ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - SM234801

PREDIKSI KETIDAKSELARASAN JUDUL DAN ISI DARI BERITA CLICKBAIT BERBAHASA INDONESIA MENGGUNAKAN MODEL INDO BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS (INDOBERT)

I GEDE FEBRY ABDI SAPUTRA
NRP 5002201008

Dosen Pembimbing
Prof. Dr. techn. Drs. Mohammad Isa Irawan, MT
NIP 19631225 198903 1 001

Program Studi Sarjana
Departemen Matematika
Fakultas Sains dan Analitika Data
Institut Teknologi Sepuluh Nopember
Surabaya
2024

TUGAS AKHIR - SM234801

PREDIKSI KETIDAKSELARASAN JUDUL DAN ISI DARI BERITA CLICKBAIT BERBAHASA INDONESIA MENGGUNAKAN MODEL INDO BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS (INDOBERT)

I GEDE FEBRY ABDI SAPUTRA

NRP 5002201008

Dosen Pembimbing

Prof. Dr. techn. Drs. Mohammad Isa Irawan, MT

NIP 19631225 198903 1 001

Program Sarjana

Departemen Matematika

Fakultas Sains dan Analitika Data

Institut Teknologi Sepuluh Nopember

Surabaya

2024

FINAL PROJECT - SM234801

PREDICTING TITLE AND CONTENT MISALIGNMENT FROM CLICKBAIT NEWS IN INDONESIAN USING INDO BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS (INDOBERT) MODEL

I GEDE FEBRY ABDI SAPUTRA

NRP 5002201008

Supervisor

Prof. Dr. techn. Drs. Mohammad Isa Irawan, MT

NIP 19631225 198903 1 001

Bachelor Program

Departement of Mathematics

Faculty of Scientics

Institut Teknologi Sepuluh Nopember

Surabaya

2024

LEMBAR PENGESAHAN

PREDIKSI KETIDAKSELARASAN JUDUL DAN ISI DARI BERITA CLICKBAIT BERBAHASA INDONESIA MENGGUNAKAN MODEL INDO BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS (INDOBERT)

TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat
memperoleh gelar Sarjana Matematika pada
Program Studi S-1 Matematika
Departemen Matematika
Fakultas Sains dan Analitika Data
Institut Teknologi Sepuluh Nopember

Oleh: **I GEDE FEBRY ABDI SAPUTRA**
NRP. 5002201008

Surabaya, Juli 2024
Disetujui oleh Tim Penguji Tugas Akhir:

Pembimbing

1. Prof. Dr. techn. Drs. Mohammad Isa Irawan, MT (.....)
   NIP. 19631225 198903 1 001

Penguji

1. Alvida Mustika Rukmi, S.Si., M.Si. (.....)
   NIP. 19720715 199802 2 001

2. Dr. Sunarsini, S.Si., M.Si. (.....)
   NIP. 19691004 199402 2 001

3. Dr. Imam Mukhlash, S.Si., MT (.....)
   NIP. 19700831 199403 1 003

Mengetahui
Kepala Departemen Matematika
Fakultas Sains dan Analitika Data

Prof. Subchan, S.Si., M.Sc. Ph.D
NIP. 19710513 199702 1 001

PERNYATAAN ORISINALITAS

Yang bertanda tangan disini:

Nama Mahasiswa / NRP	:	I Gede Febry Abdi Saputra / 5002201008
Departemen	:	Matematika
Dosen Pembimbing / NIP	:	Prof. Dr. techn. Drs. Mohammad Isa Irawan, MT / 19631225 198903 1 001

dengan ini menyatakan bahwa Tugas Akhir dengan judul "**PREDIKSI KETIDAKSELARASAN JUDUL DAN ISI DARI BERITA CLICKBAIT BERBAHASA INDONESIA MENGGUNAKAN MODEL INDO BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS (INDOBERT)**" adalah hasil karya sendiri, bersifat orisinal, dan ditulis dengan mengikuti kaidah penulisan ilmiah.

Bilamana di kemudian hari ditemukan ketidaksesuaian dengan pernyataan ini, maka saya bersedia menerima sanksi sesuai dengan ketentuan yang berlaku di Institut Teknologi Sepuluh Nopember.

Surabaya, 30 Juli 2024

Mengetahui
Dosen Pembimbing

Prof. Dr. techn. Drs. Mohammad Isa Irawan, MT
NIP. 19631225 198903 1 001

Mahasiswa

I Gede Febry Abdi Saputra
NRP. 5002201008

ABSTRAK

PREDIKSI KETIDAKSELARASAN JUDUL DAN ISI DARI BERITA CLICKBAIT BERBAHASA INDONESIA MENGGUNAKAN MODEL INDO BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS (INDOBERT)

Nama Mahasiswa / NRP	:	I Gede Febry Abdi Saputra / 5002201008
Departemen	:	Matematika FSAD -ITS
Dosen Pembimbing	:	Prof. Dr. techn. Drs. Mohammad Isa Irawan, MT

Abstrak

Istilah *clickbait* digunakan untuk menggambarkan judul berita yang menyembunyikan informasi untuk memicu rasa ingin tahu pengguna agar melakukan tindakan klik, meskipun sering mengecewakan pembaca. Dengan meningkatnya jumlah pengguna internet dan popularitas media sosial, semakin banyak berita yang tersebar di internet, sehingga penting untuk memiliki *tools* yang dapat membedakan antara berita informatif dan *clickbait*. Penelitian ini bertujuan untuk membangun model klasifikasi menggunakan model *Indo Bidirectional Encoder Representations from Transformers* (IndoBERT) untuk membedakan antara berita yang selaras (*non-clickbait*) dan tidak selaras (*clickbait*) berdasarkan judul dan isi naskah beritanya. Dataset berasal dari 26 portal penyedia berita *online*. Hasil penelitian menunjukkan bahwa model IndoBERT mampu dalam tugas klasifikasi berita dan memiliki performa unggul dengan akurasi validasi 77% dan akurasi pengujian 76% pada jenis dataset terbaik (dataset C dengan proporsi 95% data latih dan 5% data uji). Perbandingan dengan model BERT lainnya yaitu RoBERTa menunjukkan bahwa IndoBERT lebih unggul dalam *accuracy*, *precision*, *recall*, dan *F1-score*. Faktor-faktor seperti pelatihan khusus pada korpus bahasa Indonesia yang lebih relevan dan adanya *layer pooler* pada IndoBERT menjadi salah satu alasan keunggulan performa. Penelitian ini memberikan kontribusi dalam identifikasi ketidakselarasan antara isi teks dengan judul berita dengan tingkat akurasi yang tinggi, serta membuktikan bahwa IndoBERT lebih efektif dalam tugas klasifikasi berita *clickbait* dan *non-clickbait*.

Kata kunci: *Berita, Clickbait, IndoBERT, Klasifikasi.*

ABSTRACT

PREDICTING TITLE AND CONTENT MISALIGNMENT FROM CLICKBAIT NEWS IN INDONESIAN USING INDO BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS (INDOBERT) MODEL

Student Name / NRP	:	I Gede Febry Abdi Saputra / 5002201008
Departement	:	Mathematics SCIENTICS - ITS
Advisor	:	Prof. Dr. techn. Drs. Mohammad Isa Irawan, MT

Abstract

The term clickbait is used to describe headlines that hide information to trigger users' curiosity to click, although it often disappoints readers. With the increasing number of Internet users and the popularity of social media, more and more news is spread on the Internet, so it is important to have tools that can distinguish between informative news and clickbait. This study aims to build a classification model using the Indo Bidirectional Encoder Representations from Transformers (IndoBERT) model to distinguish between aligned (non-clickbait) and unaligned (clickbait) news based on the title and content of the news script. The dataset comes from 26 online news provider portals. The results show that the IndoBERT model is capable of news classification tasks and has superior performance with 77% validation accuracy and 76% testing accuracy on the best type of dataset (dataset C with a proportion of 95% training data and 5% test data). Comparison with another BERT model, RoBERTa, shows that IndoBERT is superior in accuracy, precision, recall, and F1-score. Factors such as specialized training on a more relevant Indonesian corpus and the presence of a layer pooler in IndoBERT are one of the reasons for the superior performance. This research contributes to the identification of misalignment between text content and news headlines with high accuracy, and proves that IndoBERT is more effective in the task of clickbait and non-clickbait news classification.

Keywords: *Classification, Clickbait, IndoBERT, News.*

KATA PENGANTAR

Puji syukur kehadirat Tuhan Yang Maha Esa karena atas berkah, rahmat, dan anugerah-Nya sehingga penulis dapat menyelesaikan tugas akhir dengan judul :

"PREDIKSI KETIDAKSELARASAN JUDUL DAN ISI DARI BERITA *CLICKBAIT* BERBAHASA INDONESIA MENGGUNAKAN MODEL *INDO BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS* (INDOBERT)"

sebagai salah satu persyaratan akademis dalam menyelesaikan Program Sarjana Departemen Matematika, Fakultas Sains dan Analitika Data, Institut Teknologi Sepuluh Nopember Surabaya. Tugas Akhir ini dapat diselesaikan dengan baik berkat kerja sama, bantuan, dan dukungan dari banyak pihak. Sehubungan dengan hal itu, penulis ingin mengucapkan terima kasih dan penghargaan kepada :

1. Orang tua dan keluarga yang telah memberikan dukungan, nasehat, dan motivasi sehingga penulis dapat menyelesaikan Tugas Akhir ini.
2. Kepala Departemen Matematika Institut Teknologi Sepuluh Nopember dan Sekretaris Departemen Matematika Institut Teknologi Sepuluh Nopember beserta jajarannya yang telah menyediakan fasilitas dan memberi arahan selama perkuliahan.
3. Bapak Prof. Dr. techn. Drs. Mohammad Isa Irawan, MT selaku dosen pembimbing yang telah memberikan arahan dan motivasi dengan penuh kesabaran kepada penulis.
4. Bapak Prof. Dr. Subiono, MS selaku dosen wali yang telah memberikan arahan dengan penuh kesabaran kepada penulis selama masa perkuliahan.
5. Ibu Alvida Mustika Rukmi, S.Si, M.Si; Ibu Dr. Sunarsini, S.Si, M.Si; dan Bapak Dr. Imam Mukhlash, S.Si, MT selaku dosen penguji yang telah memberikan arahan, kritik, saran, dan masukan yang membangun pada Tugas Akhir ini.
6. Bapak/Ibu dosen pengajar yang tidak bisa penulis sebutkan satu per satu, yang telah memberikan ilmu dan pengalaman yang bermanfaat kepada penulis, serta segenap Tenaga Kependidikan Departemen Matematika Institut Teknologi Sepuluh Nopember atas dukungan dan bantuannya.
7. Mahasiswi Departemen Statistika Σ31.081 yang selalu memberikan dukungan, semangat, dan berperan penting selama masa perkuliahan penulis.

Penulis menyadari bahwa tugas akhir ini masih jauh dari kesempurnaan. Oleh karena itu, penulis mengharapkan saran dan kritik dari pembaca. Akhir kata, semoga Tugas Akhir ini bermanfaat bagi semua pihak yang berkepentingan.

Surabaya, Juli 2024

I Gede Febry Abdi Saputra

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

Tugas akhir ini merupakan suatu penelitian oleh penulis. Pada bab ini dijelaskan mengenai latar belakang, batasan masalah, rumusan masalah, tujuan penelitian, dan manfaat penelitian.

1.1 Latar Belakang

Informasi yang dikemas melalui berita selalu menjadi bagian dari kehidupan sehari-hari. Media berita online membantu menyampaikan informasi kepada masyarakat tentang peristiwa-peristiwa yang terjadi di dunia setiap hari. Namun, terkadang judul berita yang ditampilkan sengaja menyesatkan pembaca, dengan tujuan untuk mendapat perhatian dan tindakan klik. Selain itu, banyak media berita membuat judul secara sensasional dan dilebih-lebihkan dari isi berita aslinya sebagai daya tarik dari sebuah berita atau artikel yang diunggah. Hal ini sering disebut dengan istilah *clickbait* (Potthast et al., 2016). *Clickbait* merupakan istilah yang digunakan untuk menggambarkan tajuk utama berita yang dengan disengaja menahan informasi untuk menciptakan kesenjangan pengetahuan dan menciptakan rasa ingin tahu pada pengguna untuk melakukan tindakan klik dan menimbulkan kekecewaan pada pembaca saat membacanya. Penggunaan judul yang sengaja menahan informasi untuk menciptakan kesenjangan pengetahuan akan menciptakan rasa ingin mencari informasi lebih bagi pengguna, sehingga pengguna melakukan tindakan klik dan berharap mendapatkan informasi yang dicarinya (Anand et al., 2019).

Dalam jurnalisme terdapat tindakan salah seperti informasi yang menyesatkan sehingga menimbulkan masalah sosial yang kritis (Yoon et al., 2019). Suatu berita cenderung lebih mengutamakan mendapatkan klik daripada memberikan informasi yang berkualitas sehingga teks isi dari berita tersebut tidak jarang membuat pengguna kecewa. Hal ini terjadi ketika isi naskah berita memiliki ketidakselarasan terhadap judulnya. Tindakan tersebut tentunya akan merugikan pengguna dalam usahanya mencari informasi yang diinginkan ketika penggunaan *clickbait* diikuti dengan konten yang tidak sesuai (Yoon et al., 2019). Ketidakselarasan terhadap naskah berita dengan judulnya merupakan kondisi di mana judul suatu berita tidak sesuai atau tidak memperlihatkan isi naskah secara tepat dan akurat. Hal tersebut dapat mempengaruhi bagaimana pembaca memahami dan menilai makna dari berita tersebut. Adapun beberapa faktor penyebab judul suatu berita memiliki ketidakselarasan dengan isi teks berita tersebut, di antaranya adalah kontradiksi antara judul dengan isi teks, penggunaan kata-kata yang tidak benar dalam judul, penggunaan judul yang terlalu *general* atau tidak mencerminkan isi dari teks secara spesifik, judul yang terlalu mengubah makna atau mempertegas aspek tertentu dari isi teks, serta judul yang menggunakan bahasa provokatif untuk memikat pembaca. Oleh karena itu, jika judul berita tidak merepresentasikan isi dari berita dengan benar, maka dapat mengakibatkan pembaca menyerap informasi yang salah (Yoon et al., 2019).

Beberapa penelitian berikut berpendapat bahwa berita bersifat *clickbait* hanya sifat yang dimiliki tajuk utamanya saja. Pada jurnal *Using Neural Network for Identifying Clickbaits in Online News Media* (Omidvar et al., 2018) yang berpendapat bahwa *clickbaits*

adalah berbagai judul berita yang menarik perhatian pengguna tetapi membuat pengguna kecewa pada akhirnya. Hasil penelitian tersebut telah mengeksplorasi pendekatan otomatis untuk melakukan deteksi berita *clickbait* dengan upaya klasifikasi menggunakan *corpus* tajuk utama berita. Penelitian ini menggunakan metode *Bidirectional Gated Recurrent Units* (GRU) dalam mengklasifikasikan data teks dan memperoleh akurasi sebesar 0.8553. Pada jurnal *Clickbait Detection of Indonesian News Headlines using Fine-Tune Bidirectional Encoder Representations from Transformers (BERT)* oleh (Putri & Pratomo, 2022) dan jurnal *Clickbait Detection in Indonesia Headline News Using IndoBERT and RoBERTa* oleh (Syahputra et al., 2023) telah dilakukan klasifikasi terhadap judul berita *clickbait* dalam Bahasa Indonesia menggunakan metode model *pre-trained* BERT. Penelitian ini menghasilkan masing-masing 0.847 dan 0.98 menggunakan dari segi akurasi model IndoBERT berdasarkan pengklasifikasian dari judul berita saja. Namun, pada jurnal *Detecting Incongruity Between News Headline and Body Text via a Deep Hierarchical Encoder* (Yoon et al., 2019) proses klasifikasi berita pada jurnal ini tidak hanya berfokus pada klasifikasi dari judulnya saja, tetapi dilakukan dengan pendekatan analisis keselarasan antara isi dan judul berita. Judul berita yang tidak benar atau tidak sesuai dianggap dapat menyesatkan pembaca untuk menganjurkan informasi yang berlebihan atau salah. Penelitian ini menghasilkan akurasi sebesar 0.904 dengan menggunakan metode *Attentive Hierarchical Dual Encoder*.

Berdasarkan masalah yang telah diidentifikasi dan penelitian sebelumnya, penulis menyimpulkan bahwa judul tidak dapat dijadikan satu-satunya acuan dalam menilai kelayakan suatu teks berita. Teks berita juga harus dianalisis secara seksama bersamaan dengan judul untuk memilah berita yang berkualitas. Prediksi berita dalam konteks penelitian ini merujuk pada kemampuan untuk memprediksi seberapa sesuai atau tidak selarasnaya isi suatu naskah dengan judul berita yang bersifat *clickbait*. Penelitian ini bertujuan untuk membantu pembaca membedakan antara berita yang mengandung konten informatif dan berita yang hanya bertujuan menarik perhatian serta klik dari pembaca, melalui model yang dikembangkan berdasarkan performa dan implementasinya. Penelitian ini membangun model klasifikasi teks berita selaras dan tidak selaras (*clickbait*) yang menerima input berupa judul dan teks berita kemudian mengembalikan skor keselarasan dari berita tersebut. Model yang dibangun menggunakan basis model *pre-trained* dari *Bidirectional Encoder Representations from Transformers* (BERT) yaitu IndoBERT. Model tersebut telat dilatih dengan basis model BERT yang canggih dalam mengklasifikasikan kata-kata berbahasa Indonesia (Putri & Pratomo, 2022). Selain itu, IndoBERT yang didasarkan pada model BERT yang sudah dilatih sebelumnya, telah menggunakan dataset Indo4B yang terdiri dari lebih dari 23 GB data teks berbahasa Indonesia (Wilie et al., 2020). Proses *pretraining* ini melibatkan eksposur terhadap jumlah besar data teks, yang membantu model untuk memahami struktur bahasa dan konteks yang beragam dalam bahasa Indonesia. Penelitian dengan penerapan model ini diharapkan mampu menghasilkan hasil klasifikasi teks yang akurat dan prediksi yang tepat.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam penelitian ini adalah sebagai berikut:

1. Apakah penerapan model IndoBERT mampu dalam memprediksi ketidakselarasan isi naskah dengan judul berita pada berita *clickbait*?

2. Bagaimana menganalisis hasil performansi model IndoBERT dalam memprediksi ketidakselarasan isi naskah dengan judul berita pada berita *clickbait*?

1.3 Batasan Masalah

Adapun batasan masalah dari penelitian Tugas Akhir ini adalah bahwa data yang digunakan merupakan hasil *scraping* dari portal berita *online* berupa data berita tekstual berbahasa Indonesia dan tidak ada kategori berita spesifik yang digunakan. Data yang digunakan berasal dari portal berita berbahasa Indonesia, di antaranya adalah ANTARA News, BBC Indonesia, Brilio.net, CNBC Indonesia, CNN Indonesia, Detik.com, Dream.co.id, IDN Times, iNews.id, JPNN.com, Kapanlagi.com, Kompas.com, Kompasiana.com, Liputan6.com, Merdeka.com, Okezone.com, Portal Pekalongan, Republika, Sindonews.com, Suara.com, TvOne News, Tribunnews.com, Tempo.co, Tirto.id, Viva.co.id, dan WowKeren.com. Proses pengumpulan data berita dilakukan mulai tanggal 18 Februari sampai 12 Maret 2024.

1.4 Tujuan

Sebagai lanjutan dari rumusan masalah, dapat dijabarkan beberapa tujuan dari penelitian ini sebagai berikut:

1. Mengetahui kemampuan dari model IndoBERT dalam memprediksi ketidakselarasan isi naskah dengan judul berita pada berita *clickbait*.

2. Menganalisis hasil performasi model IndoBERT dalam memprediksi ketidakselarasan isi naskah dengan judul berita pada berita *clickbait*.

1.5 Manfaat

Dengan adanya penelitian ini, diharapkan dapat memberikan manfaat antara lain:

1. Didapatkan informasi terkait kinerja dari metode IndoBERT dalam memprediksi ketidakselarasan berita berdasarkan judul dengan isi naskah berita.

2. Diperoleh model klasifikasi teks yang lebih stabil dengan menggunakan *pre-trained* model, sehingga dapat menjadi salah satu pertimbangan peneliti lainnya dalam memilih metode klasifikasi untuk kasus yang serupa.

BAB II
TINJAUAN PUSTAKA

Pada bab ini dijelaskan terkait penelitian terdahulu tentang memprediksi berita *clickbait* menggunakan model *machine learning*. Selain itu, dijelaskan pula mengenai *Clickbait*, Klasifikasi Teks, *Web Scrapping*, *Word Embedding*, BERT, dan IndoBERT.

2.1 Penelitian Terdahulu

Dalam mendeteksi berita *clickbait* menggunakan *machine learning* telah dilakukan pada beberapa penelitian-penelitian sebelumnya. Namun, kebanyakan penelitian terkait berita *clickbait* berbahasa Indonesia hanya berfokus pada memprediksi dari judul atau *headlines* saja. Pada penelitian yang dilakukan oleh D. U. K. Putri dan D. N. Pratomo (Putri & Pratomo, 2022) menggunakan model BERT dengan dua pendekatan berbasis vektor kata yaitu *Bag-of-Words* dan TF-IDF untuk mendeteksi berita *clickbait* menggunakan dataset berita utama Indonesia bernama CLICK-ID. Namun, penelitian ini hanya mengidentifikasi judul dari artikel berita yang termasuk *clickbait* dan *non-clickbait* dengan memperoleh nilai akurasi sebesar 0.847. Adapun penelitian serupa yang dilakukan oleh Syahputra dkk., (Syahputra et al., 2023) menggunakan dataset berisi 15.000 *headline* berita Indonesia yang telah dianotasi *clickbait* dan *non-clickbait*. Pelatihan pada dataset ini menggunakan model IndoBERT dan RoBERTa dengan variasi jenis dataset yang telah melalui proses augmentasi dan tanpa melalui proses augmentasi. Pada model IndoBERT memperoleh nilai F1-score sebesar 0.95 dan model RoBERTa (*Robustly optimized* BERT *approach*) memperoleh nilai F1-score sebesar 0.91.

