkatan dari sekitar 0,54 hingga mencapai sekitar 0,66. Meskipun peningkatannya tidak setinggi data training, tren yang ditunjukkan tetap konsisten dan stabil tanpa fluktuasi yang signifikan.

Pada grafik *loss*, terlihat bahwa nilai *loss* data *training* mengalami penurunan secara konsisten dari sekitar 0,81 pada awal pelatihan menjadi sekitar 0,68 pada akhir *epoch*. Hal ini menunjukkan bahwa kesalahan prediksi

model semakin berkurang seiring dengan proses pembelajaran. Selain itu, nilai loss pada data validasi juga mengalami penurunan dari sekitar 0,70 menjadi sekitar 0,62. Kurva *loss* validasi terlihat lebih landai, yang mengindikasikan bahwa perbaikan performa pada data yang tidak dilatih berlangsung secara stabil meskipun tidak terlalu drastis.

Secara keseluruhan, pola pada grafik *accuracy* dan *loss* menunjukkan bahwa model LSTM mampu mempelajari data dengan cukup baik dan menunjukkan kemampuan generalisasi yang memadai. Hal ini ditunjukkan oleh meningkatnya nilai accuracy serta menurunnya nilai *loss* pada data training maupun validasi. Meskipun terdapat selisih antara kurva training dan validasi, perbedaannya masih dalam batas yang wajar dan tidak mengindikasikan adanya *overfitting* yang signifikan. Dengan akurasi validasi yang mencapai sekitar 66% dan loss validasi sekitar 0,62, model dapat dikatakan cukup stabil, meskipun masih memiliki ruang untuk peningkatan performa lebih lanjut.

Selanjutnya dilakukan perhitungan K-*Fold Cross validation accuracy* untuk meningkatkan keandalan evaluasi model dengan hasil yang ditunjukkan pada gambar 5.4 tentang K-*Fold cross validation accuracy*.



Gambar 5. 4 *K-Fold Cross validation accuracy* LSTM

Berdasarkan Gambar 5.4 yang menampilkan hasil evaluasi menggunakan metode *K-Fold Cross Validation*, performa model LSTM diuji menggunakan 10 fold pengujian (F1–F10) untuk mengukur tingkat konsistensi model dalam melakukan klasifikasi. Setiap fold merepresentasikan satu kali proses pelatihan dan pengujian dengan pembagian data yang berbeda, sehingga hasil evaluasi ini memberikan gambaran yang lebih objektif mengenai kemampuan generalisasi model. Secara umum, nilai akurasi pada masing-masing *fold* menunjukkan variasi performa dalam rentang yang cukup baik, yaitu sekitar 71,83% hingga 87,32%.

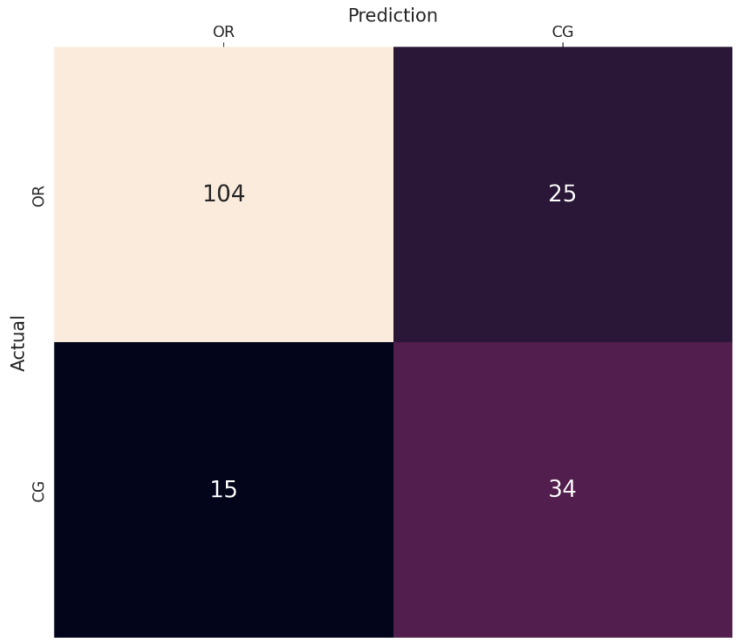
Berdasarkan grafik, akurasi tertinggi diperoleh pada fold F7 sebesar 87,32%, diikuti oleh F8 dan F9 yang masing-masing mencapai 84,51%, serta F4 sebesar 83,10%. Selain itu, F3 juga menunjukkan performa yang relatif baik dengan akurasi 83,10%, sedangkan F6 berada pada kisaran 78,87%. Hasil ini menunjukkan bahwa pada beberapa pembagian data tertentu, model LSTM

mampu mengenali pola dengan cukup baik sehingga menghasilkan tingkat akurasi yang tinggi.

Di sisi lain, terdapat beberapa *fold* dengan performa yang lebih rendah, seperti F1 (71,83%) dan F2 (71,83%), yang menjadi nilai akurasi terendah pada pengujian ini. Selain itu, F5 (77,46%) dan F10 (80,00%) juga menunjukkan akurasi yang lebih rendah dibandingkan *fold* lainnya. Variasi ini kemungkinan disebabkan oleh perbedaan distribusi dan karakteristik data pada masing-masing *fold*, seperti kompleksitas teks atau ketidakseimbangan data pada subset tertentu.

Meskipun terdapat variasi antar *fold*, secara keseluruhan tidak terlihat adanya penurunan performa yang ekstrem. Sebagian besar nilai akurasi berada di atas 78%, yang menunjukkan bahwa model masih mampu mempertahankan kestabilan prediksi pada berbagai pembagian data. Dengan demikian, hasil *K-Fold Cross Validation* ini mengindikasikan bahwa model LSTM memiliki tingkat konsistensi dan kemampuan generalisasi yang cukup baik, meskipun masih terdapat ruang untuk peningkatan performa pada beberapa subset data tertentu.

Langkah selanjutnya yaitu menghitung performance algoritma LSTM untuk *accuracy, precision, recall* dan F1-*Score* dengan menggunakan hasil confuision matrix seperti terlihat pada gambar 5.5 tentang hasil eksperimen LSTM dengan *cofussion matrix*.



Gambar 5. 5 Hasil Tabel Cunfunsion Matrix

Pada gambar 5.5 Hasil tabel *confusion matrix* diatas dapat dijelaskan berdasarkan empat komponen utama yang didapatkan pada masing-masing kolom dengan penjelasan yaitu :

- TP (*True Positive*), sebanyak 104 data original yang terklasifikasi benar.
- TN (True Negatif), sebanyak 34 data fake yang terklasifikasi benar.
- FP (*False Positive*), sebanyak 15 data original yang terklasifikasi salah.
- FN (*False Negative*), sebanyak 25 data fake yang terkasifikasi salah.

Dari tabel confusin matrix yang sudah didapat, selanjutnya adalah melakukan perhitungan *accuracy*, *precission*, *recal* dan f1-*Score*, untuk mengukur sejauh mana keberhasilan model dalam mengkhasifikasi data.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (5.1)$$

$$Accuracy = \frac{104+34}{104+34+15+25} = \frac{138}{178} = 0{,}7752$$

$$Precesion = \frac{TP}{TP+FP} \quad (5.2)$$

$$Precesion = \frac{104}{104+15} = \frac{104}{119} = 0{,}8739$$

$$Recall = \frac{TP}{TP+FN} \quad (5.3)$$

$$Recall = \frac{104}{104+25} = \frac{104}{129} = 0{,}8062$$

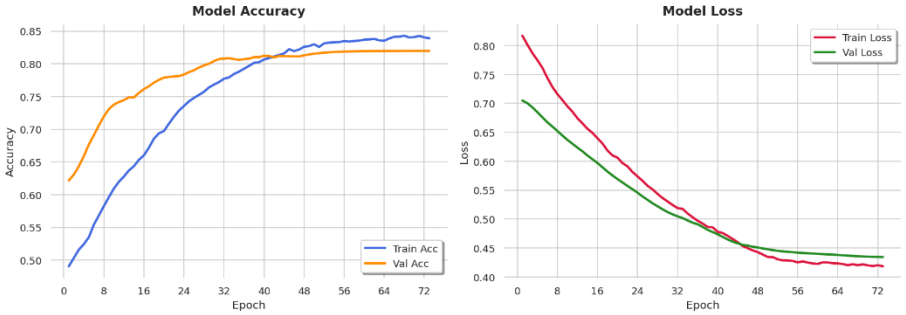
$$F1 - Score = 2 * \frac{Recall\ x\ Preceision}{Recall+Precision} \quad (5.4)$$

$$F1 - Score = 2 * \frac{0{,}8062x\ 0{,}8739}{0{,}8062+0{,}8739} = 2 * \frac{0{,}7045}{1{,}6801} = 0{,}8386$$

5.2.2 SMOTE

Pada tahap eksperimen model *Long Short-Term Memory* (LSTM) dikombinasikan dengan teknik *oversampling* SMOTE (*Synthetic Minority Over-sampling Technique*) untuk mengatasi permasalahan ketidakseimbangan kelas pada dataset. Data penelitian terlebih dahulu dibagi menjadi data pelatihan dan data pengujian dengan perbandingan 80:20, di mana 80% data digunakan untuk melatih model agar mampu mempelajari pola sekuensial dari teks ulasan, sedangkan 20% sisanya digunakan untuk mengevaluasi kinerja model terhadap data yang belum pernah dilihat sebelumnya. Penerapan SMOTE dilakukan pada data pelatihan guna meningkatkan representasi kelas minoritas secara sintetis sehingga model tidak bias terhadap kelas mayoritas.

Pada tahap awal dilakukan pelatihan model *accuracy* dan model yang di tunjukan pada gambar 5.6 tentang grafik model *accuracy* dan model *loss* pada pelatihan SMOTE +BERT+LSTM.



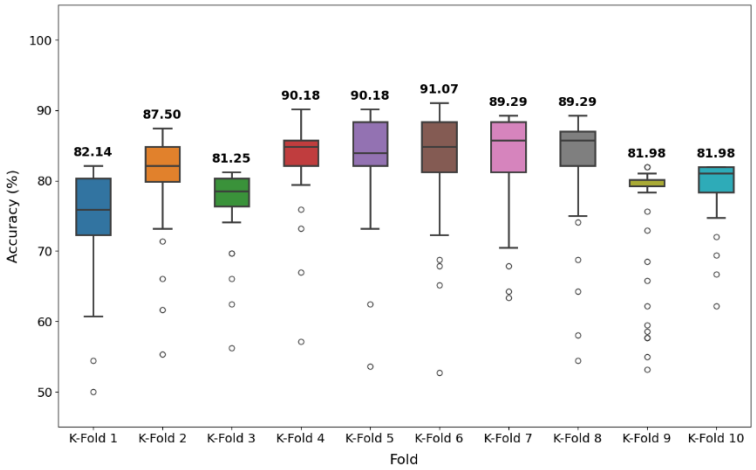
Gambar 5. 6 Grafik model *accuracy* dan model *loss* SMOTE+BERT+LSTM

Berdasarkan Gambar 5.6 terlihat bahwa akurasi model pada data latih (*train accuracy*) mengalami peningkatan yang stabil dari sekitar 0,50 pada *epoch* awal hingga mencapai sekitar 0,84 pada *epoch* akhir. Peningkatan ini menunjukkan bahwa model LSTM secara bertahap mampu mempelajari pola data dengan lebih baik selama proses pelatihan. Sementara itu, akurasi pada data validasi (*validation accuracy*) juga menunjukkan tren peningkatan dari sekitar 0,62 di awal pelatihan hingga berada di kisaran 0,82–0,83 pada *epoch* akhir. Kurva validasi cenderung meningkat dengan relatif stabil dan hanya mengalami fluktuasi kecil.

Jika dilihat dari grafik *loss*, nilai *loss* pada data latih mengalami penurunan yang konsisten dari sekitar 0,85 pada awal pelatihan hingga mendekati 0,42 pada *epoch* akhir, yang menunjukkan bahwa kesalahan prediksi model semakin berkurang. Di sisi lain, *loss* pada data validasi juga mengalami penurunan dari sekitar 0,70 menjadi sekitar 0,43–0,44. Penurunan ini berlangsung cukup stabil, meskipun pada beberapa titik terlihat sedikit pelandaian di akhir *epoch*, yang menandakan bahwa perbaikan performa mulai mencapai kondisi konvergen.

Secara keseluruhan, kombinasi SMOTE, BERT, dan LSTM mampu menghasilkan performa model yang cukup baik, ditunjukkan oleh peningkatan akurasi serta penurunan loss baik pada data latih maupun validasi. Perbedaan antara kurva training dan validasi relatif kecil, sehingga tidak menunjukkan adanya *overfitting* yang signifikan. Hal ini mengindikasikan bahwa model memiliki kemampuan generalisasi yang cukup baik dalam mengklasifikasikan data, meskipun masih terdapat peluang untuk peningkatan performa melalui optimasi parameter atau teknik regularisasi pada penelitian selanjutnya.

Kemudian dilakukan perhitungan K-*Fold Cross validation accuracy* untuk meningkatkan keandalan evaluasi model dengan hasil yang ditunjukkan pada gambar 5.7 tentang K-*Fold cross validation accuracy*.



Gambar 5. 7 K-*Fold Cross validation accuracy* SMOTE+BERT+LSTM

Selanjutnya, berdasarkan Gambar 5.7 hasil visualisasi *K-Fold Cross Validation* menunjukkan bahwa performa model SMOTE + BERT + LSTM cenderung stabil meskipun masih terdapat variasi antar *fold*. Nilai akurasi tertinggi terlihat pada *fold* F6 sebesar 91,07%, diikuti oleh F4 dan F5 yang

masing-masing mencapai 90,18%, serta F8 dan F7 yang berada pada kisaran 89,29%. Hal ini menunjukkan bahwa pada beberapa pembagian data tertentu, model mampu memberikan performa yang sangat baik dan konsisten.

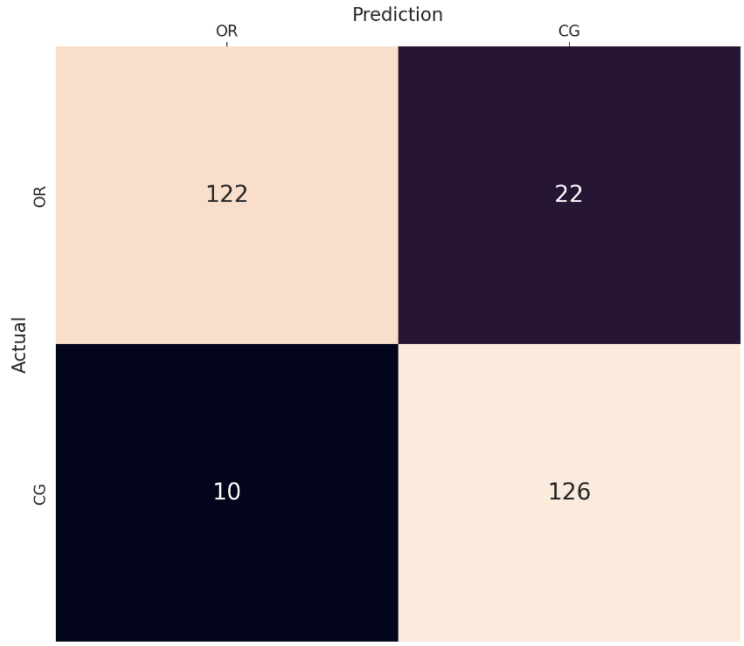
Di sisi lain, nilai akurasi terendah terdapat pada *fold* F3 sebesar 81,25%, disusul oleh F9 dan F10 yang masing-masing sebesar 81,98%, serta F1 sebesar 82,14%. Nilai ini mengindikasikan adanya penurunan performa pada beberapa subset data tertentu, yang kemungkinan disebabkan oleh perbedaan distribusi atau kompleksitas data pada masing-masing *fold*.