Adapun penelitian lain dalam mendeteksi berita *clickbait* menggunakan artikel berbahasa asing. Penelitian yang dilakukan oleh Yoon, dkk. (Yoon et al., 2019), yaitu mendeteksi kesesuaian *headline* berita dan teks isi. Penelitian ini menggunakan dataset artikel berbahasa Korea dengan nilai akurasi model tertinggi menggunakan *Attentive Hierarchical Dual Encoder* (AHDE) sebesar 0.904. Penelitian lainnya dilakukan oleh Omidvar dkk, (Omidvar et al., 2018) dalam mendeteksi judul berita *clickbait* berbahasa Inggris pada media sosial Twitter menggunakan model *Bidirectional Gated Recurrent Units* (GRU) dengan memperoleh nilai akurasi sebesar 0.8553.

2.2 *Clickbait*

Clickbait, dalam konteks maknanya, dapat diartikan sebagai strategi untuk menarik perhatian pengguna internet dengan menampilkan judul atau *headline* yang menarik sehingga mengundang untuk diklik (umpan klik). Tujuannya adalah agar pengguna internet tertarik untuk membuka dan membaca kontennya. Namun, seringkali setelah konten tersebut dibuka, isinya tidak sesuai dengan harapan atau ekspektasi awal pengguna, dan hal ini menyebabkan dampak negatif. Istilah *clickbait* kini cenderung memiliki konotasi negatif karena dinilai sebagai "jebakan klik", yang artinya menciptakan ketertarikan palsu untuk menarik penonton tetapi tidak memberikan konten yang sesuai dengan yang dijanjikan. (Romli, 2018).

2.3 Klasifikasi Teks

Konstruksi model yang dapat mengkategorikan dokumen baru ke dalam kelas yang telah ditentukan sebelumnya dikenal sebagai klasifikasi teks (Mirończuk & Protasiewicz, 2018). Algoritme klasifikasi teks adalah dasar dari berbagai sistem perangkat lunak yang memproses teks dalam skala besar. Klasifikasi teks digunakan pada forum diskusi untuk menentukan apakah komentar harus ditandai sebagai tidak pantas atau tidak, sedangkan pada perangkat lunak email menerapkannya untuk menentukan apakah email masuk dikirim ke kotak masuk atau disaring ke folder spam. Pada penelitian ini, klasifikasi teks digunakan untuk menentukan keselarasan dan ketidakselarasan antara judul dengan naskah berita berdasarkan label yang telah diberikan. Klasifikasi teks bertujuan untuk menganalisis, memproses, dan mengekstrak informasi dari teks. Dalam bidang ini, beberapa kategori umum termasuk analisis sentimen, kategorisasi berita, dan klasifikasi topik (Minaee et al., 2021).

Pada teks yang umumnya merupakan data tidak terstruktur memerlukan tahap *preprocessing* agar menjadi lebih terstruktur dan lebih mudah diekstraksi informasinya. Tujuan *preprocessing* teks adalah untuk memecah setiap dokumen menjadi kata-kata individual dan menggambarkannya sebagai vektor fitur. Dalam konteks pengindeksan dokumen, pemilihan fitur dan langkah-langkah utama *preprocessing* teks harus dilakukan untuk memilih kata-kata kunci yang relevan. Pada tahap *preprocessing* teks, dokumen teks input diproses untuk membentuk fitur yang dikenal sebagai *tokenization, words, terms* atau *attributes* (Kadhim, 2018).

2.4 *Web Scraping*

Web scraping dikenal sebagai *web extraction* atau *harvesting*, merujuk pada teknik ekstraksi data dari *World Wide Web* (WWW) dan penyimpanannya dalam sistem *file* atau *database*. Tujuannya adalah untuk mengumpulkan informasi yang dapat digunakan untuk analisis atau pengambilan keputusan di masa mendatang. Umumnya, data dari web diambil menggunakan *Hypertext Transfer Protocol* (HTTP) atau melalui browser web. Proses *web scraping* dapat dilakukan secara manual oleh pengguna atau secara otomatis oleh bot atau *web crawler* (Zhao, 2017).

2.5 *Word Embedding*

Word embedding merupakan teknik dalam pemrosesan bahasa alami (*natural language processing*) yang berfungsi untuk mewakili kata-kata sebagai vektor numerik dalam ruang multidimensional. Representasi vektor ini memberikan kemampuan kepada komputer untuk memahami makna kata dan mengintegrasikannya dengan kata-kata lain dalam konteks yang relevan (Mikolov et al., 2013). Penggunaan teknik *word embedding* sangat bermanfaat dalam berbagai aplikasi NLP, seperti klasifikasi teks, pemodelan topik, dan analisis sentimen. Dalam konteks pengolahan bahasa alami, representasi vektor kata juga dapat mempercepat proses pelatihan model dan meningkatkan tingkat akurasi. Meskipun *word embedding* umumnya dihasilkan dari pelatihan model yang memerlukan data besar dan waktu komputasi yang signifikan, tersedia banyak sumber *word embedding* yang telah terpelajari sebelumnya yang dapat digunakan untuk berbagai aplikasi.

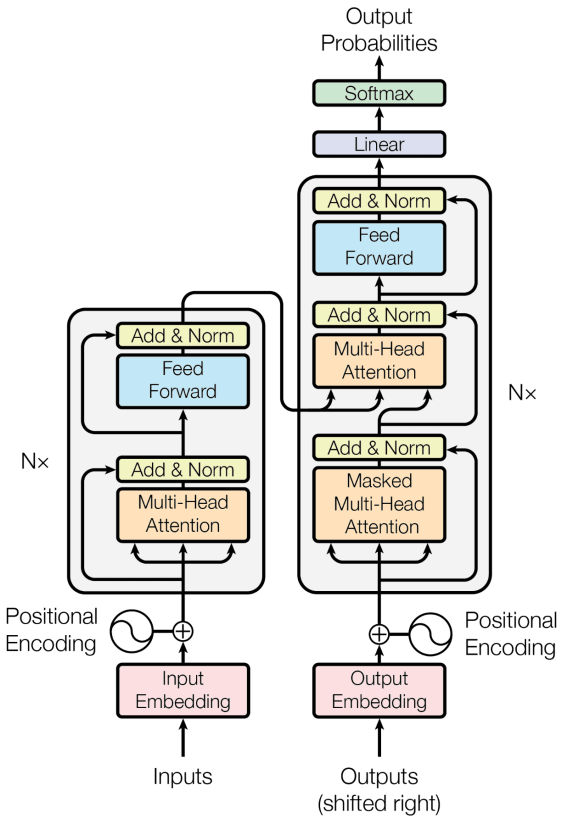
2.6 *Bidirectional Encoder Representations from Transformers* (BERT)

Bidirectional Encoder Representations from Transformers (BERT) (Luo & Wang, 2019) adalah metode berbasis jaringan saraf yang digunakan untuk pra-pelatihan bahasa alami (*natural language*). BERT bertujuan untuk membangun pemahaman yang lebih jelas terhadap bahasa, yang dirancang untuk mengatasi ambiguitas dalam kalimat

dengan memahami konteks sekitarnya melalui teks. BERT menggunakan *transformer* sebagai dasar dengan menggunakan bagian *encoder* (suatu metode untuk memahami konteks kalimat dengan menganalisis hubungan kontekstual antara kata-kata dalam teks) (Ganesh et al., 2021). Transformer memungkinkan pembelajaran dan modifikasi pemahaman dengan memanfaatkan mekanisme *self-attention*. Mekanisme *self-attention* adalah metode di mana transformer dapat memproses kata-kata terkait dan mengubahnya berdasarkan informasi yang diperoleh dari konteks. Transformer sendiri terdiri dari dua komponen utama, yaitu *encoder* dan *decoder*.

2.6.1 Transformer

Transformer (Vaswani et al., 2017) memanfaatkan mekanisme *attention* dan mengeliminasi penggunaan model berulang (*recurrent*) serta konvolusi. Diperkenalkan pada tahun 2017, model ini menggunakan *encoder* dan *decoder* dalam struktur pembentuknya berdasarkan arsitektur transformer yang ditampilkan pada Gambar 2.1.



Gambar 2.1 Arsitektur Transformer
(Vaswani et al., 2017)

Sebelum masuk ke bagian *encoder*, data teks akan melewati lapisan yang bernama *input embedding*. Pada tahap ini, kata-kata disematkan ke dalam ruang *embedding* atau ruang multidimensional sebagai vektor posisi. Kata-kata dengan makna yang serupa ditempatkan pada vektor yang berdekatan satu sama lain. Namun, posisi suatu kata dalam kalimat dapat memengaruhi makna keseluruhan kalimat tersebut. Oleh karena itu, diperlukan *positional encoder* (PE) untuk menghasilkan vektor *encoding* atau vektor numerik dalam ruang multidimensional yang mencerminkan posisi kata dalam kalimat.

Rumus untuk mendapatkan vektor *encoding* (*Positional Encoder*) menggunakan fungsi sinusoidal dan kosinusoidal ditunjukkan pada Persamaan 2.1 dan Persamaan 2.2.

$$\mathbf{PE}(pos, 2i) = \sin\left(\frac{pos}{10000^{(2i/d_{\text{model}})}}\right) \tag{2.1}$$

$$\mathbf{PE}(pos, 2i+1) = \cos\left(\frac{pos}{10000^{((2i+1)/d_{\text{model}})}}\right) \tag{2.2}$$

Persamaan tersebut merupakan komponen - komponen dari vektor *encoding* untuk posisi *pos* dan dimensi $2i$ dan $2i + 1$, serta d_{model} adalah dimensi dari vektor *embedding* di model tersebut. Nilai yang didapatkan dari masing-masing persamaan akan ternormalisasi antara -1 hingga 1. Pada Persamaan 2.1 digunakan untuk posisi kata yang genap, sedangkan Persamaan 2.2 digunakan untuk posisi kata yang ganjil.

Keadaan serupa terjadi pada lapisan *Output Embedding*, yang bertanggung jawab untuk memproses teks keluaran sebelum memasuki bagian *decoder*. Dalam blok *encoder* dan *decoder*, terdapat dua jenis lapisan, yakni lapisan *Multi-Head Attention* dan lapisan *Feed Forward*. Lapisan *Multi-Head Attention* berfokus pada tingkat ketergantungan antara kata satu dengan kata lain dalam suatu kalimat secara kontekstual yang di mana setiap *Multi-Head* memiliki 8 *self-attention head*. Namun, Transformer memperbarui mekanisme *self-attention* dengan memperkenalkan tiga matriks bobot yang dapat dilatih, yaitu *Query*, *Key*, dan *Value* (Raschka, 2021). Matriks-matriks ini kemudian dikalikan dengan kata masukan yang telah dipetakan ke ruang vektor, menghasilkan representasi *Query* ($\mathbf{Q}$), *Key* ($\mathbf{K}$), dan *Value* ($\mathbf{V}$) sebagai berikut.

$$q_i = x_i \mathbf{W}_h^Q \tag{2.3}$$

$$k_i = x_i \mathbf{W}_h^K \tag{2.4}$$

$$v_i = x_i \mathbf{W}_h^V \tag{2.5}$$

Untuk setiap kata ke-i (x_i), didapatkan matriks *similarity* S_i dengan menggunakan q_i sebagai matriks *query* ke-i, $\mathbf{W}_h^Q$ sebagai matriks bobot ke-h untuk *query*, k_i sebagai matriks *key* ke-i, $\mathbf{W}_h^K$ sebagai matriks bobot ke-h untuk *key*, v_i sebagai matriks *value* ke-i, dan $\mathbf{W}_h^V$ sebagai matriks bobot ke-h untuk *value*. Pada persamaan *similarity* terdapat proses perkalian dot product terhadap q_i dan k_j^T dengan i adalah kata saat ini dan j adalah kata lainnya. Persamaan matriks *similarity* ditunjukkan pada Persamaan 2.6.

$$S_i = \begin{bmatrix} \frac{q_i k_1^T}{\sqrt{d_k}} \\ \frac{q_i k_2^T}{\sqrt{d_k}} \\ \vdots \\ \frac{q_i k_j^T}{\sqrt{d_k}} \end{bmatrix} \tag{2.6}$$

Berdasarkan Persamaan 2.6, didapatkan representasi kata berupa matriks yang melalui tahapan *self-attention*. Rumus mencari *attention value* dari kata tersebut ditunjukkan pada Persamaan 2.7.

$$\mathit{Attention}_i(q_i, \mathbf{K}, \mathbf{V}) = \sum_{j=1}^{n} \left[\frac{\exp\left(q_i k_j^T / \sqrt{d_k}\right)}{\sum_m \exp\left(q_i k_m^T / \sqrt{d_k}\right)} \times v_j \right] \tag{2.7}$$

Proses *self-attention* ini terjadi pada setiap kata secara paralel, sehingga semua kata diproses secara bersamaan (Raschka, 2021). Pada k_j^T merupakan transpos dari vektor *key* ke-j, d_k merupakan dimensi dari vektor *key*, dan v_j merupakan vektor *value* ke-j. Pada $\mathbf{Q} = \{q_i, i = 1, 2, 3, ..., n\}$, $\mathbf{K} = \{k_j, j = 1, 2, 3, ..., n\}$, dan $\mathbf{V} = \{v_j, j = 1, 2, 3, ..., n\}$, sehingga matriks *self-attention* dari seluruh kata direpresentasikan dengan Persamaan 2.8.

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{2.8}$$

Dalam *Multi-head Attention* akan dilakukan proses penggabungan (*concatenation*) pada setiap *head*. Untuk memperoleh *Multi-Head Attention* dapat dilakukan dengan persamaan yang ditunjukkan pada persamaan sebagai berikut.

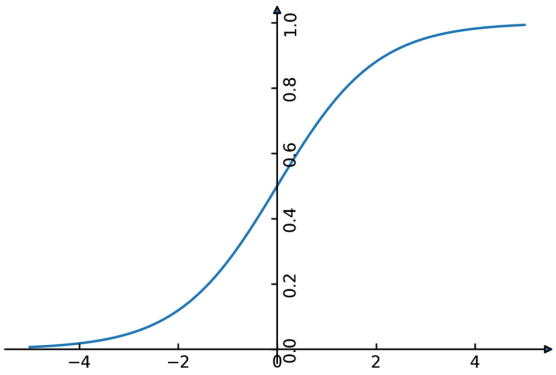
$$MultiHead(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h) \cdot \mathbf{W}^O \tag{2.9}$$

$$\text{head}_i = \text{Attention}(\mathbf{Q} \cdot \mathbf{W}_i^Q, \mathbf{K} \cdot \mathbf{W}_i^K, \mathbf{V} \cdot \mathbf{W}_i^V) \tag{2.10}$$

Pada Persamaan 2.9 terdapat $head_1$ merupakan nilai *attention* dari *self-attention* yang pertam dan dilanjutkan dengan penggabungan atau *concatenation*. Proses ini menggabungkan hasil dari masing - masing *head attention*. Selanjutnya, proses pada *Multi-Head Attention* adalah *linear layer* yaitu dengan penambahan parameter *weight* untuk pelatihan model. Selain itu, terdapat fungsi *softmax* untuk melakukan normalisasi dengan persamaan *softmax* ditunjukkan pada Persamaan 2.11.

$$softmax(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}} \tag{2.11}$$

Pada Persamaan 2.11, untuk $i \in \mathbb{N}$, x_i merupakan nilai input setiap kata ke-i, e^{x_i} merupakan fungsi eksponensial standar dari nilai input, dan N merupakan jumlah kelas dalam dataset. Fungsi aktivasi *softmax* akan melakukan perhitungan dengan menghasilkan nilai bobot atau *weight*. Representasi grafik dari fungsi aktivasi *softmax* ditunjukkan pada Gambar 2.2.



Gambar 2.2 Grafik Fungsi Aktivasi *Softmax*
(Nayak & Kumar, 2022)

Setelah melewati proses *multi-head attention*, token selanjutnya masuk ke *feed forward layer* untuk memproses ke tahap berikutnya. Perhitungan untuk *Feed Forward Layer* akan ditunjukkan pada Persamaan 2.12 berikut.

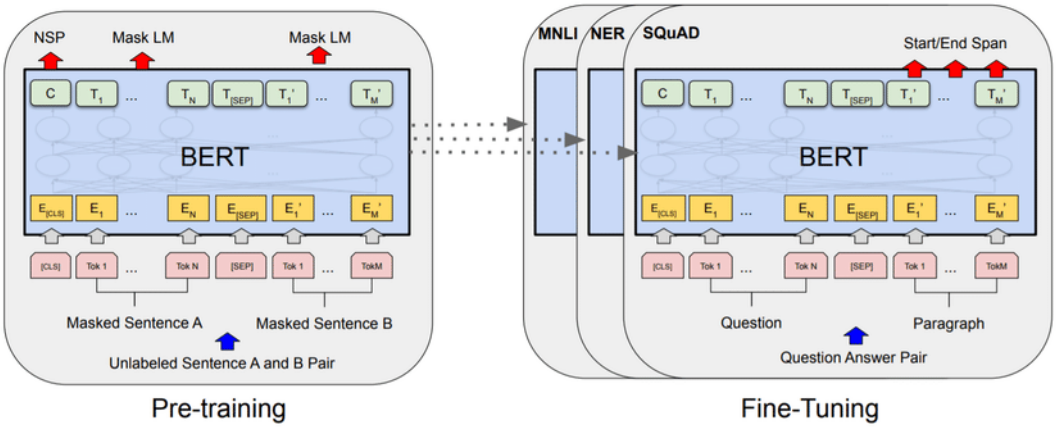
$$\mathbf{FFN}(\mathrm{x}) = \max(0, x\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2 \tag{2.12}$$

dengan x merupakan input, $\mathbf{W}_1$ dan $\mathbf{W}_2$ merupakan *weight* (matriks bobot), serta $\mathbf{b}_1$ dan $\mathbf{b}_2$ merupakan *bias*.

2.6.2 *Pre-training* Model BERT

Pada tahap *pre-trained*, BERT menggunakan dua tugas *unsupervised*, yaitu *Masked Language Modeling* (MLM) dan *Next Sentence Prediction* (NSP) (Devlin et al., 2018). Dalam tahap MLM, model dilatih untuk memprediksi token yang disamarkan secara acak dalam sebuah teks dengan menutupi beberapa persentase token input secara acak. Penggantian token dapat dilakukan dengan token [MASK], token acak, atau mempertahankan token asli. Proses ini memungkinkan BERT memperoleh representasi dua arah yang kuat. Namun, ada ketidakcocokan antara tahap *pre-training* dan *fine-tuning* karena token [MASK] tidak muncul selama *fine-tuning*. Sebagai solusi, token [MASK] tidak selalu digunakan untuk menggantikan kata yang sebenarnya.

Tahapan kedua adalah NSP, dilakukan untuk memahami hubungan antara dua kalimat. Model dilatih untuk memprediksi apakah satu kalimat merupakan kelanjutan dari kalimat lainnya. Proses ini membantu memperoleh representasi yang bermanfaat untuk tugas turunan seperti *Question Answering* (QA) dan *Natural Language Inference* (NLI) . NSP melibatkan pemilihan dua kalimat, di mana 50% dari waktu kalimat kedua adalah kelanjutan langsung dari kalimat pertama (IsNext), dan 50% dari waktu itu adalah kalimat acak (NotNext). NSP memperkuat pemahaman hubungan antar kalimat dan berkontribusi pada inisialisasi parameter *fine-tuning* pada tahap selanjutnya. Visualisasi proses *pre-training* dan *fine-tuning* ditunjukkan pada Gambar 2.3.



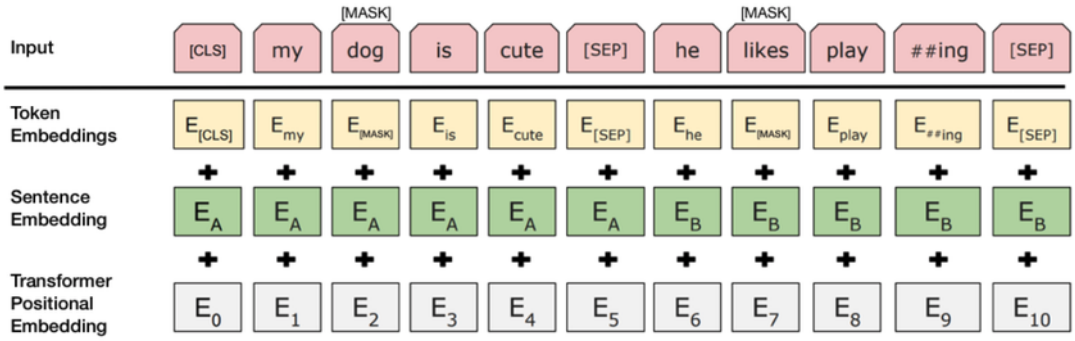
Gambar 2.3 *Pre-training* dan *Fine-tuning* pada BERT (Devlin et al., 2018)

2.6.3 *Fine-tuning* Model BERT

Proses *fine-tuning* (Devlin et al., 2018) pada BERT menjadi sangat mudah karena model ini menggunakan mekanisme *self-attention* yang memungkinkan BERT untuk beradaptasi dengan berbagai tugas turunan. Dalam aplikasi yang melibatkan pasangan teks, pola umumnya adalah melakukan *encoding* pada pasangan teks secara independen sebelum menerapkan perhatian silang dua arah. BERT menggabungkan mekanisme *self-attention* untuk menyatukan kedua tahap tersebut, menghasilkan pengkodean yang efektif

dengan menggabungkan pasangan teks melalui *self-attention*, termasuk perhatian silang dua arah antara dua kalimat. Dalam *fine-tuning* untuk setiap tugas, kita hanya perlu menyediakan *input* dan *output* khusus tugas tersebut ke model BERT, dan selanjutnya melakukan *fine-tune* pada semua parameter secara *end-to-end*.

Sebelum memasuki tahap *fine-tuning* BERT, penambahan token khusus dilakukan pada *input formatting* BERT. Penambahan ini mencakup token [SEP] di setiap akhir kalimat sebagai penanda pemisah antar kalimat dan token [CLS] di setiap awal teks sebagai penanda bahwa lapisan tersembunyi dari arsitektur Transformer milik BERT akan melakukan pembelajaran klasifikasi teks. Token ini terus mengalami pembaharuan bobot dari setiap lapisan Transformer, dan hasil pembobotan akhir dari token ini akan menjadi *output* dari model (Devlin et al., 2018). Representasi input untuk token yang diberikan dibangun dengan menjumlahkan token, segmen, dan posisi *embeddings* yang sesuai. Konstruksi ini dapat dilihat secara visual pada Gambar 2.4.