Secara umum, sebagian besar nilai akurasi berada pada kisaran di atas 85%, seperti pada F2 (87,50%), F4 (90,18%), F5 (90,18%), F6 (91,07%), F7 (89,29%), dan F8 (89,29%). Hal ini menunjukkan bahwa model memiliki tingkat konsistensi yang cukup baik dalam melakukan klasifikasi pada berbagai pembagian data. Meskipun demikian, masih terdapat beberapa *fold* dengan nilai di bawah 85%, seperti F1, F3, F9, dan F10, yang menandakan bahwa performa model masih dipengaruhi oleh variasi data.

Jika ditinjau dari bentuk boxplot, terlihat bahwa beberapa *fold* seperti F6 dan F7 memiliki rentang distribusi yang lebih lebar, yang menunjukkan variasi performa yang lebih tinggi selama proses validasi. Sebaliknya, *fold* seperti F9 dan F10 memiliki rentang yang relatif lebih sempit, yang mengindikasikan bahwa model menunjukkan performa yang lebih stabil pada pembagian data tersebut. Secara keseluruhan, hasil ini menunjukkan bahwa model cukup robust dan memiliki kemampuan generalisasi yang baik, meskipun masih terdapat

fluktuasi performa pada beberapa fold yang perlu diperhatikan untuk pengembangan selanjutnya.

Langkah selanjutnya yaitu menghitung performance algoritma LSTM untuk *accuracy*, precision, *recal*l dan F1-*Score* dengan menggunakan confuision matrix seperti terlihat pada gambar 5.8 tentang hasil eksperimen SMOTE + BERT + LSTM dengan cofussion matrix.



Gambar 5. 8 Hasil tabel confunsion matrix

Pada gambar 5.8 Hasil tabel *confusion matrix* diatas dapat dijelaskan berdasarkan empat komponen utama yang didapatkan pada masing-masing kolom dengan penjelasan yaitu :

- TP (*True Positive*), sebanyak 122 data original yang terklasifikasi benar.
- TN (True Negatif), sebanyak 126 data fake yang terklasifikasi benar.
- FP (*False Positive*), sebanyak 10 data original yang terklasifikasi salah.
- FN (*False Negative*), sebanyak 22 data fake yang terkasifikasi salah.

Dari tabel confusin matrix yang sudah didapat, selanjutnya adalah melakukan perhitungan *accuracy*, *precission*, *recal* dan f1-*Score*, untuk mengukur sejauh mana keberhasilan model dalam mengkhasifikasi data.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (5.5)$$

$$Accuracy = \frac{122+126}{122+126+10+22} = \frac{248}{280} = 0{,}8857$$

$$Precesion = \frac{TP}{TP+FP} \quad (5.6)$$

$$Precesion = \frac{122}{122+10} = \frac{122}{132} = 0{,}9242$$

$$Recall = \frac{TP}{TP+FN} \quad (5.7)$$

$$Recall = \frac{122}{122+22} = \frac{122}{144} = 0{,}8472$$

$$F1 - Score = 2 * \frac{Recall\ x\ Preceision}{Recall+Precision} \quad (5.8)$$

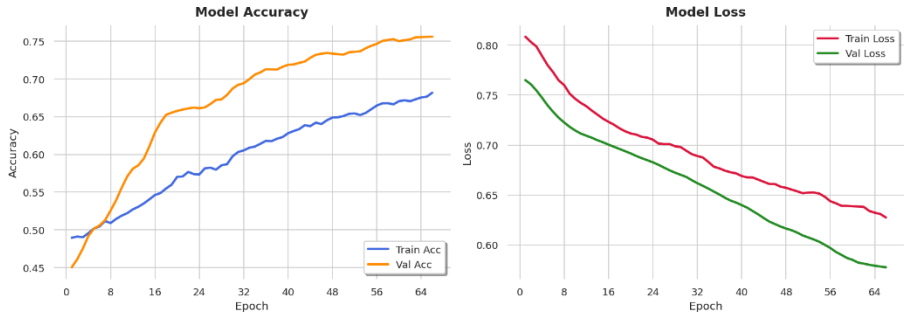
$$F1 - Score = 2 * \frac{0{,}8472x\ 0{,}9242}{0{,}8472 + 0{,}9242} = 2 * \frac{0{,}7830}{1{,}7714} = 0{,}8840$$

5.2.3 ADASYN

Pada tahap eksperimen ini, model *Long Short-Term Memory* (LSTM) dikombinasikan dengan teknik *oversampling* ADASYN (Adaptive Synthetic Sampling) untuk mengatasi permasalahan ketidakseimbangan kelas pada dataset. Data penelitian dibagi menjadi data pelatihan dan data pengujian dengan perbandingan 80:20, di mana 80% data digunakan untuk melatih model agar mampu mempelajari pola sekuensial dari teks ulasan, sedangkan 20% sisanya digunakan untuk mengevaluasi performa model terhadap data yang belum pernah dilihat sebelumnya. Penerapan ADASYN dilakukan pada data pelatihan dengan cara menghasilkan sampel sintetis secara adaptif, terutama

pada data yang sulit dipelajari, sehingga distribusi kelas menjadi lebih seimbang. Selanjutnya, model LSTM dilatih dengan jumlah *epoch* sebanyak 50 untuk mengoptimalkan proses pembelajaran dalam menangkap dependensi jangka panjang pada data teks. Pendekatan ini diharapkan mampu meningkatkan kinerja model dalam melakukan klasifikasi, khususnya dalam membedakan antara ulasan asli dan ulasan palsu secara lebih akurat.

Pada tahap awal dilakukan pelatihan model *accuracy* dan model yang di tunjukan pada gambar 5.9 tentang grafik model accurasy dan model loss pada pelatihan ADASYN + BERT + LSTM.



Gambar 5. 9 Grafik model *accuracy* dan model *loss* ADASYN+BERT+LSTM

Berdasarkan Gambar 5.9, grafik model *accuracy* dan model *loss* menunjukkan bahwa kinerja model mengalami peningkatan selama proses pelatihan. Pada grafik *accuracy*, nilai akurasi data latih (*train accuracy*) meningkat secara bertahap dari sekitar 0,46 pada *epoch* awal hingga mencapai sekitar 0,68 pada *epoch* terakhir. Hal ini menunjukkan bahwa model LSTM yang dikombinasikan dengan BERT serta penyeimbangan data menggunakan ADASYN mampu mempelajari pola pada dataset dengan cukup baik seiring bertambahnya *epoch*.

Sementara itu, akurasi pada data validasi (*validation accuracy*) juga mengalami peningkatan yang cukup signifikan, dimulai dari sekitar 0,47 dan terus meningkat hingga mencapai sekitar 0,76 pada *epoch* terakhir. Meskipun sempat mengalami sedikit fluktuasi pada beberapa *epoch*, tren kenaikan ini menunjukkan bahwa model memiliki kemampuan generalisasi yang cukup baik terhadap data yang belum pernah dilihat. Bahkan, nilai *validation accuracy* yang lebih tinggi dibanding train accuracy mengindikasikan bahwa model tidak mengalami *overfitting* yang berlebihan.

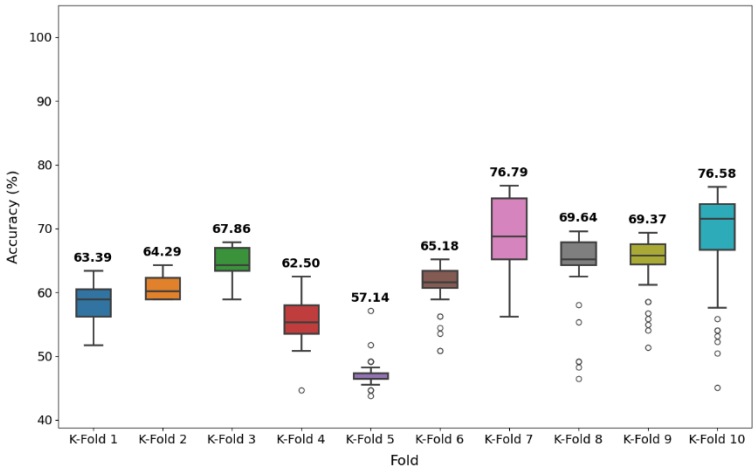
Jika ditinjau dari grafik model *loss*, nilai loss pada data latih (*train loss*) mengalami penurunan secara konsisten dari sekitar 0,81 pada *epoch* awal hingga mencapai sekitar 0,63 pada *epoch* terakhir. Penurunan ini menunjukkan bahwa kesalahan prediksi model pada data pelatihan semakin kecil seiring proses pembelajaran berlangsung.

Di sisi lain, nilai loss pada data validasi (*validation loss*) juga menunjukkan tren penurunan yang stabil, dari sekitar 0,77 hingga mendekati 0,58 pada *epoch* akhir. Nilai *validation loss* yang lebih rendah dibanding train *loss* menunjukkan bahwa model mampu mempertahankan performa yang baik pada data validasi, sehingga mengindikasikan proses pembelajaran berjalan secara optimal.

Secara keseluruhan, hasil eksperimen menunjukkan bahwa penggunaan ADASYN dalam menyeimbangkan data memberikan kontribusi positif terhadap performa model BERT+LSTM. Hal ini terlihat dari peningkatan *accuracy* serta penurunan *loss* yang stabil pada data latih maupun data validasi.

Dengan demikian, model dapat dikatakan memiliki performa yang cukup baik dan mampu melakukan klasifikasi dengan tingkat generalisasi yang optimal.

Kemudian dilakukan perhitungan K-*Fold Cross validation accuracy* untuk meningkatkan keandalan evaluasi model dengan hasil yang ditunjukkan pada gambar 5.10 tentang K-*Fold cross validation accuracy*.



Gambar 5. 10 K-*Fold Cross validation accuracy* ADASYN+BERT+LSTM

Berdasarkan Gambar 5.10, hasil evaluasi performa model menggunakan metode *K-Fold Cross Validation* pada skenario ADASYN + BERT + LSTM menunjukkan adanya variasi akurasi pada setiap fold. Nilai akurasi tertinggi diperoleh pada *fold* ke-7 (F7) sebesar 76,79%, yang menunjukkan bahwa pada pembagian data tersebut model mampu bekerja dengan cukup optimal. Selain itu, *fold* ke-10 (F10) juga menunjukkan performa yang tinggi dengan akurasi sebesar 76,58%. Sebaliknya, nilai akurasi terendah terlihat pada *fold* ke-5 (F5) sebesar 57,14%, serta *fold* ke-4 (F4) yang berada pada kisaran 62,50%, yang mengindikasikan adanya penurunan performa model pada beberapa subset data tertentu.

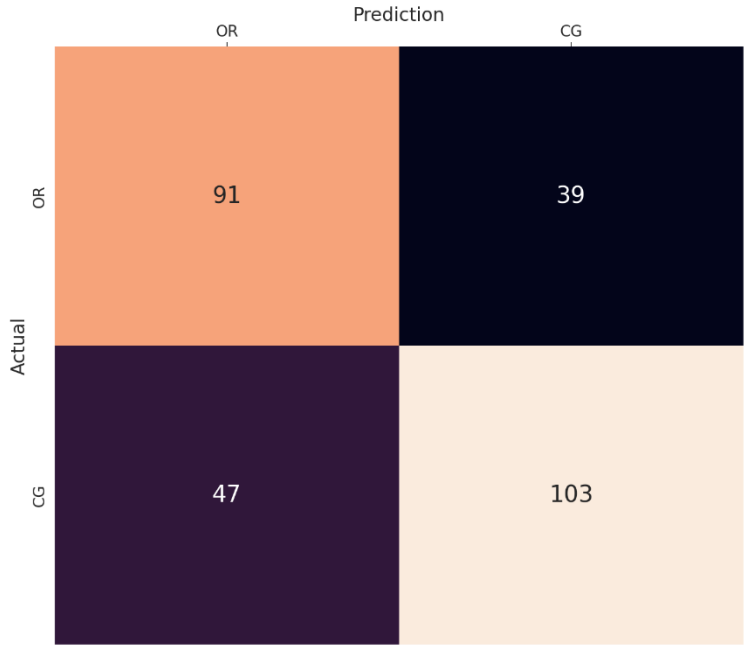
Secara umum, sebagian besar nilai akurasi berada pada rentang 63% hingga 70%, seperti pada F1 (63,39%), F2 (64,29%), F3 (67,86%), F6 (65,18%), F8 (69,64%), dan F9 (69,37%). Hal ini menunjukkan bahwa model memiliki tingkat konsistensi yang cukup baik meskipun masih terdapat fluktuasi antar *fold*. Variasi ini wajar terjadi karena setiap *fold* memiliki distribusi data latih dan data uji yang berbeda, sehingga memengaruhi hasil evaluasi pada masing-masing iterasi.

Jika dilihat dari bentuk *boxplot* pada grafik, beberapa *fold* seperti F7 dan F10 memiliki rentang nilai yang relatif lebih lebar, yang menunjukkan adanya variasi performa yang cukup tinggi selama proses validasi. Sementara itu, F5 memiliki rentang yang lebih sempit namun dengan nilai akurasi yang lebih rendah, yang mengindikasikan bahwa performa model pada *fold* tersebut lebih stabil tetapi kurang optimal dibanding *fold* lainnya. Selain itu, terdapat beberapa outlier pada beberapa *fold* seperti F4, F6, F8, F9, dan F10, yang menunjukkan adanya data tertentu yang memberikan hasil akurasi jauh berbeda dari mayoritas distribusi data.

Dengan demikian, dapat disimpulkan bahwa pendekatan ADASYN + BERT + LSTM mampu memberikan performa yang cukup baik dalam mendeteksi ulasan, dengan rata-rata akurasi yang relatif stabil pada sebagian besar *fold*. Namun, adanya perbedaan performa antar *fold* menunjukkan bahwa model masih cukup sensitif terhadap distribusi data. Oleh karena itu, diperlukan optimasi lebih lanjut seperti hyperparameter tuning, peningkatan

kualitas preprocessing, atau penambahan data untuk meningkatkan stabilitas dan konsistensi performa model.

Langkah selanjutnya yaitu menghitung performance algoritma LSTM untuk *accuracy*, precision, *recal*l dan F1-*Score* dengan menggunakan confuision matrix seperti terlihat pada gambar 5.11 tentang hasil eksperimen ADASYN + BERT+ LSTM dengan *cofussion matrix*.



Gambar 5. 11 Hasil tabel *confunsion matrix*

Pada gambar 5.11 Hasil tabel *confusion matrix* diatas dapat dijelaskan berdasarkan empat komponen utama yang didapatkan pada masing-masing kolom dengan penjelasan yaitu :

- TP (*True Positive*), sebanyak 91 data original yang terklasifikasi benar.
- TN (*True Negatif*), sebanyak 103 data fake yang terklasifikasi benar.
- FP (*False Positive*), sebanyak 47 data original yang terklasifikasi salah.
- FN (*False Negative*), sebanyak 39 data fake yang terkasifikasi salah.