Gambar 2.4 Konstruksi Input pada BERT
(Devlin et al., 2018)

2.7 IndoBERT

IndoBERT merupakan model *pre-trained* yang dibuat berdasarkan BERT untuk teks berbahasa Indonesia. Proses pelatihan IndoBERT menggunakan dataset Indo4B yang memiliki lebih dari 23 GB data teks berbahasa Indonesia. Sumber data Indo4B berasal dari berbagai platform seperti berita daring, media sosial, Wikipedia, artikel daring, subtitel dari rekaman video, dan dataset paralel. Dataset ini mencakup 4 miliar kata, baik yang bersifat formal maupun bahasa sehari-hari (Wilie et al., 2020). Model ini berakar pada arsitektur transformer dan mengadopsi konsep dari model BERT. Namun, IndoBERT mengalami proses pelatihan yang berfokus sebagai *masked language model*. Pelatihan tersebut dilakukan melalui library *Hugging Face* dengan mengikuti konfigurasi BERT-*base* (*uncased*) (Koto et al., 2020).

IndoBERT memiliki 12 lapisan tersembunyi dengan 768 *hidden size*, 12 *attention head*, dan lapisan *feed forward* tersembunyi dengan 3072 *hidden size*. Model ini mengubah kerangka kerja *Hugging Face* untuk memproses aliran teks yang berbeda untuk token dari dokumen yang berbeda, dan dilatih menggunakan 512 token.. Pada tugas klasifikasi, IndoBERT mencapai skor rata-rata tertinggi sebesar 88.46. Oleh karena itu, sebagai model monolingual, IndoBERT memiliki kemampuan yang lebih baik dalam memahami semantik bahasa formal dan kolokial dibandingkan dengan model multilingual. Namun, perlu dicatat bahwa, berbeda dengan model multilingual yang memiliki pemahaman yang lebih baik terhadap istilah asing, performa IndoBERT pada tugas *sequence labelling* tidak sebaik model XLM (Wilie et al., 2020).

2.8 *Hyperparameter*

Sejak berkembangnya *deep neural network* (DNN), teknologi ini telah memberikan dampak besar pada kehidupan sehari-hari manusia. Model DNN ini cukup efisien dalam pengaplikasiannya, tetapi proses pembuatan modelnya masih belum efisien (Yu & Zhu, 2020). Para peneliti berupaya dengan teliti untuk menyempurnakan desain model, algoritma, dan pemilihan *hyperparameter* yang tepat. *Hyperparameter* adalah sejumlah parameter yang tidak dapat diubah selama proses pelatihan model DNN. *Hyperparameter* dapat digunakan untuk membangun struktur model, seperti fungsi aktivasi, atau menentukan efisiensi dan akurasi pelatihan model, seperti *learning rate*, *batch size*, *epoch*, dan *optimizer*. *Learning rate* adalah *hyperparameter* yang mengontrol besarnya perubahan model dalam merespons estimasi kesalahan setiap kali bobot model diperbarui. Memilih *learning rate* yang tepat bisa menjadi tantangan, karena nilai yang terlalu kecil dapat memperlambat proses pelatihan, sementara nilai yang terlalu besar dapat menyebabkan pembelajaran yang tidak optimal, terlalu cepat, dan tidak stabil. Selain itu, *batch size* adalah jumlah sampel data yang dimasukkan ke dalam model selama satu iterasi pelatihan. Hal ini dilakukan karena model tidak dapat memproses semua data sekaligus, terutama jika data yang dilatih sangat besar. Misalnya, jika *batch size* bernilai 16, model akan melatih 16 data pertama dalam satu kali iterasi, kemudian 16 data berikutnya, dan seterusnya hingga seluruh data selesai dilatih. Sedangkan, satu *epoch* berarti seluruh dataset telah dilatih satu kali (Yu & Zhu, 2020).

2.9 Fungsi Optimasi Adam

Adam (*Adaptive Moment Estimation*) merupakan algoritma optimasi gradien stokastik (*stochastic gradient descent*) yang sering digunakan dalam *machine learning* untuk mencari nilai optimal dari parameter model. Adam mengintegrasikan konsep dari algoritma momentum dan RMSProp (*Root Mean Square Propagation*) untuk secara adaptif menyesuaikan laju pembelajaran (*learning rate*) bagi setiap parameter model. Dalam algoritma Adam, setiap parameter model memiliki *learning rate* yang unik dan disesuaikan secara adaptif berdasarkan momentum dari gradien sebelumnya serta RMS dari gradien saat ini. Adam melakukan beberapa langkah untuk mengoptimalkan parameter. Namun, Adam memiliki persamaan yang lebih rumit dan cenderung rentan terhadap *overfitting* (Kingma & Ba, 2017).

Pada fungsi optimasi Adam, terdapat proses menghitung gradien dari *batch* data pelatihan. Dalam tahapan ini kita perlu melakukan proses *backpropagation* pada model *neural network*. Rumus perhitungan gradien pada setiap lapisan akan bervariasi tergantung pada arsitektur model *neural network* yang digunakan dan fungsi aktivasi yang diterapkan pada setiap lapisan (*layer).* Secara umum, gradien pada suatu lapisan dihitung dengan mengalikan gradien dari lapisan berikutnya dengan turunan fungsi aktivasi lapisan tersebut terhadap inputnya (Kingma & Ba, 2017). Berikut adalah persamaan yang terdapat pada fungsi optimasi Adam ditunjukkan pada Persamaan 2.13.

$$W_{t+1} = W_t - \frac{\alpha}{\sqrt{v_t} + \epsilon} b_t \tag{2.13}$$

W_t mengacu pada *weight* dan bias pada iterasi ke-t, α merupakan *learning rate*, ϵ merupakan nilai kecil yang umumnya diatur menjadi 10^{-8}, b_t merupakan *moving average* dari gradien pada iterasi ke-t atau momentum pertama pada *epoch* sebelumnya, v_t merupakan *moving average* dari kuadrat gradien pada iterasi ke-t atau momentum kedua

pada *epoch* sebelumnya, serta t merupakan langkah waktu atau iterasi ke-t dengan rentang t adalah $t \in \{0, 1, 2, \ldots, N-1\}$ di mana N adalah jumlah total iterasi yang dilakukan dalam proses optimasi. Persamaan v_t dan b_t berturut-turut dituliskan pada Persamaan 2.14 dan Persamaan 2.15.

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) \left(\frac{\partial \mathcal{L}(W_t)}{\partial W_t} \right)^2 \tag{2.14}$$

$$b_t = \beta_1 b_{t-1} + (1 - \beta_1) \left(\frac{\partial \mathcal{L}(W_t)}{\partial W_t} \right) \tag{2.15}$$

β_1 adalah faktor pengurangan momentum pertama yang pada umumnya diatur menjadi 0,9; β_2 adalah faktor pengurangan momentum kedua yang pada umumnya diatur menjadi 0,999; dan $\mathcal{L}(W_t)$ merupakan fungsi loss yang bergantung pada parameter W_t.

BAB III
METODOLOGI

Pada bab ini dijelaskan secara umum mengenai urutan pelaksanaan Tugas Akhir dengan langkah-langkah yang dilakukan ditunjukkan pada diagram alir.

3.1 Objek dan Aspek Penelitian

Pada penelitian Tugas Akhir ini menggunakan data yang diambil dari portal penyedia berita *online* berbahasa Indonesia sebagai objek penelitian yaitu bersumber dari ANTARA News, BBC Indonesia, Brilio.net, CNBC Indonesia, CNN Indonesia, Detik.com, Dream.co.id, IDN Times, iNews.id, JPNN.com, Kapanlagi.com, Kompas.com, Kompasiana.com, Liputan6.com, Merdeka.com, Okezone.com, Portal Pekalongan, Republika, Sindonews.com, Suara.com, TvOne News, Tribunnews.com, Tempo.co, Tirto.id, Viva.co.id, dan WowKeren.com. Adapun aspek dari penelitian ini adalah memprediksi ketidakselarasan pada berita *clickbait* menggunakan model *Indonesia Bidirectional Encoder Representations from Transformers* (IndoBERT).

3.2 Pengumpulan Data

Data yang digunakan pada penelitian ini bersifat primer yang dikumpulkan dengan menggunakan teknik *web scraper* melalui *script* Python. Pada tahapan ini, *script* dari Python digunakan dalam mengatur otomatisasi dari *library* yang digunakan untuk membuka laman sebuah situs web dan memasukkan data ke dalam *form* yang telah disediakan, serta mengekstrak data. Pengambilan data berupa *library* yang digunakan adalah *Selenium* yang berfungsi dalam membuka halaman situs web dan *Beatifulsoup* yang berfungsi dalam teknik parsing kode HTML sekaligus mengekstrak data yang dibutuhkan. Adapun fitur atau variabel data yang dikumpulkan adalah judul, penerbit, isi naskah berita, dan URL berita. Proses pengumpulan data berita dilakukan mulai tanggal 18 Februari sampai 12 Maret 2024.

3.3 Spesifikasi Data

Data pada penelitian ini bersifat primer. Data yang digunakan adalah data berita berbahasa Indonesia. Fitur-fitur yang terdapat pada data yang dikumpulkan di antaranya judul, penerbit, teks, dan URL. Berikut merupakan keterangan dari fitur-fitur data yang dimaksud tertera pada Tabel 3.1.

Tabel 3.1 Fitur Data

Fitur	Tipe Data	Deskripsi
Judul (*title*)	*string*	Judul pada berita atau sebuah artikel.
Penerbit	*categorical*	Penerbit berita atau portal yang menerbitkan berita.
Text	*string*	Teks isi dari berita, seluruh halaman jika terdapat *pagination*.
URL	*string*	Alamat halaman web berita.

Sementara itu, berdasarkan fitur atau variabel yang telah dikumpulkan melalui proses *scraping*, fitur atau variabel yang digunakan dalam penelitian ini adalah judul (*title*) dan isi naskah berita (*text*).

3.4 Pembersihan Data

Data yang digunakan dibersihkan dengan cara menghilangkan nilai kosong atau *null* dan nilai duplikat yang terdapat pada fitur *title* dan *text*. Pembersihan data pada kedua fitur yang memiliki nilai kosong atau *null* dapat memengaruhi hasil analisis dan model yang dibangun. Dalam kasus nilai kosong atau *null* pada kolom judul dan teks dapat menyebabkan model tidak dapat memprediksi hasil dengan akurat. Oleh karena itu, langkah awal dalam membersihkan data adalah dengan menghilangkan nilai kosong atau *null* pada kolom tersebut. Selain itu, perlu diperiksa kembali data pada kolom teks dan judul yang memiliki nilai duplikat. Nilai duplikat dapat memengaruhi hasil analisis dan model yang dibangun. Hal ini disebabkan data duplikat dapat memberikan bobot yang tidak wajar pada model. Dengan melakukan kedua proses tersebut, menghilangkan nilai kosong atau *null* dan nilai duplikat, data yang digunakan menjadi lebih valid dalam proses analisis.

3.5 Pelabelan Data

Pelabelan data atau anotasi data adalah proses yang dilakukan dengan memberi label positif dan label negatif. Label positif biasanya ditandai dengan angka 1 yang merepresentasikan bahwa berita tersebut selaras dan label negatif biasanya ditandai dengan angka 0 yang merepresentasikan bahwa berita tersebut tidak selaras (*clickbait*). Proses pelabelan ini dilakukan secara otomatis dengan bantuan *platform* berbasis *Artificial Intelligence* (AI) dalam menilai keselarasan judul dengan isi berita.

3.6 Analisis Eksplorasi Data

EDA (*Exploratory Data Analysis*) dilakukan khusus untuk fitur data yang digunakan, yaitu kolom teks dan kolom judul. Dalam proses EDA, beberapa distribusi yang dihitung mencakup jumlah kalimat dalam kolom teks dan jumlah kata dalam kolom judul. Penghitungan distribusi data ini bertujuan untuk menentukan pola distribusi data yang tepat, yang akan berdampak positif pada kemampuan model dalam memberikan prediksi untuk data yang belum pernah dilihat sebelumnya. EDA memainkan peran penting dengan menghitung distribusi data, membantu dalam pemahaman karakteristik data, dan memberikan wawasan tambahan. Dengan mengevaluasi distribusi data, kita dapat mengidentifikasi *outlier* atau data yang tidak standar. Fokus penghitungan distribusi data dilakukan pada kolom teks dan judul, yang merupakan atribut kunci dalam pembangunan model prediksi. Menghitung jumlah kata dan kalimat pada kedua kolom ini dapat memberikan informasi tentang kompleksitas teks dan judul, serta membantu dalam mengidentifikasi karakteristik data dan mendeteksi potensi anomali.

3.7 Pra-Pemrosesan Data

Pada tahapan ini bertujuan untuk mempersiapkan data sebelum memasuki model yang dibangun, sehingga pemrosesan data teks pada tahapan-tahapan selanjutnya dapat dilakukan secara efisien. Data yang diproses adalah data pada fitur judul dan teks. Tahapan *preprocessing* adalah sebagai berikut.

1. *Lowercasing* atau *Case Folding*
   Proses awal adalah mengubah data tersebut menjadi *lowercase* atau sering disebut

tahapan *case folding.* Proses ini digunakan untuk mengubah huruf besar menjadi huruf kecil. Hal ini bertujuan untuk mempermudah analisis data dan agar tidak terjadi duplikasi pada kata atau frasa yang sama dengan karakter yang berbeda.

2. *Punctuation Removal*
   Proses selanjutnya adalah penghapusan karakter seperti tanda baca pada fitur judul dan teks. Hal ini bertujuan untuk menghindari karakter yang tidak berguna dalam proses analisis data, seperti tanda baca atau karakter khusus, sehingga nantinya hanya fokus pada kata-kata atau frasa-frasa penting dalam data.

3. *Stopwords Removal*
   Dilanjutkan dengan proses penghapusan *stopwords* pada data. *Stopwords* merupakan kata - kata umum yang sering muncul dalam teks, seperti 'pada', 'yang', 'untuk', dan sebagainya. Kata-kata ini biasanya tidak memiliki arti yang penting dalam proses analisis data. Dengan penghapusan *stopwords*, kita dapat memfokuskan analisis terhadap kata-kata penting yang dapat memberikan makna dan informasi yang lebih penting dalam data. Selain itu, penghapusan *stopwords* juga dapat membantu mengurangi dimensi data dan mempercepat waktu pemrosesan data.

4. *Filtering*
   Pada tahapan ini merupakan proses penyaringan tiap data seperti pemisahan tanda baca dengan kata dan menghapus seluruh karakter yang bukan termasuk huruf, angka, dan tanda baca yang tersisa. Dalam proses ini menggunakan bantuan *library* RegEx (*regular expression*) untuk menyaring data. Status data saat ini masih terdapat tanda baca (.) dan (,) yang diberikan *whitespace* antara karakter sebelum dan setelahnya.

3.8 *Splitting Data*

Spliting data merupakan tahapan membagi data berita menjadi data latih dan data uji. Pembagian pada dataset dibagi menjadi tiga proporsi yang berbeda. Selain itu, pada data latih dibagi kembali untuk proses data validasi yang diproses oleh model. Pembagian pertama memiliki persentase 80% data untuk pelatihan dan 20% data untuk pengujian, pembagian kedua memiliki persentase 90% data untuk pelatihan dan 10% data untuk pengujian, serta pembagian ketiga memiliki persentase 95% data untuk data pelatihan dan 5% data untuk pengujian. Setiap data latih dibagi dengan persentase pembagian 80% untuk data latih dan 20% untuk data validasi.

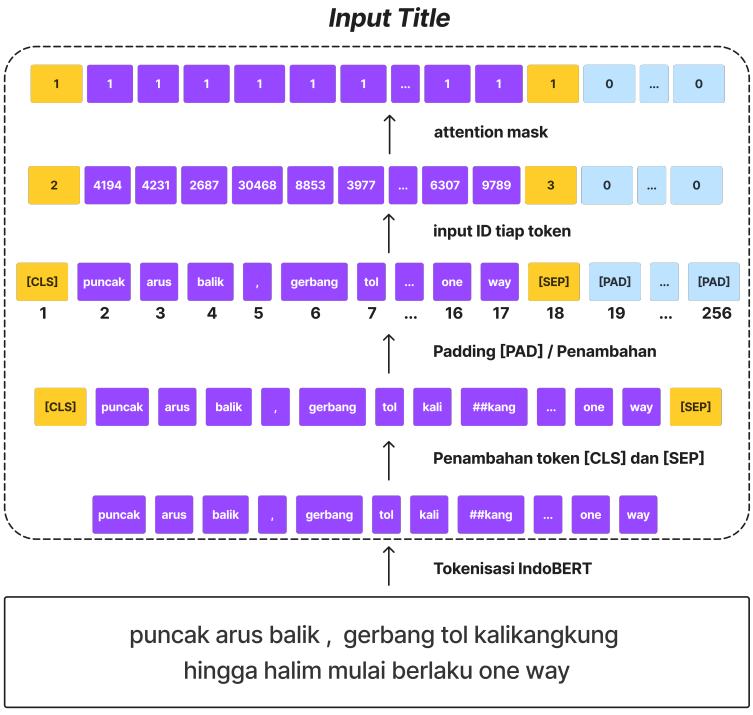
3.9 Penyeimbangan Distribusi Label

Penyeimbangan distribusi label merupakan tahapan dengan tujuan untuk mengatasi masalah ketidakseimbangan jumlah data pada setiap label (dalam kasus ini label 0 dan 1) yang dapat memengaruhi kinerja model pada tahap pelatihan. Jika salah satu label memiliki jumlah data yang lebih sedikit, maka model cenderung lebih fokus pada label tersebut dan mengabaikan label lainnya, sehingga menyebabkan ketidakseimbangan dalam hasil klasifikasi teks nantinya. Dengan melakukan tahapan penyeimbangan distribusi label, setiap label memiliki jumlah data yang seimbang, sehingga model yang dibangun belajar secara lebih baik dan menghasilkan hasil klasifikasi yang lebih akurat. Dalam kasus ini, teknik yang digunakan untuk menyeimbangkan distribusi label adalah *oversampling.* Teknik *oversampling* dilakukan dengan memperbanyak jumlah data pada

label yang jumlahnya lebih sedikit. Data yang diperoleh memiliki ketidakseimbangan dengan jumlah berita *non-clickbait* lebih banyak dibandingkan dengan yang mengandung *clickbait*. Sehingga, teknik *oversampling* digunakan untuk meningkatkan jumlah pada data berita yang berstatus *clickbait*.

3.10 Pelatihan Model IndoBERT

Setelah melalui prapemrosesan data dan pembagian data (*splitting*), selanjutnya dataset dilatih dengan model IndoBERT (*Indo Bidirectional Encoder Representations from Transformers*). Sebelum diolah ke dalam model, dataset diubah menjadi input yang dapat diterima oleh IndoBERT yaitu dalam bentuk *input formating* yang terdiri dari *input ids*, *token type ids*, dan *attention mask*. Proses ini menggunakan IndoBERT Tokenizer yang mengubah *sequence* kalimat menjadi potongan kata. Sebelum *input formating* IndoBERT dilanjutkan ke proses model, perlu dilakukan penyesuaian panjang pada teks. Pastikan semua teks memiliki panjang token yang sama atau paling banyak sesuai dengan kebutuhan model dengan menambahkan *token padding* ke teks yang terlalu pendek atau memotong teks yang terlalu panjang. Pada model IndoBERT disesuaikan dengan jumlah token agar tidak melebihi 512 sesuai dengan maksimal token IndoBERT. Parameter yang tersedia pada IndoBERT digunakan untuk menentukan bahwa teks dipotong jika jumlah token melebihi 512. Namun, pada penelitian ini setiap fitur yang digunakan, yaitu *title* dan *text* ditetapkan panjang maksimum token digunakan adalah 256. Sehingga saat penggabungan *input* antar dua fitur berjumlah sesuai maksimum token pada BERT yaitu 512. *Input formating* IndoBERT pada fitur *title* dilakukan proses *padding* saja karena data pada fitur tersebut tidak melebihi batas token maksimum. Proses pada fitur *title* tersebut ditunjukkan pada Gambar 3.1.

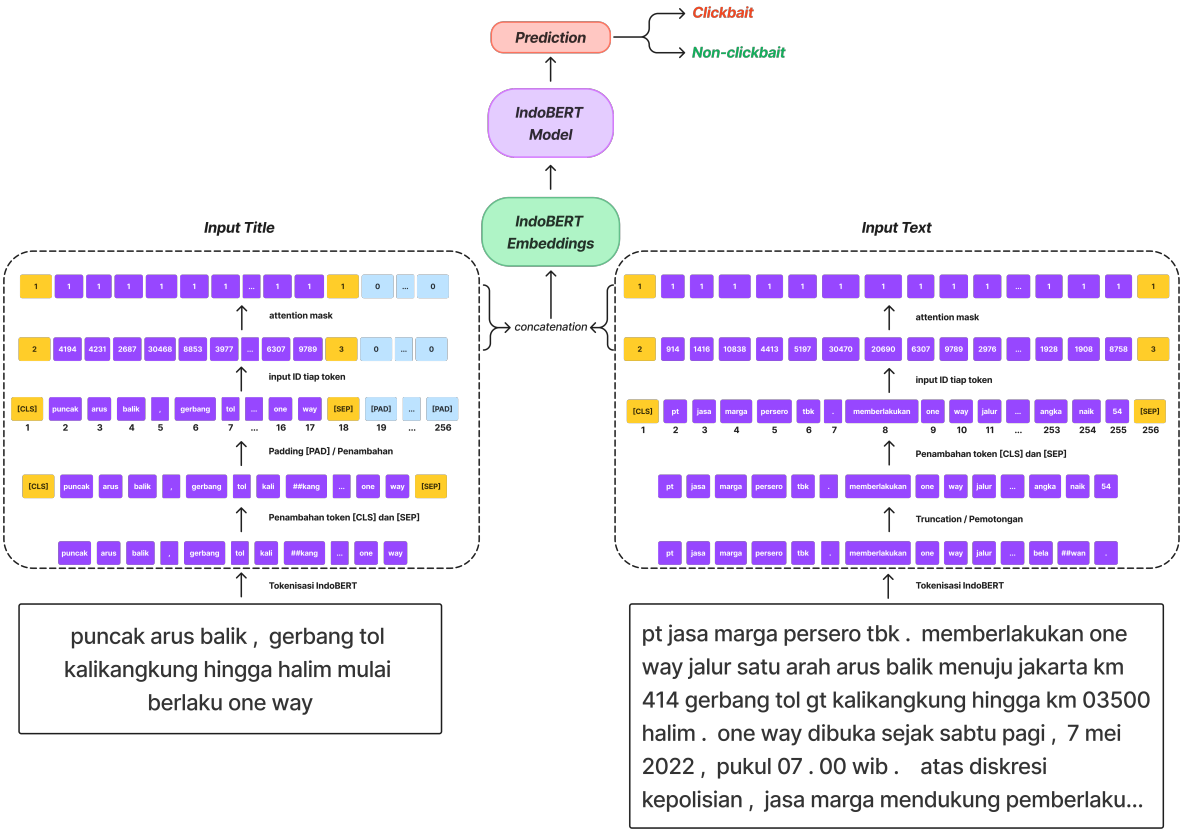


Gambar 3.1 *Input Title*

Selain fitur *title*, fitur *text* juga melalui proses *input formating*. Pada proses ini dilakukan *padding* dan *truncation* karena sebagian besar data pada fitur ini melebihi batas maksimal token yang telah ditetapkan yaitu 256. Proses pada fitur *text* tersebut ditunjukkan pada Gambar 3.2.

Gambar 3.2 *Input Text*

Kemudian dilakukan proses *fine-tuning* di mana *pre-train model* BERT diadaptasi untuk melakukan klasifikasi teks. Selama proses *fine-tuning* perlu dilakukan pemberian *hyperparameter* untuk pelatihan model IndoBERT seperti *optimizer* menggunakan *adam*, *batch size*, *epoch*, dan *learning rate* guna mengoptimalkan performa pada saat proses *fine-tuning* model. Proses *fine-tuning* model akan dilakukan pada data input *title* dan data input *text*. Hasil tensor yang diperoleh dari dua fitur tersebut digabungkan (*concatenate*) menggunakan bantuan *framework* PyTorch dan menghasilkan matriks *input ids* berdimensi $\mathbb{R}^{512\times768}$. Setelah proses *concatenation*, dilanjutkan menuju *layer embedding*, dan model IndoBERT (*layer encoder*). Proses *fine-tuning* IndoBERT untuk tugas klasifikasi dilakukan dengan menambahkan lapisan klasifikasi tambahan. *Library* Transformer menyediakan kelas bernama '*BertForSequenceClassification*' yang dirancang khusus untuk tugas klasifikasi. Kelas ini bekerja dengan mengambil *output* dari *layer pooler* dan menghitung *logits*. Nilai *logits* yang dihasilkan kemudian diproses menggunakan *softmax* untuk mendapatkan nilai probabilitas prediksi. Dalam tahap ini juga dilakukan dengan pemberian *hyperparameter* yang telah ditetapkan pada Tabel 4.9 dan melakukan kombinasi dari setiap *hyperparameter*. Model dan *tokenizer* IndoBERT yang digunakan adalah '*indobenchmark/indobert-base-p1*' yang memiliki 12 lapisan *encoder*, 12 head *attention*, 768 *hidden nodes*, dan 124 juta parameter. Gambaran dari proses tersebut ditunjukkan pada Gambar 3.3.



Gambar 3.3 Proses Input Data *title* dan *text*

3.11 Analisis dan Evaluasi Model

Pada tahap ini, tingkat keakuratan model IndoBERT dalam memprediksi keselarasan berita akan dievaluasi seberapa baik kemampuannya dalam menentukan label berita yang memiliki label *clickbait* dan *non-clickbait*. Ketika model melakukan proses prediksi pada data, dihasilkan nilai *True Positif* (TP), *False Positif* (FP), *True Negatif* (TN), dan *False Negatif* (FN) yang digunakan dalam perhitungan metrik evaluasi. Adapun beberapa parameter yang digunakan yaitu akurasi, presisi, recall, dan F1-score.

Akurasi merupakan parameter evaluasi yang menilai sejauh mana model yang digunakan mampu membuat prediksi yang benar dari total prediksi yang dihasilkan. Jumlah data yang terprediksi dengan benar dimisalkan dengan *true* dan jumlah keseluruhan data dengan *all*. Persamaan akurasi tertera pada Persamaan 3.1.

$$Accuracy = \frac{true}{all} \times 100\% \tag{3.1}$$

Presisi merupakan parameter evaluasi yang menilai sejauh mana model mampu membuat prediksi yang positif untuk kelas positif dari total prediksi positif yang dihasilkan. Persamaan presisi tertera pada Persamaan 3.2.

$$Precision = \frac{TP}{TP + FP} \times 100\% \tag{3.2}$$

Sensitivitas (*Recall*) merupakan parameter evaluasi yang menilai sejauh mana

kemampuan suatu model untuk mengenali kelas positif secara benar. Persamaan sensitivitas ditunjukkan pada Persamaan 3.3.

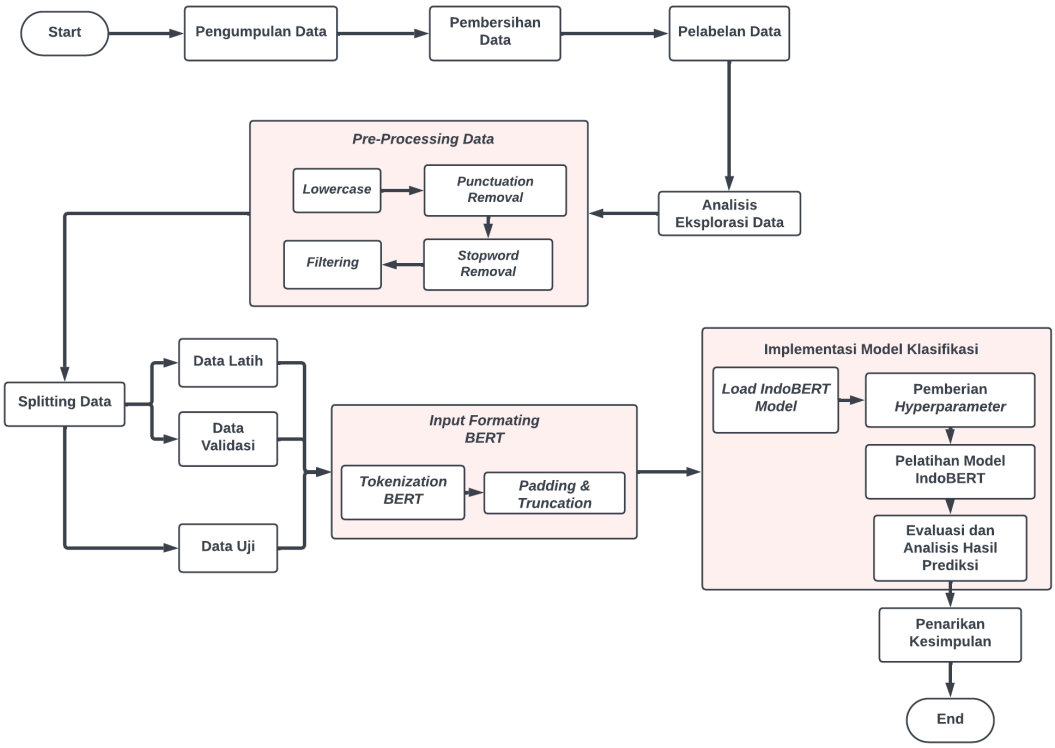
$$Recall = \frac{TP}{TP + FN} \times 100\% \tag{3.3}$$

F1-score merupakan parameter untuk mengatasi ketidakseimbangan kelas dengan menggabungkan *recall* dan presisi. Persamaan ini ditampilkan pada Persamaan 3.4.

$$F_1\text{-score} = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \times 100\% \tag{3.4}$$

3.12 Diagram Alir

Diagram alir penelitian pada tahap penelitian ini ditampilkan pada Gambar 3.4.



Gambar 3.4 Diagram Alir Penelitian

BAB IV
HASIL DAN PEMBAHASAN

Pada bagian ini dijelaskan terkait proses perancangan mulai dari tahap persiapan data hingga implementasi dalam memprediksi ketidakselarasan antara judul dengan isi berita berbahasa Indonesia menggunakan model IndoBERT.

4.1 Informasi Dataset dan Proses Pelabelan

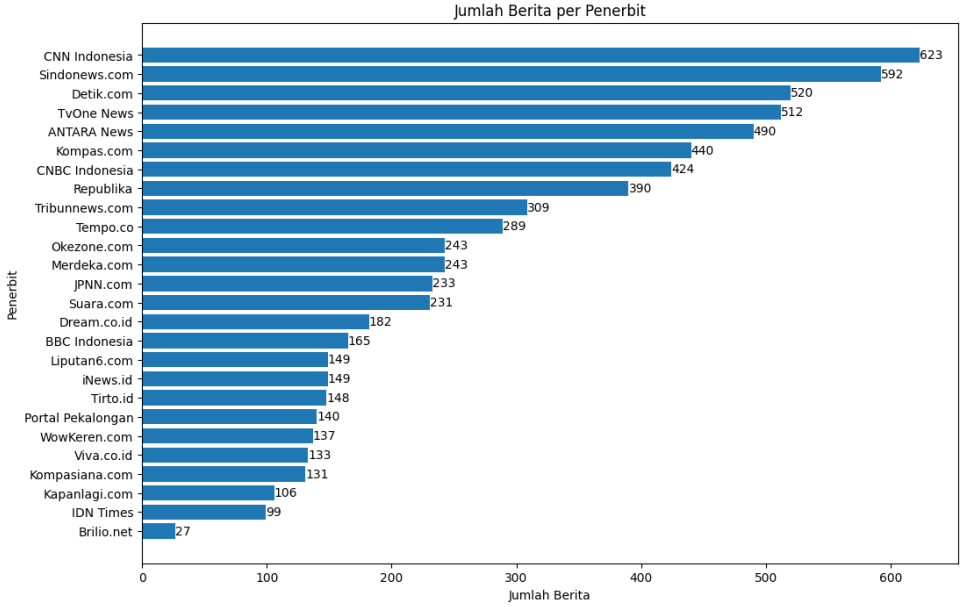
Pada tahapan awal dilakukan proses pengumpulan data dengan teknik *web scraping*. Data hasil *scraping* adalah data primer berupa kumpulan data berita berbahasa Indonesia dari 26 portal berita *online*. Terdapat 4 fitur atau variabel yang terkumpul yaitu judul, penerbit, narasi berita, dan alamat halaman web (URL). Proses pengumpulan data dengan teknik *scraping* ini dilaksanakan mulai 18 Februari 2024 sampai 12 Maret 2024 dengan jumlah mencapai 7105 data berita. *Raw dataset* atau akuisisi data hasil dari *scraping* tertera pada Tabel 4.1.

Tabel 4.1 Akusisi Data (*Raw Dataset*)

URL	**Penerbit**	***title***	***text***
https://www.tribunnews.com/regional/2022/06/02/...	Tribun News	VIRAL Video Mahasiswa soal Pasang Kateter Pasien, Pihak RSUD Wonosari dan Unisa Beri Klarifikasi	Beredar seorang mahasiswi yang merekam dirinya dengan narasi terkait pemasangan kateter urin kepada pasien laki-laki dan viral di TikTok. Adapun pengunggah video tersebut adalah akun TikTok bernama @moditabok. .....
https://www.wowkeren.com/berita/tampil/00428439.html	Wowkeren	Medina Zein Disebut Idap Bipolar Akut, Kondisi Terakhir Saat Dijenguk Mengkhawatirkan	Ade Anggareni selaku kuasa hukum keluarga Medina Zein membeberkan alasan mengapa sang pengusaha harus dirawat di Rumah Sakit Jiwa (RSJ) karena peyakit bipolarnya. ....
....	....	....	....
https://economy.okezone.com/read/2022/05/11/278/2592101/ ihsg-diserang-aksi...	Okezone	JIHSG Diserang Aksi Jual, Bagaimana Perdagangan Saham Hari Ini?	Indeks Harga Saham Gabungan (IHSG) turun tajam dalam dalam dua hari terakhir. Data penutupan bursa kemarin menunjukkan IHSG anjlok 1,30% di 6.819,79, menambah tekanan setelah babak belur 4,42% pada awal pekan. ...

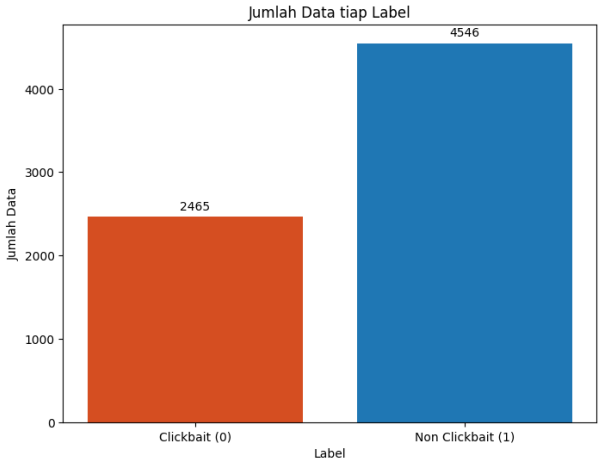
Berdasarkan fitur atau variabel dari akusisi data tersebut, terdapat dua fitur yang akan digunakan dalam proses penelitian ini yaitu *title* dan *text*. Adapun persebaran data

hasil pengumpulan data dari 26 portal berita *online* berbahasa Indonesia dengan jumlah 7.105 data yang ditampilkan pada Gambar 4.1.



Gambar 4.1 Plot Persebaran Berita Berdasarkan Penerbit

Data yang telah terkumpul akan dilakukan proses pembersihan terhadap nilai *null* dan nilai duplikat. Proses ini akan membersihkan data dari *noise* yang memengaruhi hasil analisis dan menghasilkan data bebas *noise* sebanyak 7.011 data. Dilanjutkan dengan proses pelabelan secara manual dengan menyesuaikan keselarasan dan ketidakselarasan tiap berita dengan memerhatikan judul dan narasi berita. Hasil pelabelan pada data berita tersebut masih bersifat tidak seimbang atau *imbalance* dengan jumlah yaitu 2.465 data untuk berita bersifat *clickbait* dan 4.546 data untuk berita bersifat *non-clickbait*. Proporsi pelabelan data dari seluruh berita ditampilkan pada Gambar 4.2.



Gambar 4.2 Plot Persebaran Data tiap Label

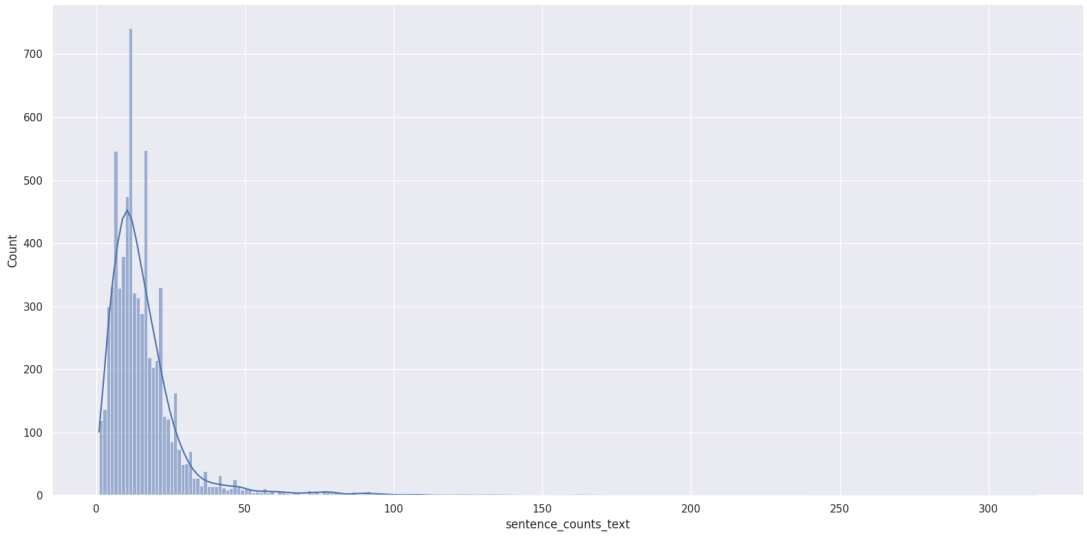
4.2 Analisis Eksplorasi Data

EDA (*Exploratory Data Analysis*) dilakukan setelah data melalui proses pembersihan dan pelabelan. Tahapan awal adalah mengidentifikasi komponen pada kedua fitur yaitu *title* dan *text*. Dalam hal ini akan dihitung banyak kalimat pada fitur *text* dan banyak kata pada fitur *title*. Diperoleh nilai *minimum*, *maximum*, dan *average* ditunjukkan pada Tabel 4.2.

Tabel 4.2 Eksplorasi terhadap Kata dan Kalimat pada Fitur Data

ket.	***sentence_counts_text***	***word_counts_title***	***max_word_counts_text***
minimum	1	2	4
maximum	316	24	277
average	16,2	9,6	31,16

Proses selanjutnya adalah penghitungan distribusi data pada fitur *text* dan *title*, yang merupakan atribut penting dalam membangun prediksi model. Proses ini dapat memberikan informasi tentang kerumitan *text* dan *title*, membantu dalam memahami karakteristik data, dan yang terpenting adalah mendeteksi keberadaan anomali pada data. Distribusi fitur *text* terhadap *sentence_counts_text* (banyak kalimat setiap *text*) sebelum dilakukan proses EDA ditunjukkan pada Gambar 4.3.



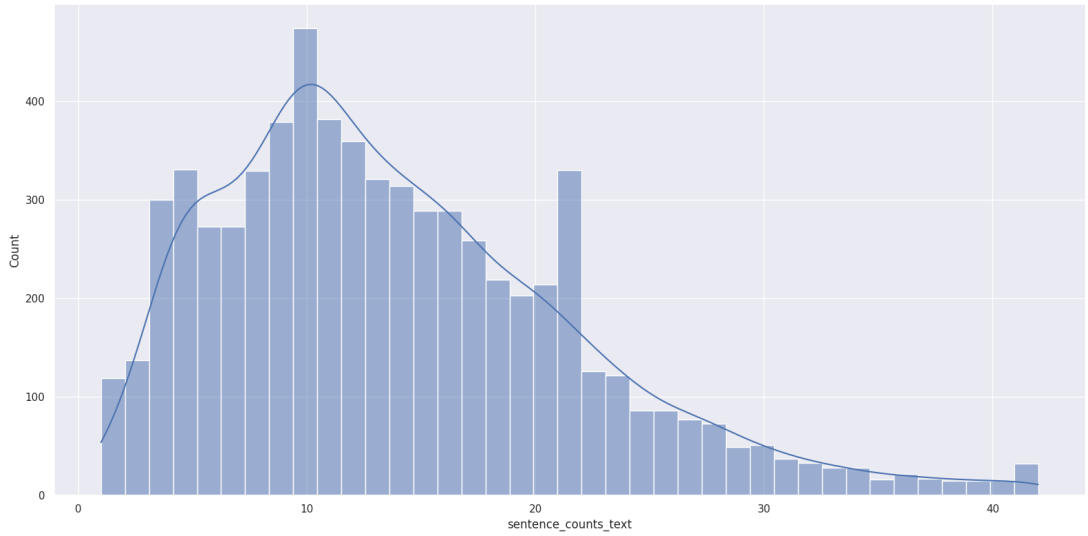
Gambar 4.3 Distribusi Jumlah Kalimat Fitur *Text* Sebelum Direduksi

Pada Gambar 4.3 tersebut terlihat beberapa *outlier* dan nilai ekstrim pada hasil distribusi. Tahapan ini akan dilakukan pembersihan data *outlier* dan nilai ekstrim dengan menerapkan pencarian data persentil. Persentase persentil yang akan diterapkan adalah 96% dengan mencari nilai persentil ke-96 dan mengabaikan sekitar 4% data yang mungkin mengandung *outlier* atau nilai ekstrim. Dari perhitungan persentil tersebut diperoleh nilai sebagai berikut.

$$\textit{Persentil ke-96} = \left(\frac{P}{100}\right) \times (N+1) = \left(\frac{96}{100}\right) \times (7011+1) = 6731,52$$

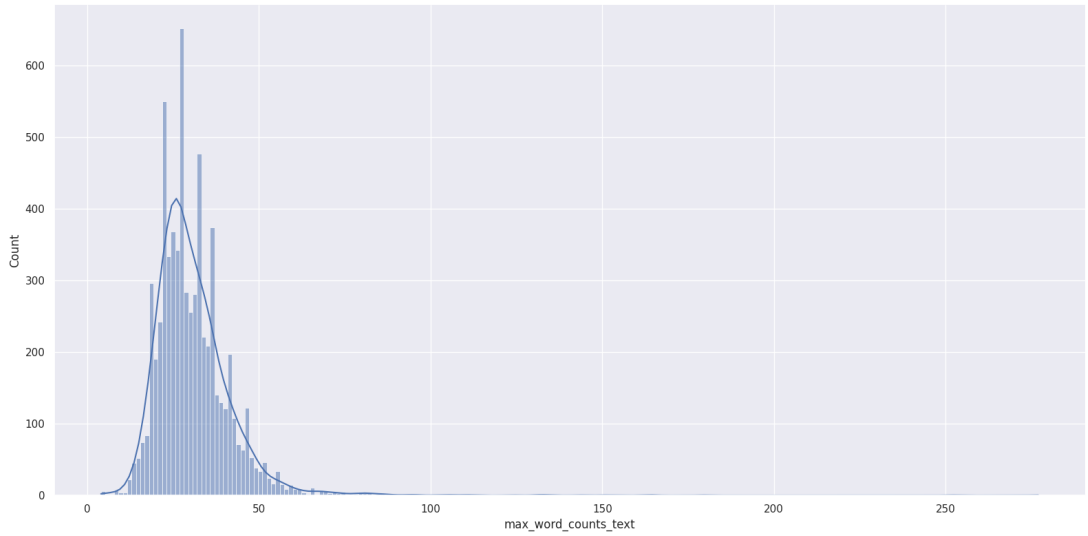
Namun, setelah dilakukan perhitungan persentil dengan persentase 96%, didapatkan nilai *threshold* yaitu 43. Nilai *threshold* tersebut digunakan untuk menyaring data.

Misalnya, jika *threshold* untuk data adalah 43, maka semua data dengan nilai *threshold* lebih dari 43 akan dihapus. Jumlah data yang tersisa setelah dilakukan filterisasi dengan *threshold* 43 adalah 6.722. Nilai 6.722 ini masih berada di sekitar hasil persentil yaitu 6.731,52. Ini berarti terdapat 289 data yang memiliki nilai lebih besar dari 43 dan akan direduksi. Hasil distribusi data setelah dilakukan analisis eksplorasi data dengan menggunakan nilai persentil ditunjukkan pada Gambar 4.4.