Dari tabel confusin matrix yang sudah didapat, selanjutnya adalah melakukan perhitungan *accuracy*, *precission*, *recal* dan f1-*Score*, untuk mengukur sejauh mana keberhasilan model dalam mengkhasifikasi data.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (5.9)$$

$$Accuracy = \frac{91+103}{91+103+47+39} = \frac{193}{280} = 0{,}6928$$

$$Precesion = \frac{TP}{TP+FP} \quad (5.10)$$

$$Precesion = \frac{91}{91+47} = \frac{91}{138} = 0{,}6594$$

$$Recall = \frac{TP}{TP+FN} \quad (5.11)$$

$$Recall = \frac{91}{91+39} = \frac{91}{130} = 0{,}7$$

$$F1 - Score = 2 * \frac{Recall\ x\ Preceision}{Recall+Precision} \quad (5.12)$$

$$F1 - Score = 2 * \frac{0{,}7x\ 0{,}6594}{0{,}7 + 0{,}6594} = 2 * \frac{0{,}4616}{1{,}3594} = 0,6791$$

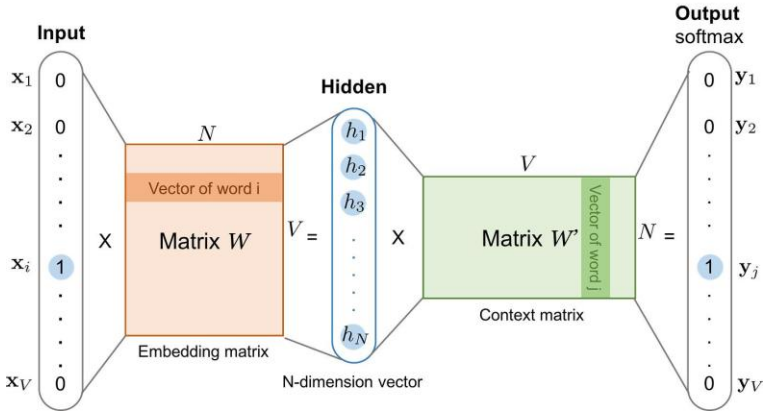
BAB VI

METODE WORD2VEC Dengan LSTM

6.1 Desain

6.1.1 *Feature extraction* dengan Word2Vec

Pada tahap feature extraction, penelitian ini mengimplementasikan model Word2Vec sebagai pendekatan utama dalam merepresentasikan teks ulasan berbahasa Indonesia ke dalam bentuk vektor numerik yang bermakna secara semantik. Pemilihan Word2Vec didasarkan pada kemampuannya dalam memetakan kata-kata yang memiliki konteks serupa ke dalam ruang vektor yang berdekatan, sehingga hubungan antar kata dapat dipelajari dengan baik. Selain itu, Word2Vec dinilai efisien dalam proses pelatihan dan mampu menghasilkan representasi kata yang relevan untuk analisis teks berbahasa Indonesia, terutama pada dataset ulasan dari berbagai domain.



Gambar 6. 1 Alur *Feature Ectraction* dengan Word2Vec

Berdasarkan Gambar 6.1 tentang alur *feature extraction* dengan Word2Vec menggunakan jaringan saraf dangkal (*shallow neural network*)

yang terdiri atas lapisan input, lapisan *embedding* (hidden layer), dan lapisan output. Setelah proses pelatihan selesai, bobot yang dipelajari pada lapisan *embedding* akan menjadi representasi vektor dari setiap kata dalam kosakata.

Pada bagian Input Layer, setiap kata direpresentasikan dalam bentuk one-hot vector dengan ukuran sebesar jumlah kosakata (*Vocabulary Size*, V). Pada gambar, hanya satu elemen bernilai 1, sedangkan elemen lainnya bernilai 0. Posisi nilai 1 menunjukkan kata target yang sedang diproses, yaitu kata ke-i. Representasi one-hot ini kemudian dikalikan dengan *Embedding Matrix* (W) yang berukuran V×N, di mana V adalah jumlah kosakata dan N adalah dimensi vektor *embedding* yang ingin dipelajari. Hasil perkalian tersebut menghasilkan sebuah vektor berdimensi N yang disebut sebagai representasi atau *embedding* dari kata target.

Lapisan tengah atau *Hidden Layer* menghasilkan vektor (h_1, h_2,...,h_N) yang merepresentasikan makna semantik kata target dalam ruang vektor berdimensi rendah. Pada Word2Vec, lapisan ini tidak menggunakan fungsi aktivasi nonlinier sehingga prosesnya hanya berupa transformasi linear. Tujuannya adalah memperoleh representasi kata yang efisien namun tetap mampu menangkap hubungan semantik antar kata. Kata-kata yang sering muncul dalam konteks yang serupa akan memiliki vektor yang berdekatan dalam ruang *embedding*.

Selanjutnya, vektor *embedding* tersebut dikalikan dengan *Context Matrix* (W') yang berukuran N×V. Hasilnya adalah skor untuk seluruh kata dalam kosakata. Pada Output Layer, fungsi Softmax digunakan untuk mengubah skor

tersebut menjadi distribusi probabilitas. Kata dengan probabilitas tertinggi dianggap sebagai kata konteks yang paling mungkin muncul di sekitar kata target. Selama proses pelatihan, bobot matriks W dan W′ akan terus diperbarui menggunakan algoritma optimasi hingga model mampu memprediksi konteks kata dengan baik.

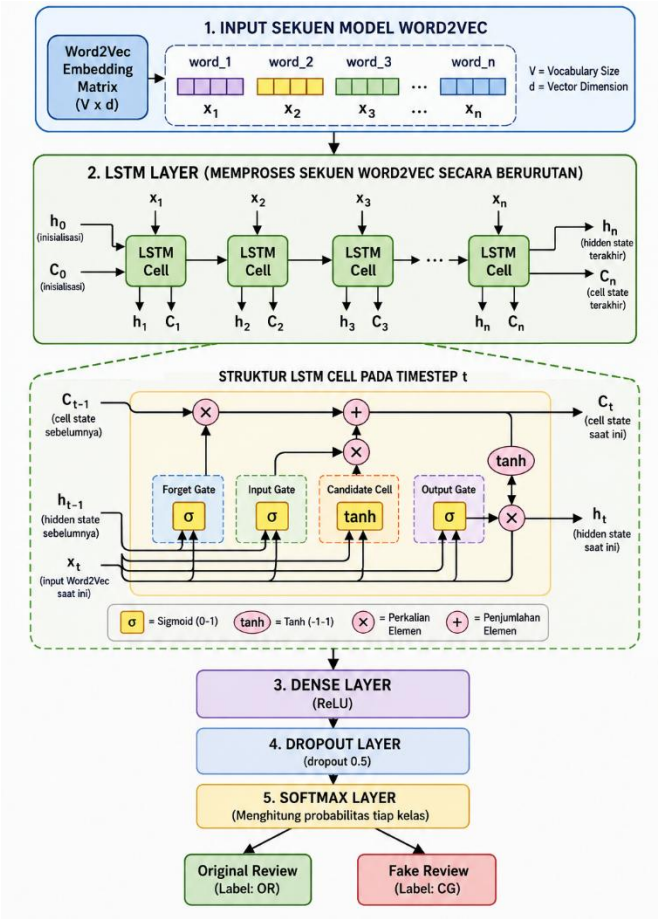
Secara keseluruhan, gambar tersebut menjelaskan bahwa Word2Vec mempelajari representasi kata melalui tugas prediksi konteks. Setelah proses pelatihan selesai, matriks *embedding* W menyimpan vektor kata yang mengandung informasi semantik dan sintaktik. Vektor inilah yang kemudian digunakan sebagai fitur masukan pada berbagai tugas pemrosesan bahasa alami, seperti klasifikasi teks, analisis sentimen, deteksi ulasan palsu, dan pengelompokan dokumen. Konsep ini didasarkan pada *Distributional Hypothesis*, yaitu kata-kata yang muncul dalam konteks yang mirip cenderung memiliki makna yang mirip pula.

6.1.2 Klasifikasi dengan Algoritma LSTM

Pada tahap klasifikasi, penelitian ini mengimplementasikan algoritma *Long Short-Term Memory* (LSTM) sebagai model utama untuk mendeteksi ulasan palsu (*fake review*) pada dataset ulasan marketplace. LSTM merupakan pengembangan dari *Recurrent Neural Network* (RNN) yang dirancang untuk mengatasi permasalahan vanishing gradient, sehingga mampu menangkap dependensi jangka panjang dalam data sekuensial seperti teks. Kemampuan

tersebut menjadikan LSTM sangat sesuai untuk menganalisis pola bahasa, susunan kata, serta hubungan konteks antar kata dalam ulasan pengguna.

Arsitektur model yang digunakan dalam penelitian ini mengadopsi pendekatan hibrida dengan mengintegrasikan representasi teks berbasis Word2Vec dan kemampuan pemodelan sekuensial dari LSTM. Word2Vec berperan dalam mengubah kata-kata pada ulasan menjadi vektor numerik yang merepresentasikan makna semantik, sedangkan LSTM memproses urutan vektor tersebut untuk mempelajari pola kalimat secara lebih mendalam. Kombinasi kedua metode ini memungkinkan model memahami keterkaitan antar kata sekaligus konteks keseluruhan ulasan.



Gambar 6. 2 Alur Model LSTM

Berdasarkan gambar 6.2 tentang alur proses klasifikasi ulasan menggunakan model *Long Short-Term Memory* (LSTM) dengan representasi fitur Word2Vec. Proses diawali dengan pembentukan Word2Vec *Embedding* Matrix yang mengubah setiap kata dalam teks menjadi vektor numerik berdimensi tertentu. Setiap kata pada ulasan, seperti $word_1$, $word_2$, $word_3$, hingga $word_n$, direpresentasikan sebagai vektor *embedding* (x_1, x_2, x_3, ..., x_n) yang kemudian disusun menjadi sebuah sekuens input. Representasi ini memungkinkan model menangkap hubungan semantik antar kata sehingga informasi penting dalam teks dapat dipertahankan sebelum memasuki tahap pemrosesan oleh LSTM.

Selanjutnya, sekuens vektor Word2Vec diproses secara berurutan oleh LSTM Layer. Pada tahap ini, setiap LSTM Cell menerima input saat ini (x_t) serta informasi dari waktu sebelumnya berupa *hidden state* (h_{t-1}) dan cell state (c_{t-1}). Mekanisme ini memungkinkan LSTM mempelajari ketergantungan jangka panjang dalam teks dan mengatasi permasalahan vanishing gradient yang sering terjadi pada *Recurrent Neural Network* (RNN) biasa. Setiap sel menghasilkan hidden state dan cell state baru yang diteruskan ke sel berikutnya hingga seluruh kata dalam kalimat selesai diproses. Hasil akhirnya adalah *hidden state* terakhir (h_n) yang merepresentasikan informasi penting dari keseluruhan ulasan.

Bagian tengah memperlihatkan struktur internal LSTM Cell yang terdiri atas tiga gerbang utama, yaitu *Forget Gate*, *Input Gate*, dan *Output Gate*. *Forget Gate* berfungsi menentukan informasi dari cell state sebelumnya yang

perlu dipertahankan atau dibuang. *Input Gate* mengatur informasi baru yang akan disimpan ke dalam memori, sedangkan *Candidate Cell* menghasilkan kandidat informasi baru menggunakan fungsi aktivasi tanh. Setelah itu, *Output Gate* menentukan informasi yang akan dikeluarkan sebagai hidden state saat ini. Melalui kombinasi fungsi sigmoid dan tanh, LSTM dapat menyaring informasi penting serta mempertahankan konteks yang relevan sepanjang urutan kata.

Setelah proses ekstraksi fitur sekuensial selesai, keluaran dari LSTM diteruskan ke *Dense Layer* dengan fungsi aktivasi ReLU untuk mempelajari pola klasifikasi yang lebih kompleks. Kemudian diterapkan *Dropout Layer* dengan nilai dropout sebesar 0,5 untuk mengurangi risiko overfitting selama proses pelatihan model. Tahap terakhir adalah *Softmax Layer* yang menghitung probabilitas masing-masing kelas. Berdasarkan probabilitas tertinggi yang dihasilkan, model mengklasifikasikan ulasan ke dalam salah satu dari dua kategori, yaitu *Original Review* (OR) sebagai ulasan asli atau *Fake review* (CG) sebagai ulasan palsu. Dengan alur tersebut, model LSTM mampu memanfaatkan representasi Word2Vec dan informasi kontekstual antar kata untuk melakukan deteksi ulasan palsu secara efektif..

6.2 Uji Coba

6.2.1 *BASELINE*

Pada tahap eksperimen, dataset yang digunakan terlebih dahulu dibagi menjadi dua bagian utama, yaitu data training dan data testing dengan

perbandingan 80:20. Sebanyak 80% data digunakan untuk melatih model agar mampu mempelajari pola dari data ulasan, sedangkan 20% sisanya digunakan sebagai data pengujian untuk mengevaluasi kinerja model terhadap data yang belum pernah dilihat sebelumnya. Pembagian ini bertujuan untuk mengukur kemampuan generalisasi model sehingga hasil evaluasi yang diperoleh lebih objektif dan tidak bias terhadap data pelatihan.

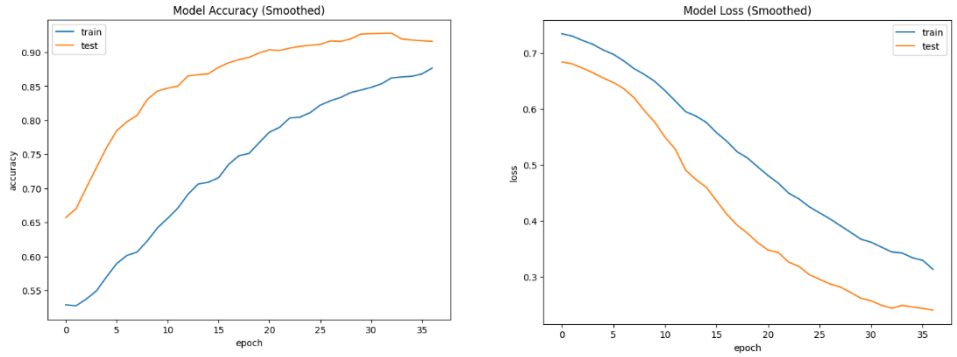
Pada skenario baseline ini, representasi teks dilakukan menggunakan Word2Vec untuk mengubah setiap kata pada ulasan menjadi vektor numerik yang memiliki makna semantik. Selanjutnya, urutan vektor kata tersebut diproses oleh algoritma *Long Short-Term Memory* (LSTM) untuk menangkap hubungan antar kata dalam bentuk data sekuensial. Kombinasi Word2Vec dan LSTM pada tahap baseline digunakan sebagai model dasar untuk mengetahui performa awal sistem sebelum dilakukan pengembangan atau optimasi lanjutan.

Selanjutnya, untuk meningkatkan keandalan evaluasi model, penelitian ini menerapkan metode K-*Fold* Cross Validation dengan nilai K = 10. Dalam metode ini, dataset dibagi menjadi 10 bagian yang sama besar, kemudian model dilatih dan diuji sebanyak 10 kali dengan kombinasi data training dan testing yang berbeda pada setiap iterasi. Pada setiap proses, satu *fold* digunakan sebagai data pengujian, sedangkan sembilan *fold* lainnya digunakan sebagai data pelatihan.

Hasil evaluasi dari seluruh iterasi kemudian dirata-ratakan untuk memperoleh nilai performa akhir yang lebih stabil dan representatif.

Penggunaan K-*Fold* Cross Validation ini diharapkan dapat mengurangi bias akibat pembagian data tunggal serta meningkatkan validitas hasil penelitian. Dengan demikian, performa model Word2Vec dengan LSTM tidak hanya bergantung pada satu skenario pembagian data saja, tetapi diuji secara menyeluruh melalui beberapa kombinasi data.

Selama proses pelatihan, Word2Vec berperan menghasilkan representasi fitur dari teks ulasan, sedangkan LSTM mempelajari pola urutan kata untuk membedakan antara *Original Review* (OR) dan *Fake review* (CG). Evaluasi model baseline dilakukan menggunakan metrik seperti *accuracy*, precision, *recal*l, dan F1-*Score*. Melalui pengujian baseline ini, diperoleh gambaran awal mengenai kemampuan model Word2Vec dengan LSTM dalam mendeteksi ulasan palsu pada marketplace sebelum dilakukan eksperimen lanjutan dengan metode optimasi lainnya.



Gambar 6. 3 Grafik model *Accuracy* dan model Loss pada LSTM

Berdasarkan Gambar 6.3 grafik *model accuracy* dan *model loss* pada proses pelatihan model Word2Vec + LSTM. Grafik ini menunjukkan perkembangan kinerja model selama proses pelatihan hingga mencapai

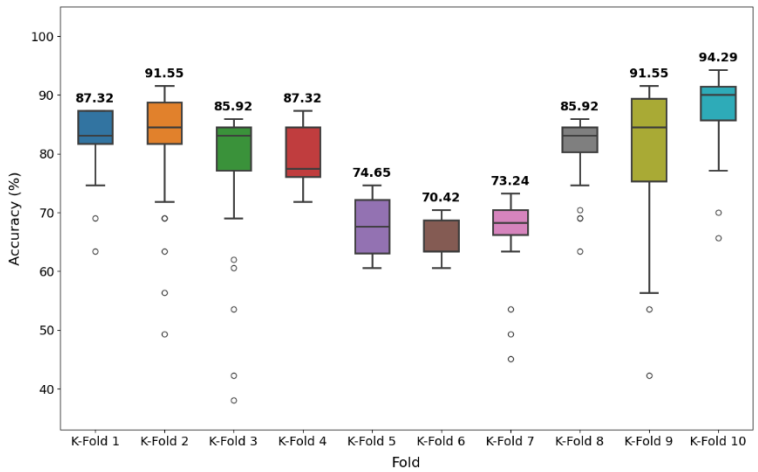
konvergensi. Grafik sebelah kiri memperlihatkan perubahan nilai akurasi (*accuracy*), sedangkan grafik sebelah kanan menunjukkan perubahan nilai kesalahan (*loss*) pada data latih (*train*) dan data uji (*test*).