Gambar 4.4 Distribusi Jumlah Kalimat Fitur *Text* Setelah Direduksi

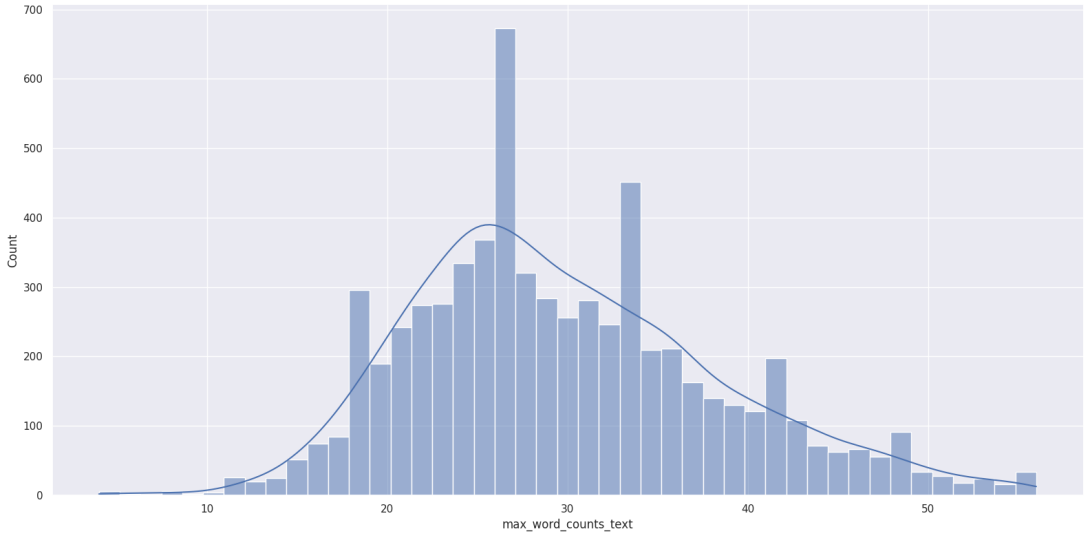
Selanjutnya, dilakukan proses EDA terhadap jumlah maksimal kata dalam kalimat pada variabel *text*. Distribusi pada fitur *text* terhadap *max_word_counts_text* (jumlah maksimal kata dalam kalimat) sebelum proses EDA ditunjukkan pada Gambar 4.5.



Gambar 4.5 Distribusi Jumlah Maksimum Kata Fitur *Text* Sebelum Direduksi

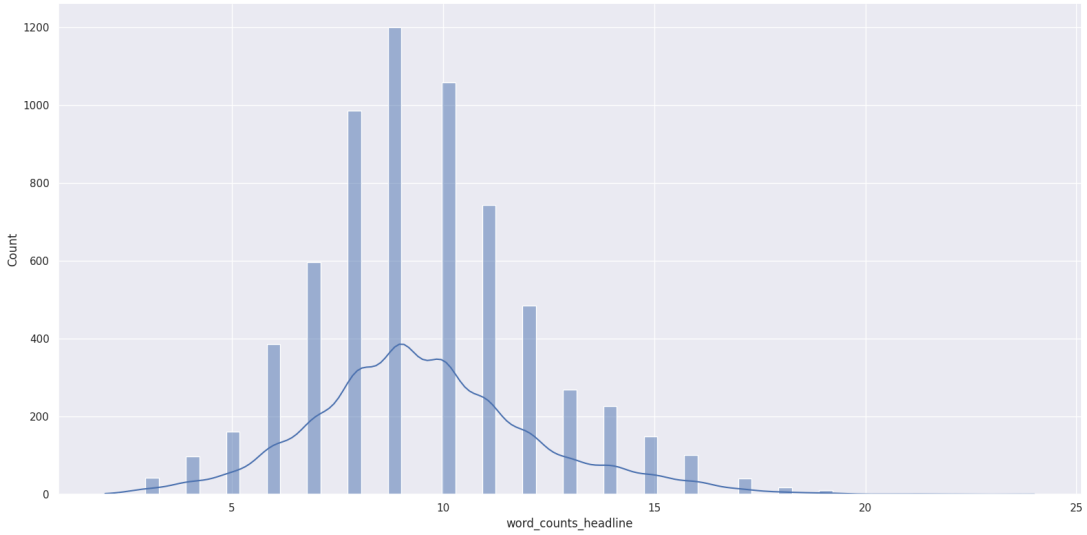
Pada Gambar 4.5 dilakukan proses yang sama yaitu dengan menerapkan pencarian data persentil ke-96 dan didapatkan nilai *threshold* yaitu 57. Jumlah data yang tersisa setelah dilakukan filterisasi pencarian nilai persentil dan penerapan nilai *threshold* adalah

6.575. Hasil distribusi data setelah dilakukan EDA pada proses tersebut ditunjukkan pada Gambar 4.6.



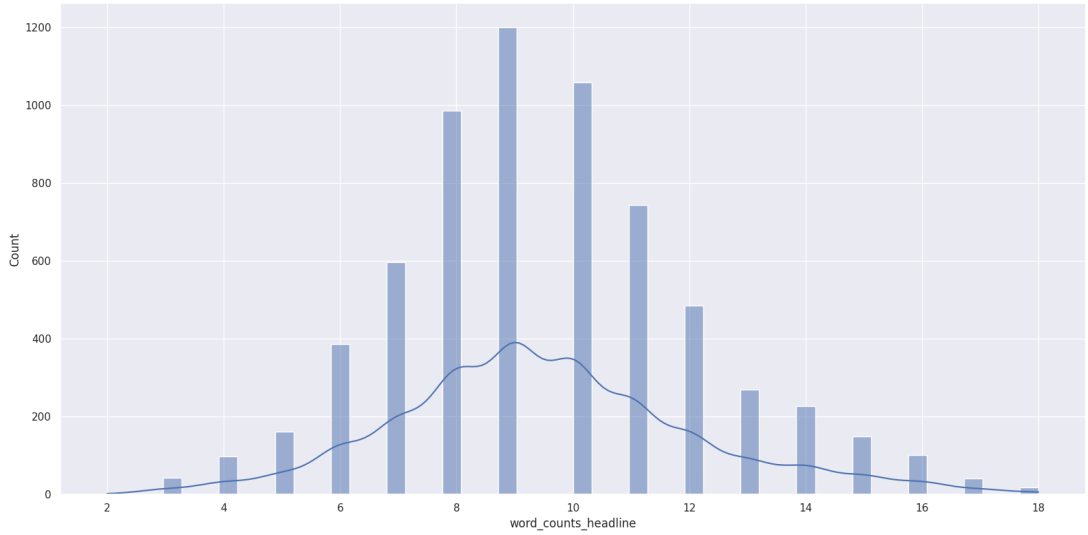
Gambar 4.6 Distribusi Jumlah Maksimum Kata Fitur *Text* Setelah Direduksi

Selain pada fitur *text*, dilakukan juga proses EDA pada fitur *title*. Pada proses ini dilakukan proses EDA pada fitur *title* terhadap *word_counts_title* (banyak kata setiap *title*). Distribusi sebelum dilakukan proses EDA pada fitur tersebut ditunjukkan pada Gambar 4.7.



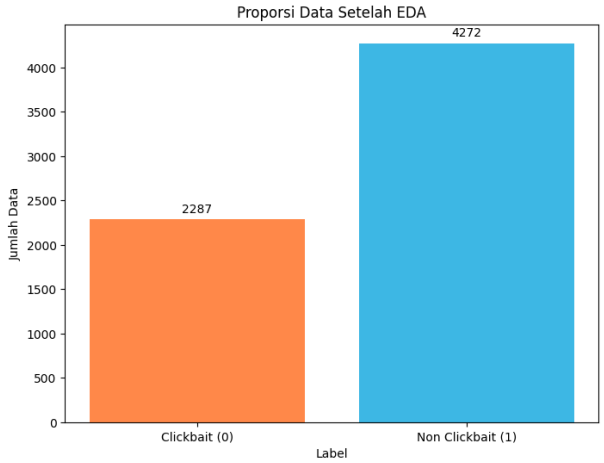
Gambar 4.7 Distribusi Jumlah Kata Fitur *Title* Sebelum Direduksi

Pada Gambar 4.7 dilakukan proses yang sama yaitu dengan menerapkan pencarian data persentil ke-96 dan didapatkan nilai *threshold* yaitu 19. Jumlah data yang tersisa setelah melalui proses ini yaitu 6.559. Hasil distribusi data setelah dilakukan EDA pada fitur *title* ditunjukkan pada Gambar 4.8.



Gambar 4.8 Distribusi Jumlah Kata Fitur *Title* Setelah Direduksi

Setelah melalui proses analisis eksplorasi data pada seluruh fitur, terdapat pengurangan data pada masing-masing label. Pada label 0 (*clickbait*) berjumlah 2.287 data dan label 1 (*non-clickbait*) berjumlah 4.272 data. Hasil persebaran data setelah melalui proses EDA atau analisis eksplorasi data ditunjukkan pada Gambar 4.9.



Gambar 4.9 Plot Persebaran Data Setelah EDA

4.3 Pra-pemrosesan Data

Data yang diperoleh dari proses *scraping* sering kali masih dalam bentuk mentah dan perlu diolah terlebih dahulu. Pra-pemrosesan data dilakukan untuk mengubah data mentah menjadi data bersih dan terstruktur. Tahapan ini penting untuk memastikan konsistensi komponen pada data yang akan digunakan dalam pelatihan model.

4.3.1 *Case Folding*

Tahap ini adalah langkah awal dalam pra-pemrosesan data yang bertujuan untuk mengubah semua karakter menjadi huruf kecil. Hal ini dilakukan karena data yang diperoleh sering kali tidak terstruktur dan konsisten dalam penggunaan huruf kapital. Hasil implementasi *case folding* pada salah satu data ditunjukkan pada Tabel 4.3.

Tabel 4.3 Implementasi *Case Folding*

Tahapan	**Teks sebelum *Case Folding***	**Teks setelah *Case Folding***
lowercasing title	**M**enteri **T**jahjo **K**umolo **S**etujui **P**emberlakuan **WFH** **B**agi **ASN** **S**epekan **S**etelah **M**udik	menteri tjahjo kumolo setujui pemberlakuan wfh bagi asn sepekan setelah mudik
lowercasing text	**M**enteri **P**endayagunaan **A**paratur **N**egara dan **R**eformasi **B**irokrasi (**PANRB**) **T**jahjo **K**umolo menyatakan setuju dengan usul **K**apolri **J**enderal **L**istyo **S**igit **P**rabowo agar instansi pemerintahan menerapkan kebijakan bekerja dari rumah atau work from home (**WFH**) selama seminggu setelah puncak arus balik lebaran pada 8 **M**ei 2022. **H**al ini untuk mencegah terjadinya kemacetan saat arus balik...	menteri pendayagunaan aparatur negara dan reformasi birokrasi (panrb) tjahjo kumolo menyatakan setuju dengan usul kapolri jenderal listyo sigit prabowo agar instansi pemerintahan menerapkan kebijakan bekerja dari rumah atau work from home (wfh) selama seminggu setelah puncak arus balik lebaran pada 8 mei 2022. hal ini untuk mencegah terjadinya kemacetan saat arus balik...

4.3.2 *Punctutation Removal*

Tahapan selanjutnya adalah *punctuation removal* atau penghapusan tanda baca. Semua tanda baca akan dihapus kecuali tanda baca (.) dan (,) untuk mempertahankan struktur kalimat. Hasil implementasi *punctutation removal* pada salah satu data ditunjukkan pada Tabel 4.4.

Tabel 4.4 Implementasi *Punctuation Removal*

Tahapan	**Sebelum *Remove Punctuation***	**Setelah *Remove Punctuation***
Punctuation Removal (text)	menteri pendayagunaan aparatur negara dan reformasi birokrasi (panrb) tjahjo kumolo menyatakan setuju dengan usul kapolri jenderal listyo sigit prabowo agar instansi pemerintahan menerapkan kebijakan bekerja dari rumah atau work from home (wfh) selama seminggu setelah puncak arus balik lebaran pada 8 mei 2022. hal ini untuk mencegah terjadinya kemacetan saat arus balik. tjahjo kumolo memberi arahan kepada seluruh pejabat pembina kepegawaian (ppk) agar mengatur jadwal wfh di instansi masing - masing...	menteri pendayagunaan aparatur negara dan reformasi birokrasi panrb tjahjo kumolo menyatakan setuju dengan usul kapolri jenderal listyo sigit prabowo agar instansi pemerintahan menerapkan kebijakan bekerja dari rumah atau work from home wfh selama seminggu setelah puncak arus balik lebaran pada 8 mei 2022. hal ini untuk mencegah terjadinya kemacetan saat arus balik. tjahjo kumolo memberi arahan kepada seluruh pejabat pembina kepegawaian ppk agar mengatur jadwal wfh di instansi masing masing...

4.3.3 *Stopwords Removal*

Pada proses ini akan dilakukan penghapusan *stopword* atau kata yang kurang bermakna ketika diproses melalui model. Dalam proses ini menggunakan bantuan *library* bernama Sastrawi untuk menghilangkan *stopword* tersebut. Daftar *stopword* yang dihilangkan pada penelitian ini ditunjukkan pada Tabel 4.5.

Tabel 4.5 Daftar *Stopword* Terdeteksi *Library* Sastrawi

Daftar Stopword
'yang', 'untuk', 'pada', 'ke', 'para', 'namun', 'menurut', 'antara', 'dia', 'dua', 'ia', 'seperti', 'jika', 'sehingga', 'kembali', 'dan', 'tidak', 'ini', 'karena', 'kepada', 'oleh', 'saat', 'harus', 'sementara', 'setelah', 'belum', 'kami', 'sekitar', 'bagi', 'serta', 'di', 'dari', 'telah', 'sebagai', 'masih', 'hal', 'ketika', 'adalah', 'itu', 'dalam', 'bisa', 'bahwa', 'atau', 'hanya', 'kita', 'dengan', 'akan', 'juga', 'ada', 'mereka', 'sudah', 'saya', 'terhadap', 'secara', 'agar', 'lain', 'anda', 'begitu', 'mengapa', 'kenapa', 'yaitu', 'yakni', 'daripada', 'itulah', 'lagi', 'maka', 'tentang', 'demi', 'dimana', 'kemana', 'pula', 'sambil', 'sebelum', 'sesudah', 'supaya', 'guna', 'kah', 'pun', 'sampai', 'sedangkan', 'selagi', 'sementara', 'tetapi', 'apakah', 'kecuali', 'sebab', 'selain', 'seolah', 'seraya', 'seterusnya', 'tanpa', 'agak', 'boleh', 'dapat', 'dsb', 'dst', 'dll', 'dahulu', 'dulunya', 'anu', 'demikian', 'tapi', 'ingin', 'juga', 'nggak', 'mari', 'nanti', 'melainkan', 'oh', 'ok', 'seharusnya', 'sebetulnya', 'setiap', 'setidaknya', 'sesuatu', 'pasti', 'saja', 'toh', 'ya', 'walau', 'tolong', 'tentu', 'amat', 'apalagi', 'bagaimanapun'

Berdasarkan Tabel 4.5 telah terdeteksi sebanyak 125 jenis *stopword* yang akan dihapus pada data penelitian yaitu pada fitur *title* dan *text*. Implementasi dari penghapusan *stopword* tersebut akan ditunjukkan pada Tabel 4.6.

Tabel 4.6 Implementasi *Stopwords Removal*

Tahapan	**Teks sebelum *Stopword Removal***	**Teks sesudah *Stopword Removal***
Stopwords Removal (text)	menteri pendayagunaan aparatur negara **dan** reformasi birokrasi panrb tjahjo kumolo menyatakan setuju **dengan** usul kapolri jenderal listyo sigit prabowo **agar** instansi pemerintahan menerapkan kebijakan bekerja **dari** rumah **atau** work from home wfh selama seminggu **setelah** puncak arus balik lebaran **pada** 8 mei 2022. **hal** ini untuk mencegah terjadinya kemacetan **saat** arus balik...	menteri pendayagunaan aparatur negara reformasi birokrasi panrb tjahjo kumolo menyatakan setuju usul kapolri jenderal listyo sigit prabowo instansi pemerintahan menerapkan kebijakan bekerja rumah work from home wfh selama seminggu puncak arus balik lebaran 8 mei 2022. ini mencegah terjadinya kemacetan arus balik...

4.3.4 *Filtering*

Selanjutnya akan dilakukan proses penyaringan atau *filtering* pada data *title* dan *text*. Tahapan ini menghapus karakter yang bukan huruf dan angka. Selain itu, akan dilakukan

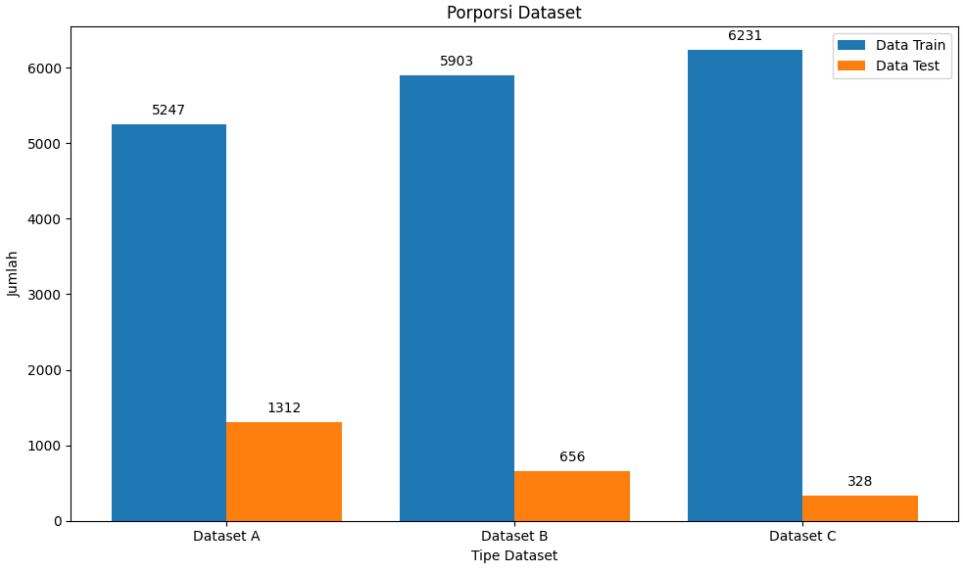
penambahan *whitespace* pada karakter yang berbeda kategori seperti tanda baca yang tersisa pada data tersebut. Implementasi dari proses *filtering* akan ditunjukkan pada Tabel 4.7.

Tabel 4.7 Implementasi *Filtering*

Tahapan	**Teks sebelum *Filtering***	**Teks sesudah *Filtering***
Filtering (text)	... tjahjo kumolo memberi arahan seluruh pejabat pembina kepegawaian ppk mengatur jadwal wfh instansi masingmasing. **â € œ**saya setuju pendapat kapolri instansi pemerintah menerapkan kebijakan wfh. seluruh ppk diharapkan mengatur pembagian jadwal penyelenggaraan pemerintahan pelayanan masyarakat tetap berjalan, **â €** ujar tjahjo lewat keterangan tertulis...	... tjahjo kumolo memberi arahan seluruh pejabat pembina kepegawaian ppk mengatur jadwal wfh instansi masing masing . saya setuju pendapat kapolri instansi pemerintah menerapkan kebijakan wfh . seluruh ppk diharapkan mengatur pembagian jadwal penyelenggaraan pemerintahan pelayanan masyarakat tetap berjalan , ujar tjahjo lewat keterangan tertulis...

4.4 Pembagian dan Penyeimbangan Distribusi Data

Pada penelitian ini menggunakan skenario pelatihan dan pengujian model dengan pembagian tiga tipe dataset yaitu Dataset **A**, Dataset **B**, dan Dataset **C**. Skenario tersebut bertujuan untuk menghasilkan kinerja model yang lebih beragam dan memudahkan dalam hal komparasi performa model. Hal yang membedakan dari ketiga jenis dataset tersebut adalah jumlah pembagiannya. Dataset **A** memiliki proporsi yaitu 80% untuk data latih dan 20% data uji, Dataset **B** memiliki proporsi yaitu 90% untuk data latih dan 10% untuk data uji, serta Dataset **C** memiliki proporsi yaitu 95% untuk data latih dan 5% untuk data uji. Ilustrasi pembagian dataset tersebut dapat ditunjukkan pada Gambar 4.10.



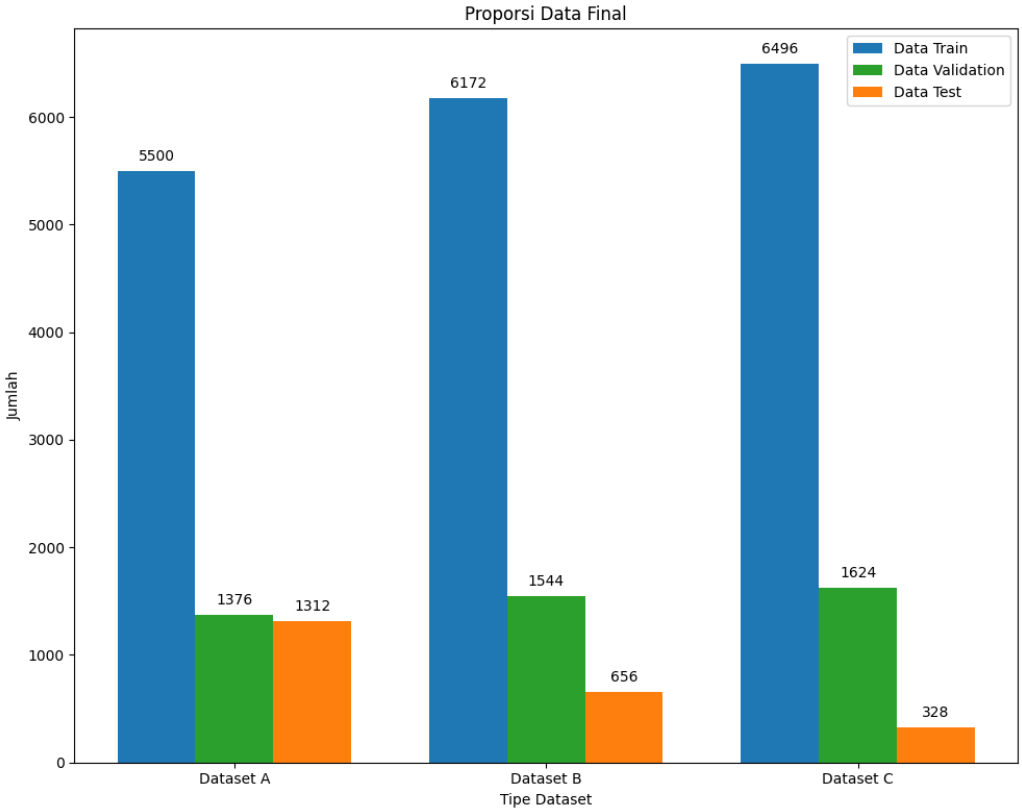
Gambar 4.10 Proporsi Awal Data Latih dan Data Uji

Secara umum, dataset pada Tugas Akhir ini yaitu dimisalkan pada $Dataset\mathbf{A} = \{(title_t, text_t, label_t) \mid 1 \leq t \leq |A|, title \in A, text \in A, label \in A\}$, menyatakan bahwa pasangan terurut data *title* dan *text* dengan labelnya. Notasi tersebut juga berlaku untuk Dataset **B** dan **C**. Namun, dataset tersebut memiliki ketidakseimbangan pada label yang disebut sebagai *imbalance data* seperti yang diilustrasikan pada Gambar 4.4. Pada tahap pelatihan model diperlukan proporsi yang seimbang khususnya pada label 0 dan 1. Oleh karena itu, dilakukan percobaan untuk menyeimbangkan data dengan metode *oversampling* pada data latih khususnya untuk menaikkan sampel pada label 0 (*clickbait*) seperti yang ditunjukkan pada Tabel 4.8.