Pada grafik model *accuracy*, terlihat bahwa akurasi data latih (*train*) meningkat secara bertahap dari sekitar 0,53 pada *epoch* awal hingga mencapai sekitar 0,88 pada *epoch* ke-35. Peningkatan yang konsisten ini menunjukkan bahwa model semakin mampu mengenali pola-pola yang terdapat pada data pelatihan. Sementara itu, akurasi data uji (*test*) juga mengalami peningkatan yang signifikan, dimulai dari sekitar 0,66 pada *epoch* awal dan terus meningkat hingga mencapai sekitar 0,92–0,93 pada akhir pelatihan. Kurva akurasi data uji yang berada di atas kurva data latih menunjukkan bahwa model mampu mempertahankan performa yang baik pada data yang tidak digunakan selama proses pelatihan.

Pada grafik model *loss*, terlihat bahwa nilai *loss* pada data latih (*train*) mengalami penurunan secara konsisten dari sekitar 0,74 pada *epoch* awal menjadi sekitar 0,21 pada *epoch* akhir. Penurunan ini menunjukkan bahwa kesalahan prediksi model semakin berkurang seiring bertambahnya jumlah *epoch.* Hal yang sama juga terlihat pada data uji (*test*), di mana nilai *loss* menurun dari sekitar 0,70 menjadi sekitar 0,14 pada akhir proses pelatihan. Kurva *loss* data uji yang terus menurun tanpa mengalami kenaikan yang signifikan menunjukkan bahwa model mampu melakukan generalisasi dengan baik terhadap data baru.

Secara keseluruhan, grafik *accuracy* dan *loss* menunjukkan bahwa model Word2Vec + LSTM berhasil mempelajari pola data secara efektif. Hal ini ditunjukkan oleh meningkatnya nilai akurasi serta menurunnya nilai *loss* pada data latih maupun data uji. Selain itu, tidak terlihat adanya perbedaan yang ekstrem antara kedua kurva sehingga dapat disimpulkan bahwa model memiliki kemampuan generalisasi yang baik dan tidak menunjukkan indikasi *overfitting* yang signifikan dalam proses deteksi ulasan.

Kemudian dilakukan perhitungan K-*Fold Cross validation accuracy* untuk meningkatkan keandalan evaluasi model dengan hasil yang ditunjukkan pada gambar 6.4 tentang K-*Fold cross validation accuracy*.



Gambar 6. 4 K-*Fold Cross validation accuracy* LSTM

Berdasarkan Gambar 6.4 grafik *K-Fold Cross Validation accuracy* pada model LSTM, terlihat bahwa performa model pada setiap fold mengalami variasi nilai akurasi. Variasi tersebut menunjukkan kemampuan model dalam melakukan generalisasi terhadap pembagian data yang berbeda pada setiap proses validasi. Berdasarkan nilai rata-rata yang ditampilkan pada masing-

masing *fold*, akurasi tertinggi diperoleh pada *Fold* 10 sebesar 94,29%, diikuti oleh *Fold* 2 dan *Fold* 9 yang masing-masing mencapai 91,55%. Sementara itu, nilai akurasi terendah terdapat pada *Fold* 7 sebesar 70,42%, diikuti oleh *Fold* 5 sebesar 74,65% dan *Fold* 6 sebesar 73,24%. Perbedaan nilai akurasi antar *fold* menunjukkan bahwa karakteristik data pada setiap pembagian masih memengaruhi kinerja model LSTM.

Secara umum, sebagian besar *fold* menghasilkan tingkat akurasi yang cukup baik. *Fold* 1 memperoleh akurasi sebesar 87,32%, *Fold* 2 sebesar 91,55%, *Fold* 3 sebesar 85,92%, *Fold* 4 sebesar 87,32%, *Fold* 8 sebesar 85,92%, *Fold* 9 sebesar 91,55%, dan *Fold* 10 sebesar 94,29%. Hasil tersebut menunjukkan bahwa model LSTM mampu mempelajari pola serta konteks yang terdapat dalam data ulasan dengan cukup efektif. Kemampuan LSTM dalam menangkap hubungan antar kata dan mempertahankan informasi jangka panjang memungkinkan model melakukan klasifikasi dengan tingkat akurasi yang tinggi pada sebagian besar subset data.

Meskipun demikian, terdapat beberapa *fold* yang menunjukkan performa lebih rendah, terutama pada *Fold* 5, *Fold* 6, dan *Fold* 7 yang berada pada rentang akurasi 70–75%. Nilai yang relatif lebih rendah tersebut mengindikasikan bahwa data pada *fold* tersebut kemungkinan memiliki tingkat kompleksitas yang lebih tinggi, seperti adanya ulasan yang ambigu, variasi kosakata yang beragam, penggunaan bahasa tidak baku, atau kemiripan karakteristik antara ulasan asli dan ulasan palsu. Kondisi tersebut

menyebabkan model mengalami kesulitan dalam membedakan kelas secara optimal.

Dari sisi kestabilan model, grafik *boxplot* menunjukkan bahwa sebagian besar *fold* memiliki median yang berada pada rentang akurasi tinggi dengan sebaran data yang relatif seimbang. *Fold* 1, *Fold* 2, *Fold* 3, *Fold* 4, *Fold* 8, *Fold* 9, dan Fold 10 memperlihatkan distribusi yang lebih konsisten dibandingkan *fold* lainnya. Namun, beberapa fold masih memiliki nilai pencilan (*outlier*) yang cukup jauh dari median, menandakan adanya variasi performa pada beberapa iterasi pengujian. Meskipun demikian, mayoritas *fold* tetap menunjukkan tingkat akurasi yang tinggi sehingga model dapat dikatakan memiliki kemampuan generalisasi yang cukup baik.

Secara keseluruhan, hasil evaluasi *K-Fold Cross Validation* pada model LSTM menunjukkan bahwa model mampu memberikan performa klasifikasi yang baik dalam mendeteksi ulasan palsu. Dominasi nilai akurasi di atas 85% pada sebagian besar *fold* mengindikasikan bahwa model berhasil memahami pola dan konteks bahasa dalam data ulasan *marketplace*. Akan tetapi, adanya beberapa *fold* dengan akurasi yang lebih rendah menunjukkan bahwa masih diperlukan pengembangan lebih lanjut, seperti optimasi *hyperparameter,* peningkatan kualitas tahap *preprocessing*, serta penerapan teknik penyeimbangan data agar performa model menjadi lebih konsisten pada seluruh *fold* pengujian.

Langkah selanjutnya yaitu menghitung performance algoritma LSTM untuk *accuracy*, *precision*, *recall* dan F1-*Score* dengan menggunakan

confuision matrix seperti terlihat pada gambar 6.5 tentang hasil eksperimen Word2Vec+LSTM dengan *cofussion matrix*.

Gambar 6. 5 Hasil tabel *confunsion matrix*

Pada gambar 6.5 Hasil tabel *confusion matrix* diatas dapat dijelaskan berdasarkan empat komponen utama yang didapatkan pada masing-masing kolom dengan penjelasan yaitu :

- TP (*True Positive*), sebanyak 33 data original yang terklasifikasi benar.
- TN (True Negatif), sebanyak 120 data fake yang terklasifikasi benar.
- FP (*False Positive*), sebanyak 20 data original yang terklasifikasi salah.
- FN (*False Negative*), sebanyak 5 data fake yang terkasifikasi salah.

Dari tabel confusin matrix yang sudah didapat, selanjutnya adalah melakukan perhitungan *accuracy*, *precission*, *recal* dan f1-*Score*, untuk mengukur sejauh mana keberhasilan model dalam mengkhasifikasi data.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (6.1)$$

$$Accuracy = \frac{33+120}{33+120+20+5} = \frac{153}{178} = 0{,}8595$$

$$Precesion = \frac{TP}{TP+FP} \quad (6.2)$$

$$Precesion = \frac{33}{33+20} = \frac{33}{53} = 0{,}6226$$

$$Recall = \frac{TP}{TP+FN} \quad (6.3)$$

$$Recall = \frac{33}{33+5} = \frac{33}{38} = 0{,}8684$$

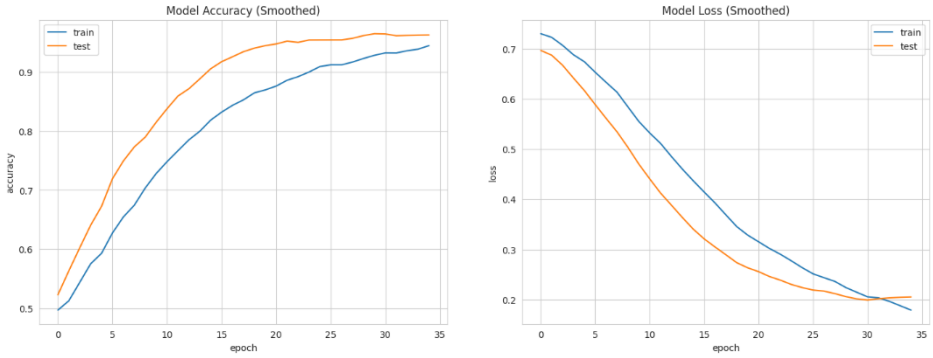
$$F1 - Score = 2 * \frac{Recall\ x\ Preceision}{Recall+Precision} \quad (6.4)$$

$$F1 - Score = 2 * \frac{0{,}8684x\ 0{,}6226}{0{,}8684 + 0{,}6226} = 2 * \frac{0{,}5406}{1{,}4910} = 0{,}7251$$

6.2.2. SMOTE

Pada tahap eksperimen, model *Long Short-Term Memory* (LSTM) dikombinasikan dengan teknik *oversampling* SMOTE (*Synthetic Minority Over-sampling Technique*) serta representasi fitur Word2Vec untuk meningkatkan performa klasifikasi pada dataset yang tidak seimbang. Data penelitian terlebih dahulu dibagi menjadi data pelatihan dan data pengujian dengan perbandingan 80:20, di mana 80% data digunakan untuk melatih model agar mampu mempelajari pola sekuensial dari teks ulasan, sedangkan 20% sisanya digunakan untuk mengevaluasi kinerja model terhadap data yang belum pernah dilihat sebelumnya. Sebelum proses pelatihan dilakukan, teks ulasan terlebih dahulu diubah ke dalam bentuk vektor numerik menggunakan Word2Vec, sehingga setiap kata memiliki representasi semantik yang dapat dipahami oleh model.

Penerapan SMOTE dilakukan pada data pelatihan setelah proses ekstraksi fitur Word2Vec, dengan tujuan meningkatkan jumlah data pada kelas minoritas secara sintetis sehingga distribusi data menjadi lebih seimbang. Dengan keseimbangan kelas yang lebih baik, model diharapkan tidak cenderung bias terhadap kelas mayoritas dan mampu mengenali pola dari kedua kelas secara lebih adil.



Gambar 6. 6 Grafik model *Accuracy* dan model Loss pada SMOTE+Word2Vec+LSTM

Berdasarkan Gambar 6.6, ditampilkan grafik model *accuracy* dan model *loss* pada proses pelatihan model yang mengombinasikan SMOTE, Word2Vec, dan LSTM. Grafik tersebut digunakan untuk mengevaluasi kemampuan model dalam mempelajari pola data selama proses pelatihan sekaligus mengukur kemampuan generalisasinya terhadap data pengujian. Grafik sebelah kiri menunjukkan perkembangan nilai *accuracy*, sedangkan grafik sebelah kanan menunjukkan perubahan nilai loss pada data *training* dan *testing.*

Pada grafik model *accuracy*, terlihat bahwa akurasi data *training* meningkat secara bertahap dari sekitar 0,50 pada awal pelatihan hingga mencapai sekitar 0,95 pada *epoch* terakhir. Peningkatan yang cukup konsisten

ini menunjukkan bahwa model semakin mampu mengenali pola yang terdapat dalam data pelatihan. Sementara itu, akurasi data testing juga mengalami peningkatan yang signifikan, dimulai dari sekitar 0,55 pada *epoch* awal dan mencapai sekitar 0,96 pada akhir proses pelatihan. Setelah memasuki sekitar *epoch* ke-20, kurva akurasi training dan testing mulai cenderung stabil dengan selisih yang relatif kecil. Kondisi ini menunjukkan bahwa model mampu mempertahankan performa yang baik pada data yang tidak digunakan selama pelatihan.

Peningkatan akurasi tersebut menunjukkan efektivitas kombinasi metode yang digunakan. Teknik SMOTE berperan dalam mengatasi ketidakseimbangan kelas dengan menghasilkan sampel sintetis pada kelas minoritas sehingga distribusi data menjadi lebih seimbang. Selanjutnya, Word2Vec mengubah teks ulasan menjadi representasi vektor numerik yang mampu menangkap hubungan semantik antar kata. *Representasi* tersebut kemudian diproses menggunakan LSTM yang memiliki kemampuan untuk mempelajari pola urutan kata dan mempertahankan informasi kontekstual dalam kalimat, sehingga menghasilkan performa klasifikasi yang lebih baik.

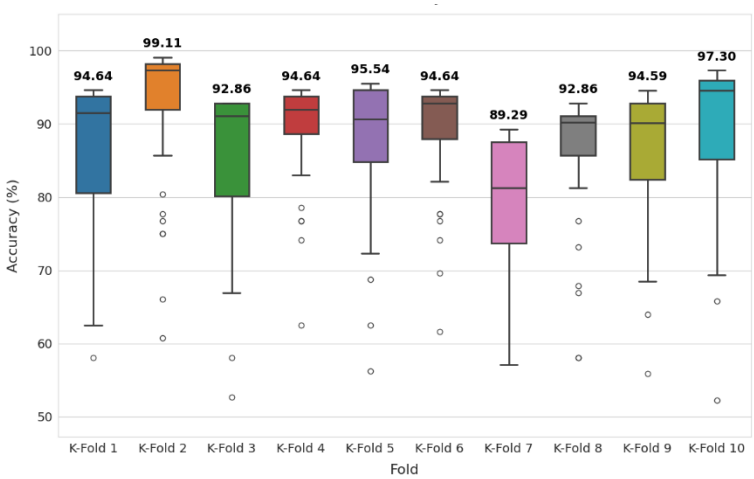
Pada grafik model *loss,* terlihat bahwa nilai *loss* pada data training mengalami penurunan secara konsisten dari sekitar 0,73 pada awal pelatihan menjadi sekitar 0,20 pada *epoch* terakhir. Penurunan ini menunjukkan bahwa tingkat kesalahan prediksi model semakin berkurang seiring bertambahnya jumlah *epoch*. Hal serupa juga terlihat pada data testing, di mana nilai *loss* menurun dari sekitar 0,70 menjadi sekitar 0,19 pada akhir pelatihan. Kurva *loss*

training dan testing memiliki pola yang hampir serupa dan terus menurun hingga mendekati titik konvergensi.

Selain itu, tidak terlihat adanya peningkatan loss yang signifikan pada data testing selama proses pelatihan. Bahkan pada beberapa *epoch* terakhir, nilai loss testing tetap berada sangat dekat dengan loss training. Kondisi ini menunjukkan bahwa model tidak mengalami *overfitting* yang berarti, karena performa pada data pengujian tetap stabil dan sejalan dengan performa pada data pelatihan.

Secara keseluruhan, hasil visualisasi menunjukkan bahwa model SMOTE + Word2Vec + LSTM memiliki performa yang baik dan stabil dalam melakukan klasifikasi teks. Hal ini ditunjukkan oleh peningkatan nilai *accuracy* hingga mencapai sekitar 95% pada data *training* dan 96% pada data *testing*, serta penurunan nilai *loss* secara konsisten hingga berada di kisaran 0,20 pada akhir pelatihan. Kedekatan antara kurva training dan testing pada kedua grafik mengindikasikan bahwa model memiliki kemampuan generalisasi yang baik dan mampu mengklasifikasikan data ulasan secara efektif. Dengan demikian, kombinasi SMOTE, Word2Vec, dan LSTM terbukti mampu menghasilkan model yang andal untuk mendeteksi ulasan palsu pada dataset *marketplace* yang digunakan dalam penelitian ini.

Kemudian dilakukan perhitungan K-*Fold Cross validation accuracy* untuk meningkatkan keandalan evaluasi model dengan hasil yang ditunjukkan pada gambar 6.7 tentang K-*Fold cross validation accuracy*.



Gambar 6. 7 Hasil Proses K-*Fold Cross validation accuracy* SMOTE+Word2Vec+LSTM

Berdasarkan Gambar 6.7, hasil evaluasi model menggunakan metode *K-Fold Cross Validation* pada kombinasi SMOTE + Word2Vec + LSTM disajikan dalam bentuk grafik boxplot untuk sepuluh fold pengujian (*Fold* 1–*Fold* 10). Grafik tersebut menggambarkan distribusi nilai akurasi pada setiap *fold* sehingga dapat digunakan untuk menilai tingkat kestabilan dan konsistensi model dalam melakukan klasifikasi pada berbagai pembagian data. Semakin tinggi nilai akurasi dan semakin konsisten distribusi data pada setiap *fold*, maka semakin baik kemampuan model dalam melakukan generalisasi terhadap data yang berbeda.

Berdasarkan hasil pengujian, nilai rata-rata akurasi pada setiap *fold* menunjukkan performa yang cukup tinggi. Akurasi tertinggi diperoleh pada *Fold* 2 sebesar 99,11%, diikuti oleh *Fold* 10 sebesar 97,30%, serta *Fold* 5 sebesar 95,54%. Sementara itu, nilai rata-rata akurasi terendah terdapat pada *Fold* 7 sebesar 89,25%, diikuti oleh *Fold* 3 dan *Fold* 8 yang masing-masing

memperoleh akurasi sebesar 92,86%. Meskipun terdapat perbedaan performa antar *fold*, seluruh hasil menunjukkan tingkat akurasi yang baik dengan mayoritas berada di atas 92%.

Sebagian besar *fold* menghasilkan akurasi yang relatif tinggi, yaitu *Fold* 1 sebesar 94,64%, *Fold* 2 sebesar 99,11%, *Fold* 3 sebesar 92,86%, *Fold* 4 sebesar 94,64%, *Fold* 5 sebesar 95,54%, *Fold* 6 sebesar 94,64%, *Fold* 8 sebesar 92,86%, *Fold* 9 sebesar 94,59%, dan *Fold* 10 sebesar 97,30%. Hasil tersebut menunjukkan bahwa model mampu mengenali pola dan karakteristik data ulasan secara konsisten pada berbagai subset data. Tingginya nilai akurasi pada hampir seluruh *fold* mengindikasikan bahwa penerapan teknik SMOTE berhasil mengurangi pengaruh ketidakseimbangan kelas sehingga model dapat mempelajari setiap kategori data dengan lebih baik.

Dari sisi distribusi data, *boxplot* menunjukkan bahwa sebagian besar *fold* memiliki median yang berada pada rentang akurasi tinggi. Namun, masih terdapat beberapa nilai pencilan (*outlier*) yang berada jauh di bawah median pada beberapa *fold*, seperti *Fold* 1, *Fold* 2, *Fold* 7, dan *Fold* 10. Keberadaan *outlier* tersebut menunjukkan bahwa pada beberapa iterasi pengujian masih terdapat data yang lebih sulit diklasifikasikan oleh model. Meskipun demikian, mayoritas distribusi nilai tetap berada pada rentang akurasi yang tinggi sehingga kestabilan model secara keseluruhan masih tergolong baik.

Performa yang diperoleh tidak terlepas dari kontribusi masing-masing metode yang digunakan. SMOTE membantu menyeimbangkan distribusi data dengan menambah sampel sintetis pada kelas minoritas, sehingga mengurangi

bias selama proses pelatihan. Word2Vec mengubah data teks menjadi representasi vektor numerik yang mampu menangkap hubungan semantik antar kata, sedangkan LSTM memanfaatkan representasi tersebut untuk mempelajari urutan kata dan konteks kalimat secara lebih efektif. Kombinasi ketiga metode tersebut memungkinkan model menghasilkan prediksi yang lebih akurat dalam proses klasifikasi ulasan.

Secara keseluruhan, hasil *K-Fold Cross Validation* menunjukkan bahwa model SMOTE + Word2Vec + LSTM memiliki performa yang baik dan relatif stabil. Mayoritas *fold* menghasilkan akurasi di atas 92%, dengan nilai tertinggi mencapai 99,11% pada *Fold* 2. Distribusi nilai yang cenderung konsisten pada sebagian besar *fold* menunjukkan bahwa model memiliki kemampuan generalisasi yang baik terhadap data baru. Oleh karena itu, kombinasi SMOTE, Word2Vec, dan LSTM dapat dianggap efektif dalam meningkatkan performa deteksi ulasan palsu pada dataset *marketplace*.

Langkah selanjutnya yaitu menghitung performance algoritma LSTM untuk *accuracy*, precision, *recal*l dan F1-*Score* dengan menggunakan confuision matrix seperti terlihat pada gambar 6.8 tentang hasil eksperimen SMOTE +Word2Vec+LSTM dengan *cofussion matrix*.

Gambar 6. 8 Hasil tabel *confunsion matrix*

Pada gambar 6.8 Hasil tabel *confusion matrix* diatas dapat dijelaskan berdasarkan empat komponen utama yang didapatkan pada masing-masing kolom dengan penjelasan yaitu :

- TP (*True Positive*), sebanyak 135 data original yang terklasifikasi benar.
- TN (True Negatif), sebanyak 121 data fake yang terklasifikasi benar.
- FP (*False Positive*), sebanyak 19 data original yang terklasifikasi salah.
- FN (*False Negative*), sebanyak 5 data fake yang terkasifikasi salah.

Dari tabel confusin matrix yang sudah didapat, selanjutnya adalah melakukan perhitungan *accuracy*, *precission*, *recal* dan f1-*Score*, untuk mengukur sejauh mana keberhasilan model dalam mengkhasifikasi data.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (6.5)$$

$$Accuracy = \frac{135+121}{135+121+19+5} = \frac{256}{280} = 0{,}9142$$

$$Precesion = \frac{TP}{TP+FP} \quad (6.6)$$

$$Precesion = \frac{135}{135+19} = \frac{135}{154} = 0{,}8766$$

$$Recall = \frac{TP}{TP+FN} \quad (6.7)$$

$$Recall = \frac{135}{135+5} = \frac{135}{140} = 0{,}9642$$

$$F1 - Score = 2 * \frac{Recall\ x\ Preceision}{Recall+Precision} \quad (6.8)$$

$$F1 - Score = 2 * \frac{0{,}9642x\ 0{,}8766}{0{,}9642 + 0{,}8766} = 2 * \frac{0{,}8452}{1{,}8408} = 0,9182$$

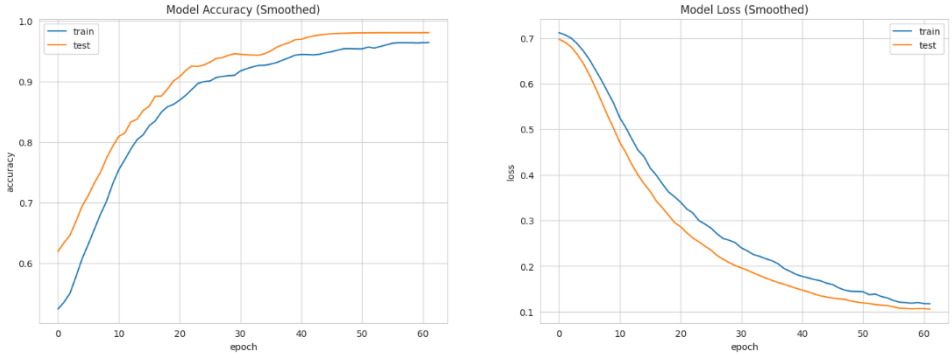
6.2.3. ADASYN

Pada tahap eksperimen, model *Long Short-Term Memory* (LSTM) dikombinasikan dengan teknik *oversampling* ADASYN (*Adaptive Synthetic Sampling*) serta representasi fitur Word2Vec untuk meningkatkan performa klasifikasi pada dataset yang tidak seimbang. Data penelitian terlebih dahulu dibagi menjadi data pelatihan dan data pengujian dengan perbandingan 80:20, di mana 80% data digunakan untuk melatih model agar mampu mempelajari pola sekuensial dari teks ulasan, sedangkan 20% sisanya digunakan untuk mengevaluasi kinerja model terhadap data yang belum pernah dilihat sebelumnya. Sebelum proses pelatihan dilakukan, teks ulasan dikonversi ke dalam bentuk vektor numerik menggunakan Word2Vec, sehingga setiap kata memiliki representasi semantik yang dapat diproses oleh model.

Penerapan ADASYN dilakukan pada data pelatihan setelah proses ekstraksi fitur Word2Vec, dengan tujuan menghasilkan data sintetis secara adaptif pada kelas minoritas berdasarkan tingkat kesulitan pembelajaran.

Berbeda dengan metode *oversampling* konvensional, ADASYN lebih fokus pada data yang sulit dipelajari, sehingga distribusi data menjadi lebih representatif dan informatif. Dengan keseimbangan kelas yang lebih baik, model diharapkan tidak bias terhadap kelas mayoritas serta mampu mengenali pola dari kedua kelas secara lebih proporsional.

Dengan integrasi ADASYN, Word2Vec, dan LSTM, diharapkan model mampu meningkatkan kinerja klasifikasi secara signifikan, terutama dalam menangani permasalahan ketidakseimbangan data. Pendekatan ini tidak hanya membantu dalam meningkatkan akurasi, tetapi juga memperbaiki kemampuan model dalam mengenali kelas minoritas, sehingga menghasilkan performa yang lebih stabil dan generalisasi yang lebih baik terhadap data baru.



Gambar 6. 9 Grafik model *Accuracy* dan model Loss pada ADASYN+Word2Vec+LSTM

Berdasarkan Gambar 6.9 grafik model *accuracy* dan model *loss* dari hasil pelatihan model yang menggunakan kombinasi ADASYN, Word2Vec, dan LSTM. Grafik tersebut menunjukkan perkembangan kinerja model pada data *training* dan *testing* selama proses pelatihan berlangsung. Analisis terhadap kedua grafik ini bertujuan untuk mengevaluasi kemampuan model dalam

mempelajari pola data serta mengukur tingkat generalisasinya terhadap data yang belum pernah digunakan sebelumnya.

Pada grafik model *accuracy,* terlihat bahwa akurasi pada data *training* dan *testing* mengalami peningkatan yang signifikan sejak *epoch* awal. Akurasi data *training* meningkat dari sekitar 0,53 pada awal pelatihan hingga mencapai sekitar 0,98 pada *epoch* ke-60. Sementara itu, akurasi data testing meningkat dari sekitar 0,63 menjadi sekitar 0,97 pada akhir pelatihan. Peningkatan yang terjadi berlangsung secara bertahap dan cenderung stabil setelah memasuki *epoch* ke-20. Selain itu, jarak antara kurva training dan testing relatif kecil sehingga menunjukkan bahwa model mampu mempertahankan performa yang baik pada data pengujian.

Peningkatan akurasi yang konsisten tersebut menunjukkan bahwa model semakin mampu mengenali karakteristik dan pola yang terdapat dalam data ulasan. Pada tahap awal pelatihan, peningkatan akurasi terjadi cukup cepat karena model mulai mempelajari pola-pola dasar dari data. Setelah itu, laju peningkatan menjadi lebih lambat dan stabil hingga mencapai kondisi konvergen pada *epoch-epoch* akhir. Kondisi ini menunjukkan bahwa proses pembelajaran berlangsung dengan baik dan model berhasil menemukan representasi yang efektif untuk melakukan klasifikasi.

Sementara itu, grafik model *loss* menunjukkan tren penurunan yang konsisten pada data *training* maupun *testing*. Nilai loss data *training* yang pada awal pelatihan berada di sekitar 0,72 terus menurun hingga mencapai sekitar 0,12 pada akhir *epoch*. Hal yang sama juga terlihat pada data *testing*, di mana

nilai *loss* menurun dari sekitar 0,70 menjadi sekitar 0,11. Penurunan *loss* yang stabil mengindikasikan bahwa tingkat kesalahan prediksi model semakin berkurang seiring bertambahnya proses pelatihan sehingga model mampu menghasilkan prediksi yang lebih akurat.

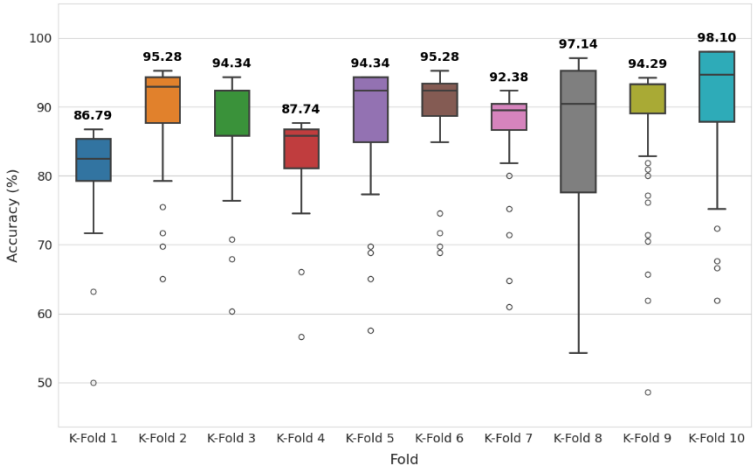
Kurva loss training dan testing juga terlihat saling berdekatan sepanjang proses pelatihan. Meskipun terdapat sedikit perbedaan pada beberapa *epoch,* tren keduanya tetap menunjukkan pola penurunan yang serupa hingga akhir pelatihan. Tidak terlihat adanya peningkatan loss yang signifikan pada data testing yang dapat mengindikasikan terjadinya *overfitting.* Sebaliknya, kedekatan kedua kurva menunjukkan bahwa kemampuan generalisasi model tetap terjaga dengan baik.

Hasil yang diperoleh tidak terlepas dari kontribusi metode yang digunakan. ADASYN (*Adaptive Synthetic Sampling*) membantu mengatasi ketidakseimbangan kelas dengan menghasilkan sampel sintetis secara adaptif pada data yang lebih sulit dipelajari. Selanjutnya, Word2Vec mengubah teks ulasan menjadi representasi vektor numerik yang mampu menangkap hubungan semantik antar kata. Representasi tersebut kemudian diproses oleh LSTM untuk mempelajari urutan kata dan konteks kalimat secara lebih mendalam, sehingga model dapat melakukan klasifikasi dengan lebih efektif.

Secara keseluruhan, menunjukkan bahwa model ADASYN + Word2Vec + LSTM memiliki performa yang sangat baik dan stabil. Hal ini ditunjukkan oleh meningkatnya nilai *accuracy* hingga mencapai sekitar 98% pada data *training* dan 97% pada data *testing*, serta menurunnya nilai *loss* secara

konsisten hingga berada pada kisaran 0,11–0,12 pada akhir pelatihan. Kedekatan antara kurva training dan testing pada kedua grafik menunjukkan bahwa model memiliki kemampuan generalisasi yang baik dan tidak mengalami *overfitting* yang signifikan. Dengan demikian, kombinasi ADASYN, Word2Vec, dan LSTM terbukti efektif dalam meningkatkan performa model untuk mendeteksi ulasan palsu pada dataset marketplace yang digunakan dalam penelitian ini.