Tabel 4.8 Jumlah Data Latih Sebelum dan Sesudah *Oversampling*

Tipe	**Sebelum *Oversampling***		**Sesudah *Oversampling***		**Total Akhir**
	Clickbait	***Non-Clickbait***	***Clickbait***	***Non-Clickbait***	
A	1809	3438	3438	3438	6876
B	2045	3858	3858	3858	7716
C	2171	4060	4060	4060	8120

Data latih pada semua tipe dataset yang telah dilakukan teknik *oversampling* akan dibagi sebanyak 20% sebagai data validasi atau *data validation*. Data uji atau *data test* tidak dilakukan *oversampling* karena akan berfungsi sebagai data baru pada saat proses prediksi dan evaluasi akhir. Proporsi data yang sudah siap untuk masuk model dapat diilustrasikan pada Gambar 4.11.



Gambar 4.11 Proporsi Akhir Dataset Uji Coba Model

Setelah mempersiapkan pembagian dataset, dalam pelatihan nanti terutama saat proses *fine-tuning* model, diperlukan *hyperparameter* untuk keberlangsungan kinerja model. Dalam penelitian ini, akan ditetapkan *hyperparameter* yang digunakan adalah *epoch, learning rate,* dan *batch size*, serta dengan bantuan *optimizer* menggunakan *Adam*. Daftar *hyperparameter* yang berperan dan akan melakukan proses *fine-tuning* pada penelitian ini ditunjukkan pada Tabel 4.9.

Tabel 4.9 Nilai *Hyperparameter* Model

No.	***Hyperparameter***	**Ukuran**
1	*epoch*	20
2	*learning rate*	1e-6, 2e-6, 5e-6, 1e-7, 2e-7, dan 5e-7
3	*batch size*	16 dan 32

4.5 Pelatihan dan Hasil Performansi Model

Setelah melewati proses pembagian data (*spliting data*) dan penetapan *hyperparameter*, selanjutnya data teks tersebut akan masuk ke dalam proses IndoBERT untuk proses pelatihan model dan memperoleh hasil performansi. Selain itu, disajikan variasi berita *clickbait* dan *non-clickbait* yang diuji coba menggunakan *web predict* hasil *deployment* dari model terbaik.

4.5.1 Representasi Input Model IndoBERT

Hal paling utama yang dilakukan adalah mengubah seluruh data teks menjadi *input formating* untuk masukan data awal yang akan disesuaikan dengan model IndoBERT. Model IndoBERT merupakan modifikasi dari model Transformer, tetapi input awal IndoBERT memiliki kriterianya sendiri dan hanya menggunakan bagian *encoder* dari Transformer. Tahapan awal adalah proses tokenisasi data teks menggunakan *IndoBERT Tokenizer*. Tokenisasi BERT ini akan memecah teks menjadi beberapa bagian sesuai dengan kamus yang dimiliki dan menambahkan token khusus pada awal kalimat yaitu [CLS] dan pada akhir kalimat yaitu [SEP]. Sebagai contoh, dilakukan tokenisasi (T) pada kalimat $S =$ *"puncak arus balik , gerbang tol kalikangkung hingga halim mulai berlaku one way"* menjadi sebagai berikut.

$T =$ [[CLS]; puncak; arus; balik; , ; gerbang; tol; kali; ##kang; ##ku; ##ng; hingga; halim; mulai; berlaku; one; way; [SEP]]

Tokenisasi (T) akan diproses untuk menghasilkan input pada model IndoBERT. Adapun beberapa input yang akan masuk pada proses *embedding* dari IndoBERT yaitu *input ids, attention mask,* dan *token type ids*. Nilai *input ids* merupakan representasi numerik dari teks (S) yang telah melalui tokenisasi menjadi

$$\textit{input ids} = \begin{bmatrix} 2 & 4194 & 4231 & 2687 & 30468 & ... & 2385 & 6307 & 9789 & 3 \end{bmatrix}$$

di mana *input ids* merupakan anggota token setiap bagian T pada kalimat S. Nilai *input ids* tersebut berdimensi $1 \times n$ di mana $n = |T|$. Selanjutnya, dalam proses tokenisasi tersebut juga akan menghasilkan input berupa *token type ids* untuk membedakan antara dua segmen dalam teks (misalnya, dalam tugas pertanyaan-jawaban). Namun, pada penelitian ini tidak menerapkan dua segmen dan hanya menghasilkan nilai 0 pada input tersebut.

$$\textit{token type ids} = \begin{bmatrix} 0 & 0 & 0 & 0 & ... & 0 & 0 & 0 & 0 \end{bmatrix}$$

dimensi yang dihasilkan oleh *token type ids* juga sama dengan *input ids*. Selain itu, proses tokenisasi juga menghasilkan *attention mask* (memiliki dimensi sama dengan *input ids*) yang digunakan dalam mekanisme *self-attention*, biasanya token *padding* bernilai 0, sehingga model hanya memperhatikan token yang relevan (fokus pada nilai 1). Pada token (T) berupa teks secara keseluruhan tanpa token *padding* sehingga *attention mask* akan memberikan nilai 1 seperti berikut.

$$attention\ mask = \begin{bmatrix} 1 & 1 & 1 & 1 & ... & 1 & 1 & 1 & 1 \end{bmatrix}$$

Selain itu, terdapat informasi posisi tiap token dalam bentuk numerik yaitu *position ids*. Informasi posisi yang didapat dari tiap token *input ids* adalah

$$position\ ids = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & ... & 14 & 15 & 16 & 17 \end{bmatrix}$$

di mana *position ids* merupakan representasi dari posisi pada tiap token (T). Proses penentuan *position ids* memungkinkan pemahaman konteks kalimat, sehingga kata yang sama dalam kalimat dapat dimaknai berbeda tergantung posisinya. Namun, pada penelitian ini, hasil ketiga input yaitu *input ids, token type ids*, dan *attention mask* akan digabungkan (*concatenation*) antar dua variabel yaitu *title* dan *text*.

4.5.2 Representasi *Embedding* dan *Encoder* IndoBERT

Proses selanjutnya adalah seluruh input yang telah dihasilkan sebelumnya akan masuk ke dalam model IndoBERT. Tahap awal adalah melalui proses *BERT Embeddings* sebagai *input embedding* yaitu penyematan ke ruang vektor. *Embedding* tersebut berisi sekumpulan vektor posisi yang mewakili setiap token sehingga membentuk suatu matriks. Model IndoBERT menggunakan ruang vektor bernilai 768 dimensi. Seluruh *embeddings* akan bernilai 768 dimensi. Dalam contoh perhitungan berikut akan ditampilkan *embedding* berukuran 5 dimensi dari penerapan 18 token (T). Hasil *Word Embeddings* menggunakan model *pretrained* IndoBERT dari *input ids* adalah

$$Word\ Embeddings = \begin{bmatrix} -0.0249 & 0.0247 & -0.0134 & -0.0117 & 0.0317 \\ -0.0273 & 0.0329 & -0.0334 & -0.0210 & -0.0337 \\ -0.0181 & -0.0154 & -0.0278 & -0.0136 & 0.0221 \\ 0.0274 & -0.0176 & 0.0244 & -0.0418 & 0.0035 \\ 0.0372 & 0.0166 & 0.0058 & 0.0004 & -0.0403 \\ -0.0091 & 0.0325 & -0.0373 & -0.0304 & 0.0121 \\ 0.0202 & 0.0002 & 0.0094 & -0.0125 & 0.0217 \\ 0.0065 & 0.0236 & -0.0314 & -0.0331 & -0.0239 \\ -0.0176 & 0.0240 & 0.0417 & -0.0572 & 0.0043 \\ 0.0300 & 0.0370 & -0.0036 & -0.0145 & -0.0162 \\ 0.0376 & -0.0045 & -0.0091 & -0.0192 & -0.0059 \\ 0.0553 & -0.0027 & -0.0492 & 0.0312 & 0.0084 \\ -0.0336 & 0.0088 & 0.0015 & -0.0476 & 0.0379 \\ -0.0042 & 0.0370 & -0.0170 & 0.0246 & 0.0073 \\ -0.0155 & -0.0292 & -0.0254 & -0.0222 & -0.0486 \\ -0.0172 & -0.0137 & -0.0075 & -0.0042 & 0.0101 \\ -0.0355 & -0.0011 & -0.0200 & -0.0087 & 0.0005 \\ 0.0067 & -0.0076 & -0.0042 & -0.0003 & 0.0305 \end{bmatrix}$$

di mana *Word Embeddings* $\in \mathbb{R}^{18\times5}$ berbentuk kumpulan vektor posisi dari *input ids* menjadi sebuah matriks yaitu pada contoh ini adalah 18. Selanjutnya, *position ids* juga berdimensi seperti *input ids* yang akan diubah ke *Position Embeddings* dengan model *pretrained* IndoBERT sebagai berikut.

$$Position\ Embeddings = \begin{bmatrix} -0.0059 & -0.0211 & 0.0248 & 0.0092 & 0.0314 \\ 0.0099 & 0.0072 & -0.0003 & -0.0117 & -0.0022 \\ -0.0058 & -0.0001 & 0.0033 & -0.0124 & -0.0050 \\ -0.0123 & -0.0004 & -0.0070 & -0.0195 & -0.0148 \\ -0.0137 & 0.0016 & -0.0042 & -0.0017 & -0.0090 \\ -0.0183 & 0.0022 & -0.0098 & -0.0119 & -0.0104 \\ -0.0073 & 0.0036 & -0.0000 & 0.0050 & 0.0025 \\ -0.0091 & 0.0039 & -0.0037 & -0.0110 & -0.0002 \\ -0.0066 & 0.0040 & -0.0008 & 0.0029 & -0.0001 \\ -0.0114 & 0.0042 & -0.0104 & -0.0121 & -0.0084 \\ -0.0025 & 0.0097 & -0.0041 & 0.0022 & -0.0079 \\ -0.0092 & 0.0024 & -0.0085 & -0.0044 & -0.0145 \\ 0.0053 & 0.0082 & -0.0001 & -0.0100 & -0.0090 \\ -0.0005 & -0.0006 & 0.0017 & -0.0004 & -0.0088 \\ -0.0008 & -0.0029 & -0.0044 & -0.0159 & -0.0053 \\ 0.0041 & -0.0014 & 0.0014 & -0.0043 & 0.0029 \\ -0.0053 & -0.0076 & -0.0078 & -0.0097 & -0.0022 \\ 0.0026 & -0.0005 & -0.0063 & -0.0093 & -0.0039 \end{bmatrix}$$

di mana *Position Embeddings* $\in \mathbb{R}^{18\times5}$ dengan setiap baris mewakili anggota token dalam *position ids* yaitu pada contoh ini adalah 18. *Embeddings* yang ketiga adalah *Token Type Embeddings* representasi ruang vektor dari token yang dimiliki *token type ids* menggunakan model *pretrained* IndoBERT yaitu

$$Token\ Type\ Embeddings = \begin{bmatrix} 0.0005 & 0.0002 & -0.0038 & 0.0114 & 0.0028 \\ 0.0005 & 0.0002 & -0.0038 & 0.0114 & 0.0028 \\ 0.0005 & 0.0002 & -0.0038 & 0.0114 & 0.0028 \\ 0.0005 & 0.0002 & -0.0038 & 0.0114 & 0.0028 \\ 0.0005 & 0.0002 & -0.0038 & 0.0114 & 0.0028 \\ 0.0005 & 0.0002 & -0.0038 & 0.0114 & 0.0028 \\ 0.0005 & 0.0002 & -0.0038 & 0.0114 & 0.0028 \\ 0.0005 & 0.0002 & -0.0038 & 0.0114 & 0.0028 \\ 0.0005 & 0.0002 & -0.0038 & 0.0114 & 0.0028 \\ 0.0005 & 0.0002 & -0.0038 & 0.0114 & 0.0028 \\ 0.0005 & 0.0002 & -0.0038 & 0.0114 & 0.0028 \\ 0.0005 & 0.0002 & -0.0038 & 0.0114 & 0.0028 \\ 0.0005 & 0.0002 & -0.0038 & 0.0114 & 0.0028 \\ 0.0005 & 0.0002 & -0.0038 & 0.0114 & 0.0028 \\ 0.0005 & 0.0002 & -0.0038 & 0.0114 & 0.0028 \\ 0.0005 & 0.0002 & -0.0038 & 0.0114 & 0.0028 \\ 0.0005 & 0.0002 & -0.0038 & 0.0114 & 0.0028 \\ 0.0005 & 0.0002 & -0.0038 & 0.0114 & 0.0028 \end{bmatrix}$$

di mana *Token Type Embeddings* $\in \mathbb{R}^{18\times5}$ dengan setiap baris mewakili anggota token dalam *token type ids* yaitu pada contoh ini adalah 18. Setelah dilakukan proses *input embeddings*, ketiga *embedding* dalam bentuk tensor tersebut akan digabungkan menjadi satu kesatuan *embedding*. Hasil penggabungan *embedding* dituliskan seperti berikut.

$$\textit{Summed Embeddings} = \begin{bmatrix} -0.030240 & 0.003763 & 0.007626 & 0.008865 & 0.065863 \\ -0.016821 & 0.040294 & -0.037387 & -0.021312 & -0.033182 \\ -0.023392 & -0.015365 & -0.028249 & -0.014592 & 0.019873 \\ 0.015605 & -0.017830 & 0.013653 & -0.049923 & -0.008574 \\ 0.024048 & 0.018453 & -0.002189 & 0.010083 & -0.046534 \\ -0.026942 & 0.034816 & -0.050838 & -0.030919 & 0.004487 \\ 0.013436 & 0.003951 & 0.005656 & 0.003895 & 0.026980 \\ -0.002048 & 0.027637 & -0.038773 & -0.032715 & -0.021273 \\ -0.023730 & 0.028241 & 0.037053 & -0.042868 & 0.007032 \\ 0.019138 & 0.041387 & -0.017757 & -0.015191 & -0.021807 \\ 0.035606 & 0.005374 & -0.016963 & -0.005614 & -0.010959 \\ 0.046682 & -0.000034 & -0.061418 & 0.038242 & -0.003305 \\ -0.027799 & 0.017133 & -0.002332 & -0.046159 & 0.031672 \\ -0.004174 & 0.036523 & -0.018996 & 0.035590 & 0.001293 \\ -0.015781 & -0.031931 & -0.033648 & -0.026780 & -0.051182 \\ -0.012602 & -0.014915 & -0.009866 & 0.002923 & 0.015793 \\ -0.040249 & -0.008581 & -0.031629 & -0.006971 & 0.001037 \\ 0.009839 & -0.007841 & -0.014326 & 0.001764 & 0.029318 \end{bmatrix}$$

Hasil penggabungan *embedding* (*summed embedding*) dimisalkan dengan x. Pada proses awal *self-attention*, untuk setiap $x_i \in x$ di mana $0 < i \leq 18$ dan untuk setiap $x_j \in x$ di mana $0 < j \leq 18$ akan dilakukan proses *dot-product* untuk mendapatkan nilai *similarity* (S_i) dari x_i. Proses ini juga melibatkan nilai bobot untuk setiap *query* (q) pada kata ke-i (q_i). Diberikan matriks bobot untuk q yaitu $\mathbf{W}_h^Q \in \mathbb{R}^{5\times5}$ dan matriks bobot untuk *key* (k) yaitu $\mathbf{W}_h^K \in \mathbb{R}^{5\times5}$. Tinjau bahwa $x_i \in \mathbb{R}^{1\times5}$ dan $x_j \in \mathbb{R}^{1\times5}$, sehingga diperoleh $q_i = x_i\mathbf{W}_h^Q \in \mathbb{R}^{1\times5}$ dan $k_j = x_j\mathbf{W}_h^K \in \mathbb{R}^{1\times5}$. Tahap awal mencari nilai *similarity* (S_i) dengan memisalkan untuk $h = 1$ (*head* pertama) sehingga $\mathbf{W}_1^Q$ dan $\mathbf{W}_1^K$ merupakan matriks identitas, sedemikian hingga $q_i = x_i$ dan $k_j = x_j$. Contoh implementasi mencari nilai *similarity* pertama (S_1) menggunakan Persamaan 2.6 adalah

$$\begin{aligned} S_{1;1} &= \left(\frac{q_1 k_1^T}{\sqrt{d_k}}\right) \\ &= [-0.03024 \quad 0.003763 \quad 0.007626 \quad 0.008865 \quad 0.065863] \begin{bmatrix} -0.03024 \\ 0.003763 \\ 0.007626 \\ 0.008865 \\ 0.065863 \end{bmatrix} / \sqrt{5} \\ &= \mathbf{0.002416} \end{aligned}$$

$$S_{1;2} = \left(\frac{q_1 k_2^T}{\sqrt{d_k}}\right)$$

$$= [-0.03024 \quad 0.003763 \quad 0.007626 \quad 0.008865 \quad 0.065863] \begin{bmatrix} -0.016821 \\ 0.040294 \\ -0.037387 \\ -0.021312 \\ -0.033182 \end{bmatrix} /\sqrt{5}$$

$$= \textbf{-0.000894}$$

$$S_{1;3} = \left(\frac{q_1 k_3^T}{\sqrt{d_k}}\right)$$

$$= [-0.03024 \quad 0.003763 \quad 0.007626 \quad 0.008865 \quad 0.065863] \begin{bmatrix} -0.023392 \\ -0.015365 \\ -0.028249 \\ -0.014592 \\ 0.019873 \end{bmatrix} /\sqrt{5}$$

$$= \textbf{0.0007216}$$

$$S_{1;4} = \left(\frac{q_1 k_3^T}{\sqrt{d_k}}\right)$$

$$= [-0.03024 \quad 0.003763 \quad 0.007626 \quad 0.008865 \quad 0.065863] \begin{bmatrix} 0.015605 \\ -0.017830 \\ 0.013653 \\ -0.04992 \\ -0.00857 \end{bmatrix} /\sqrt{5}$$

$$= \textbf{-0.000645}$$

$$\vdots$$

$$S_{1;18} = \left(\frac{q_1 k_{18}^T}{\sqrt{d_k}}\right)$$

$$= [-0.03024 \quad 0.003763 \quad 0.007626 \quad 0.008865 \quad 0.065863] \begin{bmatrix} 0.009839 \\ -0.007841 \\ -0.014326 \\ 0.001764 \\ 0.029318 \end{bmatrix} /\sqrt{5}$$

$$= \textbf{0.000675}$$

dari perhitungan *similarity* pada *query* pertama, seluruh hasil dapat dilihat pada Tabel 4.10.

Tabel 4.10 Hasil Perhitungan *Similarity* pada *Query* Pertama

Similarity (S_1)		
$S_{1;1} = 0,002416$	$S_{1;7} = 0,000654$	$S_{1;13} = 0,001467$
$S_{1;2} = -0,000894$	$S_{1;8} = -0,000814$	$S_{1;14} = 0,000232$
$S_{1;3} = 0,000722$	$S_{1;9} = 0,000532$	$S_{1;15} = -0,001569$
$S_{1;4} = -0,000645$	$S_{1;10} = -0,000952$	$S_{1;16} = 0,000588$
$S_{1;5} = -0,001632$	$S_{1;11} = -0,000875$	$S_{1;17} = 0,000425$
$S_{1;6} = 0,000259$	$S_{1;12} = -0,000786$	$S_{1;18} = 0,000675$

Berdasarkan hasil perhitungan nilai *similarity* pada Tabel 4.10 akan dilanjutkan ke perhitungan *softmax* untuk mendapatkan bobot *query* pertama (a_1) dengan menggunakan Persamaan 2.11. Jumlah token (T) adalah 18, sehingga $n = 18$. Perhitungan mencari *weight* atau bobot (a_1) adalah

$$a_{1;1} = softmax(S_{1;1}) = \frac{exp(S_{1;1})}{\sum_{j=1}^{n} exp(S_{1j})} = 0,055690$$

$$a_{1;2} = softmax(S_{1;2}) = \frac{exp(S_{1;2})}{\sum_{j=1}^{n} exp(S_{1j})} = 0,055506$$

$$a_{1;3} = softmax(S_{1;3}) = \frac{exp(S_{1;3})}{\sum_{j=1}^{n} exp(S_{1j})} = 0,0555962$$

$$a_{1;4} = softmax(S_{1;4}) = \frac{exp(S_{1;4})}{\sum_{j=1}^{n} exp(S_{1j})} = 0,0555203$$

$$\vdots$$

$$a_{1;18} = softmax(S_{1;18}) = \frac{exp(S_{1;18})}{\sum_{j=1}^{n} exp(S_{1j})} = 0,0555937$$

dari perhitungan bobot untuk *query* pertama (a_1), seluruh hasil dapat dilihat pada Tabel 4.11.