Selanjutnya dilakukan perhitungan K-*Fold Cross validation accuracy* untuk meningkatkan keandalan evaluasi model dengan hasil yang ditunjukkan pada gambar 6.10 tentang K-*Fold cross validation accuracy*.



Gambar 6. 10 Hasil Proses K-*Fold Cross validation accuracy* ADASYN+Word2Vec+LSTM

Berdasarkan Gambar 6.10, hasil evaluasi model menggunakan metode *K-Fold Cross Validation* pada kombinasi ADASYN + Word2Vec + LSTM divisualisasikan dalam bentuk grafik candlestick untuk sepuluh *fold* pengujian (F1–F10). Grafik tersebut menunjukkan distribusi nilai akurasi pada setiap *fold*

sehingga dapat digunakan untuk mengevaluasi tingkat kestabilan dan konsistensi model dalam melakukan klasifikasi pada berbagai pembagian data. Evaluasi ini penting untuk menilai kemampuan generalisasi model ketika diterapkan pada data yang berbeda dari data pelatihan.

Berdasarkan grafik, seluruh *fold* menghasilkan nilai akurasi yang tinggi dengan rentang antara 86,79% hingga 98,10%. Nilai akurasi tertinggi diperoleh pada *fold* F10 sebesar 98,10%, diikuti oleh F8 sebesar 97,14%, serta F2 dan F6 yang masing-masing mencapai 95,28%. Sementara itu, nilai akurasi terendah terdapat pada F1 sebesar 86,79%, diikuti oleh F4 sebesar 87,74%, yang masih menunjukkan performa klasifikasi yang cukup baik. Rentang akurasi yang luas namun tetap dominan tinggi ini menunjukkan bahwa model mampu bekerja secara efektif pada berbagai subset data.

Sebagian besar *fold* memperoleh akurasi di atas 92%, yaitu F2 (95,28%), F3 (94,34%), F5 (94,34%), F6 (95,28%), F7 (92,38%), F8 (97,14%), F9 (94,29%), dan F10 (98,10%). Hasil ini menunjukkan bahwa model memiliki kemampuan yang baik dalam mengenali pola dan karakteristik ulasan pada berbagai pembagian data. Tingginya akurasi tersebut mengindikasikan bahwa kombinasi ADASYN, Word2Vec, dan LSTM mampu menghasilkan representasi data yang efektif untuk proses klasifikasi.

Meskipun demikian, terdapat beberapa *fold* yang menunjukkan performa lebih rendah dibandingkan *fold* lainnya, khususnya F1 (86,79%) dan F4 (87,74%). Perbedaan ini kemungkinan disebabkan oleh variasi distribusi data pada masing-masing *fold*, seperti adanya data yang lebih kompleks atau sulit

diklasifikasikan. Namun demikian, selisih akurasi antar *fold* masih dalam batas wajar dan tidak menunjukkan penurunan performa yang signifikan.

Dari sisi kestabilan, *boxplot* pada setiap *fold* memperlihatkan distribusi nilai yang relatif terkonsentrasi dengan median yang berada pada tingkat akurasi tinggi. Hal ini mengindikasikan bahwa model memiliki konsistensi yang baik selama proses pengujian. Selain itu, tidak terdapat penurunan akurasi yang ekstrem pada *fold* tertentu, sehingga kemampuan generalisasi model dapat dikatakan cukup baik dalam menghadapi variasi data.

Performa tersebut tidak terlepas dari kontribusi metode yang digunakan. ADASYN (*Adaptive Synthetic Sampling*) membantu menyeimbangkan distribusi kelas dengan menghasilkan data sintetis pada area yang sulit dipelajari. Word2Vec mengubah teks menjadi representasi vektor yang mampu menangkap hubungan semantik antar kata, sedangkan LSTM memanfaatkan representasi tersebut untuk memahami pola urutan kata dan konteks kalimat secara mendalam. Kombinasi ketiga metode ini menghasilkan pemodelan yang lebih optimal dalam klasifikasi teks.

Secara keseluruhan, hasil *K-Fold Cross Validation* menunjukkan bahwa pendekatan ADASYN + Word2Vec + LSTM menghasilkan model yang stabil dan memiliki performa sangat baik. Dominasi nilai akurasi di atas 92% serta capaian akurasi maksimum sebesar 98,10% menunjukkan bahwa model mampu melakukan klasifikasi teks secara efektif dan konsisten. Dengan demikian, kombinasi metode ini terbukti mampu meningkatkan kemampuan

deteksi ulasan palsu dan memiliki potensi yang kuat untuk diterapkan pada permasalahan klasifikasi teks serupa.

Langkah selanjutnya yaitu menghitung performance algoritma LSTM untuk *accuracy*, precision, *recal*l dan F1-*Score* dengan menggunakan confuision matrix seperti terlihat pada tabel 6.11 tentang hasil eksperimen ADASYN + Word2Vec + LSTM dengan cofussion matrix.

Gambar 6. 11 Hasil tabel *confunsion matrix*

Pada gambar 6.11 Hasil tabel *confusion matrix* diatas dapat dijelaskan berdasarkan empat komponen utama yang didapatkan pada masing-masing kolom dengan penjelasan yaitu :

- TP (*True Positive*), sebanyak 123 data original yang terklasifikasi benar.
- TN (True Negatif), sebanyak 121 data fake yang terklasifikasi benar.
- FP (*False Positive*), sebanyak 19 data original yang terklasifikasi salah.
- FN (*False Negative*), sebanyak 1 data fake yang terkasifikasi salah.

Dari tabel confusin matrix yang sudah didapat, selanjutnya adalah melakukan perhitungan *accuracy*, *precission*, *recal* dan f1-*Score*, untuk mengukur sejauh mana keberhasilan model dalam mengkhasifikasi data.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (6.9)$$

$$Accuracy = \frac{123+121}{123+121+19+1} = \frac{244}{264} = 0{,}9242$$

$$Precesion = \frac{TP}{TP+FP} \quad (6.10)$$

$$Precesion = \frac{123}{123+19} = \frac{123}{142} = 0{,}8661$$

$$Recall = \frac{TP}{TP+FN} \quad (6.11)$$

$$Recall = \frac{123}{123+1} = \frac{123}{124} = 0{,}9919$$

$$F1 - Score = 2 * \frac{Recall\ x\ Preceision}{Recall+Precision} \quad (6.12)$$

$$F1 - Score = 2 * \frac{0{,}9919x\ 0{,}8661}{0{,}9919 + 0{,}8661} = 2 * \frac{0{,}8591}{1{,}8580} = 0,9247$$

BAB VII

EVALUASI

7.1 Komparasi Skenario Performan Algoritma LSTM

Pada bab ini dilakukan komparasi atau perbandingan terhadap performa model *Long Short-Term Memory* (LSTM) dalam mendeteksi ulasan palsu dengan membandingkan enam skenario eksperimen, yaitu baseline, SMOTE, dan ADASYN pada *feature extraction* IndoBert dan juga baseline, SMOTE, ADASYN pada *feature extraction* Word2Vec. Evaluasi dilakukan menggunakan beberapa metrik kinerja utama, yaitu *accuracy*, precision, *recall*, dan F1-*Score*. Selain itu, pengujian juga mempertimbangkan dua label kelas, yaitu OR (*Original Review*) dan CG (*Computer Generated*), sehingga dapat memberikan gambaran yang lebih komprehensif terhadap kemampuan model dalam menangani masing-masing kelas pada penelitian ini, seperti terlihat pada tabel 7.1. tentang perbandingan performa eksperimen Algoritma LSTM.

Berdasarkan pada tabel 7.1 yang menyajikan perbandingan performa eksperimen model klasifikasi teks menggunakan algoritma LSTM dengan dua pendekatan feature extraction, yaitu IndoBERT dan Word2Vec, serta tiga teknik *oversampling*, yaitu baseline (tanpa *oversampling*), SMOTE, dan ADASYN. Evaluasi dilakukan menggunakan metrik *accuracy*, *precision*, *recall,* dan F1-*Score*. Secara umum, tabel ini memberikan gambaran komprehensif mengenai pengaruh kombinasi teknik terhadap performa model.

Tabel 7. 1 Perbandingan performa eksperimen IndoBert dan Word2Vec

Skenario	Feature Extraction	Teknik *Oversampling*	*Accuracy*	Precision	*Recal*l	F1-*Score*
1	IndoBert	BASELINE	0,7752	0,8739	0,8062	0,8386
2	IndoBert	SMOTE	0,8857	0,9242	0,8472	0,8840
3	IndoBert	ADASYN	0,6928	0,6594	0,7000	0,6791
4	Word2Vec	BASELINE	0,8595	0,6226	0,8684	0,7251
5	Word2Vec	SMOTE	0,9142	0,8766	0,9642	0,9182
6	Word2Vec	ADASYN	0,9242	0,8661	0,9919	0,9247

Berdasarkan Tabel 7.1, pada skenario IndoBERT tanpa teknik *oversampling* (baseline) diperoleh nilai *accuracy* sebesar 0,7752, *precision* 0,8739, *recall* 0,8062, dan F1-*Score* 0,8386. Hasil ini menunjukkan bahwa model memiliki performa yang cukup baik dalam mengidentifikasi ulasan, terutama pada nilai precision yang lebih tinggi dibanding recall. Hal ini mengindikasikan bahwa model cukup akurat dalam memprediksi kelas positif, meskipun masih terdapat beberapa data pada kelas target yang belum terdeteksi secara optimal.

Pada skenario IndoBERT dengan SMOTE, performa model mengalami peningkatan yang signifikan dibandingkan baseline. Nilai *accuracy* meningkat menjadi 0,8857, dengan *precision* 0,9242, *recall* 0,8472, dan F1-*Score* 0,8840. Peningkatan pada hampir seluruh metrik evaluasi menunjukkan bahwa penggunaan SMOTE mampu membantu model dalam mengatasi ketidakseimbangan data. Nilai *precision* yang tinggi menunjukkan bahwa prediksi model menjadi lebih akurat, sedangkan peningkatan F1-*Score* menunjukkan keseimbangan yang lebih baik antara *precision* dan *recall*.

Dengan demikian, SMOTE terbukti efektif dalam meningkatkan performa model berbasis IndoBERT.

Sebaliknya, pada skenario IndoBERT dengan ADASYN, performa model mengalami penurunan. Nilai *accuracy* hanya mencapai 0,6928, dengan *precision* 0,6594, *recall* 0,7000, dan F1-*Score* 0,6791. Penurunan ini menunjukkan bahwa data sintetis yang dihasilkan ADASYN kurang mampu merepresentasikan pola data asli pada embedding IndoBERT, sehingga model mengalami kesulitan dalam melakukan klasifikasi secara optimal.

Pada skenario Word2Vec tanpa *oversampling* (baseline), model memperoleh *accuracy* sebesar 0,8595, *precision* 0,6226, *recall* 0,8684, dan F1-*Score* 0,7251. Hasil ini menunjukkan bahwa Word2Vec mampu menghasilkan akurasi yang cukup tinggi, namun nilai precision yang relatif rendah menunjukkan bahwa masih terdapat cukup banyak prediksi positif yang kurang tepat. Meskipun demikian, nilai *recall* yang tinggi menunjukkan kemampuan model dalam mengenali sebagian besar data pada kelas target.

Pada skenario Word2Vec dengan SMOTE, diperoleh performa yang lebih baik dibandingkan baseline dengan nilai *accuracy* sebesar 0,9142, *precision* 0,8766, *recall* 0,9642, dan F1-*Score* 0,9182. Hasil ini menunjukkan bahwa kombinasi Word2Vec dan SMOTE mampu meningkatkan performa model secara signifikan, terutama pada *recall* dan F1-*Score*. Tingginya nilai *recall* menunjukkan bahwa hampir seluruh data pada kelas target berhasil dikenali oleh model, sementara F1-*Score* yang tinggi menunjukkan keseimbangan yang baik antara *precision* dan *recall*.

Pada skenario Word2Vec dengan ADASYN, model memperoleh hasil terbaik dari seluruh eksperimen dengan *accuracy* sebesar 0,9242, *precision* 0,8661, *recall* 0,9919, dan F1-*Score* 0,9247. Nilai *recall* yang sangat tinggi menunjukkan bahwa hampir seluruh data pada kelas target berhasil dideteksi oleh model. Selain itu, nilai accuracy dan F1-*Score* yang juga tertinggi menunjukkan bahwa kombinasi ini memberikan performa yang paling optimal dalam penelitian.

Berdasarkan perbandingan enam skenario eksperimen tersebut, dapat disimpulkan bahwa teknik *oversampling* memberikan dampak yang berbeda pada masing-masing representasi fitur. Pada kelompok IndoBERT, konfigurasi terbaik diperoleh pada IndoBERT + SMOTE dengan *accuracy* 0,8857 dan F1-*Score* 0,8840, yang menunjukkan bahwa SMOTE lebih efektif dibanding ADASYN pada embedding IndoBERT. Sementara itu, pada kelompok Word2Vec, konfigurasi terbaik diperoleh pada Word2Vec + ADASYN dengan *accuracy* 0,9242 dan F1-*Score* 0,9247, yang menunjukkan bahwa ADASYN lebih unggul dibanding SMOTE pada representasi Word2Vec.

Terkait kinerja metode LSTM, hasil penelitian menunjukkan bahwa model mampu bekerja dengan baik pada kedua representasi fitur, baik IndoBERT maupun Word2Vec. Namun, berdasarkan nilai *accuracy* dan F1-*Score* tertinggi, kombinasi Word2Vec + ADASYN + LSTM menjadi konfigurasi yang paling optimal dalam penelitian ini. Temuan ini menunjukkan bahwa keberhasilan model LSTM tidak hanya dipengaruhi oleh jenis embedding yang digunakan, tetapi juga sangat bergantung pada pemilihan teknik

penyeimbangan data yang sesuai. Dengan demikian, kombinasi Word2Vec dan ADASYN menjadi pendekatan yang paling efektif untuk meningkatkan performa deteksi ulasan pada penelitian ini.

7.2 *Fake review* Detection terhadap Ulasan Aplikasi *Marketplace* dalam pandangan Al Qur'an

Perkembangan teknologi informasi dan perdagangan elektronik (*e-commerce*) telah mengubah pola interaksi masyarakat dalam melakukan transaksi jual beli. Salah satu fitur utama dalam platform *marketplace* adalah sistem ulasan (*review*) yang memungkinkan konsumen memberikan penilaian terhadap produk atau layanan yang telah digunakan. Ulasan tersebut berperan sebagai sumber informasi penting bagi calon konsumen dalam proses pengambilan keputusan. Namun, dalam praktiknya, tidak sedikit ditemukan ulasan palsu (*fake review*) yang dibuat dengan tujuan meningkatkan citra produk secara tidak objektif atau menjatuhkan reputasi pesaing. Keberadaan *fake review* ini berpotensi menyesatkan konsumen serta menurunkan tingkat kepercayaan terhadap sistem perdagangan digital.

Dalam perspektif Islam, penyampaian informasi yang tidak didasarkan pada kebenaran merupakan perbuatan yang dilarang karena dapat menimbulkan dampak negatif bagi individu maupun masyarakat. Prinsip ini tercermin dalam QS. An-Nur ayat 15:

إِذْ تَلَقَّوْنَهُۥ بِأَلْسِنَتِكُمْ وَتَقُولُونَ بِأَفْوَاهِكُم مَّا لَيْسَ لَكُم بِهِۦ عِلْمٌ وَتَحْسَبُونَهُۥ هَيِّنًا وَهُوَ عِندَ ٱللَّهِ عَظِيمٌ

Artinya : (Ingatlah) ketika kamu menerima (berita bohong) itu dari mulut ke mulut; kamu mengatakan dengan mulutmu apa yang tidak kamu ketahui sedikit pun; dan kamu menganggapnya remeh, padahal dalam pandangan Allah itu masalah besar. (QS. An-Nur Ayat 15).

Ayat ini menurut para mufasir seperti Ibnu Katsir menjelaskan peringatan keras terhadap perilaku menyebarkan informasi tanpa verifikasi (tabayyun). Peristiwa yang melatarbelakangi ayat ini berkaitan dengan penyebaran berita bohong (hadits al-ifk) yang diterima dan disebarkan secara berantai tanpa klarifikasi. Dalam konteks modern, termasuk marketplace, ayat ini mengandung prinsip bahwa setiap informasi—termasuk ulasan produk—harus didasarkan pada pengetahuan yang benar. Menyampaikan ulasan palsu berarti menyebarkan informasi yang tidak diketahui kebenarannya, yang dalam pandangan Allah merupakan dosa besar meskipun dianggap sepele oleh manusia. Prinsip ini juga diperkuat oleh hadis Rasulullah :

كَفَى بِالْمَرْءِ كَذِبًا أَنْ يُحَدِّثَ بِكُلِّ مَا سَمِعَ

"Cukuplah seseorang dianggap berdusta apabila ia menceritakan setiap apa yang ia dengar" (HR. Muslim).

Hadis tersebut menegaskan bahwa seorang Muslim wajib melakukan verifikasi terhadap informasi sebelum menyampaikan atau menyebarkannya. Oleh karena itu, praktik pemberian ulasan palsu (*fake review*) bertentangan dengan nilai-nilai Islam karena mengandung unsur penyampaian informasi yang tidak dapat dipertanggungjawabkan kebenarannya serta berpotensi merugikan pihak lain. (Ibnu Katsir, Tafsir al-Qur'an al-'Azhim, tafsir QS. An-Nur [24]: 15; HR. Muslim No. 5).

Secara akademik, ayat ini dapat dimaknai sebagai larangan terhadap penyebaran informasi yang tidak terverifikasi. Dalam konteks marketplace, hal ini menunjukkan bahwa setiap ulasan yang disampaikan seharusnya bersifat valid, objektif, dan dapat dipertanggungjawabkan. Dengan demikian, keberadaan *fake review* bertentangan dengan prinsip kehati-hatian dalam menerima dan menyampaikan informasi.

Selain itu, praktik *fake review* juga dapat dikategorikan sebagai bentuk kecurangan dalam aktivitas bisnis. Dalam QS. Asy-Syu'ara ayat 181–183 :

أَوْفُوا الْكَيْلَ وَلَا تَكُونُوا مِنَ الْمُخْسِرِينَ ﴿١٨١﴾ وَزِنُوا بِالْقِسْطَاسِ الْمُسْتَقِيمِ ﴿١٨٢﴾ وَلَا تَبْخَسُوا النَّاسَ أَشْيَاءَهُمْ وَلَا تَعْثَوْا فِي الْأَرْضِ مُفْسِدِينَ ﴿١٨٣﴾

Artinya : Sempurnakanlah takaran dan janganlah kamu termasuk orang-orang yang merugikan. Dan timbanglah dengan timbangan yang benar. Dan janganlah kamu merugikan manusia pada hak-haknya dan janganlah kamu merajalela di bumi dengan membuat kerusakan.. (QS. Asy-Syu'ara ayat 181-183)

Dalam tafsir klasik seperti Al-Qurthubi dan At-Thabari, ayat ini ditujukan kepada kaum Nabi Syu'aib yang melakukan kecurangan dalam takaran dan timbangan. Namun, maknanya bersifat universal dan mencakup seluruh bentuk ketidakjujuran dalam transaksi ekonomi. "Takaran" dan "timbangan" tidak hanya dimaknai secara fisik, tetapi juga mencakup keadilan dalam informasi dan nilai yang disampaikan kepada pihak lain. Dalam konteks e-commerce, *fake review* dapat dipandang sebagai bentuk "pengurangan takaran" dalam informasi karena konsumen tidak memperoleh gambaran produk yang sebenarnya. Hal ini berpotensi merugikan pihak lain dan melanggar prinsip

keadilan dalam muamalah. Prinsip tersebut juga diperkuat oleh sabda Rasulullah : "Barang siapa menipu kami, maka ia bukan termasuk golongan kami" (HR. Muslim No. 102). Hadis ini menegaskan bahwa segala bentuk penipuan dan penyampaian informasi yang tidak sesuai dengan kenyataan merupakan perbuatan yang dilarang dalam Islam. Oleh karena itu, praktik pemberian ulasan palsu (*fake review*) yang bertujuan memengaruhi keputusan konsumen secara tidak jujur bertentangan dengan prinsip kejujuran (shidq), amanah, dan keadilan yang menjadi landasan utama dalam etika bisnis Islam.Secara konseptual, ayat ini menegaskan pentingnya keadilan dan kejujuran dalam transaksi ekonomi. Dalam konteks e-commerce, *fake review* merupakan bentuk manipulasi informasi yang dapat mempengaruhi persepsi konsumen secara tidak adil, sehingga berpotensi menghasilkan keputusan pembelian yang tidak berdasarkan kondisi yang sebenarnya. (At-Thabari, Jami' al-Bayan, tafsir QS. Asy-Syu'ara [26]: 181–183; HR. Muslim No. 102).

Lebih lanjut, Islam juga menekankan pentingnya kejujuran dalam komunikasi dan penyampaian informasi. Hal ini tercermin dalam QS. Al-Ahzab ayat 70 :

يَا أَيُّهَا الَّذِينَ آمَنُوا اتَّقُوا اللَّهَ وَقُولُوا قَوْلًا سَدِيدًا

Artinya : Wahai orang-orang yang beriman, bertakwalah kamu kepada Allah dan ucapkanlah perkataan yang benar. (QS. Al Ahzab ayat 70)

Menurut tafsir Ibnu Katsir, ayat ini mengandung perintah untuk senantiasa berkata benar (qaulan sadidan), yaitu perkataan yang lurus, jujur, dan tidak menyimpang dari fakta. Perkataan yang benar tidak hanya terbatas pada ucapan lisan, tetapi juga mencakup seluruh bentuk komunikasi, termasuk tulisan

seperti ulasan di marketplace. Kejujuran dalam berkata merupakan bagian dari ketakwaan kepada Allah dan menjadi dasar dalam membangun kepercayaan sosial. Dalam konteks ini, *fake review* jelas bertentangan dengan perintah tersebut karena mengandung unsur kebohongan dan manipulasi informasi. Prinsip berkata benar ini juga diperkuat oleh sabda Rasulullah : "Hendaklah kalian berlaku jujur, karena sesungguhnya kejujuran membawa kepada kebaikan, dan kebaikan membawa ke surga. Seseorang yang senantiasa berkata jujur dan berusaha berlaku jujur akan dicatat di sisi Allah sebagai orang yang jujur". Hadis tersebut menegaskan bahwa kejujuran merupakan nilai fundamental dalam Islam yang harus diterapkan dalam setiap bentuk komunikasi dan penyampaian informasi. Oleh karena itu, praktik pemberian ulasan palsu (*fake review*) tidak hanya merugikan konsumen, tetapi juga bertentangan dengan prinsip qaulan sadidan yang mengharuskan setiap Muslim menyampaikan informasi secara benar, jujur, dan dapat dipertanggungjawabkan. (Ibnu Katsir, Tafsir al-Qur'an al-'Azhim, tafsir QS. Al-Ahzab [33]: 70, HR. Al-Bukhari No. 6094; Muslim No. 2607)

Dalam perspektif akademik, ayat ini dapat diinterpretasikan sebagai landasan etis dalam penyampaian informasi yang akurat dan dapat dipercaya. Oleh karena itu, praktik *fake review* dapat dikategorikan sebagai bentuk penyimpangan terhadap nilai kejujuran karena mengandung unsur informasi yang tidak sesuai dengan realitas.

Berdasarkan uraian sebelumnya, fenomena *fake review* tidak hanya menjadi permasalahan dalam bidang teknologi informasi dan bisnis digital,

tetapi juga memiliki implikasi etis dan normatif dalam perspektif Islam. Keberadaan ulasan palsu berpotensi menyesatkan konsumen, merusak mekanisme pasar yang adil, serta menurunkan tingkat kepercayaan terhadap sistem perdagangan digital. Oleh karena itu, penelitian mengenai deteksi *fake review* menjadi penting untuk dikembangkan, tidak hanya dalam rangka meningkatkan kualitas sistem e-commerce, tetapi juga sebagai upaya implementasi nilai-nilai etika Islam dalam penyebaran informasi di era digital.

2. Hubungan dengan Etika Bisnis Islam

Etika bisnis Islam dibangun atas prinsip-prinsip fundamental seperti kejujuran (ṣidq), kepercayaan (amanah), keadilan (‘adl), serta tanggung jawab (mas’uliyyah) dalam setiap aktivitas ekonomi (Hassan & Aliyu, 2020; Huda et al., 2021). Prinsip-prinsip tersebut tetap relevan dalam konteks perdagangan digital, di mana interaksi antara penjual dan pembeli tidak lagi dilakukan secara langsung, melainkan melalui perantara sistem berbasis teknologi.

Prinsip kejujuran (ṣidq) dalam Islam dapat dirujuk pada QS. At-Taubah ayat 119:

يَا أَيُّهَا الَّذِينَ آمَنُوا اتَّقُوا اللَّهَ وَكُونُوا مَعَ الصَّادِقِينَ

Artinya: “Wahai orang-orang yang beriman, bertakwalah kepada Allah dan hendaklah kamu bersama orang-orang yang jujur.” (QS. At-Taubah : 119).

Tafsir Ayat ini menegaskan pentingnya bersikap jujur dan bergaul dengan orang-orang yang memiliki integritas tinggi (ṣādiqīn). Menurut para mufasir, kejujuran tidak hanya terbatas pada ucapan, tetapi juga mencakup perbuatan dan niat. Dalam konteks e-commerce, ulasan yang diberikan oleh pengguna harus mencerminkan kondisi yang sebenarnya. Praktik *fake review*

bertentangan dengan nilai ini karena mengandung unsur kebohongan dan manipulasi informasi (Rahman et al., 2022). Prinsip untuk bersama orang-orang yang jujur juga diperkuat oleh sabda Rasulullah: "Seseorang itu mengikuti agama (kebiasaan) teman dekatnya, maka hendaklah salah seorang di antara kalian memperhatikan dengan siapa ia berteman". Hadis ini menunjukkan bahwa lingkungan pergaulan memiliki pengaruh besar terhadap perilaku dan karakter seseorang, termasuk dalam menjaga kejujuran dan integritas. Oleh karena itu, dalam ekosistem perdagangan digital diperlukan budaya kejujuran dan transparansi agar informasi yang beredar, termasuk ulasan produk, dapat dipercaya serta tidak menyesatkan konsumen. Dengan demikian, upaya mendeteksi dan mencegah *fake review* sejalan dengan nilai Islam yang mendorong umatnya untuk berpegang pada kejujuran dan berada dalam lingkungan yang menjunjung tinggi kebenaran. (HR. Abu Dawud No. 4833; HR. At-Tirmidzi No. 2378).

Selain itu, prinsip amanah (kepercayaan) dijelaskan dalam QS. An-Nisa ayat 58:

إِنَّ اللَّهَ يَأْمُرُكُمْ أَنْ تُؤَدُّوا الْأَمَانَاتِ إِلَىٰ أَهْلِهَا وَإِذَا حَكَمْتُمْ بَيْنَ النَّاسِ أَنْ تَحْكُمُوا بِالْعَدْلِ ۚ إِنَّ اللَّهَ نِعِمَّا يَعِظُكُمْ بِهِ ۗ إِنَّ اللَّهَ كَانَ سَمِيعًا بَصِيرًا

Artinya: ‘"Sesungguhnya Allah menyuruh kamu menyampaikan amanat kepada yang berhak menerimanya, dan apabila kamu menetapkan hukum di antara manusia hendaklah kamu menetapkannya dengan adil. Sungguh, Allah sebaik-baik yang memberi pengajaran kepadamu. Sungguh, Allah Maha Mendengar lagi Maha Melihat." (QS. An-Nisa [4]: 58).

Tafsir ayat ini mengandung makna bahwa setiap individu memiliki tanggung jawab moral dalam menyampaikan sesuatu secara benar dan adil.

Dalam konteks marketplace, ulasan merupakan bentuk amanah informasi kepada konsumen lain. Oleh karena itu, penyampaian ulasan palsu dapat dikategorikan sebagai bentuk pengkhianatan terhadap amanah karena berpotensi merugikan pihak lain (Azmi et al., 2023). Prinsip amanah tersebut juga diperkuat oleh sabda Rasulullah ﷺ: "Tunaikanlah amanah kepada orang yang mempercayaimu dan janganlah kamu mengkhianati orang yang mengkhianatimu" (HR. Abu Dawud No. 3534; HR. At-Tirmidzi No. 1264). Hadis ini menegaskan bahwa setiap amanah yang diberikan harus dijaga dan disampaikan dengan penuh tanggung jawab. Dalam transaksi digital, ulasan yang diberikan oleh pengguna merupakan bentuk amanah informasi yang dapat memengaruhi keputusan konsumen lain. Oleh karena itu, praktik *fake review* bertentangan dengan prinsip amanah dalam Islam karena menyajikan informasi yang tidak sesuai dengan kenyataan dan berpotensi menimbulkan ketidakadilan serta kerugian bagi pihak lain.

Lebih lanjut, prinsip keadilan ('adl) dalam Islam mengharuskan setiap transaksi dilakukan secara transparan dan tidak merugikan pihak lain. Praktik *fake review* dapat dikategorikan sebagai bentuk gharar (ketidakjelasan informasi) dan tadlis (penipuan), yang secara tegas dilarang dalam Islam (Ismail & Rashid, 2021). Hal ini menunjukkan bahwa fenomena *fake review* tidak hanya berdampak secara ekonomi, tetapi juga bertentangan dengan nilai-nilai moral dalam Islam.

Dalam konteks teknologi modern, penerapan sistem deteksi *fake review* berbasis kecerdasan buatan (Artificial Intelligence), seperti algoritma deep

learning *Long Short-Term Memory* (LSTM), menjadi solusi yang efektif dalam menjaga integritas sistem ulasan (Li et al., 2021; Kumar & Gupta, 2024). Teknologi ini mampu mengidentifikasi pola bahasa dan perilaku yang mencurigakan dalam ulasan sehingga dapat membantu meminimalkan penyebaran informasi yang tidak valid.

3. Relevansi Penelitian dalam Perdagangan Digital (*E-Commerce*)

Perkembangan e-commerce yang pesat menjadikan marketplace sebagai salah satu pilar utama dalam ekonomi digital. Dalam ekosistem ini, ulasan konsumen (online *review*s) memiliki peran penting dalam mempengaruhi keputusan pembelian. Penelitian terbaru menunjukkan bahwa mayoritas konsumen sangat bergantung pada ulasan sebelum melakukan transaksi, sehingga kualitas dan keakuratan informasi menjadi faktor krusial (Chen et al., 2022; Wang et al., 2023).

Namun demikian, meningkatnya jumlah *fake review* menjadi tantangan serius bagi keberlanjutan sistem e-commerce. Ulasan yang tidak autentik dapat menurunkan kepercayaan pengguna terhadap platform, serta menciptakan distorsi dalam persaingan pasar (Jiang et al., 2024). Oleh karena itu, pengembangan sistem deteksi *fake review* menjadi sangat relevan dalam menjaga kredibilitas marketplace.

Dalam perspektif Islam, menjaga kepercayaan dalam transaksi merupakan bagian dari nilai moral yang harus dijunjung tinggi. Allah SWT berfirman dalam QS. Al Baqarah ayat 283 :

وَإِنْ أَمِنَ بَعْضُكُمْ بَعْضًا فَلْيُؤَدِّ الَّذِي اؤْتُمِنَ أَمَانَتَهُ وَلْيَتَّقِ اللَّهَ رَبَّهُ

Artinya: "*Jika sebagian kamu mempercayai sebagian yang lain, maka hendaklah yang dipercayai itu menunaikan amanatnya (utangnya) dan hendaklah dia bertakwa kepada Allah, Tuhannya."* (QS. Al-Baqarah [2]: 283).

Hal ini juga diperkuat oleh hadis Nabi Muhammad SAW:

التَّاجِرُ الصَّدُوقُ الْأَمِينُ مَعَ النَّبِيِّينَ وَالصِّدِّيقِينَ وَالشُّهَدَاءِ

"Pedagang yang jujur dan terpercaya akan bersama para nabi, orang-orang yang benar (ṣiddiqin), dan para syuhada." (HR. Tirmidzi).