Tabel 4.11 Hasil Perhitungan Bobot pada *Query* Pertama

Weight (a_1)		
$a_{1;1} = 0,055690$	$a_{1;7} = 0,0555925$	$a_{1;13} = 0,0556377$
$a_{1;2} = 0,0555065$	$a_{1;8} = 0,0555109$	$a_{1;14} = 0,055569$
$a_{1;3} = 0,0555962$	$a_{1;9} = 0,0555857$	$a_{1;15} = 0,0554691$
$a_{1;4} = 0,0555203$	$a_{1;10} = 0,0555033$	$a_{1;16} = 0,0555888$
$a_{1;5} = 0,0554655$	$a_{1;11} = 0,0555075$	$a_{1;17} = 0,0555798$
$a_{1;6} = 0,0555705$	$a_{1;12} = 0,0555125$	$a_{1;18} = 0,0555937$

Bobot dari *query* pertama (a_1) tersebut digunakan dalam perhitungan *attention value*. Diberikan $x_k \in x$ di mana $0 < k \leq 18$ dan matriks bobot untuk *value* (v) yaitu $\mathbf{W}_h^V \in \mathbb{R}^{5\times5}$. Berdasarkan penjelasan sebelumnya terkait *query* dan *key*, matriks *value* akan dimisalkan $h = 1$, sehingga $\mathbf{W}_h^V$ adalah matriks identitas. Diketahui $x_k \in \mathbb{R}^{1\times5}$ dan

$v_k = x_k \mathbf{W}_h^V \in \mathbb{R}^{1\times 5}$, sedemikian hingga $v_k = x_k$. Perhitungan *attention value* untuk *query* pertama (A_1) dengan menggunakan Persamaan 2.7 adalah sebagai berikut.

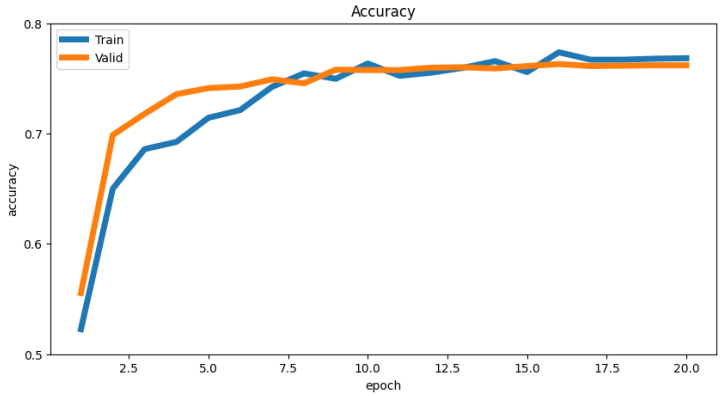
$$
\begin{aligned}
A_1 &= \sum_{k=1}^{18} (a_{1:k} \times v_k) \\
&= 0.05569\,[-0.03024 \quad 0.003763 \quad 0.007626 \quad 0.008865 \quad 0.065863] + 0,0555065 \\
&\quad [-0.016821 \quad 0.040294 \quad -0.037387 \quad -0.021312 \quad -0.033182] + 0.0555962 \\
&\quad [-0.023392 \quad -0.015365 \quad -0.028249 \quad -0.014592 \quad 0.019873] + 0,0555203 \\
&\quad [0.015605 \quad -0.017830 \quad 0.013653 \quad -0.049923 \quad -0.008574] + \mathbf{...} + 0.0555937 \\
&\quad [0.009839 \quad -0.007841 \quad -0.014326 \quad 0.001764 \quad 0.029318] \\
&= [\mathbf{-0.003315} \quad \mathbf{0.008947} \quad \mathbf{-0.016679} \quad \mathbf{-0.033762} \quad \mathbf{0.016251}]
\end{aligned}
$$

Perhitungan dari *similarity* hingga *attention value* dilakukan untuk seluruh *query* x_j. Proses ini dijalankan secara paralel untuk semua *head* menggunakan matriks bobot yang berbeda. Setiap token yang keluar dari setiap *head* memiliki dimensi yang sama seperti input x, yaitu $\mathbb{R}^{(n\times d)}$ dalam $d = 5$ dan $n = 18$. Langkah berikutnya dalam mekanisme *attention* adalah menggabungkan semua *output* dari setiap *head*, atau disebut juga *concatenation*. Jika *output* dari setiap *head* memiliki dimensi $\mathbb{R}^{18\times 5}$ dan pada model IndoBERT memiliki 12 *head*, maka hasil dari penggabungan ini akan memiliki dimensi $\mathbb{R}^{18\times 60}$.

4.5.3 Analisis dan Evaluasi Model IndoBERT pada Klasifikasi Berita

Pada subbab ini akan dijelaskan terkait analisis hasil model dalam memprediksi berita *clickbait* dan *non-clickbait* berdasarkan dari model IndoBERT dengan *hyperparameter* pada Tabel 4.9. Berdasarkan hasil *training*, akan ditampilkan hasil performansi saat pelatihan model berupa perbandingan grafik akurasi data pelatihan dan grafik akurasi data validasi. Selain itu, perhitungan evaluasi model terhadap data uji akan ditampilkan dalam bentuk laporan klasifikasi. Proses *fine-tuning* model dilakukan sebanyak 36 kali dan akan diambil hasil performansi terbaik dari tiap jenis dataset.

Pertama, pada Dataset **A**, grafik akurasi dapat dilihat pada Gambar 4.12. Terlihat bahwa akurasi pada data latih meningkat dari *epoch* 1 hingga *epoch* 8, yaitu mencapai 75%. Namun, setelahnya terdapat flukstuasi hingga *epoch* 17. Akurasi pada data validasi juga meningkat, tetapi mulai stabil dan sedikit flukatuasi setelah *epoch* 6.



Gambar 4.12 Grafik Akurasi Dataset **A** (*Batch Size* = 16, *LR* = 1e-7, *Epoch* = 20)

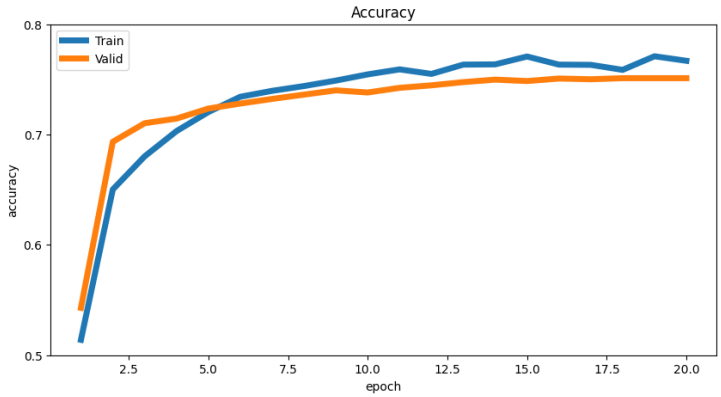
Performa akurasi model ini mendekati konvergen mulai *epoch* 17 sampai *epoch* 20. Berdasarkan grafik tersebut, performa model terindikasi *overfitting* yang cukup kecil. Setelah melalui tahap pelatihan, data uji pada Dataset **A** akan dilakukan proses evaluasi model dengan menerima data baru. Hasil laporan klasifikasi Dataset **A** terlihat pada Gambar 4.13. Terlihat pada laporan klasifikasi bahwa model bekerja lebih baik untuk mendeteksi berita berlabel *non-clickbait*, dengan nilai metrik lebih tinggi dibandingkan berita berlabel *clickbait*.

Classification Report:

	precision	recall	f1-score	support
Clickbait	0.60	0.63	0.62	478
Non Clickbait	0.78	0.76	0.77	834
accuracy			0.71	1312
macro avg	0.69	0.69	0.69	1312
weighted avg	0.72	0.71	0.71	1312

Gambar 4.13 Laporan Klasifikasi Dataset **A** (*Batch Size* = 16, *LR* = 1e-7, *Epoch* = 20)

Kedua, grafik akurasi pada Dataset **B** dapat dilihat pada Gambar 4.14. Akurasi pelatihan pada data latih meningkat dengan signifikan mencapai sekitar 76% pada *epoch* 11 tetapi terdapat fluktuasi pada *epoch* berikutnya. Sedangkan, akurasi validasi meningkat mencapai 74% pada *epoch* 9 dengan sedikit fluktuasi setelahnya.



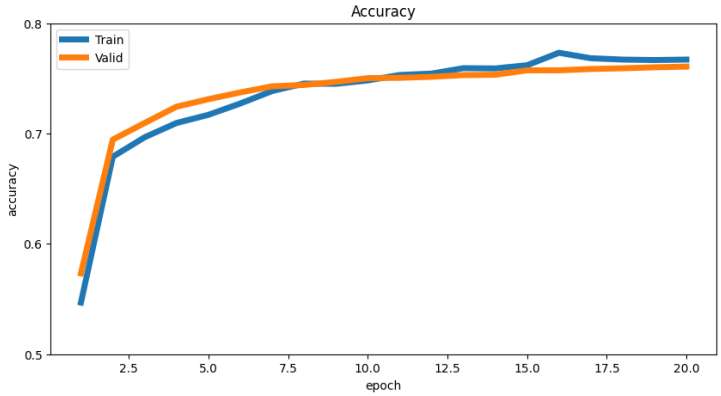
Gambar 4.14 Grafik Akurasi Dataset **B** (*Batch Size* = 16, *LR* = 1e-7, *Epoch* = 20)

Pada pelatihan Dataset **B** terlihat adanya *gap* pada proses pelatihan dan validasi. Performa dari kombinasi *hyperparameter* yang lebih unggul sama seperti yang digunakan pada Dataset **A**. Berdasarkan Gambar 4.14, hasil performa mengindikasikan bahwa model mengalami *overfitting* lebih besar daripada yang dialami Dataset **A**. Evaluasi model menggunakan data uji pada Dataset **B** dapat dilihat pada Gambar 4.15 dalam bentuk laporan klasifikasi. Terjadi peningkatan akurasi mencapai 72% dibandingkan dengan Dataset **A** dan memiliki performa lebih baik dengan nilai presisi, *recall*, serta *f1 score* yang lebih tinggi daripada Dataset **A**. Secara keseluruhan, performa yang lebih baik dilakukan pada Dataset **B** dengan lebih banyak *epoch* dan *batch size* yang lebih besar.

Classification Report:				
	precision	recall	f1-score	support
Clickbait	0.61	0.63	0.62	242
Non Clickbait	0.78	0.77	0.77	414
accuracy			0.72	656
macro avg	0.70	0.70	0.70	656
weighted avg	0.72	0.72	0.72	656

Gambar 4.15 Laporan Klasifikasi Dataset **B** (*Batch Size* = 16, *LR* = 1e-7, *Epoch* = 20)

Ketiga, pada Dataset **C**, grafik akurasi dapat dilihat pada Gambar 4.16. Pada akurasi data latih mengalami peningkatan mencapai sekitar 75% pada *epoch* 8. Akurasi pada data validasi juga meningkat hingga mencapai 76% pada *epoch* 11. Dataset **C** menunjukkan peningkatan yang baik pada pelatihan dan validasi. Berdasarkan grafik, grafik pelatihan dan validasi selalu bersinggungan sehingga menghasilkan model yang konvergen, tetapi terdapat sedikit *overfitting* karena grafik validasi belum berada di atas grafik pelatihan.



Gambar 4.16 Grafik Akurasi Dataset **C** (*Batch Size* = 16, *LR* = 1e-7, *Epoch* = 20)

Proses selanjutnya yaitu mengevaluasi model menggunakan data uji pada Dataset **C**. Hasil evaluasi model pada Dataset **C** dalam bentuk laporan klasifikasi dapat dilihat pada Gambar 4.17. Seluruh metrik evaluasi lebih unggul dibandingkan dengan performa Dataset **A** dan Dataset **B**.

Classification Report:				
	precision	recall	f1-score	support
Clickbait	0.68	0.62	0.65	116
Non Clickbait	0.80	0.84	0.82	212
accuracy			0.76	328
macro avg	0.74	0.73	0.73	328
weighted avg	0.76	0.76	0.76	328

Gambar 4.17 Laporan Klasifikasi Dataset **C** (*Batch Size* = 16, *LR* = 1e-7, *Epoch* = 20)

Performa yang lebih baik dalam memprediksi berita *clickbait* dan *non-clickbait* terdapat pada pelatihan Dataset **C**. Khususnya pada data berlabel *clickbait*, memperoleh evaluasi terbaik pada F1-Score mencapai 73% pada Dataset **C**. Performa yang dihasilkan dari seluruh skenario menujukkan bahwa kombinasi *hyperparameter* (*learning rate* 1e-7, *batch size* 16, dan *epoch* 20) mampu menghasilkan hasil performa yang optimal terutama pada konvergensi model. Pada proses evaluasi pada data uji, semua metrik (presisi, *recall*, dan *F1-score*) sedikit lebih tinggi pada Dataset **C**. Selain itu, proporsi data latih yang lebih besar cenderung meningkatkan akurasi. Pada Dataset **C**, yang memiliki proporsi data pelatihan lebih besar dibandingkan Dataset **A** dan Dataset **B**, terlihat adanya peningkatan akurasi baik pada tahap validasi maupun pengujian.

Performa model yang lebih konsisten juga terlihat pada Dataset **C**. Hal ini menunjukkan bahwa model IndoBERT mampu memanfaatkan jumlah data pelatihan yang lebih besar untuk mencapai performa yang stabil dan mengurangi indikasi *overfitting* pada model. Dalam proporsi data validasi juga konsisten, dengan setiap dataset menggunakan 20% dari data pelatihan sebagai data validasi. Hal ini dapat menjaga validasi yang cukup representatif untuk setiap ukuran dataset, memastikan bahwa performa model dievaluasi secara akurat selama proses pelatihan. Secara keseluruhan, Dataset **C** memberikan performa terbaik dengan akurasi data uji dan *F1-score* tertinggi. Performa akurasi yang dihasilkan pada model secara keseluruhan masih terbilang cukup rendah, tetapi konvergensi pada model yang dihasilkan sangat kecil terindikasi *overfitting*. Berdasarkan proses *fine-tunning* model yang telah dilakukan sebanyak 36 kali dengan kombinasi tiap *hyperparameter*, model IndoBERT dalam penerapan semua jenis dataset pada penelitian ini menunjukkan performa terbaik pada Dataset **C**.

Performansi pada model IndoBERT tersebut akan dibandingkan dengan performansi salah satu model *pre-trained* turunan BERT yang dapat memproses bahasa Indonesia yaitu model RoBERTa dari *Hugging Face* oleh Cahya '*cahya/roberta-base-indonesian-522M*', di mana model tersebut telah dilatih dengan dataset berbahasa Indonesia. Perbandingan hasil evaluasi terhadap kedua model ditunjukkan pada Tabel 4.12.

Tabel 4.12 Perbandingan Performa Metrik (%) Model IndoBERT dan RoBERTa

Model	***Accuracy***		***Precision***	***Recall***	***F1-score***
	Validation	***Test***			
IndoBERT model: *indobenchmark/indobert-base-p1*					
Dataset **A**	75%	71%	69%	69%	69%
Dataset **B**	76%	72%	70%	70%	70%
Dataset **C**	**77%**	**76%**	**74%**	**73%**	**73%**
RoBERTa model: *cahya/roberta-base-indonesian-522M*					
Dataset **A**	61%	54%	60%	60%	54%
Dataset **B**	63%	53%	60%	59%	52%
Dataset **C**	62%	59%	60%	61%	59%

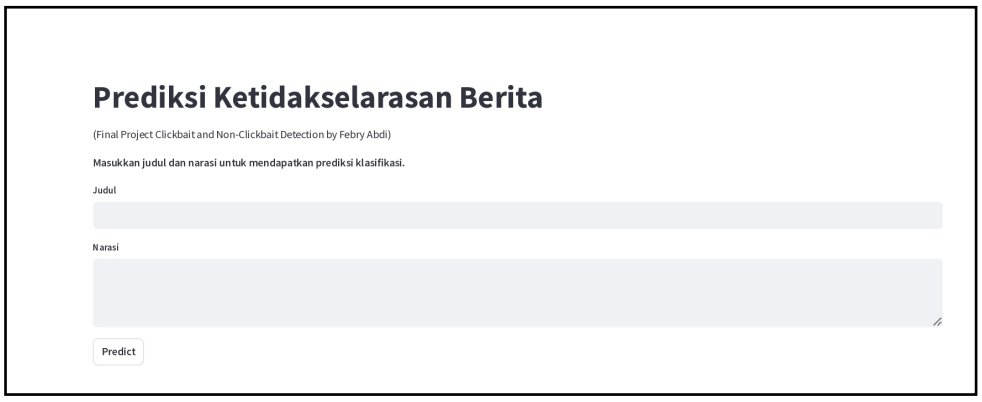
Berdasarkan pada Tabel 4.12, menunjukkan hasil performa yang dihasilkan oleh model IndoBERT lebih unggul dari model RoBERTa dalam mengklasifikasikan berita *clickbait* dan *non-clickbait*. Hal ini menujukkan bahwa model IndoBERT secara konsisten mengungguli model RoBERTa. Terlihat dari segi akurasi yang dihasilkan, IndoBERT lebih unggul dari pada RoBERTa. Hal ini dapat disebabkan oleh beberapa faktor seperti pelatihan pada model IndoBERT telah dilatih secara khusus pada

korpus bahasa Indonesia yang lebih relevan dan representatif untuk tugas-tugas yang berhubungan dengan bahasa Indonesia. Selain itu, IndoBERT memiliki *layer pooler* yang mempermudah tugas klasifikasi dengan menyediakan vektor keluaran yang relevan dan konsisten, sedangkan pada RoBERTa tidak ada.

Penelitian oleh Syahputra dkk. Syahputra et al. (2023) menunjukkan bahwa model IndoBERT memiliki performa lebih unggul dibandingkan dengan RoBERTa. Penelitian Syahputra dkk. Syahputra et al. (2023) menghasilkan nilai *F1-score* mencapai 95% untuk IndoBERT dan 91% untuk RoBERTa, membuktikan keunggulan pada model IndoBERT. Sementara itu, penelitian Tugas Akhir ini menghasilkan *F1-score* tertinggi mencapai 73% untuk IndoBERT dan 59% untuk RoBERTa. Perbedaan hasil ini disebabkan oleh beberapa faktor, termasuk perbedaan dataset yang digunakan. Penelitian oleh Syahputra dkk. Syahputra et al. (2023) menggunakan satu variabel, yaitu prediksi pada judul berita *clickbait*, sedangkan penelitian Tugas Akhir ini menggunakan data primer dengan dua variabel, yaitu judul dan narasi untuk memprediksi ketidakselarasan pada berita *clickbait*. Hal ini juga memicu penurunan performa pada penelitian Tugas Akhir ini dibandingkan dengan penelitian oleh Syahputra dkk. Selain itu, meskipun terdapat kesamaan arsitektur dasar antara IndoBERT dan RoBERTa, perbedaan strategi *pre-training* dapat menyebabkan perbedaan performa dalam tugas spesifik seperti prediksi berita *clickbait* dan *non-clickbait*.

4.6 Percobaan Prediksi Data Baru Pada Website Clickbait

Berdasarkan perolehan model terbaik, akan dilanjutkan dengan proses implementasi (*deployment*) model ke bentuk website sebagai *tools* untuk memprediksi ketidakselarasan (*clickbait*) dan keselarasan berita (*non-clickbait*). Pada tahap ini akan menggunakan *library* bernama Streamlit untuk membangun sebuah website prediksi. Streamlit adalah *tools* inovatif dan mudah digunakan untuk membangun situs website interaktif menggunakan bahasa pemrograman Python. Tampilan utama atau *dashboard* prediksi berita berupa input judul, input narasi, dan tombol prediksi yang ditunjukkan pada Gambar 4.18.



Gambar 4.18 Dashboard Prediksi Berita

Pada tahapan ini terdapat 2 buah data berita (data baru) yang akan diuji pada website prediksi tersebut. Data berita sebagai data uji tersebut diambil dari website portal berita *online*. Berita pertama yaitu dari Brilio.net ditunjukkan pada Tabel 4.13.

Tabel 4.13 Berita Prediksi 1

URL:	`https://www.brilio.net/selebritis/dituding-jadi-penyebab-kematian-mendiang-stevie-agnecya-icha-annisa-faradila-beri-reaksi-mengejutkan-240324y.html`
Penerbit:	Brilio.net
Judul:	Dituding jadi penyebab kematian mendiang Stevie Agnecya, Icha Annisa Faradila beri reaksi mengejutkan
Isi Naskah Berita:	Publik turut berduka atas kepergian Stevie Agnecya, artis yang juga merupakan mantan istri dari aktor Samuel Rizal. Ia meninggal dunia di RS Cipto Mangunkusumo (RSCM) pada 21 Maret 2024 lalu. Publik turut berduka atas kepergian Stevie Agnecya, artis yang juga merupakan mantan istri dari aktor Samuel Rizal. Ia meninggal dunia di RS Cipto Mangunkusumo (RSCM) pada 21 Maret 2024 lalu. Sebelum meninggal dunia, Stevie Agnecya sempat bercerita bahwa dirinya kena santet. Pengakuan ini diumbar Stevie di akun TikTok pribadinya pada September 2023 lalu. Bukan tanpa bukti, dalam unggahannya, istri dari Anggi Pratama ini membagikan hasil rontgennya, dimana ditemukan benda aneh mirip paku di dalam tubuhnya. Bahkan, ia juga mengaku sudah 4 tahun lamanya merasakan sakit, tanpa mengetahui penyebab penyakitnya. ”Aku dan suamiku yang selalu ada disampingku nggak akan tahu memang ternyata makin kita dekat sama Allah makin ditunjukkin semua bukti santet itu yang ada di tubuhku sudah 4 tahun,”...

Berdasarkan data berita tersebut, judul berita dan isi naskah berita akan diuji ke website prediksi sehingga menghasilkan nilai prediksi yang ditunjukkan pada Gambar 4.19.