Hadis ini menunjukkan tingginya kedudukan pedagang yang menjunjung nilai kejujuran dan amanah dalam menjalankan aktivitas ekonomi. Dalam konteks perdagangan elektronik (*e-commerce),* kejujuran dalam memberikan ulasan produk merupakan bentuk amanah informasi yang harus dijaga. Oleh karena itu, praktik *fake review* bertentangan dengan nilai amanah dan kepercayaan karena dapat menyesatkan konsumen serta merusak integritas transaksi digital. (QS. Al-Baqarah : 283; HR. At-Tirmidzi No. 1209)

Dengan demikian, penelitian ini memiliki kontribusi ganda, yaitu dalam bidang teknologi informasi melalui pengembangan model deteksi ulasan palsu, serta dalam bidang etika bisnis Islam melalui penguatan nilai kejujuran, keadilan, dan tanggung jawab. Pengembangan sistem *Fake review Detection* pada aplikasi *marketplace* diharapkan mampu menciptakan lingkungan transaksi yang lebih adil, transparan, dan terpercaya, sejalan dengan prinsip-prinsip yang diajarkan dalam Al-Qur'an.

BAB VIII

KESIMPULAN

8.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan mulai dari identifikasi masalah, studi pustaka, perancangan sistem, *implementasi*, hingga pengujian model deteksi ulasan palsu (*fake review*) pada *marketplace* Tokopedia menggunakan metode *Long Short-Term Memory* (LSTM) dengan fitur ekstraksi IndoBERT dan Word2Vec, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Penelitian ini berhasil membangun sistem deteksi ulasan palsu berbahasa Indonesia melalui tahapan *preprocessing* yang meliputi *case folding*, *punctuation removal*, *word normalization*, dan *stemming*. Selanjutnya data direpresentasikan menggunakan IndoBERT dan Word2Vec, kemudian diklasifikasikan menggunakan metode LSTM. Sistem yang dibangun mampu mengklasifikasikan ulasan ke dalam dua kategori, yaitu Original *Review* (OR) dan *Computer Generated Review* (CG).
2. Hasil penelitian menunjukkan bahwa teknik *oversampling* memberikan pengaruh yang berbeda terhadap masing-masing representasi fitur. Pada fitur IndoBERT, metode SMOTE terbukti mampu meningkatkan performa model dibandingkan baseline, dengan nilai *accuracy* meningkat dari 77,52% menjadi 88,57%. Sebaliknya, penggunaan ADASYN justru menurunkan performa menjadi 69,28%.

3. Pada fitur Word2Vec, penerapan teknik *oversampling* juga memberikan peningkatan performa yang cukup signifikan. Baseline Word2Vec memperoleh *accuracy* sebesar 85,95%, kemudian meningkat menjadi 91,42% menggunakan SMOTE, dan mencapai hasil terbaik sebesar 92,42% menggunakan ADASYN.
4. Berdasarkan perbandingan enam skenario pengujian, konfigurasi terbaik diperoleh pada kombinasi Word2Vec + ADASYN + LSTM dengan nilai *accuracy* 92,42%, *precision* 86,61%, *recall* 99,19%, dan F1-*Score* 92,47%. Hasil ini menunjukkan bahwa kombinasi tersebut paling optimal dalam mendeteksi ulasan palsu pada dataset penelitian.
5. Pada kelompok fitur IndoBERT, konfigurasi terbaik diperoleh pada IndoBERT + SMOTE + LSTM dengan nilai *accuracy* 88,57%, *precision* 92,42%, *recall* 84,72%, dan F1-*Score* 88,40%. Hal ini menunjukkan bahwa LSTM mampu memanfaatkan representasi kontekstual dari IndoBERT dengan cukup baik.
6. Berdasarkan hasil analisis, tujuan penelitian ini telah tercapai, yaitu untuk mengetahui fitur representasi teks yang paling optimal dalam deteksi *fake review* serta mengukur kinerja metode LSTM. Dari hasil eksperimen, Word2Vec terbukti lebih optimal dibandingkan IndoBERT pada dataset penelitian ini karena menghasilkan performa klasifikasi yang lebih tinggi dan lebih stabil.
7. Dari perspektif etika bisnis Islam, penelitian ini mendukung implementasi nilai kejujuran (*ṣidq*), keadilan (‘*adl*), dan tabayun dalam transaksi digital.

Sistem deteksi ulasan palsu yang dikembangkan dapat membantu meminimalkan praktik manipulasi informasi yang berpotensi merugikan konsumen, serta mendukung terciptanya ekosistem *e-commerce* yang lebih transparan, aman, dan terpercaya sesuai prinsip muamalah Islam.

8.2 Saran

Berdasarkan hasil penelitian yang telah diperoleh, beberapa saran untuk pengembangan penelitian selanjutnya adalah sebagai berikut:

1. Penelitian selanjutnya dapat menggunakan dataset yang lebih besar dan lebih beragam, baik dari sisi jumlah data, kategori produk, maupun platform *e-commerce* lain seperti Shopee, Lazada, dan Bukalapak agar model memiliki kemampuan generalisasi yang lebih baik.
2. Penelitian berikutnya dapat mengeksplorasi model *deep learning* yang lebih kompleks seperti Bidirectional LSTM (BiLSTM), GRU, CNN-LSTM, maupun model transformer modern seperti RoBERTa, IndoBERT Large, atau XLM-RoBERTa untuk memperoleh performa yang lebih optimal.
3. Selain fitur teks, penelitian mendatang dapat mengintegrasikan fitur non-teks seperti rating produk, metadata pengguna, pola aktivitas akun, dan histori ulasan agar sistem deteksi dapat bekerja lebih komprehensif.
4. Perlu dilakukan pengujian pada teknik penanganan ketidakseimbangan data lainnya seperti Borderline-SMOTE, *Random Oversampling*, Cost-

Sensitive Learning, maupun Hybrid Sampling untuk mengetahui metode yang paling sesuai pada karakteristik data ulasan berbahasa Indonesia.

5. Penelitian selanjutnya dapat mengembangkan sistem deteksi *fake review* secara real-time yang dapat diintegrasikan langsung pada platform *marketplace*, sehingga proses identifikasi ulasan palsu dapat dilakukan secara otomatis sebelum ulasan dipublikasikan.
6. Pengembangan penelitian berbasis multimodal yang menggabungkan teks, gambar, video, dan metadata transaksi dapat menjadi arah penelitian yang menjanjikan mengingat bentuk manipulasi informasi pada platform digital semakin kompleks.
7. Dari sisi implementasi praktis, hasil penelitian ini dapat dimanfaatkan oleh pengelola *marketplace* sebagai mekanisme pendukung untuk meningkatkan kualitas sistem ulasan, menjaga kepercayaan konsumen, dan menciptakan lingkungan perdagangan elektronik yang lebih jujur, adil, dan berkelanjutan sesuai prinsip etika bisnis Islam.

DAFTAR PUSTAKA

Joni Salminen, Chandrashekhar Kandpal, Ahmed Mohamed Kamel, Soon-gyo Jung, Bernard J. Jansen. 2022. Creating and detecting *fake review* of online products. *Journal of Retailing and Consumer Services 64*. doi.org/10.1016/j.jretconser.2021.102771.

Nour Qandos, Ghadir Hamad, Maitha Alharbi, Shatha Alturki, Waad Alharbi, Arwa A. Albelaihi. 2024. Multiscale cascaded domain-based approach for Arabic *fake review*s detection in e-commerce platforms. *Journal of King Saud University - Computer and Information Sciences* 36. doi.org/10.1016/j.jksuci.2024.101926.

Richa Gupta, Indu Kashyap, Vinita Jindal. 2024. SBiLM: Siamese Bi-LSTM model for handling imbalance in *fake review* detection. *International Conference on Machine Learning and Data Engineering* 235: 1157-1166. DOI: 10.1016/j.procs.2024.04.110.

Jinhao Liu, Pei Quan, Wen Zhang. 2024. A Study on *Fake review* Detection Based on RoBERTa and Behavioral Features. *International Conference on Information Technology and Quantitative Management* 242: 1323-1330. DOI: 10.1016/j.procs.2024.08.131.

Maged Nasser, Noreen Izza Arshad, Abdulalem Ali, Hitham Alhussian, Faisal Saeed, Aminu Da'u, Ibtehal Nafea. 2025. A systematic *review* of multimodal fake news detection on social media using deep learning models. *Results in Engineering* 26. doi.org/10.1016/j.rineng.2025.104752.

Rami Mohawesh, Haythem Bany Salameh, Yaser Jararweh, Mohannad Alkhalaileh, Sumbal Maqsood. 2024. *Fake review* detection using transformer-based enhanced LSTM and RoBERTa. *International Journal of Cognitive Computing in Engineering* 5: 250-258. doi.org/10.1016/j.ijcce.2024.06.001.

Ammar Aod, Muhammad Hamza Farooq, Amna Zafar, Beenish Ayesha Akram, Ruogu Zhou, And Feng Dong. 2024. Fake News Classification Methodology with Enhanced BERT. *the Natural Science Foundation of Hunan Province Project under Grant*. DOI: 10.1109/ACCESS.2024.3491376. E-ISSN: 2169-3536.

Gulsum Kayabasi Koru, And Celebi Uluyol. 2024. Detection of Turkish Fake News from Tweets with BERT Models. Doi: 10.1109/ACCESS.2024.3354165. E-ISSN: 2169-3536.

Juanjuan Peng. 2024. Identification of Fake Comments in e-commerce Based on Triplet Convolutional Twin Networkvand CatBoost Model. *the Philosophy and Social Science Research Project of Jiangsu Universities.* DOI : 10.1109/ACCESS.2024.3520717. E-ISSN: 2169-3536.

Nishant rai, Deepika Kumar, Naman Kaushik, Chandan Raj, Ahad Ali. 2022. Fake News Classification using transformer based enhanced LSTM and BERT. *International Journal of Cognitive Computing in Engineering* 3: 98-105. doi.org/10.1016/j.ijcce.2022.03.003.

Anna Clazkova. 2020. A Comparison of Synthetic *Oversampling* Methods for Multi-class Text Classification. *the Russian Foundation for Basic Research*. Doi:10.48550/arXiv.2008.04636.

Mohammed Zakariah, Salman A. AlQahtani, Mabrook S. Al-Rakhami. 2023. Machine Learning-Based Adaptive Synthetic Sampling Technique for Intrusion Detection. *Appl. Sci.* 13, 6504. doi.org/10.3390/app13116504.

Maysam Jalal Abd, and Mohsin Hasan Husein. 2024. *Fake review*s detection in e-commerce using machine learning techniques: a comparative survey. *BIO Web of Conferences* 97, 00099. https://doi.org/10.1051/bioconf/20249700099.

Pengfei Sun, Weihong Bi, Yifan Zhang, Qiuyu Wang, Feifei Kou, Tongwei Lu, Jinpeng Chen. 2024. *Fake review* Detection Model Based on Comment Content and *Review* Behavior. *Electronics*, 13, 4322. https://doi.org/10.3390/electronics13214322.

Habib Alamsyah, Yana Cahyana, Adi Rizky Pratama. 2023. Deteksi *Fake review* Menggunakan Metode Support Vector Machine dan Naïve Bayes Di Tokopedia. *Jurnal Ilmiah Teknik Informatika dan Sistem Informasi.* E-ISSN : 2685-0893.

Khoirotulmuadiba Purifyregalia, Khothibul Umam, Nur Cahyo Hendro Wibowo, Maya Rini Handayani. 2025. Detecting *Fake review*s in E-Commerce: A Case Study on Shopee Using Support Vector Machine and Random Forest. Journal of Applied Informatics and Computing. E-ISSN : 2548-6861.

Jiang, C., Zhang, X., & Jin, A. (2020). Detecting online *fake review*s via hierarchical neural networks and multivariate features. Neural Information Processing Conference (ICONIP).

Mohawesh, R., Tran, S., Ollington, R., & Xu, S. (2020). Analysis of concept drift in *fake review*s detection. Expert Systems with Applications.

Abri, F., Gutierrez, L. F., Namin, A. S., et al. (2020). *Fake review*s detection through analysis of linguistic features. Journal of Information Security and Applications.

Neisari, A., Rueda, L., & Saad, S. (2021). Spam *review* detection using self-organizing maps and convolutional neural networks. Computers & Security.

Khan, H., Asghar, M. U., Srivastava, G., et al. (2021). *Fake review* classification using supervised machine learning. Pattern Recognition Workshops.

Mohawesh, R., Xu, S., Springer, M., et al. (2021). *Fake review*s detection: A survey. IEEE Access.

Salunkhe, A. (2021). Attention-based Bidirectional LSTM for deceptive opinion spam classification. International Conference on Artificial Intelligence Research.

Wang, N., Gao, Y., et al. (2022). Research on false *review* detection methods: A state-of-the-art *review*. Journal of King Saud University – Computer and Information Sciences.

Singh, I., Chandel, A., et al. (2022). Deep learning approaches for online *review* spam detection. Information Sciences.

Wang, X., et al. (2022). Fraudulent *review* detection model focusing on emotional expressions and explicit aspects. Decision Support Systems.

Yousif, A., & Buckley, J. (2022). Impact of sentiment analysis in *fake review* detection. Journal of Big Data Analytics.

Mohawesh, R., Al-Hawawreh, M., Maqsood, S., et al. (2023). Learning textual representations for fake online *review* detection. Cluster Computing.

Lu, J., Zhan, X., Liu, G., et al. (2023). *Fake review* detection model based on pre-trained language models and CNN. Electronics.

Kumar, V., et al. (2023). Artificial intelligence applications in *fake review* detection: Bibliometric analysis and research trends. Journal of Business Research.

Fake review detection in e-commerce platforms using aspect-based sentiment analysis. (2023). Journal of Business Research.

Mohawesh, R., Maqsood, S., Jararweh, Y., et al. (2023). Explainable ensemble multi-view deep learning model for *fake review* detection. Journal of King Saud University – Computer and Information Sciences.

Krishnamoorthy, P., Sathiyanarayanan, M., & Proença, H. (2024). Hybrid deep neural network for text classification and detection tasks. International Journal of Cognitive Computing in Engineering.

Cheng, L.-C., Wu, Y.-T., et al. (2024). Detecting *fake review*ers using graph neural networks. Decision Support Systems.

Bigdeli, F. (2025). Cross-platform *fake review* detection: Comparative analysis of machine learning models. International Journal of Information Technology and Computer Science.

An analysis of graph neural networks for *fake review* detection: A systematic literature *review*. (2025). Neurocomputing.

Azmi, F., Rahman, A., & Yusuf, M. (2023). Islamic Business Ethics in Digital Marketplace: A Study on Trust and Transparency. Journal of Islamic Economics, 15(2), 120–135.

Chen, L., Wang, X., & Li, Y. (2022). The Impact of Online *Review*s on Consumer Behavior in E-Commerce. Electronic Commerce Research and Applications, 52.

Hassan, M. K., & Aliyu, S. (2020). A Contemporary Survey of Islamic Banking Literature. Journal of Financial Stability, 46.

Huda, N., Rini, N., & Mardoni, Y. (2021). Etika Bisnis Islam di Era Digital. Jurnal Ekonomi Syariah Indonesia, 11(1), 45–60.

Ismail, A., & Rashid, M. (2021). Gharar and Fraud in Modern Business Transactions. International Journal of Islamic Finance, 13(1), 67–82.

Jiang, Q., Liu, H., & Zhang, T. (2024). Detecting *Fake review*s in E-Commerce Using Deep Learning Models. IEEE Access, 12.

Kumar, S., & Gupta, R. (2024). Deep Learning Approaches for *Fake review* Detection. Expert Systems with Applications, 235.

Li, F., Huang, M., Yang, Y., & Zhu, X. (2021). *Fake review* Detection Using LSTM. Knowledge-Based Systems, 212.

Rahman, M., Karim, R., & Islam, S. (2022). Ethical Implications of *Fake review*s. Journal of Business Ethics, 180(3).

Wang, Y., Chen, H., & Zhao, L. (2023). Trust in Online Marketplace *Review*s. Information Systems Frontiers, 25(4).