Gambar 4.19 Hasil Prediksi Berita 1

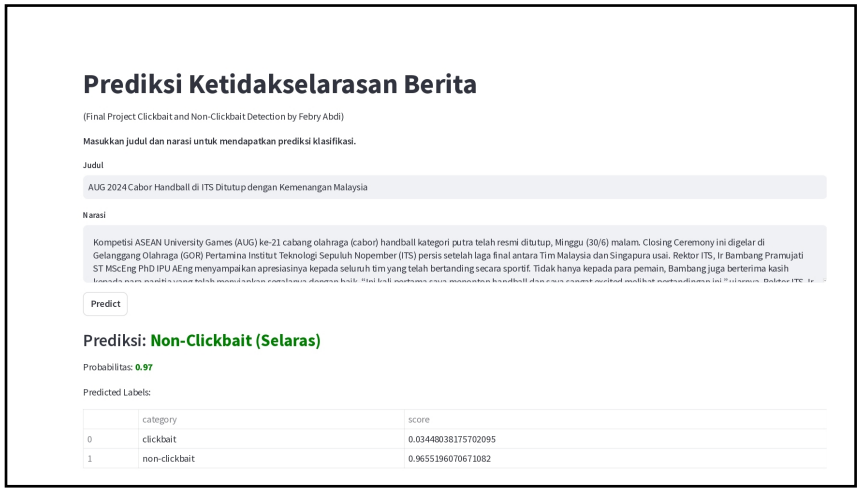
Hasil prediksi yang ditunjukkan pada Gambar 4.19 adalah berita berstatus *clickbait* dengan nilai probabilitas yaitu 0,98819. Salah satu faktor penyebab berita tersebut terprediksi *clickbait* dengan nilai probabilitas yang cukup tinggi yaitu pada penggunaan judul menggunakan kalimat sensasional seperti “... reaksi mengejutkan”.

Selanjutnya, akan dilakukan prediksi terhadap berita kedua yang diambil dari portal berita ITS News. Data berita tersebut ditunjukkan pada Tabel 4.14.

Tabel 4.14 Berita Prediksi 2

URL:	`https://www.its.ac.id/news/2024/07/01/aug-2024-cabor-handball-di-its-ditutup-dengan-kemenangan-malaysia/`
Penerbit:	ITS News
Judul:	AUG 2024 Cabor Handball di ITS Ditutup dengan Kemenangan Malaysia
Isi Naskah Berita:	Kompetisi ASEAN University Games (AUG) ke-21 cabang olahraga (cabor) handball kategori putra telah resmi ditutup, Minggu (30/6) malam. Closing Ceremony ini digelar di Gelanggang Olahraga (GOR) Pertamina Institut Teknologi Sepuluh Nopember (ITS) persis setelah laga final antara Tim Malaysia dan Singapura usai. Rektor ITS, Ir Bambang Pramujati ST MScEng PhD IPU AEng menyampaikan apresiasinya kepada seluruh tim yang telah bertanding secara sportif. Tidak hanya kepada para pemain, Bambang juga berterima kasih kepada para panitia yang telah menyiapkan segalanya dengan baik. "Ini kali pertama saya menonton handball dan saya sangat excited melihat pertandingan ini," ujarnya. Rektor ITS, Ir Bambang Pramujati ST MScEng PhD IPU AEng saat menyampaikan pidato pada Closing Ceremony AUG ke 21 cabor indoor handball kategori putra di ITS. Selain itu, Bambang juga mengucapkan selamat kepada Tim Malaysia...

Judul berita dan isi naskah berita pada Tabel 4.14 akan diuji ke website prediksi yang ditunjukkan pada Gambar 4.20.



Gambar 4.20 Hasil Prediksi Berita 2

Hasil prediksi pada Gambar 4.20 menunjukkan bahwa berita tersebut berstatus *non-clickbait* atau selaras dengan nilai probabilitas yaitu 0,9655. Hal tersebut disebabkan karena tidak adanya kata bersifat sensasional yang menyebabkan *clickbait* pada judul serta informasi yang disajikan pada berita sesuai antara judul dengan isi naskahnya.

BAB V
KESIMPULAN DAN SARAN

Pada bab ini disajikan kesimpulan yang diperoleh dari penelitian Tugas Akhir ini, serta memberikan saran untuk perbaikan penelitian selanjutnya.

5.1 Kesimpulan

Adapun kesimpulan yang diperoleh dari Tugas Akhir ini adalah sebagai berikut.

1. Implementasi model IndoBERT dengan tiga proporsi berbeda dalam pembagian dataset dan kombinasi *hyperparameter* menghasilkan akurasi rata-rata di atas 75% untuk proses validasi dan di atas 60% untuk pengujian dengan data baru. Berdasarkan eksperimen, hasil uji coba terbaik yaitu pada Dataset **C** dengan *epoch* 20 menghasilkan akurasi pelatihan sebesar 77% dan akurasi pada data uji sebesar 76%. Berdasarkan performa pada data berlabel *clickbait*, performa metrik khususnya F1-score tertinggi terdapat pada Dataset **C** dengan kombinasi *hyperparameter* serupa mencapai 73%. Hasil evaluasi model IndoBERT menunjukkan bahwa model ini mampu mengklasifikasikan data teks berita dalam memprediksi ketidakselarasan antara judul dan isi berita *clickbait*.

2. Dari hasil performa model, diperoleh bahwa IndoBERT memberikan hasil yang belum cukup baik karena konvergensi model masih terindikasi *overfitting* yang di mana model IndoBERT belum mampu generalisasi dengan baik dari data pelatihan ke data yang belum pernah dilihat sebelumnya (data uji) serta nilai akurasi yang dihasilkan juga masih rendah. Namun, dibandingkan dengan model *pre-trained* BERT lainnya seperti RoBERTa, model IndoBERT memiliki performa lebih unggul dari segi akurasi, presisi, recall, dan F1-score.

5.2 Saran

Adapun penulis memberikan beberapa saran untuk pengembangan penelitian selanjutnya guna mencapai hasil yang lebih optimal sebagai berikut.

1. Melakukan penelitian dengan jumlah data yang seimbang dan data dengan kualitas yang baik seperti memastikan data terdistribusi normal agar kinerja model lebih optimal. Selain itu, dapat melakukan proses pelabelan yang lebih akurat seperti dilakukan secara manual dengan para ahli bahasa atau ahli sastra.

2. Melakukan penelitian menggunakan model IndoBERT lain seperti IndoBERT-*large*, IndoBERT-*lite*, atau model *pre-trained* turunan BERT yang telah dilatih dengan kamus bahasa Indonesia dan dapat digunakan dalam kasus klasifikasi berurut (*sequence classification*) seperti ALBERT, XLM-RoBERTa, BERT-*multilingual*, dan DistilBERT.

DAFTAR PUSTAKA

Anand, A., Chakraborty, T., & Park, N. (2019). *We used neural networks to detect clickbaits: You won't believe what happened next!*

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ganesh, P., Chen, Y., Lou, X., Khan, M. A., Yang, Y., Sajjad, H., ... Winslett, M. (2021). Compressing large-scale transformer-based models: A case study on bert. *Transactions of the Association for Computational Linguistics*, *9*, 1061–1080.

Kadhim, A. I. (2018). An evaluation of preprocessing techniques for text classification. *International Journal of Computer Science and Information Security (IJCSIS)*, *16*(6), 22–32.

Kingma, D. P., & Ba, J. (2017). *Adam: A method for stochastic optimization.*

Koto, Fajri, Rahimi, Afshin, Lau, Han, J., ... Timothy (2020). Indolem and indobert: A benchmark dataset and pre-trained language model for indonesian nlp. *arXiv preprint arXiv:2011.00677*.

Luo, L., & Wang, Y. (2019). Emotionx-hsu: Adopting pre-trained bert for emotion classification. *arXiv preprint arXiv:1907.09669*.

Mikolov, T., Mikolov, T., Chen, K., Chen, K., Chen, K., Corrado, G. S., ... Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv: Computation and Language*.

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep learning–based text classification: a comprehensive review. *ACM computing surveys (CSUR)*, *54*(3), 1–40.

Mirończuk, M. M., & Protasiewicz, J. (2018). A recent overview of the state-of-the-art elements of text classification. *Expert Systems with Applications*, *106*, 36–54.

Nayak, C., & Kumar, A. (2022, 11). Next word prediction using machine learning techniques. *Cybersecurity*, *54*, 5161-5171.

Omidvar, A., Jiang, H., & An, A. (2018). *Using neural network for identifying clickbaits in online news media.*

Potthast, M., Köpsel, S., Stein, B., & Hagen, M. (2016). Clickbait detection. In (Vol. 9626, p. 810-817). doi: 10.1007/978-3-319-30671-1_72

Putri, D. U. K., & Pratomo, D. N. (2022, Jul.). Clickbait detection of indonesian news headlines using fine-tune bidirectional encoder representations from transformers (bert). *Inform : Jurnal Ilmiah Bidang Teknologi Informasi dan Komunikasi*, *7*(2), 162-168. Retrieved from `https://ejournal.unitomo.ac.id/index.php/inform/article/view/4686` doi: 10.25139/inform.v7i2.4686

Raschka, S. (2021). *L19.4.2 self-attention and scaled dot-product attention.* Retrieved from `https://www.youtube.com/watch?v=0PjHri8tc1c` (Diakses pada 16 Mei 2024)

Romli, A. S. M. (2018). *Jurnalistik online: Panduan mengelola media online.* Nuansa Cendekia.

Syahputra, M. E., Kemala, A. P., & Ramdhan, D. (2023, Jun.). Clickbait detection in indonesia headline news using indobert and roberta. *Jurnal Riset Informatika*, *5*(3), 425–430. Retrieved from `https://ejournal.kresnamediapublisher.com/index.php/jri/article/view/237` doi: 10.34288/jri.v5i4.237

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Wilie, B., Vincentio, K., Winata, G. I., Cahyawijaya, S., Li, X., Lim, Z. Y., ... others (2020). Indonlu: Benchmark and resources for evaluating indonesian natural language understanding. *arXiv preprint arXiv:2009.05387*.

Yoon, S., Park, K., Shin, J., Lim, H., Won, S., Cha, M., & Jung, K. (2019). *Detecting incongruity between news headline and body text via a deep hierarchical encoder.*

Yu, T., & Zhu, H. (2020). *Hyper-parameter optimization: A review of algorithms and applications.* Retrieved from `https://arxiv.org/abs/2003.05689`

Zhao, B. (2017). Web scraping. *Encyclopedia of big data*, *1*.

DAFTAR LAMPIRAN

Lampiran 1
Hasil Pelatihan dan Evaluasi Dataset A

Dataset	Batch Size	LR	Train Acc	Valid Acc	Test Acc	Recall	Prec.	F1-score
Dataset **A** (80%; 20%)	16	1e-6	0,912	0,809	0,68	0,73	0,71	0,68
		2e-6	0,977	0,843	0,67	0,68	0,67	0,66
		5e-6	0,999	0,840	0,59	0,64	0,64	0,59
		1e-7	0,768	0,762	0,71	0,69	0,69	0,69
		2e-7	0,706	0,682	0,57	0,64	0,65	0,57
		5e-7	0,807	0,757	0,68	0,70	0,68	0,67
	32	1e-6	0,845	0,778	0,60	0,66	0,67	0,60
		2e-6	0,950	0,822	0,69	0,69	0,68	0,68
		5e-6	0,998	0,859	0,65	0,68	0,67	0,64
		1e-7	0,643	0,637	0,62	0,64	0,63	0,61
		2e-7	0,683	0,676	0,59	0,64	0,64	0,59
		5e-7	0,766	0,723	0,60	0,65	0,65	0,60

Lampiran 2
Hasil Pelatihan dan Evaluasi Dataset B

Dataset	Batch Size	LR	Train Acc	Valid Acc	Test Acc	Recall	Prec.	F1-score
Dataset **B** (90%; 10%)	16	1e-6	0,893	0,808	0,68	0,69	0,67	0,67
		2e-6	0,972	0,827	0,67	0,67	0,66	0,66
		5e-6	0,999	0,856	0,59	0,63	0,62	0,59
		1e-7	0,767	0,751	0,72	0,70	0,70	0,70
		2e-7	0,713	0,686	0,63	0,67	0,67	0,63
		5e-7	0,805	0,756	0,65	0,68	0,67	0,65
	32	1e-6	0,871	0,803	0,54	0,62	0,64	0,54
		2e-6	0,948	0,839	0,66	0,68	0,67	0,65
		5e-6	0,996	0,856	0,70	0,69	0,69	0,69
		1e-7	0,654	0,642	0,53	0,60	0,63	0,52
		2e-7	0,691	0,667	0,61	0,65	0,64	0,61
		5e-7	0,779	0,735	0,64	0,67	0,66	0,64

Lampiran 3
Hasil Pelatihan dan Evaluasi Dataset C

Dataset	Batch Size	LR	Train Acc	Valid Acc	Test Acc	Recall	Prec.	F1-score
Dataset **C** (95%; 5%)	16	1e-6	0,911	0,813	0,69	0,69	0,67	0,67
		2e-6	0,973	0,848	0,73	0,71	0,71	0,71
		5e-6	0,997	0,857	0,72	0,69	0,69	0,69
		1e-7	0,767	0,761	0,76	0,73	0,74	0,73
		2e-7	0,707	0,687	0,67	0,69	0,67	0,67
		5e-7	0,811	0,766	0,67	0,69	0,68	0,67
	32	1e-6	0,849	0,775	0,66	0,67	0,66	0,65
		2e-6	0,947	0,840	0,71	0,70	0,69	0,69
		5e-6	0,997	0,857	0,59	0,64	0,63	0,59
		1e-7	0,652	0,646	0,61	0,65	0,64	0,61
		2e-7	0,688	0,681	0,56	0,63	0,65	0,56
		5e-7	0,764	0,739	0,65	0,68	0,67	0,64

Lampiran 4
Syntax Website Predict Model

```
import streamlit as st
import pandas as pd
import torch
from transformers import AutoTokenizer, BertForSequenceClassification
from keras.preprocessing.sequence import pad_sequences
import re
from Sastrawi.StopWordRemover.StopWordRemoverFactory
import StopWordRemoverFactory
import string
import torch.nn.functional as F

# Define the model and tokenizer
MODEL_PATH = 'finetuned_IndoBERT_epoch_4.model'
MODEL_NAME = 'indobenchmark/indobert-base-p1'
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

# Initialize the model
model = BertForSequenceClassification.from_pretrained(MODEL_NAME,
        num_labels=2)
model.load_state_dict(torch.load(MODEL_PATH, map_location=device))
model.to(device)
model.eval()

# Initialize the tokenizer
tokenizer = AutoTokenizer.from_pretrained('indobenchmark/
```

```
------------indobert-base-p1')

# Preprocessing functions
def lower_case(text):
    return text.lower()

def clean_and_remove_non_alphanumeric(text):
    text = re.sub(r'([^\w\s])', r'-\1-', text)
    return re.sub(r'[^A-Za-z0-9\s.,?!]', '', text)

def remove_punctuation(text):
    translator = str.maketrans('', '', string.punctuation.replace
                ('.', '').replace(',', ''))
    return text.translate(translator)

def remove_stopwords(text):
    factory = StopWordRemoverFactory()
    stopword_remover = factory.create_stop_word_remover()
    return stopword_remover.remove(text)

def preprocess(text):
    text = lower_case(text)
    text = remove_punctuation(text)
    text = remove_stopwords(text)
    text = clean_and_remove_non_alphanumeric(text)
    return text

# Streamlit app
st.title("Prediksi-Ketidakselarasan-Berita")
st.write("(Final-Project-Clickbait-and-Non-Clickbait
--------Detection-by-Febry-Abdi)")
st.write("**Masukkan-judul-dan-narasi-untuk-mendapatkan
--------prediksi-klasifikasi.**")

title = st.text_input("**Judul**")
text = st.text_area("**Narasi**")

if st.button("**Predict**"):
    # Preprocess inputs
    processed_title = preprocess(title)
    processed_text = preprocess(text)

    # Tokenize sentences
    input_ids_title = tokenizer.encode(processed_title[:512],
                      add_special_tokens=True)
    input_ids_text = tokenizer.encode(processed_text[:512],
                     add_special_tokens=True)
```

```
# Pad sequences
MAX_LEN = 256
input_ids_title = pad_sequences([input_ids_title],
                  maxlen=MAX_LEN,
                  dtype="long",
                  value=0,
                  truncating="post",
                  padding="post")
input_ids_text = pad_sequences([input_ids_text],
                  maxlen=MAX_LEN,
                  dtype="long",
                  value=0,
                  truncating="post",
                  padding="post")

# Create attention masks
attention_masks_title = [[int(token_id > 0)
                          for token_id
                          in input_ids_title[0]]]
attention_masks_text = [[int(token_id > 0)
                         for token_id
                         in input_ids_text[0]]]

# Convert inputs and masks to tensors
input_ids_title = torch.tensor(input_ids_title).to(device)
attention_masks_title = torch.tensor(attention_masks_title)
                        .to(device)
input_ids_text = torch.tensor(input_ids_text).to(device)
attention_masks_text = torch.tensor(attention_masks_text)
                       .to(device)

# Concatenate title and text inputs and masks
input_ids = torch.cat((input_ids_title,
            input_ids_text), dim=1)
attention_masks = torch.cat((attention_masks_title,
                  attention_masks_text), dim=1)

# Perform prediction
with torch.no_grad():
    inputs = {'input_ids': input_ids,
              'attention_mask': attention_masks}
    outputs = model(**inputs)
    logits = outputs.logits

# Apply softmax to get probabilities
probs = F.softmax(logits, dim=1).detach().cpu().numpy()
```

```
    # Get the predicted class (positive or negative)
    pred_class = "Non-Clickbait (Selaras)"
                 if probs[0][1] > probs[0][0]
                 else "Clickbait (Tidak Selaras)"
    pred_prob = probs[0][1]
                 if pred_class == "Non-Clickbait (Selaras)"
                 else probs[0][0]

     # Set color based on prediction
    color = "green" if pred_prob == probs[0][1] else "red"

    # Display result with color & probability values
    st.markdown(f"### Prediksi: <span style='color:{color};
----------------font-weight:bold;'>{pred_class}</span>",
                 unsafe_allow_html=True)
    st.markdown(f"Probabilitas: <span style='color:{color};
----------------font-weight:bold;'>{pred_prob:.2f}</span>",
                 unsafe_allow_html=True)

    chart_data = pd.DataFrame(
            {
                "category": ['clickbait', 'non-clickbait'],
                "score": probs[0]
            })
    st.markdown(f"Predicted Labels:")
    st.table(chart_data)
```

UCAPAN TERIMA KASIH

Penyelesaian penulisan tugas akhir ini tidak lepas dari orang-orang terdekat penulis yang telah mendukung dan memotivasi penulis. Oleh sebab itu, penulis mengucapkan terima kasih kepada:

1. Kak Anditya yang telah membantu dalam memberikan masukan terkait Tugas Akhir ini terutama mengajari dalam hal *machine learning* dan memberikan rekomendasi ide pada proses ini.

2. Teman-teman seperjuangan anak bimbingan Pak Isa (Sulthan, Trisna, Safana, Aqila, Chika, Naila, dan Oryza) yang saling menguatkan dan memberikan motivasi dalam menyelesaikan Tugas Akhir dengan tepat waktu.

3. Sobat Zoo (Denis, Riana, dan Siska) selaku sahabat yang selalu ada untuk berkeluh kesah, saling mendukung, dan memberi semangat.

4. Teman-teman SOKIN beranggotakan 19 orang yang telah berjuang bersama dan mengisi hari - hari penulis dengan penuh suka duka.

5. Teman-teman seperjuangan, Matematika ITS Angkatan 2020 yaitu angkatan MATRIXX STI-55 yang telah mengisi hari - hari penulis dengan penuh keceriaan, motivasi, dan pengalaman.

6. Mas, Mbak, dan seluruh Rekan Media Center - ITS TV (Kabinet Bahagia, Kabinet Kokoh, E13, E14, E15, E16, dan E17) yang telah memberikan pengalaman, ilmu, cerita, dan kekeluargaan bagaikan rumah kecil yang mengesankan selama berkuliah.

7. Teman-teman kepanitiaan dan organisasi UKM EXPO 2021, MABA CUP 2021, Eskalator Cita Ini Lho ITS! 2022, GERIGI ITS 2022, Kabinet Heroes IBC 2021, Kabinet Metamorfosa IBC 2022, semeton Wibisana TPKH ITS 2020, serta teman-teman KM ITS dan seluruh Civitas Akademika ITS yang telah memberikan berbagai pengalaman dan ilmu yang mengesankan selama berikuliah.

Penulis juga mengharapkan kritik dan saran yang membangun dari berbagai pihak untuk penyempurnaan isi tugas akhir ini. Akhir kata, semoga tugas akhir ini bermanfaat bagi semua pihak yang bersangkutan.

Surabaya, Juli 2024

I Gede Febry Abdi Saputra

BIODATA PENULIS

Nama lengkap penulis adalah I Gede Febry Abdi Saputra, dengan nama panggilan Febry. Saat pembuatan laporan ini, penulis berstatus sebagai Mahasiswa Departemen Matematika Institut Teknologi Sepuluh Nopember angkatan 2020. Lahir di Denpasar, 24 Februari 2024 dan berasal dari Nusa Dua, Bali. Pendidikan formal yang telah ditempuh sebelum di bangku perkuliahan mulai dari TK Pelangi Jimbaran (2006-2008), SD Negeri 8 Benoa (2008-2014), SMP Negeri 1 Kuta (2014-2017), dan SMA Negeri 1 Kuta (2017-2020). Pertengahan tahun 2020, penulis melanjutkan pendidikan menempuh Strata-1 di Departemen Matematika Institut Teknologi Sepuluh Nopember melalui jalur SNMPTN 2020. Penulis memiliki hobi berolahraga khususnya olahraga badminton dan turut bergabung dalam komunitas badminton di kampus yaitu ITS Badminton Community (IBC) sekaligus menjadi pengurus IBC sebagai Bendahara selama dua periode (2021-2023). Selain itu, penulis juga aktif dalam kepanitiaan yang diikuti antara lain Staf Acara UKM EXPO 2021, Ketua Divisi Acara MABA CUP 2021, Staf *Campaign Marketing* Eskalator Cita Ini Lho ITS! 2022, dan Wakil Ketua Divisi Acara GERIGI ITS 2022. Penulis juga bergabung dalam komunitas Media Center Kampus ITS (ITS TV) sebagai kru magang aktif angkatan E15. Selama dua periode di ITS TV, penulis turut aktif sebagai Staf Bincang dan Kepala Divisi HRM (Human Resource Management). Selain itu, selama di ITS TV, penulis juga aktif berperan sebagai *freelancer* Voice Over Talent dan Presenter di konten YouTube Institut Teknologi Sepuluh Nopember. Penulis juga berkesempatan mengikuti kegiatan studi independen kampus merdeka yaitu pada program Bangkit Academy 2023 Batch 1 dengan mengambil konsentrasi *Machine Learning*. Bagi pembaca yang ingin berdiskusi lebih lanjut atau memberikan kritik serta saran terkait Tugas Akhir ini, dapat mengirimkan pesan melalui email: febry24.saputra@gmail.com.